2018 IEEE 5th International Conference on Data Science and Advanced Analytics

MixDir: Scalable Bayesian Clustering for High-Dimensional Categorical Data

Constantin Ahlmann-Eltze
Heidelberg University
Heidelberg, Germany
ahlmann-eltze@stud.uni-heidelberg.de

Christopher Yau
Centre for Computational Biology
University of Birmingham
Birmingham, United Kingdom

Alan Turing Institute
London, United Kingdom
c.yau@bham.ac.uk

***Abstract*—Multivariate analysis of high-dimensional datasets with multiple categorical variables (e.g. surveys, questionnaires) is a challenging task but can reveal patterns of responses that are masked from univariate analyses. In this paper we propose a novel variational inference algorithm to cluster high-dimensional categorical observations into latent classes. Variational inference is an approximate Bayesian inference algorithm, which combines fast optimization methods with the ability to propagate the uncertainty to the clustering (soft clustering). The model is robust to misspecification of the number of latent classes and can infer a reasonable number from the data. We assess the performance on synthetic and real world data and show that our algorithm has similar performance to the best other tested method if the correct number of classes is known and outperforms the other methods if it the number of classes needs to be inferred. An `R`-package implementing our algorithm is available at the Comprehensive `R` Archive Network[1].**

***Index Terms*—High-dimensional, categorical variables, variational inference, Bayesian, clustering**

I. Introduction

High-dimensional categorical datasets can be challenging to handle because the correlation structure grows exponentially with the number of variables. Consider a questionnaire which has J questions, where each question has R different categories of response. If we collect the responses of I individuals, which are stored in a matrix X with dimensions $I \times J$ and every cell contains one categorical value $(A, B, C, \ldots)$, the correlation structure (i.e. the contingency tensor $\Pi_{R_1 \times \ldots \times R_J}$) grows exponentially with every additional question and becomes too complex to inspect manually for any dataset with more than a handful of variables.

Clustering is a popular approach to identify low-dimensional structures embedded within high-dimensional datasets, but relatively few methods have been proposed to specifically handle the clustering of categorical datasets in comparison to the wealth of methods available for continuous data (for example: [1]–[4]). Our work is motivated by analyses of large-scale population studies such as the Young Lives study [5], the OSMI Mental Health in Tech Survey [6], and the UK National Cancer Patient Experience Survey (NCPES) [7], where large collections of questionnaire data are available for many thousands of individuals. The Young Lives study collects data on childhood poverty over 15 years in four different countries, the OSMI study collected data on the experience with mental health issues in the tech sector, and the NCPES has collected data on the experience and satisfaction of a large number of British cancer patients with the treatment they received by the UK National Health Service (NHS). These studies use questionnaires that are predominantly composed of categorical questions and the general ambition is to be able to identify groups of individuals with similar response profiles. In the case of NCPES, this would enable policy makers to develop strategies to improve the quality of cancer care in the UK.

At present, such analyses are typically performed using

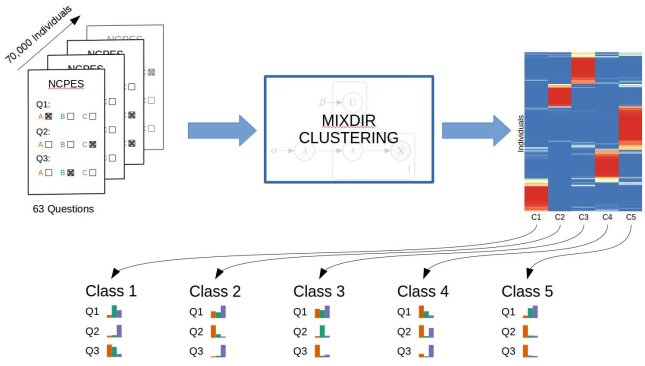

Fig. 1. **Overview of MixDir algorithm**. We have a high-dimensional dataset with categorical values (e.g. the NCPES). We run our MixDir clustering algorithm and obtain a soft clustering of the individuals into five classes. Each class has a particular distribution of values for each question. But this information can still be confusing, so to focus on the most telling response we look can either look at the questions that best explain the clustering or for responses r that maximize $p(z|X_j = r)$. In the example above this would for example be answer B for question 2 which is highly predictive for an individual to be in class 3. But also answer B for question 3 is highly predictive for class 1, although it is not the most common answer in class 1 for question 3.

[1]CRAN.R-project.org/package=mixdir Addional links for reproducing the figures are available at cwcyau.github.io/publications.html

978-1-5386-5090-5/18/$31.00 ©2018 IEEE
DOI 10.1109/DSAA.2018.00068

univariate analyses [7] which attempt to associate responses to individual questions with some outcome of interest. This approach limits the ability to identify complex, multivariate response patterns that may manifest as a *joint probability distribution* over responses to a number of questions. In the following, we first summarize the pre-existing approaches for clustering of high-dimensional categorical data before proposing a scalable Bayesian latent class model (which we call MixDir) for modelling high-dimensional categorical data. We will demonstrate the utility of the approach for the analysis of the Young Lives, the OSMI and NCPES survey data.

II. Existing work

Existing approaches for clustering of high-dimensional categorical data can be grouped into three approaches: (i) multivariate clustering approaches, (ii) latent class models and (iii) latent mixed membership models.

Multivariate clustering approaches adapt standard clustering techniques for continuous data by specifying similarity distance measures developed for categorical data explicitly. For example, k-mode [8] based methods are a variation of the popular k-means clustering approach [9], where the Euclidean distance is replaced by an alternative distance metric (for example the Hamming distance) and the center of a cluster is not the mean of its member but a vector with the most common feature for each attribute (i.e. the mode of the members) [8]. ROCK (short for RObust Clustering using linKs) performs agglomerative clustering [10]. Similarity is measured by the number of common neighbors of a cluster and in each step the two most similar clusters are merged, until a threshold is reached.

Latent class models (LCM) [11]–[13] are mixture models that assign the set of multivariate categorical observations to a latent class z. The idea is that within each latent class the observed variables are statistically independent. LCMs estimate the class probabilities λ and the probability of observing a particular response for a question conditioned on the latent class. [13] proposed a nonparametric extension of the model, where they use a Dirichlet Process prior [14] for the classes, which they call mixture of product multinomial distributions. Their model allows them to infer an appropriate number of latent classes depending on the dataset. They fit their model using a Gibbs Slice sampling algorithm, but this has two disadvantageous for clustering: first MCMC algorithms do not scale to large datasets and second they suffer from the label switching problem [15]. To address those issues we have developed a variational inference (VI) method to estimate the parameters in the basic latent class model and its nonparametric extension. VI does not randomly sample from the posterior, but solves an optimization problem of fitting the complex posterior, by approximating it with a manageable distribution. This is much faster and has the additional advantage that it converges to a unique solution for clustering, where the labels of the clusters are interchangeable. A recent work by [16] demonstrated that the variational approximation of tempered posteriors is consistent for mixtures of Gaussian and simple multinomial distributions.

It is important to distinguish our model from the mixed membership models [17], which are related but not identical to our model. In text processing mixed membership models are also called latent Dirichlet allocation (LDA) [18]. Mixed membership models differ from latent class models because they assume that every response from an individual can come from different latent classes. In LCMs each response of one individual must come from the same latent class. This means that the mixed membership model is more flexible, which can be helpful if for example a text document discusses multiple topics, but on the other hand can complicate the interpretation, because it sets the focus on the questions and not on the individuals.

III. Model

We now propose a variant of the LCM structure where we want to cluster the individuals into K classes depending on their answers. Our model can be summarized as follows:

$$\boldsymbol{\lambda}|\alpha \sim \text{Dirichlet}(\alpha) \text{ or } \text{DirichletProcess}(\alpha) \quad (1)$$

$$z_i|\boldsymbol{\lambda} \sim \text{Multinomial}(\boldsymbol{\lambda}) \quad (2)$$

$$U_{j,k}|\beta \sim \text{Dirichlet}(\beta) \quad (3)$$

$$X_{i,j}|U_j, z_i = k \sim \text{Multinomial}(U_{j,k}). \quad (4)$$

α and β are hyper-parameters that are defined externally and govern the sparsity of the model. Eq. 1 defines that the size of the classes is governed by a Dirichlet (in the case of a simple LCM) or by a Dirichlet Process (in the case of a nonparametric LCM); for now we will describe the derivation for the simple LCM and will later present how to extend it to the nonparametric case. $\boldsymbol{z}$ is a vector that contains the latent class assignment for each individual. U is a 3-way tensor of size $J \times K \times R$ and contains the probability for response r from an individual from class k for question j. Eq. 4 specifies that the response of an individual i that belongs to class k is a draw from a Multinomial distribution according to the probability vector $U_{j,k}$.

The joint distribution of the model is defined as follows

$$p(\boldsymbol{\lambda}, \boldsymbol{z}, U, X|\alpha, \beta) = p(\boldsymbol{\lambda}|\alpha) \prod_{i=1}^{I} p(z_i|\boldsymbol{\lambda}) \prod_{j=1}^{J} \prod_{k=1}^{K} p(U_{j,k}|\beta) \times \prod_{i=1}^{I} \prod_{j=1}^{J} \prod_{k=1}^{K} p(X_{i,j}|U_{j,k})^{\mathbb{1}(z_i=k)}. \quad (5)$$

and Figure 2 shows the plate notation of the model.

Finding the maximum likelihood solution would, for this model result, in an EM algorithm similar to the one described by [12], but to properly propagate uncertainty through the model and to be able to infer an appropriate number of latent classes, we develop a variational inference method that can address those challenges.

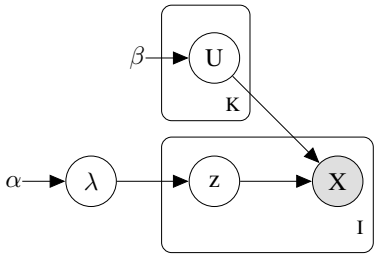

Fig. 2. **The latent class model in plate notation.** Each node represents a random variable and X is shaded gray because it is the only variable which is observed. The arrows represent the dependency structure and the plates represent repeated values. K means that we have one U_k for each cluster 1 to K. Analogous we have one cluster assignment and the corresponding set of observation for each individual 1 to I. λ is the cluster size proportion and drawn from a $\mathrm{Dirichlet}(\alpha)$.

A. Variational Inference

The idea of VI is to define a simplified probability model q and tune its parameters to approximate the original model p. We choose q as the mean field approximation of p, which allows us to write down the variational distribution:

$$\begin{aligned} q(\boldsymbol{\lambda}, \boldsymbol{z}, U) &= q(\boldsymbol{\lambda}) q(\boldsymbol{z}) q(U), \\ q(\boldsymbol{\lambda}, \boldsymbol{z}, U) &= q(\boldsymbol{\lambda}; \boldsymbol{\omega}) \prod_{i=1}^{I} q(z_i; \zeta_i) \prod_{k=1}^{K} \prod_{j=1}^{J} q(U_{j,k}; \phi_{j,k}) \end{aligned} \tag{6}$$

where $\boldsymbol{\omega}$, ζ and ϕ are the free variational parameters, that are subsequently optimized. We also define that

$$\begin{aligned} q(\boldsymbol{\lambda}; \boldsymbol{\omega}) &= \mathrm{Dirichlet}(\boldsymbol{\omega}) \\ q(z_i = k; \zeta_i) &= \zeta_{i,k} \\ q(U_{j,k}; \phi_{j,k}) &= \mathrm{Dirichlet}(\phi_{j,k}). \end{aligned} \tag{7}$$

Using this definition we can derive the update equations for the variational parameters (Appendix A). We will measure the approximation with the KL-divergence, which allows us to maximize the evidence lower bound (ELBO). We find that iterating between the following equations maximizes the ELBO and thus also minimizes the KL divergence:

$$\omega_k = \alpha + \sum_{i=1}^{I} \zeta_{i,k}. \tag{8}$$

$$\begin{aligned} \zeta_{i,k} \propto \exp \Bigg(& \psi(\omega_k) - \psi\Big(\sum_{k=1}^{K} \omega_k \Big) \\ & + \sum_{j=1}^{J} \Big[\psi(\phi_{j,k,X_{i,j}}) - \psi\Big(\sum_{r=1}^{R_j} \phi_{j,k,r} \Big) \Big] \Bigg), \end{aligned} \tag{9}$$

$$\phi_{j,k,r} = \beta + \sum_{i=1}^{I} \zeta_{i,k} \mathbb{1}[X_{i,j} = r]. \tag{10}$$

The eq. 8 and 10 have an intuitive interpretation. They are just the weighted number of individuals per class and the weighted number of observation with a particular feature, respectively. Note that $\psi(\omega_k)$ in eq. 9 is the digamma function.

B. Nonparametric extension

The strength of the latent class models is that it is straightforward to extend them to more complicated settings. For example if the true number of latent classes K is not known, one can use an approximation that assumes a potentially infinite number of classes of which only a finite number is ever observed for a finite number of observations. Mathematically this is expressed with a Dirichlet Process.

A constructive interpretation of the Dirichlet Process is the stick breaking process, which is very helpful as it allows us to construct a truncated approximation where we stop after making K_{max} breaks [19]. We apply this truncated stick breaking process as a prior for λ to give $\lambda_k = v_k \prod_{k'=1}^{k-1} (1 - v_{k'})$.

As already mentioned each v_k is drawn from a Beta distribution $q(v_k; \kappa_{k,1}, \kappa_{k,2}) = \mathrm{Beta}(\kappa_{k,1}, \kappa_{k,2})$ where $\kappa_{k,1}$ and $\kappa_{k,2}$ are the variational parameters that are optimized in the Dirichlet Process instead of the ω_k in the simple Dirichlet model.

The new joint distribution for this model thus is

$$\begin{aligned} p(\boldsymbol{\lambda}, \boldsymbol{z}, U, X | \alpha, \beta) = & \prod_{k=1}^{K_{max}-1} p(v_k|\alpha) \prod_{i=1}^{I} p(z_i|\boldsymbol{\lambda}) \\ & \times \prod_{j=1}^{J} \prod_{k=1}^{K} p(U_{j,k}|\beta) \\ & \times \prod_{i=1}^{I} \prod_{j=1}^{J} \prod_{k=1}^{K} p(X_{i,j}|U_{j,k})^{\mathbb{1}(z_i=k)}, \end{aligned} \tag{11}$$

which differs from eq. 5 in the sense that the first term has been replaced with the truncated stick breaking formulation.

We derive the updates for the free variational parameters (Appendix B) and find that iteratively running the following equations maximizes the ELBO for the nonparametric model.

$$\kappa_{k,2} = \alpha_2 + \sum_{i=1}^{I} \sum_{k'=k+1}^{K_{max}} \zeta_{i,k'}, \;\; \kappa_{k,1} = \alpha_1 + \sum_{i=1}^{I} \zeta_{i,k}, \tag{12}$$

$$\begin{aligned} \zeta_{i,k} \propto \exp \Bigg(& \psi(\kappa_{k,1}) - \psi(\kappa_{k,1} + \kappa_{k,2}) \\ & + \sum_{k'=1}^{k-1} [\psi(\kappa_{k',2}) - \psi(\kappa_{k',1} + \kappa_{k',2})] \\ & + \sum_{j=1}^{J} \Big[\psi(\phi_{j,k,X_{i,j}}) - \psi\Big(\sum_{r=1}^{R_j} \phi_{j,k,r} \Big) \Big] \Bigg) \end{aligned} \tag{13}$$

The update equation for $\phi_{j,k,r}$ (eq. 10) does not differ from the one in the parametric model, but the updates for the Dirichlet Process parameters and $\zeta_{i,k}$ change.

C. Handling missing data

We can reformulate the joint distribution of eq. 5 to incorporate missing data

$$p(\boldsymbol{\lambda}, \boldsymbol{z}, U, X|\alpha, \beta) = p(\boldsymbol{\lambda}|\alpha) \prod_{i=1}^{I} p(z_i|\boldsymbol{\lambda}) \prod_{j=1}^{J} \prod_{k=1}^{K} p(U_{j,k}|\beta) \times \prod_{k=1}^{K} \prod_{(i,j)\in S^o} p(X^o_{i,j}|U_{j,k})^{\mathbb{1}(z_i=k)} \times \prod_{k=1}^{K} \prod_{(i,j)\in S^m} p(X^m_{i,j}|U_{j,k}, z_i = k)$$

where S^o is the set answer that were observed and S^m is the set of answers that are missing for each individual.

If we assume that the data is missing completely at random (MCAR), which means that the chance of missing a value is unrelated to the latent class, the unobserved answer or any other previous answer, then $p(X^m_{i,j}|U_{j,k}, z_i = k) = \text{const}$. The estimation of the free variational parameters is thus independent of the missing values and they can be skipped during the variational updates. To impute the missing values, one would simply draw a latent class based on the observed data and draw replacements for the missing values from $p(X^m_{i,j}|U_{j,k}, z_i = k)$. If on the other hand we believe that the missingness of a data point contains useful information for the inference, it is best recoded as an additional possible response r.

IV. Applications

In this section we first want to analyze the performance of our proposed algorithm using a simulation study and then we used the temporal consistency of the inferred clusters of the Young Lives survey as a real-world example. Lastly, we apply our model to analyze the latent structure of the OSMI Mental Health in Tech and the 2015 UK National Cancer Patient Experience survey.

A. Simulation study

To demonstrate that that our algorithm is able to identify latent structure in a high-dimensional dataset and to give an idea how it performs compared to other clustering algorithms for categorical data, we generated a dataset with a known latent structure. First, we generate four latent classes and for each class a prototypical list p_k of length 5. Then we assign each individual randomly to one of the four classes. Each element in this list p_k is a vector of length R, which contains the proportions to draw response r if an individual belongs to class k. Most of the entries have a roughly equal chance for each response or just a slight bias, but a few of them are highly specific for one class. Those are the ones that need to be picked up by a methods to produce a good clustering result. In each experimental run we vary the signal to noise ratio of the dataset, to test the performance with a range different settings.

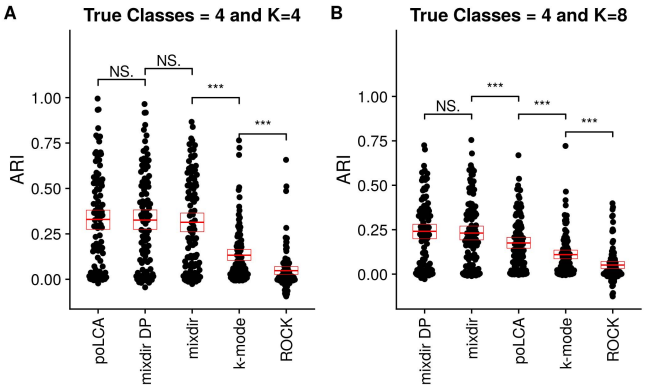

Fig. 3. **Performance comparison on a synthetic dataset** of the k-mode, ROCK, the EM algorithm for the latent class model (poLCA) and our implementation with a Dirichlet Process prior (mixdir DP) and the simple Dirichlet prior (mixdir). The performance is measured with the adjusted Rand index (ARI) that calculates the overlap between the inferred clustering and the ground truth on the synthetic dataset. The significance test is a two-sided paired Wilcoxon rank sum test and NS. indicates a p-value > 0.05, one star indicates $p > 0.01$. two stars $p > 0.001$ and three stars indicate $p < 0.001$. The red box shows the mean and the bootstrapped confidence limits. The algorithms are tested in two settings one where K is the correct number of latent classes in the model (A) and one where K is an overestimate of the number of latent classes (B).

We compared the parametric and nonparametric variants of MixDir algorithm using the Dirichlet and Dirichlet Process priors respectively (the latter we refer to as MixDir-DP) with three other algorithms: the ROCK algorithm [20], one for k-mode [21] and an EM inference implementation of the latent class model [12] called poLCA. We chose these three algorithms, because they were all readily available as packages for the popular and widely used `R` statistical computing platform [22], which is also the platform we used for implementing our algorithm. This should be kept in mind when comparing for example the runtimes of the algorithms, where a bad performance could just be explained by an inefficient implementation. In terms of time complexity with respect to the number of observations n ROCK has a worst case time complexity of $O(n^2 \log(n))$. The k-mode algorithm is linear to n, as are the three latent class models. Interestingly although ROCK has the worst theoretical time complexity of the compared methods, it consistently ran the fastest.

We measure the performance of the clustering algorithms using the adjusted Rand index (ARI) [23], which is a popular measure for comparing two clustering results. In our case we compare the proposed clustering of each algorithm to the ground truth. The ARI is 0 when the proposed clustering is as good as random and 1 if the algorithm recovered the ground truth. It is important to note that our algorithm produces a probabilistic output for each observation to belong to each class (also called soft clustering). For comparison with the other methods we assign each individual to the latent class for which it has the highest probability.

Figure 3 shows the performance on 100 differently initial-

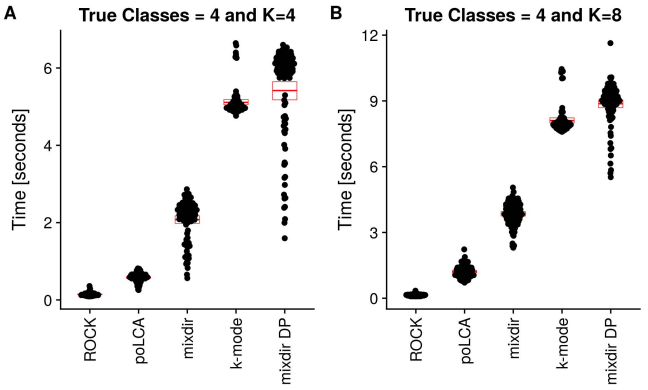

Fig. 4. **Run-time analysis** of the k-mode, ROCK, the EM algorithm for the latent class model (poLCA) and our implementation with a Dirichlet Process prior (mixdir DP) and the simple Dirichlet prior (mixdir) on a dataset with 1000 observations for 5 questions with 5 to 15 categories each. The red box shows the mean and the bootstrapped confidence limits. The runtime was measured on a 4 year old laptop with a Intel Core i7-3635QM processor.

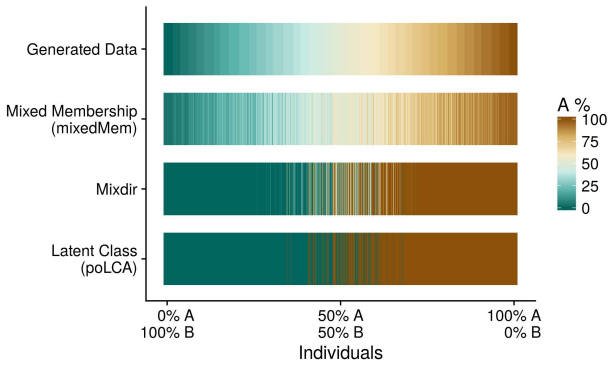

Fig. 5. **Clustering of mixed-membership data using mixdir, poLCA and mixedMem.** The data consist of 2000 individuals with 40 features which are assigned to one of two classes. The top row shows the percentage for each individual how many of its features are assigned to class A.

ized datasets for all five clustering methods with varying signal to noise ratios. We tuned the signal to noise ratios such that we cover the whole range of results for the methods ranging from cases where all methods achieve a decent clustering to cases where none of the methods is able produce a clustering better than random. In Figure 3A we can see that if K is set to the correct number of latent classes the three LCMs outperform k-mode and ROCK and that there is no significant difference between our methods and poLCA. Only if we misspecify K (Figure 3B), something that can easily happen on a real world dataset, we see that our models outperform the other approaches. When we actually look into the inferred clusters we can see that for high performing examples of MixDir with the Dirichlet Process prior the method is able to recover the correct number of four classes even if $K = 8$. Figure 4 shows that the increased performance comes at the cost of an increased runtime.

A particular challenge with the ROCK algorithm is that it relies on a user-defined parameter θ that defines the minimum similarity so that two elements are considered a neighbor. The resulting clustering strongly depends on this parameter, but there is little guidance to choosing it, so we simply used the suggested value of $\theta = 0.5$ in all of the above examples. Our method also has hyper-parameters (we used $\alpha = 1$ and $\beta = 0.1$), but they have less of an effect on the result, especially when a lot of data is provided. The hyper-parameters serve as pseudo-counts in eq. 8 and 10 and thus usually need to be within the same magnitude as the number of observations per latent class and category, respectively, to affect the clustering.

The above test is to a certain extend self serving, because we use the same model to generate the data that we also use to classify it. So it is important to see how the model behaves if the model is misspecified. We will test the performance of our model on data generated from a mixed membership model, which also emphasizes how our model differs. We generate a dataset of 2000 individuals with 40 features that can take one out of three different values. We have two latent classes A and B, but instead of assigning each individual to one of the classes, each feature of every individual is assigned to a class. This means that an individual can be truly a 50/50 mix of class A and B, namely when 20 features are from A and 20 from B. This is the generative model that is assumed by mixed membership models. We cluster this data using MixDir, poLCA and an R implementation for fitting mixed membership models (mixedMem) [24].

Unsurprisingly the mixed membership model performs best in the classification task and is able to recover for nearly every individual the correct fraction of membership in class A and B (Fig. 5). In contrast, the latent class models (poLCA and MixDir) assume that every individual belongs *exclusively* to class A or B. They are still able to classify most individuals correctly whether they are mostly from class A or B, but for individuals with mixed response profiles, poLCA makes some classification errors due to the hard assignments it reports. However, the probabilistic output of MixDir means that individuals with a mixed response profile will receive an uncertain posterior class assignment. This is a good example where the probabilistic nature of our clustering algorithm can be an helpful indicator of model mis-specification.

B. Young Lives

Next, we consider performance using a real-world data set from The Young Lives Survey. This survey is an international study of childhood poverty. It follows children in Ethiopia, India, Peru and Vietnam over 15 years tracking indicators about the health of the children, literacy, wealth of the household and many more indicators. So far four rounds of surveys have been conducted (in 2002, 2006, 2009 and 2013) following the same children from birth to their teens. Initially, we focused on the Ethiopian dataset and specifically the younger cohort with children aged 1, 5, 8 and 12 years in the rounds respectively. We took several steps to clean the data: removal of unique identifiers, binning of continuous variables, removal

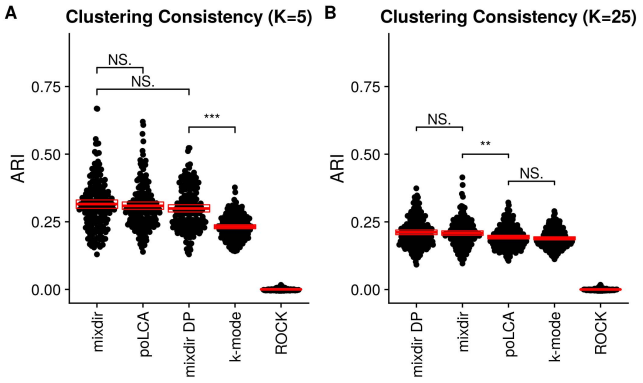

Fig. 6. **Performance comparison on the Young Lives Ethiopia datasets.** The performance is the adjusted Rand index (ARI) between the clusterings of the four rounds and the algorithms were run 25 times on 20% randomly sampled individuals. The significance test is a two-sided paired Wilcoxon rank sum test, NS. indicates a p-value > 0.05 and the stars indicate p-values of > 0.01, > 0.001 and < 0.001. The red box shows the mean and the bootstrapped confidence limits. We once set K to a small number of latent classes (A) and once we to an overestimate of the number of latent classes (B).

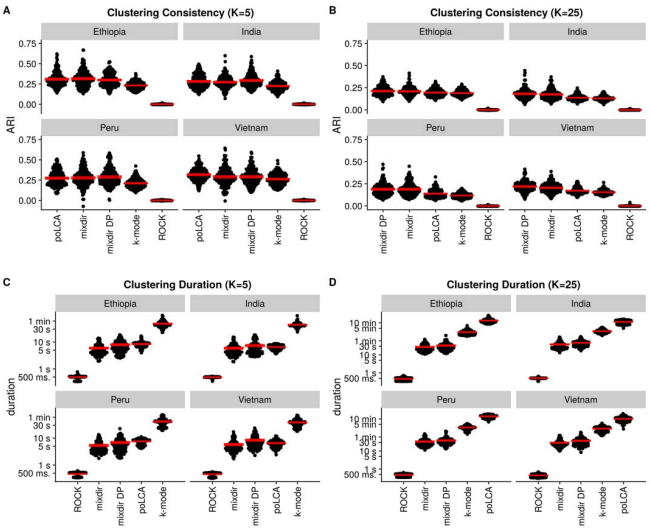

Fig. 7. **Cluster consistency and runtime for all algorithms on all four countries.** The performance of each method is the adjusted Rand index (ARI) between the clustering of the first and second, first and third, first and fourth, second and third, second and fourth, and third and fourth round. The process was run 25 times on 20% randomly sampled individuals. The methods are ordered by mean ARI. The red boxes show the mean and the bootstrapped confidence limits. We compare two different settings $K = 5$ (A, C) and $K = 25$ (B, D), which are likely under- and overestimations of the true number of latent classes in the data. To ensure that ROCK divides the dataset into more than one cluster we had to set $\theta = 0.1$.

of children that dropped out and removal of columns without any variation. In the end we worked with a dataset of about 1200 children, 52 questions and a median 4.5 different answers per question.

We then wanted to compare the temporal consistency of the clusters that different algorithms identify in the four rounds. We use the ID of the children as ground truth and calculate the adjusted Rand index (ARI) [23] between the first and second, first and third, first and fourth, second and third, second and fourth, and third and fourth rounds. To have sufficient statistical power to detect performance differences and ensure consistent results, we repeat the procedure 25 times and each time randomly sample 20% of the individuals. The assumption in this analysis is that the groupings of children across the years do not change dramatically.

Since only MixDir-DP provides a means for *automatically* selecting the number of latent classes, we first considered an analysis where we prefix the number of latent classes to fixed values ($K = 5$ and 25) for all algorithms. With a smaller number of classes, we find that the latent class methods (`poLCA`, `mixdir` and `mixdir DP`) outperform k-mode and ROCK (Figure 6A). With a larger number of classes (i.e. $K = 25$) we find that our algorithm outperforms all other methods including `poLCA` (Figure 6B). To ensure that our result is reproducible we also ran the same experiment on the datasets from India, Peru and Vietnam (Figure 7A,B). In terms of the runtime, we find that on the Young Lives dataset our method outperforms the others (except for ROCK, but which had the worst performance) (Figure 7C,D).

C. OSMI Mental Health in Tech Survey 2016

Open Sourcing Mental Illness (OSMI) is a non-profit corporation that is dedicated to mental wellness in the technology industry. Part of this effort is the collection of data on the state of mental health in the technology sector, including a survey from 2016 with responses of 1,400 individuals. The survey consists of a questionnaire with 63 different questions [6]. We applied MixDir to explore latent structure within this survey data.

Before running the clustering algorithm we needed to clean the data: we removed free-form answers, filtered out self-employed individuals, who were asked different questions, and summarized the responses which mental health disorder individuals had into single consistent values. In the end we worked with a dataset of 1,146 individuals with responses for 46 different questions and about 4% missing values.

The number of latent classes is unknown and our inference of this quantity will be determined by our prior beliefs. We therefore explored a range of hyperparameters for the Dirichlet Process prior with an increasingly penalization on the creation of new classes by setting α_1 to 1, 10, 100 and 1000. We always set $K_{max} = 25$ which is enough to not hamper the fitting but keeps the algorithm tractable. Figure 8A shows an alluvial plot tracing how each individual's class assignment changes as a function of increasing α_1 - the smaller classes are merged as this parameter grows. This visualisation provides a very useful means of understanding how the clustering structures alters as a function of our prior beliefs (encoded in the hyperparameter α_1) allowing us to partially objectify otherwise subjective beliefs about the number of latent classes. We decided to use $\alpha_1 = 100$ for the subsequent analysis because it capture

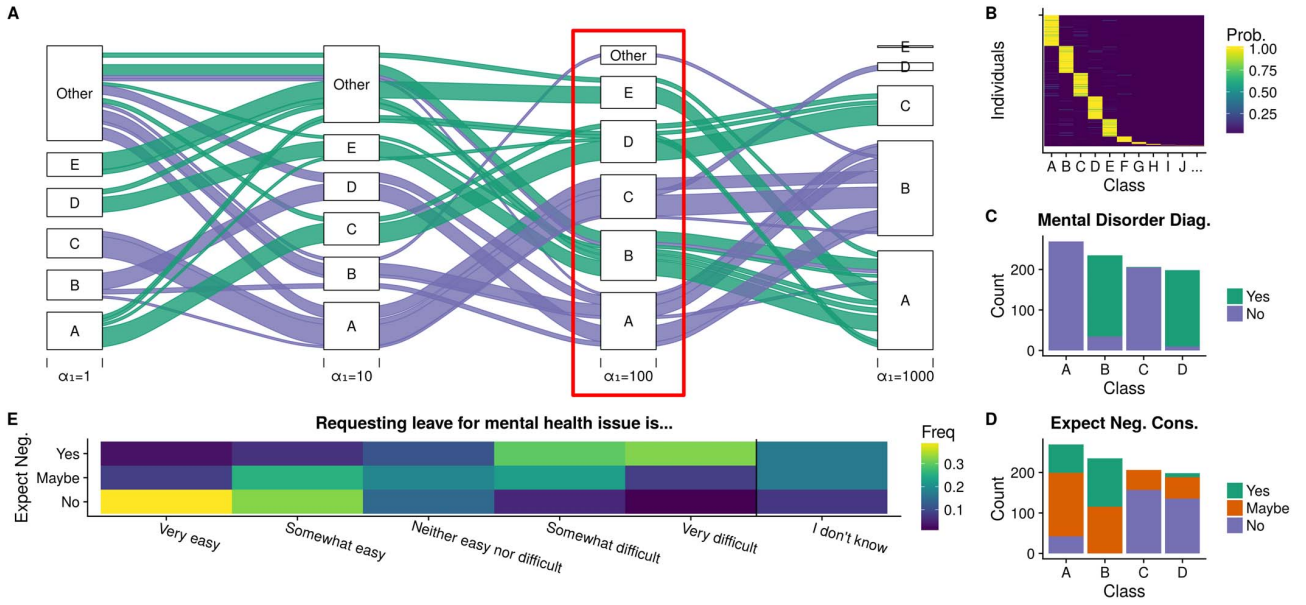

Fig. 8. **Analysis of the 2016 OSMI Mental Health in Tech Survey.** A is an alluvial plot that shows the effect of clustering the data with an increasing level of α_1. If a mental health disorder has been diagnosed the band is colored green, if not purple. Flows with less than 10 individuals are suppressed to reduce the visual clutter. Individuals that were assigned to classes bigger than E are summarized in the "Other" block. The red rectangle highlights the parameter setting $\alpha_1 = 100$ that was used in the other plots B-E for the more detailed analysis. B shows the class assignment probabilities for each individual. C is a bar plot that shows how many individuals in the first 4 classes have a diagnosis of a mental disorder. D is a bar plot that shows how many individuals in the first 4 classes answered the question "Do you think that discussing a mental health disorder with your employer would have negative consequences?". E is a plot of the contingency table for the questions "If a mental health issue prompted you to request a medical leave from work, asking for that leave would be:" and "Do you think that discussing a mental health disorder with your employer would have negative consequences?".

the main structure of the data, with the majority of survey participants grouped into five main classes. Note that this is *not* a statement that there are in fact five latent classes.

We focus on the four biggest classes (A, B, C and D) which together cover 88% of the individuals (Figure 8B). We find that two groups (B and D) mainly consist of individuals that answered the question "Have you been diagnosed with a mental health condition by a medical professional?" with "Yes" and two groups (A and C) with individuals that mainly answered the question with "No" (Figure 8C). It is of course interesting that the algorithm creates two groups (A and C vs. B and D) for the major phenotypic characteristic. To find out what is the main difference between A and C, and B and D, we look at the predictive features for each of the classes (the question-answer pairs that maximize the probability for class k: $\text{argmax}_{X_j=r}\, p(z = k|X_j = r)$). We notice that for group C and D answering the question "Do you think that discussing a mental health disorder with your employer would have negative consequences?" with "No" is a predictive feature ($p = 41\%$ and $p = 29\%$, respectively), whereas answering with "Yes" is predictive for group A and B ($p = 35\%$ and $p = 56\%$, respectively) (Figure 8D). This suggests, that there are differences between the individuals who expect that being open about their mental health disorder will have negative consequences.

We looked for other features about the employer that differ by the expectation of discussing mental health and found, for example, that people are less likely to expect negative consequences if the employer has mental health care under the employer-provided coverage (Chi-squared test $p = 0.0061$). We also found that people have more negative expectations about asking for mental health-related leave, if their general expectations about discussing mental health with their employer is negative (Wilcoxon rank-sum test $p < 2.2 \times 10^{-16}$, Figure 8E).

Interestingly post-traumatic stress disorder (PTSD) is also a predictive feature for group B. This leads us to propose the hypothesis that individuals who are affected by PTSD, might have a more negative expectation about the consequences of discussing it with their employer. We check the hypothesis with a Fisher's exact test and reject the null hypothesis that individuals with PTSD have the same expectation as individuals with other mental disorder ($p = 0.0417$). On the other hand having a diagnosis for attention deficit hyperactivity disorder (ADHD) is a predictive feature for group D, but we did not find a significant relation of ADHD with having less negative expectations (Fisher's exact test $p = 0.6341$). This is a good example how the unsupervised clustering can help uncover interesting underlying structures, but on the other hand one must be careful not to over-interpret the data and check if trends can be confirmed with the whole dataset.

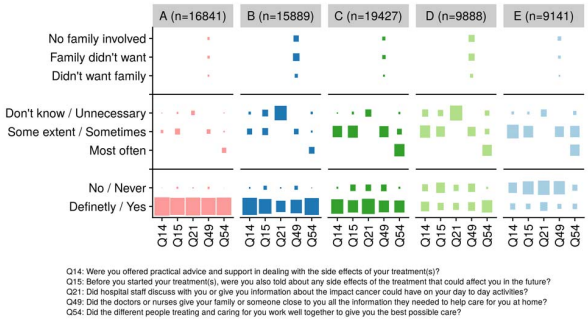

Fig. 9. **Defining features plot of the NCPES dataset, with the questions that are most informative for distinguishing the inferred clusters.** The area of each square indicates the fraction of individuals in a group that answered the question a specific way. The sum of the area of all squares in each column is 1. In the original questionnaire the answers for the question 14, 15, 21, 49 and 54 differed slightly in their formulation, so we grouped them into common categories. The vertical bars have no additional meaning and only serve to enhance the visual clarity of the plot.

D. 2015 National Cancer Patient Experience Survey (NCPES)

Lastly, we wanted to analyze the 2015 NCPES [7] to demonstrate the usefulness of our algorithm on a big dataset. The NCPES is an annual survey that has been conducted since 2010 and is commissioned by the British National Health Service (NHS) to monitor the state of care for cancer patients. The data consists of the responses from 71,186 individuals and has additional data about the gender, the tumor and the age of the patient. The data available from the questionnaire consists of 67 questions, of which all but three are categorical single choice questions. Those three that are not, were ignored in the subsequent analysis. The dataset also has a considerable number of missing values, in total more than 16% of all entries are not available. The NCPES is an interesting example of a high-dimensional categorical dataset, but so far most of the analysis has focused predominantly on univariate features of the dataset. Researchers have looked on the distribution of responses for individual questions (e.g. "87% of respondents said that, overall, they were always treated with dignity and respect while they were in hospital." [7]).

Ideally we would want to infer an adequate number of latent classes on this dataset, but due to its inherent complexity and size, this would lead to too many classes for any manual downstream analysis (Figure 10). For brevity in this paper, we decided for the sake of simplicity and interpretability to analyze the dataset using the simple Dirichlet prior with $K = 5$. We run our algorithm multiple times to check if the clusterings are consistent and find that the average agreement between ten iterations is $\mathrm{ARI} = 0.998$.

The challenge with this dataset is that it contains such a large number of questions, which makes it difficult to decide which are the interesting variables that are important for the clustering. To find a manageable number of questions, we reduce the dimensionality of the dataset. We iteratively remove variables and test how much this affects the predicted clustering. We measure the loss of information using the

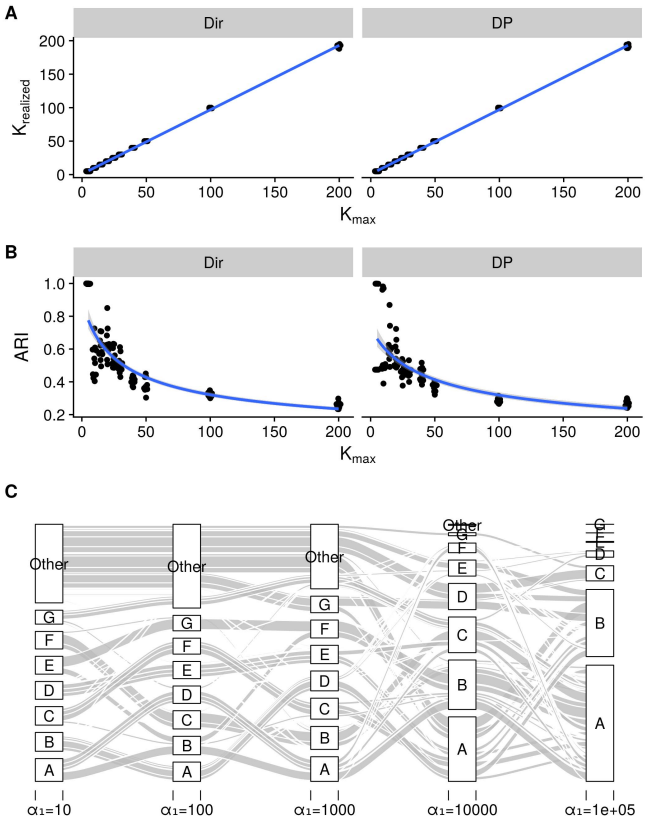

Fig. 10. **Challenges of inferring the number of classes on the NCPES data.** A shows the number of realized clusters for the MixDir model with a Dirichlet or a Dirichlet Process prior, depending on the maximum number of possible clusters. B shows the adjusted Rand index between the relicates depending on the maximum number of available clusters. The blue line shows a linear and an inverse Gaussian model. D shows an alluival plot demonstrating the effect of increasing α_1 with a Dirichlet Process prior.

Jensen-Shannon divergence on the predicted and the original class probability matrix at each step and remove the variable that least affects the clustering. This way we can narrow down the original set of 63 questions to five which are the most informative for distinguishing the clusters (Figure 9).

We find interesting distinct groups: cluster A is the second largest group and contains individuals that answered very positively throughout all 5 questions, cluster B is still mostly positive, but the answers are more nuanced, cluster C is the largest cluster and is defined by somewhat positive responses (e.g. "Yes, to some extent" or "Yes, some of the time"), cluster D is more negative and is also defined by individuals answering "Don't know", lastly, cluster E is the smallest cluster and the most negative with people answering the questions more often negative than positive. To validate that the overall satisfaction is a major driver of the clustering, we look at question 59, which asks the individuals to rate their overall experience from 1 to 10. We removed this question during the data cleaning, because it is not categorical, and can thus use it to demonstrate that we were nonetheless able to

recover this information. We perform an Wilcoxon rank sum test on the ratings comparing that $\text{ratings}(A) > \text{ratings}(B) > \text{ratings}(C) > \text{ratings}(D) > \text{ratings}(E)$, which is in all four cases highly significant ($p < 2.2 \times 10^{-16}$).

An important feature of our method is that it produces probabilistic assignments of individuals to the latent classes. As we just described in the case of the NCPES we find that the latent classes have a linear relationship, so one can easily imagine that some individuals might be in between two classes, but are rarely considered a mix of more than two classes. Accordingly we find that less than 4% of the individuals have more than 10% probability for at least three classes.

When we focus on question 49 which asks about the involvement of the family and/or friends, we can see that they were less involved in clusters C, D and E. To see if indeed missing involvement of the family leads to less overall satisfaction, we test if overall individuals which answered "Yes" or "Yes, to some extend" were more satisfied than individuals that answered "No" or "No family or friends were involved" (Wilcoxon rank sum test $p < 2.2 \times 10^{-16}$). On the other hand this needs to be qualified because individuals that deliberately decided against involvement of their family are overall more satisfied than individuals that just stated their social network was not involved (Wilcoxon rank sum test $p < 2.2 \times 10^{-16}$). This underlines the importance of social networks during cancer treatment, but on the other hand which role the ability to make deliberate choices can play.

To summarize, we are able to cluster the large NCPES dataset and uncover interesting latent structure. We identify the overall satisfaction as a major underlying feature of the dataset and show how it can be related to the support patients get from their family and friends. This demonstrates that our algorithm can be a useful tool for handling large and high-dimensional categorical datasets.

V. Discussion

There is no universally best clustering technique without context, but we find that our method has several desirable features. It can deal with large datasets of more than 70,000 observations, it has a principled approach to handle missing data thanks to the Bayesian framework and it can handle datasets where the true number of latent classes is not known. We developed two related versions of the algorithm, one for a finite number of classes and the nonparametric version where we assume that the number of latent classes keeps increasing as long as gather more observations. One limitation is that for the analysis of the performance of the different methods using simulations, we only quantified it using data generated from a model that has the same independence assumptions as the model we developed here. Another limitation could be that for the nonparametric extension we use a Dirichlet Process prior, which has the known problem of overestimating the true number of latent classes [25]. This issue should be kept in mind, but in our experience this has not been a problem for the datasets we looked at.

In this paper we have presented a variational inference algorithm for Bayesian latent class models and their nonparametric extension. We demonstrate on high-dimensional categorical data that our clustering algorithm is able infer good results on synthetic and real world datasets. We also show that its performance is comparable to the best competitor (poLCA) if the correct number of latent class is known *a priori*, and actually outperforms the other methods if the number of classes is not known, which is a common problem.

Acknowledgment

The authors acknowledge the support of the UK Medical Research Council Grant No. MR/P02646X/1. Some of the data used in this study comes from Young Lives, a 15-year study of the changing nature of childhood poverty in Ethiopia, India, Peru and Vietnam (www.younglives.org.uk). Young Lives is funded by UK aid from the Department for International Development (DFID). The views expressed here are those of the author(s). They are not necessarily those of Young Lives, the University of Oxford, DFID or other funders. This paper provides a secondary analysis of data obtained through the UK Data Service for the Young Lives Study (7483) and NCPES (8163) data.

References

[1] D. Arthur and S. Vassilvitskii, "K-Means++: the Advantages of Careful Seeding," *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, vol. 8, pp. 1027–1025, 2007.

[2] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[3] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[4] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An Efficient Data Clustering Databases Method for Very Large," *ACM SIGMOD International Conference on Management of Data*, vol. 1, pp. 103–114, 1996.

[5] J. Boyden, T. Woldehanna, S. Galab, A. Sanchez, M. Penny, and L. Duc, "Young Lives: an International Study of Childhood Poverty: Round 4, 2013-2014." *UK Data Service*, 2016.

[6] Open Sourcing Mental Illness Ltd., "OSMI Mental Health in Tech Survey," 2016. [Online]. Available: https://osmihelp.org/research

[7] NHS England Quality Health, "National Cancer Patient Experience Survey, 2015," pp. –, 2017.

[8] Z. Huang, "Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values," *Data Mining and Knowledge Discovery*, vol. 2, no. 3, pp. 283–304, 1998. [Online]. Available: http://link.springer.com/article/10.1023/A:1009769707641

[9] J. Macqueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 233, pp. 281–297, 1967.

[10] S. Guha, R. Rastogi, and K. Shim, "ROCK: A Robust Clustering Algorithm for Categorical Attributes," *Information Systems*, vol. 25, no. 5, pp. 345–366, 2000.

[11] P. Lazarsfeld and N. Henry, *Latent structure analysis*. Houghton Mifflin, Boston, 1968.

[12] D. A. Linzer and J. B. Lewis, "poLCA: An R Package for Polytomous Variable Latent Class Analysis," *Journal of Statistical Software*, vol. 42, no. 10, pp. 1–29, 2011.

[13] D. B. Dunson and C. Xing, "Nonparametric Bayes Modeling of Multivariate Categorical Data," *Journal of the American Statistical Association*, vol. 104, no. 487, pp. 1042–1051, 2009. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1198/jasa.2009.tm08439

[14] T. S. Ferguson, "A Bayesian Analysis of Some Nonparametric Problems," *Annals of Statistics*, vol. 1, no. 2, pp. 209–230, 1973.

[15] M. Stephens, "Dealing with label switching in mixture models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 62, no. 4, pp. 795–809, 2000. [Online]. Available: http://doi.wiley.com/10.1111/1467-9868.00265
[16] B.-E. Chérief-Abdellatif and P. Alquier, "Consistency of Variational Bayes Inference for Estimation and Model Selection in Mixtures," no. arxiv, pp. 1–33, 2018. [Online]. Available: http://arxiv.org/abs/1805.05054
[17] E. M. Airoldi, D. M. Blei, E. A. Erosheva, and S. E. Fienberg, "Introduction to Mixed Membership Models and Methods," *Handbook of Mixed Membership Models and Their Applications*, vol. c, pp. 3–14, 2014.
[18] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Macine Learning Research*, vol. 3, no. 4-5, pp. 993–1022, 2003.
[19] D. M. Blei and M. I. Jordan, "Variational inference for Dirichlet process mixtures," *Bayesian Analysis*, vol. 1, no. 1 A, pp. 121–144, 2006.
[20] C. Buchta and M. Hahsler, "cba: Clustering for Business Analytics," 2017. [Online]. Available: https://cran.r-project.org/package=cba
[21] C. Weihs, U. Ligges, K. Luebke, and N. Raabe, "klaR Analyzing German Business Cycles," in *Data Analysis and Decision Support*, D. Baier, R. Decker, and L. Schmidt-Thieme, Eds. Berlin: Springer-Verlag, 2005, pp. 335–343.
[22] R Core Team, "R: A Language and Environment for Statistical Computing," Vienna, Austria, 2017. [Online]. Available: https://www.r-project.org/
[23] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.
[24] Y. S. Wang and E. A. Erosheva, "Fitting Mixed Membership Models using mixedMem," pp. 1–21, 2015.
[25] J. Miller and M. Harrison, "A simple example of Dirichlet process mixture inconsistency for the number of components," *Advances in Neural Information Processing ...*, vol. 1, pp. 1–8, 2013.
[26] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational Inference: A Review for Statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.
[27] H. Ishwaran and L. James, "Markov chain Monte Carlo in approximate Dirichlet and beta two parameter process hierarchical models," *Biometrika*, vol. 83, pp. 371–390, 2000.

APPENDIX A
VARIATIONAL INFERENCE DERIVATION

In this section we want to give a short introduction to variational inference (VI) and the explicit derivation of the updates for the variational parameters.

VI is an approximate method to do inference in Bayesian models [3], [26]. It is an alternative to the well known MCMC algorithms that randomly sample from the model until the stationary distribution of the samples correspond to the posterior of the model and the data. Instead VI converts the inference problem into an optimization problem, which can be solved much more efficiently.

In the Bayesian framework we are interested in learning about the distribution of the parameters given the observed data. Mathematically this can be written as

$$p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{z},\boldsymbol{x})}{p(\boldsymbol{x})}, \tag{14}$$

where $\boldsymbol{z}$ are all the parameters of the model and $\boldsymbol{x}$ is the data. This is just a reformulation of the famous Bayes rule and means that the conditional distribution equals the joint distribution of data and parameters divided by the marginal $p(\boldsymbol{x})$. Calculating this marginal is the big challenge in Bayesian inference because to calculate the probability of observing a particular dataset $\boldsymbol{x}$ one would need to consider all possible configurations of the parameters $\boldsymbol{z}$. Or again in mathematical notation

$$p(\boldsymbol{x}) = \int p(\boldsymbol{z},\boldsymbol{x})\,\mathrm{d}\boldsymbol{z}. \tag{15}$$

Only for very simple models it is possible to calculate this integral analytically, for complex models it is necessary to find approximations for this integral for example with MCMC or VI.

In VI we choose a family $\mathfrak{F}$ of distributions, which is easier to handle, and try to find a setting where our approximate distribution $q(\boldsymbol{z}) \in \mathfrak{F}$ is as close as possible to the posterior $p(\boldsymbol{z}|\boldsymbol{x})$. The closeness is measured with the Kullback-Leibler (KL) divergence

$$\begin{aligned}\mathrm{KL}(q(\boldsymbol{z})\|p(\boldsymbol{z}|\boldsymbol{x})) &= \int q(\boldsymbol{z}) \log \frac{q(\boldsymbol{z})}{p(\boldsymbol{z}|\boldsymbol{x})}\,\mathrm{d}\boldsymbol{z} \\ &= \mathbb{E}[\log q(\boldsymbol{z})] - \mathbb{E}[\log p(\boldsymbol{z}|\boldsymbol{x})]\end{aligned} \tag{16}$$

where all expectations $\mathbb{E}$ are taken with respect to $q(\boldsymbol{z})$. The KL divergence is not symmetric and favors $q(\boldsymbol{z})$ to be smaller and to underestimate the variance of $p(\boldsymbol{z}|\boldsymbol{x})$, but on the other hand it is has nice mathematical properties that makes it useful for approximating complex models.

As discussed earlier the term $p(\boldsymbol{z}|\boldsymbol{x})$ in eq. 16 is usually not available, but we can re-arrange the equation so that it is not needed:

$$\begin{aligned}\mathrm{KL}(q(\boldsymbol{z})\|p(\boldsymbol{z}|\boldsymbol{x})) = {}& \mathbb{E}[\log q(\boldsymbol{z})] - \mathbb{E}[\log p(\boldsymbol{z},\boldsymbol{x})] \\ &+ \log p(\boldsymbol{x}) \\ -\mathrm{KL}(q(\boldsymbol{z}\|p(\boldsymbol{z}|\boldsymbol{x})) + \log p(\boldsymbol{x}) &= \mathbb{E}[\log p(\boldsymbol{z},\boldsymbol{x})] - \mathbb{E}[\log q(\boldsymbol{z})] \\ \mathrm{ELBO}[q] &= \mathbb{E}[\log p(\boldsymbol{z},\boldsymbol{x})] - \mathbb{E}[\log q(\boldsymbol{z})] \\ \mathrm{ELBO}[q] &= \mathbb{E}[\log p(\boldsymbol{z},\boldsymbol{x})] + \mathbb{H}[q(\boldsymbol{z})],\end{aligned} \tag{17}$$

where ELBO is short for evidence lower bound and $\mathbb{H}$ is the entropy of a function ($\mathbb{H}[p] = -\int p(x)\log p(x)\,\mathrm{d}x$). Looking at eq. 17 we can see that maximizing the ELBO is equivalent to minimizing the KL divergence up to an additive constant. So the goal of VI is to find $q^*(\boldsymbol{z})$ such that

$$q^*(\boldsymbol{z}) = \underset{q(\boldsymbol{z})\in\mathfrak{F}}{\operatorname{argmin}}\,\mathrm{KL}(q(\boldsymbol{z})\|p(\boldsymbol{z}|\boldsymbol{x})) = \underset{q(\boldsymbol{z})\in\mathfrak{F}}{\operatorname{argmax}}\,\mathrm{ELBO}[q]. \tag{18}$$

In theory all kind of distributions $\mathfrak{F}$ could be applicable here, but in practice the one that is most commonly chosen in VI is the so called mean field variational family. It assumes that the latent variables are all independent, so that $q(\boldsymbol{z})$ factorizes to

$$q(\boldsymbol{z}) = \prod_{j=1}^{m} q_j(z_j) \tag{19}$$

and each density $q_j(z_j)$ can be chosen independently to maximize the ELBO. Our factorization of $q(\boldsymbol{z})$ is provided in eq. 6.

Now we can start to derive the update equations. First we will write down the expectation of the joint $\mathbb{E}[\log p(\boldsymbol{z}, \boldsymbol{x})]$ from eq. 17

$$\begin{aligned}\mathbb{E}\left[\log p(\boldsymbol{z}, \boldsymbol{x})\right] = &\mathbb{E}\left[\log p(\boldsymbol{\lambda}|\alpha)\right] \\ &+ \sum_{i=1}^{I} \mathbb{E}\left[\log p(z_i|\boldsymbol{\lambda})\right] \\ &+ \sum_{j=1}^{J}\sum_{k=1}^{K} \mathbb{E}[\log p(U_{j,k}|\beta)] \\ &+ \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} \mathbb{E}[\log p(X_{i,j}|U_{j,k}, z_i = k)]\end{aligned} \tag{20}$$

The first line of equation eq. 20 is simply the expectation of a log Dirichlet distribution, which we can look up in [18] as

$$\begin{aligned}\mathbb{E}\left[\log p(\boldsymbol{\lambda}|\alpha)\right] = &\log\Gamma\Big(\sum_k \alpha_k\Big) \\ &- \sum_k \log\Gamma(\alpha_k) \\ &+ \sum_k (\alpha_k - 1)\Big(\psi(\omega_k) - \psi\Big(\sum_k \omega_k\Big)\Big)\end{aligned} \tag{21}$$

where $\psi(\omega_k)$ is the digamma function.

The expectation in the second line of eq. 20 we can again look up in [18] as

$$\begin{aligned}\mathbb{E}[\log p(z_i|\boldsymbol{\lambda})] &= \sum_{k=1}^{K} \zeta_{i,k}\, \mathbb{E}_q[\log \lambda_k] \\ &= \sum_{k=1}^{K} \zeta_{i,k}\Big(\psi(\omega_k) - \psi\Big(\sum_k \omega_k\Big)\Big).\end{aligned} \tag{22}$$

The expectation in the third line of eq. 20 is again just a log Dirichlet

$$\begin{aligned}\mathbb{E}[\log p(U_{j,k}|\beta)] = &\log\Gamma\Big(\sum_r \beta_r\Big) \\ &- \sum_r \log\Gamma(\beta_r) \\ &+ \sum_r \Big((\beta_r - 1)\psi(\phi_{j,k,r}) - \psi\Big(\sum_r \phi_{j,k,r}\Big)\Big).\end{aligned} \tag{23}$$

The expectation in the fourth line of eq. 20 is not readily available, so we will have to derive it

$$\begin{aligned}\mathbb{E}[\log p(X_{i,j}|U_{j,k}, z_i = k)] &= \mathbb{E}\left[\mathbb{1}[z_i = k] \log p(X_{i,j}|U_{j,k})\right] \\ &= \mathbb{E}\left[\mathbb{1}[z_i = k]\right] \mathbb{E}[\log p(X_{i,j}|U_{j,k})]] \\ &= \zeta_{i,k}\Big(\sum_{r=1}^{R_j} \mathbb{1}[X_{i,j} = r] \\ &\quad \times \Big(\psi(\phi_{j,k,r}) - \psi\Big(\sum_r \phi_{j,k,r}\Big)\Big)\Big)\end{aligned} \tag{24}$$

Now that we have all the necessary expectations to calculate $\mathbb{E}[p(\boldsymbol{z}, \boldsymbol{x})]$, we can derive the Entropies for $\mathbb{H}[q]$ (the later half of eq. 17).

$$\begin{aligned}\mathbb{H}\left[q(\boldsymbol{\lambda}, \boldsymbol{z}, U; \boldsymbol{\omega}, \zeta, \phi)\right] = &\mathbb{H}\left[q(\boldsymbol{\lambda}; \boldsymbol{\omega})\right] \\ &+ \sum_{i=1}^{I} \mathbb{H}\left[q(z_i; \zeta_i)\right] \\ &+ \sum_{j=1}^{J}\sum_{k=1}^{K} \mathbb{H}[q(U_{j,k}; \phi_{j,k})]\end{aligned} \tag{25}$$

The first line of eq. 25 is just the entropy of a Dirichlet distribution which we can look up as

$$\begin{aligned}\mathbb{H}\left[q(\boldsymbol{\lambda}; \boldsymbol{\omega})\right] = &-\log\Gamma\Big(\sum_k \omega_k\Big) + \sum_k \log\Gamma(\omega_k) \\ &- \sum_k (\omega_k - 1)\Big(\psi(\omega_k) - \psi\Big(\sum_k \omega_k\Big)\Big).\end{aligned} \tag{26}$$

The second line of eq. 25 is the entropy of a multinomial:

$$\mathbb{H}\left[q(z_i; \zeta_i)\right] = -\sum_k \zeta_{i,k} \log \zeta_{i,k} \tag{27}$$

Analogous to the entropy in eq. 26, we can write down the entropy for the third line of eq. 25

$$\begin{aligned}\mathbb{H}[q(U_{j,k}; \phi_{j,k})] = &-\log\Gamma\Big(\sum_r \phi_r\Big) + \sum_r \log\Gamma(\phi_r) \\ &- \sum_r (\phi_r - 1)\Big(\psi(\phi_r) - \psi\Big(\sum_r \phi_r\Big)\Big)\end{aligned} \tag{28}$$

Now we have all elements in place to actually optimize the free variational parameters $\boldsymbol{\omega}$, ζ and ϕ to maximize the ELBO. One approach would to apply a general purpose optimizer like BFGS, but the number of parameters in our model can grow very quickly, so that this approach becomes inefficient. Instead we will use a coordinate ascent strategy (CAVI [3]), where we iteratively optimize each single free parameter while the other are hold constant until the ELBO has converged. To achieve efficient updates, we will derive analytical updates for each of the parameters, by taking the derivative of the ELBO and setting it to zero.

First we will derive the update for latent group mixture parameter ω_k. This step is equivalent to the derivation of the updates of γ_i in the LDA model described in the appendix

A.3.2 of [18].

$$\begin{aligned}\frac{\partial \,\text{ELBO}}{\partial \omega_k} &= \frac{\partial}{\partial \omega_k}\Big(\sum_k(\alpha_k-1)\Big(\psi(\omega_k)-\psi\Big(\sum_k\omega_k\Big)\Big)\\ &\quad+\sum_k\sum_i\zeta_{i,k}\Big(\psi(\omega_k)-\psi\Big(\sum_k\omega_k\Big)\Big)\\ &\quad-\log\Gamma\Big(\sum_k\omega_k\Big)+\log\Gamma(\omega_k)\\ &\quad-\sum_k(\omega_k-1)\Big(\psi(\omega_k)-\psi\Big(\sum_k\omega_k\Big)\Big)\Big)\\ &= \frac{\partial}{\partial \omega_k}\Big(\psi(\omega_k)(\alpha_k+\sum_i\zeta_{i,k}-\omega_k)\\ &\quad-\psi\Big(\sum_k\omega_k\Big)\sum_k(\alpha_k+\sum_i\zeta_{i,k}-\omega_k)\\ &\quad-\log\Gamma\Big(\sum_k\omega_k\Big)+\log\Gamma(\omega_k)\Big)\\ &= \psi'(\omega_k)(\alpha_k+\sum_i\zeta_{i,k}-\omega_k)\\ &\quad-\psi'\Big(\sum_k\omega_k\Big)\sum_k(\alpha_k+\sum_i\zeta_{i,k}-\omega_k)\end{aligned} \tag{29}$$

If we assume that all α_k are equal, because our prior is symmetric, we can see that the whole term is zero when $(\alpha+\sum_i\zeta_{i,k}-\omega_k)=0$ and thus we can conclude that the ELBO is maximized when ω_k is set to

$$\omega_k=\alpha+\sum_i\zeta_{i,k}. \tag{30}$$

We will now derive the update for $\zeta_{i,k}$ in a similar fashion:

$$\begin{aligned}\frac{\partial \,\text{ELBO}}{\partial \zeta_{i,k}} &= \frac{\partial}{\partial \zeta_{i,k}}\Big(\zeta_{i,k}\Big(\psi(\omega_k)-\psi\Big(\sum_k\omega_k\Big)\Big)\\ &\quad+\zeta_{i,k}\sum_{j=1}^{J}\Big(\psi(\phi_{j,k,X_{i,j}})-\psi\Big(\sum_r\phi_{j,k,r}\Big)\Big)\\ &\quad-\zeta_{i,k}\log\zeta_{i,k}\Big)\\ &= \psi(\omega_k)-\psi\Big(\sum_k\omega_k\Big)\\ &\quad+\sum_{j=1}^{J}\Big(\psi(\phi_{j,k,X_{i,j}})\\ &\quad-\psi\Big(\sum_r\phi_{j,k,r}\Big)\Big)-\log(\zeta_{i,k})-1\end{aligned} \tag{31}$$

Setting this to zero and solving for $\zeta_{i,k}$ we find that

$$\begin{aligned}\zeta_{i,k}\propto\exp\Big(&\Big(\psi(\omega_k)-\psi\Big(\sum_k\omega_k\Big)\Big)\\ &+\sum_{j=1}^{J}\Big(\psi(\phi_{j,k,X_{i,j}})-\psi\Big(\sum_r\phi_{j,k,r}\Big)\Big)\Big),\end{aligned} \tag{32}$$

where the solution is only correct up to a proportional constant, because of the constraint that $\sum_k\zeta_{i,k}=1$.

Finally we will derive the update for $\phi_{j,k,r}$:

$$\begin{aligned}\frac{\partial \,\text{ELBO}}{\partial \phi_{j,k,r}} &= \frac{\partial}{\partial \phi_{j,k,r}}\Big(\sum_r(\beta_r-1)\Big(\psi(\phi_{j,k,r})-\psi\Big(\sum_r\phi_{j,k,r}\Big)\Big)\\ &\quad+\sum_r\sum_{i=1}^{I}\zeta_{i,k}\mathbb{1}[X_{i,j}=r]\\ &\quad\times\Big(\psi(\phi_{j,k,r})-\psi\Big(\sum_r\phi_{j,k,r}\Big)\Big)\\ &\quad-\log\Gamma\Big(\sum_r\phi_{j,k,r}\Big)+\sum_r\log\Gamma(\phi_{j,k,r})\\ &\quad-\sum_r(\phi_{j,k,r}-1)\Big(\psi(\phi_{j,k,r})-\psi\Big(\sum_r\phi_{j,k,r}\Big)\Big)\Big)\\ &= \frac{\partial}{\partial \phi_{j,k,r}}\Big(\psi(\phi_{j,k,r})\\ &\quad\times(\beta_r+\sum_{i=1}^{I}\zeta_{i,k}\mathbb{1}[X_{i,j}=r]-\phi_{j,k,r})\\ &\quad-\psi\Big(\sum_r\phi_{j,k,r}\Big)\\ &\quad\times\sum_r(\beta_r+\sum_{i=1}^{I}\zeta_{i,k}\mathbb{1}[X_{i,j}=r]-\phi_{j,k,r})\\ &\quad-\log\Gamma\Big(\sum_r\phi_{j,k,r}\Big)+\sum_r\log\Gamma(\phi_{j,k,r})\Big)\\ &= \psi'(\phi_{j,k,r})(\beta_r+\sum_{i=1}^{I}\zeta_{i,k}\mathbb{1}[X_{i,j}=r]-\phi_{j,k,r})\\ &\quad-\psi'\Big(\sum_r\phi_{j,k,r}\Big)\\ &\quad\times\sum_r(\beta_r+\sum_{i=1}^{I}\zeta_{i,k}\mathbb{1}[X_{i,j}=r]-\phi_{j,k,r})\end{aligned} \tag{33}$$

When we set this to zero and solve for $\phi_{j,k,r}$ we can again see that the whole term is zero when $(\beta_r+\sum_{i=1}^{I}\zeta_{i,k}\mathbb{1}[X_{i,j}=r]-\phi_{j,k,r})=0$ and if we again assume that all β_r are equal, that thus

$$\phi_{j,k,r}=\beta+\sum_{i=1}^{I}\zeta_{i,k}\mathbb{1}[X_{i,j}=r]. \tag{34}$$

Appendix B
Nonparametric extension

In this section we want to derive the update equation of the variational parameters for the nonparametric model.

The first term of the ELBO that obviously changes is the expectation of the Dirichlet, which now is the expectation of the Dirichlet Process

$$\begin{aligned}\mathbb{E}[\log p(\boldsymbol{v}|\alpha)] &= \sum_{k=1}^{K_{max}-1}\mathbb{E}[\log p(v_k|\alpha)]\\ &= \sum_{k=1}^{K_{max}-1}\big((\alpha_1-1)(\psi(\kappa_{k,1})-\psi(\kappa_{k,1}+\kappa_{k,2}))\\ &\qquad+(\alpha_2-1)(\psi(\kappa_{k,2})-\psi(\kappa_{k,1}+\kappa_{k,2}))\big)\end{aligned}. \tag{35}$$

The second line of eq. 20 we can look up in [19]

$$\mathbb{E}[\log p(z_i|\boldsymbol{v})] = \sum_{k=1}^{K_{max}} \big(q(z_i > k)\,\mathbb{E}[\log(1-v_k)] + q(z_i = k)\,\mathbb{E}[\log v_k]\big) \tag{36}$$

where

$$\begin{aligned} q(z_i > k) &= \sum_{k'=k+1}^{K_{max}} \zeta_{i,k'} \\ q(z_i = k) &= \zeta_{i,k} \\ \mathbb{E}[\log(1-v_k)] &= \psi(\kappa_{k,2}) - \psi(\kappa_{k,1}+\kappa_{k,2}) \\ \mathbb{E}[\log v_k] &= \psi(\kappa_{k,1}) - \psi(\kappa_{k,1}+\kappa_{k,2}). \end{aligned} \tag{37}$$

The expectations in line 3 and 4 of eq. 20 are unchanged and still

$$\begin{aligned} \mathbb{E}[\log p(U_{j,k}|\beta)] &= \log\Gamma\Big(\sum_r \beta_r\Big) - \sum_r \log\Gamma(\beta_r) \\ &\quad + \sum_r (\beta_r - 1)\Big(\psi(\phi_{j,k,r}) - \psi\Big(\sum_r \phi_{j,k,r}\Big)\Big) \\ \mathbb{E}[\log p(X_{i,j}|U_{j,k}, z_i = k) &= \zeta_{i,k} \sum_{r=1}^{R_j} \mathbb{1}[X_{i,j} = r] \\ &\quad \times \Big(\psi(\phi_{j,k,r}) - \psi\Big(\sum_r \phi_{j,k,r}\Big)\Big) \end{aligned} \tag{38}$$

We also need to update the entropy for the Dirichlet in eq. 25.

$$\begin{aligned} \mathbb{H}[q(\boldsymbol{v};\boldsymbol{\kappa}_1,\boldsymbol{\kappa}_2)] = -\sum_{k=1}^{K_{max}} \big(&\log\Gamma(\kappa_{k,1}) + \log\Gamma(\kappa_{k,2}) \\ &- \log\Gamma(\kappa_{k,1}+\kappa_{k,2}) \\ &+ (\kappa_{k,1}-1)\psi(\kappa_{k,1}) \\ &+ (\kappa_{k,2}-1)\psi(\kappa_{k,2}) \\ &- (\kappa_{k,1}+\kappa_{k,2}-2)\psi(\kappa_{k,1}+\kappa_{k,2})\big) \end{aligned} \tag{39}$$

We now again have all the elements to derive the new update equations. We will first derive the updates for $\kappa_{k,1}$

$$\begin{aligned} \frac{\partial\,\text{ELBO}}{\partial\kappa_{k,1}} = \frac{\partial}{\partial\kappa_{k,1}}\Bigg(&(\alpha_1 - 1)(\psi(\kappa_{k,1}) - \psi(\kappa_{k,1}+\kappa_{k,2})) \\ &- (\alpha_2 - 1)\psi(\kappa_{k,1}+\kappa_{k,2}) \\ &+ \sum_{i=1}^{I}\Big((\psi(\kappa_{k,2}) - \psi(\kappa_{k,1}+\kappa_{k,2}))\sum_{k'=k+1}^{K_{max}}\zeta_{i,k'} \\ &\qquad + (\psi(\kappa_{k,1}) - \psi(\kappa_{k,1}+\kappa_{k,2}))\zeta_{i,k}\Big) \\ &- \Big(\log\Gamma(\kappa_{k,1}) + \log\Gamma(\kappa_{k,1}+\kappa_{k,2}) \\ &\quad + (\kappa_{k,1}-1)\psi(\kappa_{k,1}) \\ &\quad + (\kappa_{k,1}+\kappa_{k,2}-2)\psi(\kappa_{k,1}+\kappa_{k,2})\Big)\Bigg) \\ = \psi(\kappa_{k,1})&(\alpha_1 - 1 + \sum_i \zeta_{i,k} - \kappa_{k,1} + 1) \\ &+ \psi(\kappa_{k,1}+\kappa_{k,2}) \\ &\times (\ -\alpha_1 + 1 - \alpha_2 + 1 \\ &\quad - \sum_i \sum_{k'=k+1}^{K_{max}} \zeta_{i,k'} \\ &\quad - \sum_i \zeta_{i,k} + \kappa_{k,1} + \kappa_{k,2} - 2) \end{aligned} \tag{40}$$

This term is zero if $\kappa_{k,2}$ removes the additional terms in the second parentheses and $\kappa_{k,1}$ is just equal to the remaining terms. The update equations for $\kappa_{k,1}$ and $\kappa_{k,2}$ are thus

$$\begin{aligned} \kappa_{k,2} &= \alpha_2 + \sum_i \sum_{k'=k+1}^{K_{max}} \zeta_{i,k'} \\ \kappa_{k,1} &= \alpha_1 + \sum_i \zeta_{i,k}, \end{aligned} \tag{41}$$

which matches the results of [19]. In this equation we see that unlike the classical Dirichlet Process our model has two hyper-parameters: α_1 and α_2. This model is also the called the Beta two parameter process [27]. It is equivalent to the Dirichlet Process if $\alpha_1 = 1$. A large value for α_2 encourages the opening of more classes, whereas a large value for α_1 penalizes the opening of new classes.

Specifically for the Beta two parameter process the ratio of $\kappa_{k,1}$ to $\kappa_{k,2}$ determines how much of the remaining probability mass is assigned to class k. On average class k will cover $\frac{\kappa_{k,1}}{\kappa_{k,1}+\kappa_{k,2}}$ of the remaining space. The priors α_1 and α_2 serve as additional pseudo counts in that ratio. So if we believe *a priori* that each class should cover 90% of the remaining space there are in theory two ways to achieve this. We can either fix $\alpha_1 = 1$ and make α_2 smaller (i.e. $\alpha_2 = 1/9$) or fix $\alpha_2 = 1$ and make α_1 larger (i.e. $\alpha_1 = 9$). But in the first case if we actually have observed $\sum_i \sum_{k'=k+1}^{K_{max}} \zeta_{i,k'} > 1$ this would easily overpowers our prior believe, whereas in the second case the regularization is much stronger.

We will now similarly derive the update for $\zeta_{i,k}$

$$\begin{aligned}
\frac{\partial \,\text{ELBO}}{\partial \zeta_{i,k}} = \; & \frac{\partial}{\partial \zeta_{i,k}} \Bigg(\zeta_{i,k}(\psi(\kappa_{k,1}) - \psi(\kappa_{k,1} + \kappa_{k,2})) \\
& + \zeta_{i,k} \sum_{k'=1}^{k-1} (\psi(\kappa_{k',2}) - \psi(\kappa_{k',1} + \kappa_{k',2})) \\
& + \zeta_{i,k} \sum_{j=1}^{J} \Big(\psi(\phi_{j,k,X_{i,j}}) - \psi\Big(\sum_r \phi_{j,k,r} \Big) \Big) \\
& - \zeta_{i,k} \log \zeta_{i,k} \Bigg) \\
= \; & \psi(\kappa_{k,1}) - \psi(\kappa_{k,1} + \kappa_{k,2}) \\
& + \sum_{k'=1}^{k-1} (\psi(\kappa_{k',2}) - \psi(\kappa_{k',1} + \kappa_{k',2})) \\
& + \sum_{j=1}^{J} \Big(\psi(\phi_{j,k,X_{i,j}}) - \psi\Big(\sum_r \phi_{j,k,r} \Big) \Big) \\
& - \log(\zeta_{i,k}) - 1
\end{aligned} \tag{42}$$

We can easily set this to zero and solve this for $\zeta_{i,k}$ again up to an proportional constant

$$\begin{aligned}
\zeta_{i,k} \propto \exp \Bigg(& \psi(\kappa_{k,1}) - \psi(\kappa_{k,1} + \kappa_{k,2}) \\
& + \sum_{k'=1}^{k-1} (\psi(\kappa_{k',2}) - \psi(\kappa_{k',1} + \kappa_{k',2})) \\
& + \sum_{j=1}^{J} \Big(\psi(\phi_{j,k,X_{i,j}}) - \psi\Big(\sum_r \phi_{j,k,r} \Big) \Big) \Bigg)
\end{aligned} \tag{43}$$

The update equation for $\phi_{j,k,r}$ does not change, so we now have all the elements to maximize the ELBO.