Austrian Journal of Statistics
2025, Volume 54, 100–123.
http://www.ajs.or.at/
doi:10.17713/ajs.v54i3.2083

Ordinal Clustering with the flex-Scheme

Dominik Ernst
BOKU University Vienna

Lena Ortega Menjivar
BOKU University Vienna

Theresa Scharl
BOKU University Vienna

Bettina Grün
WU Vienna

Abstract

We investigate suitable methods for clustering multivariate ordinal data assuming that the data are collected in surveys with item batteries on the same ordinal answer format and that respondents are to be grouped to characterize different answering patterns and tendencies. We consider heuristic partitioning methods and model-based methods which fit within the *flex*-scheme proposed by Fritz Leisch for clustering, in combination with different variants of scale handling (numerical coding, nominalization and respecting the ordinal scale). The performance of the methods to extract the true clustering structure is assessed in an illustrative simulation study using artificial data where the number of observations, the number of variables and the number of response levels as well as the difficulty of the clustering problem are systematically varied to highlight in which situations certain methods might be preferable. By extending the *flex*-scheme with new methods, which we provide in our R package **flexord**, we help pave the way for future research on ordinal data clustering with even more complex and diverse data-generating processes.

Keywords: ordinal data, partitioning clustering, model-based clustering, *flex*-scheme.

1. Introduction

Finding groups in data is a common research aim in many fields, including for example market research, education, or health care (Ghosal, Nandy, Das, Goswami, and Panday 2020). The data used in these applications usually come from surveys where answers to item batteries are collected on ordinal scales. Ordinal scales are valued for their ease of administration amd various types of ordinal scales are commonly used, in particular with respect to the number of response options offered to respondents typically ranging from two-point to eleven-point options, see, for example, Taherdoost (2019). However, data collection using ordinal scales leads to challenges during data analysis as many of the usual default analysis methods have been developed for either metric or nominal data.

"Clustering" is an umbrella term for a wide array of unsupervised methods designed to find groups in data. These methods are commonly differentiated into distance-based (hierarchical or partitioning) and model-based clustering methods (Dolnicar, Grün, and Leisch 2018,

pp. 75–142). The first approach assigns cluster memberships by minimizing distances between cluster members, i.e., partitioning the observations into groups such that they are similar within groups, and the second via cluster-specific distributions.

In their extensive work on partitioning and model-based clustering, Leisch (2004, 2006); Grün and Leisch (2007, 2008) indicated that the choice of algorithm (specifically, the distance measure and centroid determination mechanism used for partitioning; and the distribution used for model-based clustering) has a significant influence on any – inherently exploratory – clustering result. For this reason, Fritz Leisch developed the *flex*-scheme to provide toolboxes for partitioning and model-based clustering where distance/centroid determination and cluster distribution can be easily switched. This scheme is implemented for the R environment for statistical computing and graphics (R Core Team 2024) in the packages **flexclust** and **flexmix** (Leisch 2006; Grün and Leisch 2025). These implementations allow to easily consider extensions and variants and compare their results. Additionally, these packages contain functionalities that mitigate the risk of choosing solutions which correspond to local optima via automatized re-runs of the algorithm; and they provide tailored visualization tools. The packages contain eight pre-implemented distance functions[1], five centroid functions[2] and 13 distributions[3] for clustering. Most of these functions are tailored for interval data, and only few options for ordinal data are available. However, the packages are designed in such a way that they can easily be extended to other algorithms.

Different approaches for clustering ordinal data have been pursued in the literature. One extreme approach is to treat ordinal variables as nominal, i.e., ignoring the ordering of categories. With this strategy, one can be sure not to violate any assumptions towards the data. However, treating ordinal data as nominal induces a loss of available methods, and a loss of power in the methods that are available (Agresti 2010, pp. 2–3). The approach at the other extreme is to treat ordinal variables as interval scaled, i.e., by assuming categories as equidistant or by assigning suitable scores. Inspired by the terminology coined by Foss, Markatou, and Ray (2019) for mixed-type data, we will refer to these two extremes as *nominalization* and *numerical coding*. Between these two extremes lie strategies that attempt to *respect ordinal scales*, for example by restricting parametric analyses to rely only on the ordering information of categories (Agresti 2010, p. 4). In this area, approaches differ with respect to the stringency of assumptions they impose on ordinal scales. For example, Podani (1999) states that subtractions, multiplications or divisions even of ranks of ordinal raw data, are only permissible if there are no tied ranks (which will frequently be present for ordinal response scales in surveys which are inherently limited). Thus, rank operations must be extended to account for ties. Walesiak and Dudek (2010) established an even more stringent rule stating that only comparison functions such as *equal*, *greater than*, or *smaller than* are permissible for ordinal data. In general, ordinal data analysis clearly needs to be performed within the spectrum between the two extremes of *numerical coding* and *nominalization*.

In the field of partitioning algorithms, the best known and most commonly applied methods that attempt to *respect ordinal scales* are K-medians and *Partitioning Around Medoids* using Gower's distance extended for ordinal data as proposed by Kaufman and Rousseeuw (1990). Alternatively, Szepannek, Aschenbruck, and Wilhelm (2024) proposed an adaptation of the *K-prototypes* algorithm (which is a K-centroids clustering algorithm adapted to handle both numeric and categorical variables). Their approach handles ordinal categorical variables via the extension of Gower's distance proposed by Podani (1999). In their 2020 paper, Zhang and Cheung lament the dearth of distance measures available for ordinal variables, and therefore develop an automatized distance calculation method, where inter-category distances are

[1]Euclidean distance, Manhattan distance, maximum distance, Minkowski distance, Canberra distance, Jaccard distance, the angle between observations, and one minus the correlation between observations.

[2]Mean, standardized mean, median, a numerical optimizer, and a numerical optimizer with constraints 0 and 1.

[3]Multivariate Gaussian, inverse Gaussian, lognormal, exponential, gamma, Weibull, Burr, inverse Burr, multivariate binary, combinations of Gaussian and binary data, Poisson, and factor analyzers.

iteratively adjusted to optimize clustering efficiency.

Several approaches have also been considered in the field of model-based clustering of ordinal data. For example, Hennig and Liao (2013) use an adjacent category logit model as an extension to the latent class analysis model. McParland and Gormley (2016) pursue a latent variable approach where a multivariate Gaussian latent variable is mapped to an ordinal scale using suitable thresholds. Furthermore, Jacques, Biernacki, and Selosse developed model-based clustering algorithms based on their proposed Binary Ordinary Search distribution (2018; 2020). The extension of the proportional odds model to various clustering applications was proposed by Costilla, Liu, Arnold, and Fernández (2019) and Preedalikit, Fernández, Liu, McMillan, Nai Ruscone, and Costilla (2024). Anders and Batchelder (2015) consider a clustering approach tailored towards work within the Cultural Consensus Theory that is based on an ordered polytomous distribution (which is a generalization of the proportional odds model that allows for category-specific intercepts).

In the literature, clustering of ordinal data has often been addressed in the context of clustering mixed-type data. In addition, the contributions often take an approach based on mapping the data to a different scale level or only taking into account either the partitioning or the model-based framework. In this paper, we consider a clustering task for ordinal data collected on a set of items using the same scale, i.e., the same number of response levels is available for all variables. We provide an overview of suitable partitioning and model-based methods in this situation with the *flex*-scheme allowing for different mappings to scale levels, and we compare their performance in an illustrative simulation study using artificial data. The simulation study focuses on the ability of the clustering methods to extract the *true clustering* solution in situations with different sample sizes, different numbers of variables, different lengths of response levels and different difficulty of the cluster structure in the data. We provide the algorithms behind the methods that had not previously been implemented into the *flex*-scheme (or, in some cases, had not yet been applied to partitioning/model-based clustering of ordinal data at all) in our R package **flexord**, available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=flexord` (Ortega Menjivar and Ernst 2025), and share the scripts behind our simulations in the reproduction repository available from Ortega Menjivar, Ernst, Scharl, and Grün (2025).

2. Methods

Table 1 provides an overview on methods for clustering ordinal data which make use of the *flex*-scheme. The methods are split by clustering type (*partitioning* and *model-based*) and categorized with regard to their approach to scale handling (*numerical coding*, *respecting ordinal scales* and *nominalization*). In addition to the methods listed in Table 1, we also consider partitioning method for ordinal data which do not fit within the *flex*-scheme: Partitioning Around Medoids (PAM) in combination with Gower's distance as well as the GDM2 distance. The following sections provide more details on all these methods.

2.1. Partitioning methods

Partitioning methods aim at grouping a set of observations into a prespecified number of disjoint groups K where their union corresponds to the total set of the observations. In the context of K-centroids clustering, the clustering problem consists of determining a set of centroids C_K such that the average dissimilarity of each point to the closest centroid is minimal (Leisch 2006):

$$D(X_N, C_K) = \frac{1}{N} \sum_{n=1}^{N} d(\boldsymbol{x}_n, c(\boldsymbol{x}_n)) \to \min_{C_K},$$

where $X_N = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ is the set of observations, $C_K = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\}$ is the set of centroids

Table 1: Overview on ordinal methods within the *flex*-scheme, with partitioning methods on top and model-based methods at the bottom by scale handling type and distance measure & centroid/distribution

Partitioning based on K-centroids clustering			
Scale handling	Method	Distance measure	Centroid
Numerical coding	`kmeans`	squared Euclidean distance	mean
Nominalization	`kmodes`	1 – Simple Matching Coefficient	mode
Respecting ordinal scale	`kmedians`	Manhattan distance	median
	`kGower`	Gower's distance for ordinal data	numerical optimizer median
	`kGDM2`	Generalized distance measure for ordinal data	mode
Model-based using finite mixtures			
Scale handling	Distribution		
Numerical coding	normal		
Nominalization	multinomial		
Respecting ordinal scale	binomial beta-binomial		

and

$$c(\boldsymbol{x}) = \arg\min_{\boldsymbol{c} \in C_K} d(\boldsymbol{x}, \boldsymbol{c})$$

assigns each point to its closest centroid.

To perform K-centroids clustering, one requires a dissimilarity measure between observations $\boldsymbol{x}$ and potential centroids $\boldsymbol{c}$. In addition, the space considered for observations and centroids needs to be specified.

Determining the globally optimal solution is difficult and usually an iterative algorithm is employed to find a solution. The iterative algorithm consists of the following steps:

Step 1: Choose initial cluster centers ("centroids") at random.

Step 2: Assign data points to the centroid where the dissimilarity is minimal.

Step 3: Determine cluster centroids based on the given partition.

Step 4: Repeat Steps 2 and 3 until the centroids no longer change.

This algorithm solves the K-means clustering problem, when the squared Euclidean distance is used as dissimilarity measure and the centroids in Step 3 are determined in closed form as the column-wise mean values across all observations currently assigned to the cluster.

Step 3 results in *canonical* centroid estimates if the cluster centroids minimize the sum of dissimlarities to the observations in the cluster. In this case the algorithm converges because the objective function is monotonically decreasing. However, convergence is only to a *local* optimum and depends on the initialization.

Determining canonical centroids requires solving an optimization problem, where the solution is available in closed form for some cases. If the solution to this optimization problem is not available in closed form (e.g., when using the Jaccard dissimilarity for binary data), faster clustering versions are obtained by plugging in different ways to determine the centroids,

e.g., by inserting the usual K-means step and determining the cluster-specific component-wise means as centroids. We will to refer to these types of centroids as *pragmatic* centroids, as they are chosen for practical reasons, such as speedup or interpretability. For *pragmatic* centroids, however, convergence is no longer guaranteed. For more information see Leisch (2006).

Leisch (2006) presents and compares several variants of K-centroids clustering algorithms, distinguishing in particular the space $\mathscr{X}$ of an observation $\boldsymbol{x}$ and the admissible space $\mathscr{C}$ for a centroid $\boldsymbol{c}$. For p-dimensional metric spaces where observations and centroids are elements in $\mathbb{R}^p$, in particular the following two versions result:

`kmeans`: uses squared Euclidean dissimilarity $d(\boldsymbol{x}, \boldsymbol{c}) = \sum_{j=1}^{p}(x_j - c_j)^2$ and has the cluster-specific component-wise means as canonical centroids.

`kmedians`: uses Manhattan dissimilarity $d(\boldsymbol{x}, \boldsymbol{c}) = \sum_{j=1}^{p}|x_j - c_j|$ and has the cluster-specific component-wise medians as canonical centroids.

Package **flexclust** provides them by calling `kccaFamily()` with argument `which = "kmeans"` or `"kmedians"` (Leisch 2006). Both variants may be used for clustering ordinal data. When using `kmeans`, numerical coding is assumed for the observations. Using `kmedians` respects the ordinal scale when relying on the use of ranks (see the Gower's coefficient for ordinal data below and the implementation in package **cluster**; Maechler, Rousseeuw, Struyf, Hubert, and Hornik 2024).

In the context of p-dimensional nominal data, the following version results:

`kmodes`: uses 1 minus the Simple Matching Coefficient as dissimilarity measure, i.e.,

$$d(\boldsymbol{x}, \boldsymbol{c}) = \frac{\#\{j = 1, \ldots, p | x_j \neq c_j\}}{p}$$

and has for each cluster the cluster-wise modes (i.e., the most frequent values) as canonical centroids.

Package **klaR** (Weihs, Ligges, Luebke, and Raabe 2005) provides an R implementation of this algorithm. This clustering approach is provided in package **flexord** for the *flex*-scheme by providing a function for determining the dissimilarity between data points and centers and a function determining the centers and by combining them with functions from package **flexclust**.

Leisch (2006) also discusses a K-centroids clustering version based on the Jaccard dissimilarity for binary data where either canonical or pragmatic centroids are determined. However, the clustering of ordinal data is not considered. To obtain K-centroids clustering for ordinal data, suitable dissimilarity measures for ordinal variables need to be specified. In the context of mixed-type data, Gower (1971) proposes a coefficient of (dis)similarity for p-dimensional observations containing dichotomous, qualitative and quantitative variables where the dissimilarity is obtained as a weighted sum over the variable-specific dissimilarities:

$$d(\boldsymbol{x}_i, \boldsymbol{x}_k) = \frac{\sum_{j=1}^{p} \delta_{ikj} d(x_{ij}, x_{kj})}{\sum_{j=1}^{p} \delta_{ikj}},$$

where $\delta_{ikj} = 1$ if x_{ij} and x_{kj} are not missing and if not both are 0 for a binary variable j (corresponding to an asymmetric dissimilarity measure for binary variables).

Kaufman and Rousseeuw (1990, pp. 35–36) build on Gower's coefficient of (dis)similarity using the same weighted sum approach and extend it to also include ordinal variables. They propose to replace ordinal observations with their ranks before determining the dissimilarity for variable j as:

$$d(x_{ij}, x_{kj}) = \frac{|r(x_{ij}) - r(x_{kj})|}{R_j},$$

where the ranks assigned take values in $\{1, 2, \ldots, R_j\}$ such that equal measurements have equal ranks and each rank occurs at least once and R_j represents the range of ranks of variable j in the sample.

Package **cluster** implements the calculation of the matrix of pairwise dissimilarity measures between N observations in function `daisy()` based on these ideas. However, the function uses the internal codes $\{1, \ldots, M\}$ for an ordinal variable with M levels instead of the ranks and determines the range based on the sample. This implies that potentially not all ranks, i.e., not all values in $\{1, \ldots, R_j\}$ may be observed in the data, and that the range is determined based on the range of the observed internal codes. An alternative extension of Gower's dissimilarity to ordinal data is considered in Podani (1999, pp. 335–336) who uses sample ranks with ties and includes a correction for ties in the dissimilarity measure.

Leisch (2006) points out the need to define the space for the centroids when solving the K-centroids clustering problem which can correspond to the space of the observations but can also differ, e.g., in the binary case considered. This aspect is of particular importance in the context of mixed-type data cases and for ordinal data. To address this issue, Kaufman and Rousseeuw introduced the notion of a *medoid* as centroids for partitioning clustering (Kaufman and Rousseeuw 1990, pp. 68–123). "Medoids" correspond to observations in the data set, i.e., refer to a row $\boldsymbol{x}_n$ in the data set, which is then used as centroid $\boldsymbol{c}_k$. They thus in general represent *pragmatic* instead of *canonical* centroids.

Restricting centroids to observed data points implies that PAM does only require a pairwise dissimilarity matrix between the N observations as input, i.e., indeed no space for the observations needs to be defined. If observations are provided, the PAM algorithm starts with calculating the pairwise dissimilarity matrix. The PAM implementation also includes a *build* phase where a good set of initial centroids is identified implying that the clustering obtained with PAM relies less on a specific random initialization. For these reasons, a distinction will be made from now on between "classical" K-centroid procedures and PAM.

PAM in combination with Gower's coefficient of dissimilarity is an established method for clustering mixed-type or ordinal data and can directly be performed with the implementation in package **cluster**. In addition, Gower's dissimilarity coefficient for ordinal data may also be combined with a "classical" K-centroids implementation using the *flex*-scheme. In the implementation available in package **flexord**, the centroids do not necessarily need to correspond to an observation in the data set. No closed form formulas for determining the canonical centroids for this dissimilarity are available and an exhaustive search approach may be pursued where the centroids are determined separately for each variable j by selecting the response level out of all possible response levels which minimizes the dissimilarity to all cluster members.

Walesiak (1993) and Walesiak and Dudek (2010) propose a *generalized distance measure* for ordinal variables (GDM2). GDM2 only relies on the properties of ordinal variables, i.e., the notion that two values of an ordinal variable are either equal or the first one is greater or the first one is smaller. For a given set of observations X_N, they propose to determine the distance between observations $\boldsymbol{x}_i$ and $\boldsymbol{x}_k$ using

$$d_N(\boldsymbol{x}_i, \boldsymbol{x}_k) = \frac{1}{2}\left(1 - \frac{\left(\sum_{j=1}^{p}\sum_{l=1}^{N} a_{ilj}a_{klj} + \sum_{j=1}^{p} a_{ikj}a_{kij}\right)}{\sqrt{\left(\sum_{j=1}^{p}\sum_{l=1}^{N} a_{ilj}^2\right)\left(\sum_{j=1}^{p}\sum_{l=1}^{N} a_{klj}^2\right)}}\right) \tag{1}$$

with

$$a_{ikj} = \begin{cases} 1 & \text{if } x_{ij} > x_{kj}, \\ 0 & \text{if } x_{ij} = x_{kj}, \\ -1 & \text{if } x_{ij} < x_{kj}, \end{cases}$$

where $i, k = 1, \ldots, N$ and $j = 1, \ldots, p$. This implies that $d_N(\boldsymbol{x}_i, \boldsymbol{x}_k) \in [0, 1]$ for all $\boldsymbol{x}_i, \boldsymbol{x}_k \in X_N$.

This distance measure is implemented in the R package **clusterSim** (Walesiak and Dudek 2020) such that a distance matrix is returned given a data matrix with N observations and p variables. The resulting matrix can then be used as input for partitioning clustering with the PAM algorithm.

Equivalently, the GDM2 distance measure for an ordinal variable j can be defined based on the relative frequencies of each level of the ordinal variable j in X_N and the empirical cumulative distribution given by

$$\hat{f}_{j,N}(x) = \frac{1}{N}\#\{i = 1, \ldots, N | x_{ij} = x\}, \qquad \hat{F}_{j,N}(x) = \frac{1}{N}\#\{i = 1, \ldots, N | x_{ij} \leq x\}.$$

By reformulating the GDM2 formula in Equation (1) in this way, we can define for arbitrary values $\boldsymbol{x} \in \mathscr{X}$ and $\boldsymbol{c} \in \mathscr{C}$:

$$d_N(\boldsymbol{x}, \boldsymbol{c}) = \frac{1}{2}\left(1 - \frac{\left(\sum_{j=1}^{p} 1 - \delta_{\{x_j \neq c_j\}}(\frac{1}{N} + 2\left|\tilde{F}_{j,N}(x_j) - \tilde{F}_{j,N}(c_j)\right|) - \delta_{\{x_j = c_j\}}\hat{f}_{j,N}(x_j)\right)}{\sqrt{\left(\sum_{j=1}^{p}(1 - \hat{f}_{j,N}(x_j))\right)\left(\sum_{j=1}^{p}(1 - \hat{f}_{j,N}(c_j))\right)}}\right),$$

with $\tilde{F}_{j,N}$ defined as

$$\tilde{F}_{j,N}(x) = \hat{F}_{j,N}(x) - \frac{1}{2}\hat{f}_{j,N}(x)$$

and with $\delta_{\text{condition}}$ being 1 if the condition holds and 0 otherwise. For the detailed derivations, see Appendix A.

With this alternative definition, which extends the calculation of the distance to two arbitrary data points, we are now able to evaluate the GDM2 distance not only in combination with medoids as centroids, but also in the general K-centroids context where the impact of different types of centroids can be evaluated. We can thus extend package **flexclust** to use also GDM2 as distance measure. The implementation in package **flexord** allows to choose for Step 3 that either a general purpose optimizer is used to obtain the canonical centroids, or a pragmatic centroid is chosen, such as the mode as in `kmodes`.

2.2. Model-based methods

The model-based approach to clustering assumes that the data X_N are generated from a finite mixture distribution:

$$f(X_N) = \prod_{n=1}^{N}\sum_{k=1}^{K}\tau_k f_k(\boldsymbol{x}_n|\boldsymbol{\theta}_k). \tag{2}$$

Clusters are also referred to as components in the mixture model setting. That is, each cluster $k \in \{1, \ldots, K\}$ is represented by a component distribution f_k with parameters $\boldsymbol{\theta}_k$. τ_k represents the prior probability of each observation belonging to a certain component and thus corresponds to the cluster size, with $\tau_k > 0$ and $\sum_{k=1}^{K}\tau_k = 1$.

We define the component distribution for the p-dimensional observations based on the conditional independence assumption to be given by

$$f_k(\boldsymbol{x}_n|\boldsymbol{\theta}_k) = \prod_{j=1}^{p} f_{kj}(x_{nj}|\boldsymbol{\theta}_{kj}).$$

For the purpose of this paper we assume that each f_{kj} is from the same parametric family, i.e., across variables as well as components. Generally this assumption is not necessary for mixture models. In particular the distribution f_{kj} does not have to be the same across variables. A more flexible specification is in general useful for example when modeling mixed-type data. Note that f_{kj} could also be multivariate if the variables are split into p blocks with

the conditional independence function applying to the blocks and a multivariate distribution, such as for example a multivariate normal distribution, used for a block.

When using mixture models for clustering, the corresponding estimation problem is twofold: (1) the parameters $\boldsymbol{\theta}_k$ of each component and the prior weights τ_k need to be estimated and (2) the cluster memberships for each observation need to be determined. Each problem individually would be easily solvable if the solution to the other were available, i.e., given the cluster memberships parameter estimation could be performed on subsets of the original data, and if the parameters were known one could assign observations to the component where they were most likely generated from.

This aspect is exploited by the Expectation-Maximization (EM) algorithm (Dempster, Laird, and Rubin 1977). The EM algorithm is often used for maximum likelihood estimation but is equally applicable for maximum a-posteriori estimation. In general, the EM algorithm is useful in a missing data context where the complete-data log-likelihood or posterior is easier to maximize. In mixture models, the missing data correspond to the cluster memberships. The likelihood of a mixture distribution can be unbounded leading to spurious solutions. A Bayesian regularization approach corresponding to maximum a-posteriori estimation can be employed to avoid this (Fraley and Raftery 2007).

The EM algorithm is an iterative estimation scheme where each iteration consists of both, an E-step and an M-step. The E-step determines the expected complete-data log-likelihood or posterior given the observed data and current parameter estimates. The M-step maximizes the expected complete-data log-likelihood or posterior with respect to the parameters.

For mixture models, the complete-data log-likelihood or posterior is linear in the missing information. The E-step consists of computing for each observation $n = 1, \ldots, N$ the conditional probability that object n belongs to the kth component:

$$\hat{z}_{nk} = \frac{\tau_k f_k(\boldsymbol{x}_n|\boldsymbol{\theta}_k)}{\sum_{j=1}^{K} \tau_j f_j(\boldsymbol{x}_n|\boldsymbol{\theta}_j)}. \tag{3}$$

The M-step then estimates parameters $\boldsymbol{\theta}_k$ and τ_k given the conditional probabilities $\hat{z}_{nk}$. Both steps are iteratively computed until convergence. Based on the parameter estimates obtained at convergence, observations are assigned to the cluster with the highest conditional probability $\hat{z}_{nk}$. Given its iterative nature, the results of the EM algorithm heavily depend on its initial values. As such it is common to re-run the algorithm multiple times with randomly chosen starting values in order to alleviate this dependence (Biernacki, Celeux, and Govaert 2003).

The R package **flexmix** implements the EM algorithm in a way that is easily extendable. In order to extend the package with a new distribution one needs to implement two ingredients: (1) the likelihood contribution of each observation and (2) a weighted estimator for the parameters of the component distributions. For the purpose of this paper this was done for the distributions listed in the following and made available in package **flexord**.

Normal distribution

When treating ordinal data as interval scaled, any mixture model for metric data may be used, with the normal distribution representing the default choice as cluster distribution (Scrucca, Fraley, Murphy, and Raftery 2023). Based on the conditional independence assumption, a univariate normal distribution is assumed for each variable and cluster with density given by:

$$f_{kj}(x_{nj}|\mu_{kj}, \sigma^2_{kj}) = \frac{1}{\sqrt{2\pi\sigma^2_{kj}}} \exp\left(-\frac{(x_{nj} - \mu_{kj})^2}{2\sigma^2_{kj}}\right)$$

with means μ_{kj} and variances σ^2_{kj} for each variable and cluster. For small values of σ^2_{kj}, the density takes large values which tend towards infinity in case the variance tends towards zero. I.e., the density is unbounded for zero values of the variance (Fraley and Raftery 2007).

In order to avoid this problem, we use Bayesian regularization and we impose Bayesian regularization on both parameters. For the mean, a normal prior is used conditional on the variance

$$\mu|\sigma^2 \sim N(\mu_{\mathcal{P}}, \sigma^2/\kappa_{\mathcal{P}}).$$

For the variance, an inverse gamma prior is used

$$\sigma^2 \sim \text{inverseGamma}(\nu_{\mathcal{P}}/2, \zeta_{\mathcal{P}}^2/2).$$

This is the conjugate prior for the parameters of the univariate normal distribution. The hyperparameters $\mu_{\mathcal{P}}$, $\kappa_{\mathcal{P}}$, $\nu_{\mathcal{P}}$ and $\zeta_{\mathcal{P}}^2$ are the *mean*, *shrinkage*, *degrees of freedom* and *scale*.

The weighted maximum a-posteriori (MAP) estimators for the regularized Bayesian estimation in the M-step are given by:

$$\hat{\mu}_{kj} = \frac{\kappa_{\mathcal{P}}\mu_{\mathcal{P}} + n_k\bar{x}_{kj}}{\kappa_{\mathcal{P}} + n_k}$$

with the sum of posterior weights given by $n_k = \sum_{n=1}^{N} \hat{z}_{nk}$ and the weighted mean by $\bar{x}_{kj} = 1/n_k \sum_{n=1}^{N} \hat{z}_{nk}x_{nj}$. The MAP estimator for the variance is:

$$\hat{\sigma}_{kj}^2 = \frac{\zeta_{\mathcal{P}}^2 + \frac{\kappa_P n_k}{(\kappa_P + n_k)}(\bar{x}_{kj} - \mu_{\mathcal{P}})^2 + \sum_{n=1}^{N} \hat{z}_{nk}(x_{nj} - \bar{x}_{kj})^2}{\nu_{\mathcal{P}} + n_k + 3}.$$

We use the same choices for the prior hyperparameters as Fraley and Raftery (2007):

- $\mu_{\mathcal{P}}$: the mean of the data.
- $\kappa_{\mathcal{P}}$: 0.01. For the mean estimator above this hyperparameter can be seen as adding $\kappa_{\mathcal{P}}$ observations with value $\mu_{\mathcal{P}}$ to each group in the data. The value of 0.01 was derived by experimentation in Fraley and Raftery (2007).
- $\nu_{\mathcal{P}}$: 3.
- $\zeta_{\mathcal{P}}^2$: var(data)/K^2 that is the empirical variance divided by the square of the number of components.

Multinomial distribution

With the multinomial distribution each ordinal variable is treated as unordered, i.e., the ordinal property is not taken into account. The density is given by

$$f_{kj}(x_{nj}|\boldsymbol{\pi}_{kj}) = \prod_{c=1}^{r} \pi_{kj,c}^{\delta_{\{x_{nj}=c\}}}$$

with probabilities for all categories $\boldsymbol{\pi}_{kj} = (\pi_{kj,1}, \ldots, \pi_{kj,r})$ with r corresponding to the reponse level length and the indicator function δ which is either 1 or 0 depending whether the condition in the subscript is fulfilled or not.

During estimation, parameter estimates may become numerically zero. This violates the condition that the parameters of the multinomial distribution are positive. Again Bayesian regularization helps to avoid degenerate solutions by performing MAP estimation after imposing a proper conjugate prior (Galindo Garre and Vermunt 2006). The conjugate prior of the multinomial distribution is the Dirichlet distribution. The parameters of the Dirichlet prior are selected to correspond to the marginal distribution of the variable across all observations. These parameters are obtained using unweighted maximum likelihood estimation for the aggregate data:

$$\hat{\pi}_{jc}^{*} = \frac{1}{N}\sum_{n=1}^{N} \delta_{\{x_{nj}=c\}}$$

for each variable j and category $c \in \{1, \ldots, r\}$. The weighted MAP estimator of the regularized estimation in the M-step is then given by

$$\hat{\pi}_{kjc} = \frac{\alpha \hat{\pi}^*_{jc} + \sum_{n=1}^{N} \hat{z}_{nk} \delta_{\{x_{nj}=c\}}}{\alpha + n_k}.$$

The hyperparameter α can be seen as adding α observations with values $\hat{\pi}^*_{jc}$ to each component. In this paper we use as default for regularization $\alpha = 1$ such that the "distortion" by the regularization is rather mild but still boundary estimates are avoided.

Binomial distribution

The binomial distribution is a parsimonious way to model ordinal data. The distribution is characterized by a single parameter corresponding to the mean and has a unimodal shape. This makes this distribution in particular appealing for a model-based clustering context by facilitating the interpretation of the clusters.

The component distribution is given as

$$f_{kj}(x_{nj}|\pi_{kj}) = \binom{r-1}{x_{nj}} \pi_{kj}^{x_{nj}} (1-\pi_{kj})^{r-x_{nj}-1}$$

with the number of categories r and π_{kj} the success probability in component k and variable j with $x_{nj} \in \{0, \ldots, r-1\}$.

We use again a regularized parameter estimator, which results from adding artificial observations to the data set corresponding to the population mean. Firstly, we need the overall estimates, which correspond to the unweighted maximum likelihood estimator:

$$\hat{\pi}^*_j = \frac{1}{N} \sum_{n=1}^{N} \frac{x_{nj}}{r-1},$$

which is the same as the column means of the original data set with entries re-scaled to $[0, 1]$ by dividing by the number of categories r minus one.

The MAP estimator for each cluster k and variable j is obtained using the prior $\pi_{kj} \sim \text{Beta}(\alpha\hat{\pi}^*_j, \alpha(1-\hat{\pi}^*_j))$ using

$$\hat{\pi}_{kj} = \frac{\alpha \hat{\pi}^*_j + \sum_{n=1}^{N} \hat{z}_{nk} \frac{x_{nj}}{r-1}}{\alpha + n_k}.$$

α may again be interpreted as the prior sample size, i.e., the number of artificial data points added to each component. This avoids degenerate solutions by "pulling" the parameter estimates toward the aggregate mean in the sample. As a side effect, this causes the estimated clusters to be slightly more similar to one another which is negligible for small values of α but can also be done on purpose using larger values, as discussed in Section 3.2.

Beta-binomial distribution

The beta-binomial distribution is an extension to the binomial distribution. Instead of a single parameter π, this distribution is parameterized by two shape parameters a, b making it more flexible than the binomial distribution where both, the mean and variance, are fixed by a single parameter. The density is given by

$$f_{kj}(x_{nj}|a_{kj}, b_{kj}) = \binom{r-1}{x_{nj}} \frac{B(x_{nj} + a_{kj}, r - 1 - x_{nj} + b_{kj})}{B(a_{kj}, b_{kj})}$$

with the beta function

$$\mathrm{B}(z_1, z_2) = \frac{\Gamma(z_1)\Gamma(z_2)}{\Gamma(z_1 + z_2)}.$$

While maximum likelihood estimation is not available in closed form for the beta-binomial distribution, one can employ a simple gradient descent method (Kondofersky 2008). The partial derivatives of the log density are given by

$$\frac{\partial}{\partial a_{kj}} \log f_{kj}(x_{nj}|a_{kj}, b_{kj}) = \psi(x_{nj} + a_{kj}) - \psi(r - 1 + a_{kj} + b_{kj}) - \psi(a_{kj}) + \psi(a_{kj} + b_{kj}),$$
$$\frac{\partial}{\partial b_{kj}} \log f_{kj}(x_{nj}|a_{kj}, b_{kj}) = \psi(x_{nj} + b_{kj}) - \psi(r - 1 + a_{kj} + b_{kj}) - \psi(b_{kj}) + \psi(a_{kj} + b_{kj}),$$

where the digamma function is defined as the logarithmic derivative of the gamma function

$$\psi(z) = \frac{\mathrm{d}}{\mathrm{d}z} \log \Gamma(z).$$

To avoid degenerate solutions, we can impose regularization in a similar vein as before, i.e., corresponding to adding α observations equal to the population mean to each component. We obtain estimates for the population mean simply by maximizing the unweighted log-likelihood. We then regularize the weighted log-likelihood by adding to the weights $\hat{z}_{nk}$ a constant weight α/N and multiplying these with the component-specific likelihood contribution of each observation. Thus we obtain the regularized log-likelihood contribution $\log L_{kj}$ for component k and variable j as

$$\log L_{kj}(\boldsymbol{x}_j|a_{kj}, b_{kj}) = \sum_{n=1}^{N} \left(\hat{z}_{nk} + \frac{\alpha}{N}\right) \log f_{kj}(x_{nj}|a_{kj}, b_{kj}),$$

where $\boldsymbol{x}_j = (x_{1j}, \ldots, x_{Nj})$.
This results in the following partial derivatives of the regularized log-likelihood contributions:

$$\frac{\partial}{\partial a_{kj}} \log L_{kj}(x_j|a_{kj}, b_{kj}) = \sum_{n=1}^{N} \left(\hat{z}_{nk} + \frac{\alpha}{N}\right) \left(\frac{\partial}{\partial a_{kj}} \log f_{kj}(x_{nj}|a_{kj}, b_{kj})\right),$$
$$\frac{\partial}{\partial b_{kj}} \log L_{kj}(x_j|a_{kj}, b_{kj}) = \sum_{n=1}^{N} \left(\hat{z}_{nk} + \frac{\alpha}{N}\right) \left(\frac{\partial}{\partial b_{kj}} \log f_{kj}(x_{nj}|a_{kj}, b_{kj})\right).$$

This implies that only the weights $\hat{z}_{nk}$ used for the maximum likelihood estimator need to be changed to impose this kind of regularization. Hence the (unregularized) maximum likelihood estimator results as a special case of the regularized maximum likelihood estimator and the same implementation of the estimator can be used in both cases. Given that this approach only changes the weights and is agnostic of the specific component density f_{kj} used, one could also implement this for other component-specific distributions.

3. Simulation study

3.1. Input data

Fop, Smart, and Murphy (2017) conducted latent class analysis with variable selection on a data set of 464 patients containing the information whether 38 different symptoms of lower back pain were detected for each patient. They clustered the multivariate binary data using a finite mixture of independent Bernoulli distributions. They were able to retrieve the three diagnosis types (*Central Neuropathic*, *Peripheral Neuropathic*, and *Nociceptive*) assigned by experts to the patients (Smart, Blake, Staines, and Doody 2010, 2011), after selecting a subset of 11 symptom variables out of the 38 variables. These 11 symptom variables contain the essential information for retaining the cluster structure while approximately fulfilling the conditional independence assumption. We use this data set as the basis for our simulation

study because of its reasonably well-defined cluster structure along roughly two axes, for which the *true cluster memberships* are known through the expert diagnoses.

3.2. Study design

In our present work, we fit mixture models to the data set presented in Fop *et al.* (2017) using the 11 variables they selected. We use these fitted models as generative models to create artificial data with a known clustering structure in the simulation study. Similar to Fop *et al.* (2017), we also fit finite mixtures with three components where the component distributions are multivariate independent Bernoulli distributions. In order to be able to obtain clustering structures of different difficulty, we impose different degrees of regularization on the component distributions to shrink the component distributions to a common mean distribution to a varying extent. We then generate data by drawing from the fitted mixture models using binomial distributions for the components where the parameter for the number of trials is varied to reflect different lengths of the ordinal response scales.

In this way, we obtain simulated data sets consisting of multivariate ordinal variables that have the following characteristics:

(1) The data sets are obtained based on a realistic setup for data collected in surveys with multi-item ordinal answer scales, given that pain scales in medical research are a common ordinal measuring tool in this context.

(2) The data sets exhibit a clustering structure similar to a structure encountered in real data.

(3) The data sets may vary regarding their sample size N, the number of levels of the ordinal response scale r, the number of available variables m, and the difficulty of extracting the cluster structure via the regularization parameter α.

The true cluster memberships of the observations are known in the artificial data sets via the simulation process where first the cluster memberships are drawn from a multinomial distribution.

Fitting unregularized component distributions to the original binary data set resulted in a generative model where the clustering structure is quite pronounced and the three groups can easily be extracted regardless of the clustering method applied. In order to be able to adjust the difficulty of the clustering problem we fitted regularized component distributions shrinking the component means and thus inducing a higher overlap between components. In Section 2.2, we considered regularized versions of the component distributions in order to avoid boundary or degenerate solutions. For such a use case, one usually only adds a small degree of regularization corresponding to adding only a small number of artificial observations. For the simulation study, the intention is to impose suitable regularization such that the components of the resulting mixture models vary considerably in their degree of separateness. To obtain rather difficult clustering problems, a large degree of distortion of the original cluster structure needs to be achieved and we have to select rather large values of up to $\alpha = 150$ for the regularization parameter to obtain data sets with a rather diffuse and challenging cluster structure. After having fit regularized binomial mixtures for each selected difficulty (i.e., corresponding to a specific α value), we simulate data sets from the generative process resulting from the fitted parameters. This strong degree of regularization is only imposed when fitting the finite mixture model to determine the generative process, but not when assessing the performance of the different clustering approaches in the simulation study.

We vary the number of variables in the simulation study by using the subset of the most important variables determined by a ranking of the variables for the given 11 variables. We determine a simple variable importance measure for each variable by fitting simple multinomial regression models where the true clustering of the original data is the dependent variable

and one of the 11 variables is the independent variable. The log-likelihood of the fitted models is used to rank variables in their importance. The importance measure is thus only computed once on the original data set and the ranking of variables is identical across all iterations of the simulation. For each simulated data set, we select the m variables of highest "importance" in the original data set.

We investigate the influence of varying the following characteristics:

- the regularization parameter $\alpha \in \{0, 75, 150\}$, where more regularization results in more diffuse clusters;
- the sample size $N \in \{50, 200, 500\}$;
- the number of response levels of each ordinal variable $r \in \{2, \ldots, 11\}$; and
- the number of variables $m \in \{3, 6, 11\}$, with the clustering problem being easier with more variables.

We create 100 data sets for each combination of these factors. This results in total in 27,000 data sets to which we applied the methods presented in Section 2 using $K = 3$, i.e., assuming that the true number of clusters is known. Each method based on the *flex*-scheme is applied using the default settings in packages **flexclust** and **flexmix** for initialization as well as assessing convergence.

In package **flexclust**, the K-centroid algorithm is initialized by randomly selecting K distinct observations from the data set. Package **flexmix** assigns a-posteriori probabilities based on random component assignments to K components where a weight of 0.9 is assigned to this component and 0.1 to all other components and these weights are then re-scaled to sum to one. Each algorithm is initialized randomly 10 times and the best solution obtained is retained (corresponding to within-cluster sum of distances in the partitioning case and the log-likelihood in the model-based case). The maximum number of iterations is set to the default value of 200. In contrast to the *flex*-implementations, the number of re-starts is set to one when applying the PAM algorithm. Each application of a clustering procedure results in a partition of the data and the clustering performance is measured by comparing the obtained partition to the *true cluster memberships* based on the Adjusted Rand Index (ARI; Hubert and Arabie 1985).

3.3. Results

The simulation results are summarized in Figures 1, 2 and 3. Each figure shows box plots of the ARI values across the 100 clustering results against the *true clusters* obtained for each method and data set characteristics defined by a specific combination of N, m, and r. The regularization strength α is varied across the three figures, i.e., the difficulty of the clustering problem increases from Figure 1 to Figure 3. Outliers outside 1.5 times the inter-quartile range are omitted from the box plots to improve readability.

Each of the figures shows the results for the partitioning methods on the left and for the model-based methods on the right. Among the partitioning algorithms, we only show in Figures 1, 2 and 3 the version of the `kgower` algorithm where we use the *canonical* centroid, as the alternative version shows results that are virtually equal to `kmeans`, for more details on this see below. Thus, we show 11 of the 12 evaluated algorithms in the figures. The sample size is varied in the columns and the number of variables in the rows. In each panel, the response level length is on the x-axis and the box plots showing the performance of the different methods for the same data setting are grouped together. In general, the clustering information in the data increases with the number of observations, the number of variables and the response level length.

Figure 1 shows the "easy" clustering case, where a regularization parameter of $\alpha = 0$ is used to obtain the generative model and the resulting clusters are well separated. In this "easy"

Figure 1: Simulation results on the data sets with regularization parameter $\alpha = 0$ (i.e., the "easy", well separated data case) for the evaluated methods split by data set characteristics sample size N, number of variables m, and response level length r. Listed by algorithm type (partitioning/model-based), and colored by method and scale handling type (brown shade: numerical coding; pink shade: nominalization; blue-green shade: respecting ordinal scales).

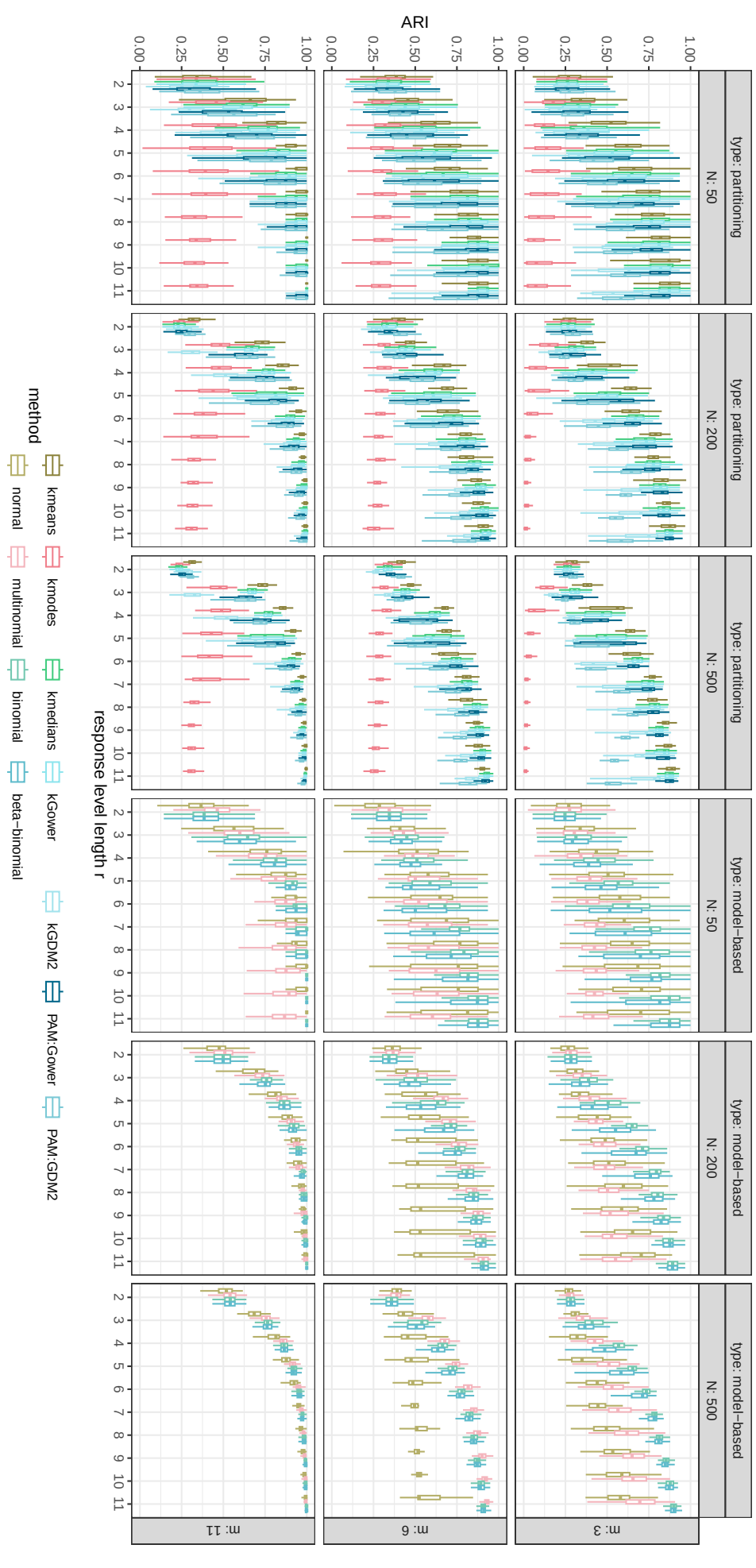

Figure 2: Simulation results on the data sets with regularization parameter $\alpha = 75$ (i.e., the "intermediate", medium-well separated data case) for the evaluated methods split by data set characteristics sample size N, number of variables m, and response level length r. Listed by algorithm type (partitioning/model-based), and colored by method and scale handling type (brown shade: numerical coding; pink shade: nominalization; blue-green shade: respecting ordinal scales).

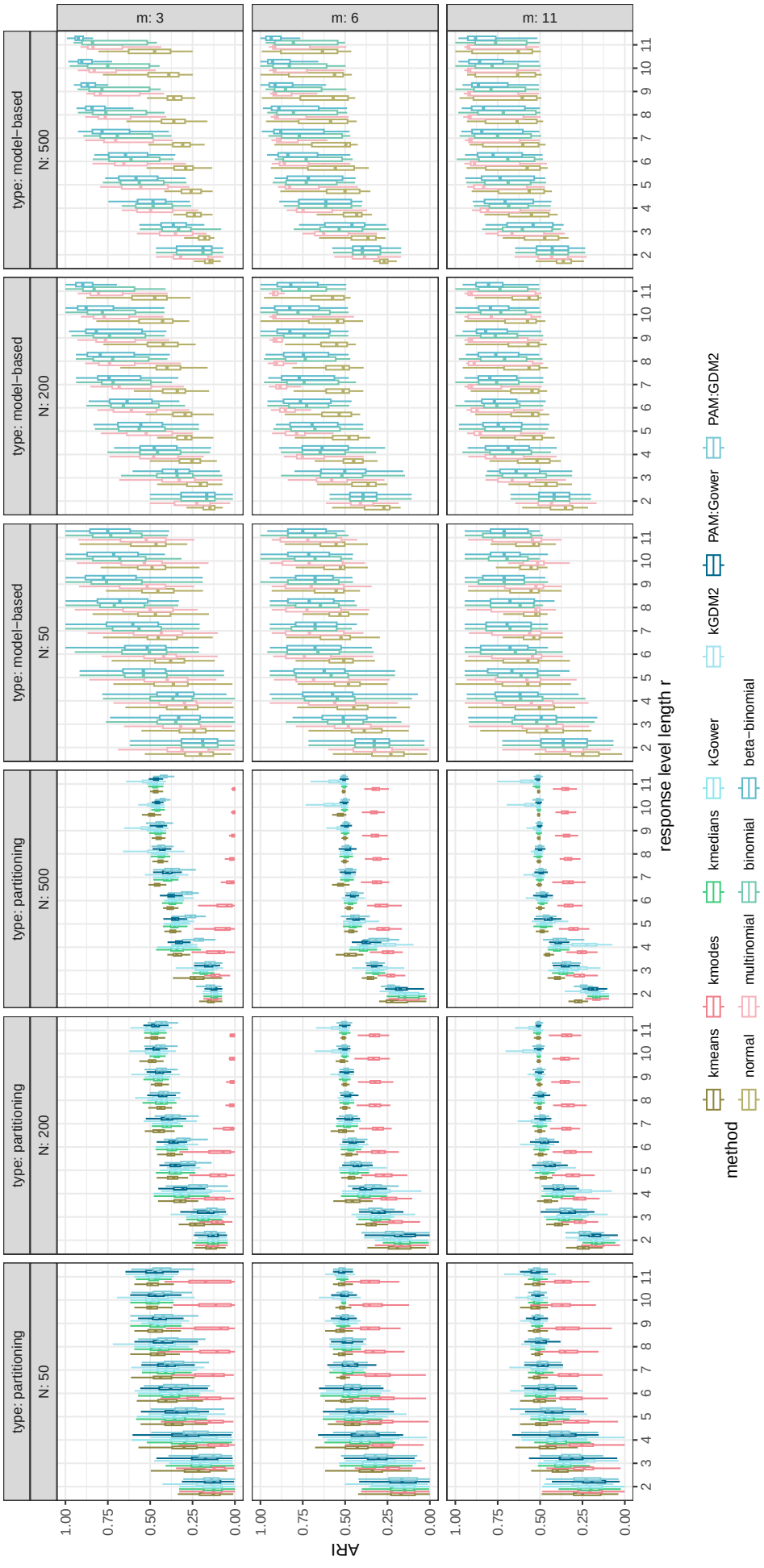

Figure 3: Simulation results on the data sets with regularization parameter $\alpha = 150$ (i.e., the "difficult", more diffuse data case) for the evaluated methods split by data set characteristics sample size N, number of variables m, and response level length r. Listed by algorithm type (partitioning/model-based), and colored by method and scale handling type (brown shade: numerical coding; pink shade: nominalization; blue-green shade: respecting ordinal scales).

case, we can see that most algorithms manage to retain the true cluster structures well (with a median ARI of around 1). This holds in particular true for data sets containing many variables and a high response level length regardless of the sample size. In these cases only using `kmodes`, i.e., imposing a nominal scale, for the partitioning methods induces solutions where the ARI is not around 1. In general retaining the true cluster structure is harder when fewer variables are available.

With respect to the partitioning methods, it can be observed that `kmodes` struggles with higher response levels regardless of sample size and number of variables and also the methods based on the GDM2 distance perform considerably worse than those using either numeric coding or making stronger assumptions about differences between scale levels for the same number of observations and variables, and response level length. Overall, `kmeans` as well as `kmedians` and `kGower` provide good results in case the response level length is at least five.

The model-based methods in general perform poorly for three variables in combination with a multinomial distribution for the components, while the performance is very good in case the response level length is at least five and a method is used which either uses numeric coding or respects the ordinal scale. In contrast, for six variables using regularized normal distributions for the components consistently results in poor clustering performance. In this case using either a nominal scale or respecting the ordinal scale gives good results in case the response level length is at least five.

The tendencies described above are very similar for the "intermediate" clustering case with regularization parameter $\alpha = 75$, which is shown in Figure 2. Naturally, ARIs are on average lower and show more variability, with a higher variability for partitioning methods than for model-based ones. In general, clustering solutions obtained with methods where ordinal scales are respected tend to outperform numerical coding and/or nominalization. Again, for partitioning clustering, nominalization via `kmodes` gives extremely poor results. Also imposing strict assumptions regarding ordinality leads to worse results than numerical coding or applying more lax assumptions regarding ordinality. In the intermediate case, `PAM:Gower` (partitioning around medoids via Gower's distance) is slightly more unstable in several data situations than `kmeans` or `kmedians`, but overall, the performance is very comparable.

For the "difficult" case with more diffuse clusters and a regularization parameter of $\alpha = 150$ (see Figure 3), model-based methods tend to outperform partitioning methods and numerical coding does a bit worse on average than either nominalization or respecting ordinal scales. Among the partitioning methods, nominalization should be avoided. The methods that respect ordinal scales all show very similar performance (with `PAM:GDM2` doing slightly worse), while numerical coding via `kmeans` tends to perform similarly if not better.

In summary, the results obtained across the three evaluated levels for α (i.e., the different difficulty levels for clustering) as well as across the different values for N, m, and r indicate that respecting ordinal scales via model-based clustering using either a binomial or beta-binomial distribution shows good clustering performance in most of the evaluated data scenarios when taking both the median ARI and the range of the ARI into account. The results of these two methods slightly differ depending on the specific data situation but overall, the results are very similar (with equal ARIs for 49.1% of the data sets, and ARIs that are within ± 0.05 of each other for 73.0% of the data sets).

Also, nominalization via model-based clustering with a multinomial distribution tends to produce good results. In comparison to `kmeans`, which is the best contender among the partitioning methods, model-based clustering via a multinomial distribution produces clustering with an equal or higher ARI for 58.2% of the data sets.

Among the partitioning methods, numerical coding via `kmeans` is the algorithm that provides the best results in most data situations, with `kmedians` in general being a close second. For the partitioning methods that rely on respecting the ordinal scale, we take a closer look at the results using Gower's distance and the GDM2 distance. We applied Gower's distance in combination with canonical centroids calculated via a numerical optimizer and pragmatic

centroids via medians for K-centroids clustering; as well as via medoids as pragmatic centroids via PAM. `kGower` using medians as centroids brings the same results as `kmedians` in 96.4% of cases and can thus be considered equivalent in this context (and is thus excluded from the figures). `kGower` using a canonical centroid outperforms its PAM pendant for 43.9% of the cases. This happens especially for data sets with a higher response level length r.

Regarding the GDM2 distance, we have results using PAM as well as K-centroids clustering with the mode as pragmatic centroid. As already discussed, very stringent assumptions regarding ordinality bring worse results in the partitioning case, and the overall comparison shows that these two algorithms only manage to beat `kmodes`. Comparing the two indicates that `kGDM2` has a very slight edge of being equal or better for 56.4% of the cases. The K-centroids clustering approach tends to do better in situations with a higher response level length r, while PAM tends to do better in situations with a lower r.

3.4. Discussion

We evaluated the following methods, categorized by their assumptions towards ordinal scales, in our simulation study:

(a) *Numerical coding:* We used a regularized normal distribution for model-based clustering and `kmeans` in the partitioning case.

(b) *Nominalization:* We used a regularized multinomial distribution for model-based clustering and `kmodes` in the partitioning case.

(c) *Respecting ordinal scales:* We compare two distributions, specifically binomial and beta-binomial distributions for model-based clustering; and among the partitioning methods we compare 4 K-centroids algorithms (K-medians; K-centroids clustering with ordinal Gower's distance, both in combination with a numerically optimized centroid, as well as with cluster-wise medians as centroids; K-centroids clustering with GDM2 distance in combination with modes as centroids) and 2 PAM algorithms (using ordinal Gower's distance and GDM2 distance).

The following methods have previously not been implemented in the *flex*-scheme and are now available in package **flexord**:

Model-based clustering: regularizations for beta-binomial, normal and multinomial distributions.

Partitioning clustering: `kmodes` based on the simple matching distance and modes as centroids, `kGower` using the ordinal Gower's distance with adapted numerical optimizers for obtaining the centroids, and `kGDM2` based on the pairwise GDM2 distance. Furthermore, this is the first work where the GDM2 distance has been reformulated to be applied in the context of K-centroids clustering with arbitrary centroids.

In our simulation study, model-based methods tend to outperform partitioning methods, and among the model-based methods, using distributions that respect ordinal scales bring the best results. However, we have to keep in mind that the simulated data sets were drawn from this data generation process, giving these methods an advantage over the other methods. Regarding the two model-based methods respecting ordinal scales, the simulation study indicates that – while results are very similar across all evaluated data situations –, using a binomial distribution tends to bring better results in cases where the generative model corresponds to binomial component distributions which are either fitted not regularized at all, or only slightly regularized, while using a beta-binomial distribution provides superior results when the simulated data sets are clustered more diffusely.

Other methods that did comparably well in retaining the *true cluster memberships* were model-based clustering via a multinomial distribution, and partitioning clustering via `kmeans`

(i.e., using a numerical coding) followed by `kmedians`. Performance degrades for model-based clustering via a regularized normal distribution; and partitioning clustering with distances and centroids that respect ordinal scales. The worst results are obtained when performing `kmodes` clustering (i.e., using a partitioning method that relies on nominalization).

For model-based clustering via a multinomial distribution, the ARIs obtained are very close to those obtained when clustering using a binomial or beta-binomial distribution for the components. There is only one noticeable exception: The multinomial distribution severely underperforms in data cases that were simulated without any regularization and use only three variables. When investigating the distribution of ARIs more closely in these cases, we can see that the ARIs for the multinomial distribution are bimodal. This may be due to identifiability issues in case only three variables are observed; and the multinomial distribution might be most strongly affected by this as it is the distribution that uses the least information from the data and contains a relatively high number of parameters.

Another interesting observation is that the conclusions regarding the scale treatment of ordinal variables completely differ between the partitioning and the model-based clustering approaches. Respecting ordinal scales and nominalization are a good choice for model-based clustering, but less so for partitioning clustering. For partitioning clustering, algorithms that apply numerical coding or have comparably lax assumptions regarding ordinality tend to perform better. Mitigating factors here might be (1) that, when clustering via a regularized normal distribution, the variance is underestimated, and thus numerical coding produces relatively subpar results in the model-based case. And (2), for the partitioning methods, it could be that in cases where the differences between the response levels of the ordinal variables are indeed not equidistant, the methods more geared towards ordinal data would have a higher chance to show their strengths.

4. Conclusion

In the present study, we exploited and extended the *flex*-scheme for partitioning and model-based clustering to investigate different clustering methods for multivariate ordinal data. We provided an overview of different options available for analysis in dependence of assumptions made regarding the scale level. We indicated how they may be included in the *flex*-scheme by extending the R packages **flexclust** and **flexmix** via package **flexord**. In total, this resulted in 12 different model-based and partitioning algorithms. We reviewed their clustering performance for ordinal variables in a simulation study evaluating 270 different combinations of data characterstics on 27,000 data sets. Our contribution to clustering research is thus twofold: on the one hand, we provide an overview and schematic categorization of both partitioning and model-based methods for clustering ordinal data and we extend the *flex*-scheme to handle new methods in this context. On the other hand, we compare their performance in an extensive simulation study.

In our simulation study, model-based methods generally outperform partitioning methods. Among the model-based methods, the strongest performers for clustering ordinal data are those that respect ordinal scales by using binomial or beta-binomial distributions. Among the partitioning methods, K-means with numerical coding and K-medians provided the best performance. The results of the simulation study also highlight the nuanced performance differences obtained depending on the scale type imposed for analysis. In general, model-based clustering benefits from respecting ordinal scales or using nominalization, while partitioning methods perform better with numerical coding. These findings underline the importance of selecting suitable clustering methods and scale handling strategies based on the specific characteristics of the data.

We designed the simulation study to use synthetic data sets generated with a realistic clustering structure where we could easily vary aspects like number of variables and response level length, but also the number of observations and the difficulty of extracting the clustering

structure. This specific study design, however, might have given the model-based approach respecting ordinal scales an unfair advantage. This aspect should be taken into consideration when assessing the results. This also suggests that further research is needed to assess if these conclusions might be generalizable to other data structures and data generating processes.

Further use and extension of the *flex*-scheme will be helpful in these future endeavors. To further aid this, we provide the methods that are newly implemented into the *flex*-scheme in the R package **flexord** (Ortega Menjivar and Ernst 2025), and share the scripts behind our simulations in the reproduction repository available from Ortega Menjivar *et al.* (2025).

Acknowledgments

The first two authors Dominik Ernst and Lena Ortega Menjivar contributed equally to this manuscript.

We would like to acknowledge the use of the BOKU nonas high-performance computing clusters to implement the simulation study.

A. Further details on the GDM2 distance

Walesiak (1993) and Walesiak and Dudek (2010) define the GDM2 distance matrix for a data set X_N with p-dimensional ordinal variables as follows for a pair of observations $\boldsymbol{x}_i$ and $\boldsymbol{x}_k$ from X_N:

$$d_N(\boldsymbol{x}_i, \boldsymbol{x}_k) = \frac{1}{2}\left(1 - \frac{\left(\sum_{j=1}^{p}\sum_{l=1}^{N} a_{ilj}a_{klj} + \sum_{j=1}^{p} a_{ikj}a_{kij}\right)}{\sqrt{\left(\sum_{j=1}^{p}\sum_{l=1}^{N} a_{ilj}^2\right)\left(\sum_{j=1}^{p}\sum_{l=1}^{N} a_{klj}^2\right)}}\right)$$

with

$$a_{ikj} = \begin{cases} 1 & \text{if } x_{ij} > x_{kj}, \\ 0 & \text{if } x_{ij} = x_{kj}, \\ -1 & \text{if } x_{ij} < x_{kj}, \end{cases}$$

where $i, k = 1, \ldots, N$ and $j = 1, \ldots, p$. This implies that $d_N(\boldsymbol{x}_i, \boldsymbol{x}_k) \in [0, 1]$ for all $\boldsymbol{x}_i, \boldsymbol{x}_k \in X_N$.

Alternatively, the GDM2 distance can also be defined based on the relative frequencies of each level of the ordinal variable j in X_N and the empirical cumulative distribution of variable j given by

$$\hat{f}_{j,N}(x) = \frac{1}{N}\#\{i = 1, \ldots, N | x_{ij} = x\}, \qquad \hat{F}_{j,N}(x) = \frac{1}{N}\#\{i = 1, \ldots, N | x_{ij} \leq x\}.$$

Re-writing the formulas containing a based on the relative and empirical frequency distributions gives for $x_{ij} = x_{kj}$:

$$\frac{1}{N}\left(\sum_{l=1}^{N} a_{ilj}^2 + a_{iij}^2\right) = \frac{1}{N}\left(\sum_{l=1}^{N} a_{ilj}^2\right) = 1 - \hat{f}_{j,N}(x_{ij})$$

and for $x_{ij} < x_{kj}$:

$$\begin{aligned}\frac{1}{N}\left(\sum_{l=1}^{N} a_{ilj}a_{klj} + a_{ikj}a_{kij}\right) &= \\ &= \hat{F}_{j,N}(x_{ij}) - \hat{f}_{j,N}(x_{ij}) + 1 - \hat{F}_{j,N}(x_{kj}) - (\hat{F}_{j,N}(x_{kj}) - \hat{f}_{j,N}(x_{kj}) - \hat{F}_{j,N}(x_{ij})) - \frac{1}{N} \\ &= 1 + 2\hat{F}_{j,N}(x_{ij}) - \hat{f}_{j,N}(x_{ij}) - 2\hat{F}_{j,N}(x_{kj}) + \hat{f}_{j,N}x_{kj} - \frac{1}{N} \\ &= 1 - 2\left(\left(\hat{F}_{j,N}(x_{kj}) - \frac{1}{2}\hat{f}_{j,N}(x_{kj})\right) - \left(\hat{F}_{j,N}(x_{ij}) - \frac{1}{2}\hat{f}_{j,N}(x_{ij})\right)\right) - \frac{1}{N}.\end{aligned}$$

When combining the case $x_{ij} < x_{kj}$ with the case $x_{kj} < x_{ij}$, this results in the following for $x_{ij} \neq x_{kj}$:

$$\begin{aligned}\frac{1}{N}\left(\sum_{l=1}^{N} a_{ilj}a_{klj} + a_{ikj}a_{kij}\right) &= \\ &= 1 - 2\left|\left(\hat{F}_{j,N}(x_{kj}) - \frac{1}{2}\hat{f}_{j,N}(x_{kj})\right) - \left(\hat{F}_{j,N}(x_{ij}) - \frac{1}{2}\hat{f}_{j,N}(x_{ij})\right)\right| - \frac{1}{N} \\ &= 1 - 2\left|\tilde{F}_{j,N}(x_{kj}) - \tilde{F}_{j,N}(x_{ij})\right| - \frac{1}{N},\end{aligned}$$

with $\tilde{F}_{j,N}$ defined as

$$\tilde{F}_{j,N}(x) = \hat{F}_{j,N}(x) - \frac{1}{2}\hat{f}_{j,N}(x).$$

Inserting this into the distance measure gives

$$\begin{aligned}&d_N(\boldsymbol{x}_i, \boldsymbol{x}_k) = \\ &= \frac{1}{2}\left(1 - \frac{\left(\sum_{j=1}^{p}(1 - \delta_{\{x_{ij}\neq x_{kj}\}}(\frac{1}{N} + 2\left|\tilde{F}_{j,N}(x_{kj}) - \tilde{F}_{j,N}(x_{ij})\right|) - \delta_{\{x_{ij}=x_{kj}\}}\hat{f}_{j,N}(x_{ij}))\right)}{\sqrt{\left(\sum_{j=1}^{p}(1 - \hat{f}_{j,N}(x_{ij}))\right)\left(\sum_{j=1}^{p}(1 - \hat{f}_{j,N}(x_{kj}))\right)}}\right),\end{aligned}$$

where $\delta_{\text{condition}}$ is 1 if the condition holds and 0 otherwise.
Based on this, one can define for arbitrary values $\boldsymbol{x} \in \mathscr{X}$ and $\boldsymbol{c} \in \mathscr{C}$:

$$d_N(\boldsymbol{x}, \boldsymbol{c}) = \frac{1}{2}\left(1 - \frac{\left(\sum_{j=1}^{p} 1 - \delta_{\{x_j\neq c_j\}}(\frac{1}{N} + 2\left|\tilde{F}_{j,N}(x_j) - \tilde{F}_{j,N}(c_j)\right|) - \delta_{\{x_j=c_j\}}\hat{f}_{j,N}(x_j)\right)}{\sqrt{\left(\sum_{j=1}^{p}(1 - \hat{f}_{j,N}(x_j))\right)\left(\sum_{j=1}^{p}(1 - \hat{f}_{j,N}(c_j))\right)}}\right).$$

Based on this alternative definition, the GDM2 distance may be determined for new data points, and can thus be used in a K-centroids clustering context.

References

Agresti A (2010). *Analysis of Ordinal Categorical Data*. John Wiley & Sons. doi:10.1002/9780470594001.

Anders R, Batchelder WH (2015). "Cultural Consensus Theory for the Ordinal Data Case." *Psychometrika*, **80**(1), 151–181. doi:10.1007/s11336-013-9382-9.

Biernacki C, Celeux G, Govaert G (2003). "Choosing Starting Values for the EM Algorithm for Getting the Highest Likelihood in Multivariate Gaussian Mixture Models." *Computational Statistics & Data Analysis*, **41**(3–4), 561–575. doi:10.1016/S0167-9473(02)00163-9.

Costilla R, Liu I, Arnold R, Fernández D (2019). "Bayesian Model-Based Clustering for Longitudinal Ordinal Data." *Computational Statistics*, **34**(3), 1015–1038. doi:10.1007/s00180-019-00872-4.

Dempster AP, Laird NM, Rubin DB (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society: Series B (Methodological)*, **39**(1), 1–22. doi:10.1111/j.2517-6161.1977.tb01600.x.

Dolnicar S, Grün B, Leisch F (2018). *Market Segmentation Analysis: Understanding It, Doing It, and Making It Useful*. Management for Professionals. Springer-Verlag, Singapore. doi:10.1007/978-981-10-8818-6.

Fop M, Smart K, Murphy TB (2017). "Variable Selection for Latent Class Analysis with Application to Low Back Pain Diagnosis." *The Annals of Applied Statistics*, **11**(4), 2080–2110. doi:10.1214/17-aoas1061.

Foss AH, Markatou M, Ray B (2019). "Distance Metrics and Clustering Methods for Mixed-Type Data." *International Statistical Review*, **87**(1), 80–109. doi:10.1111/insr.12274.

Fraley C, Raftery AE (2007). "Bayesian Regularization for Normal Mixture Estimation and Model-Based Clustering." *Journal of Classification*, **24**(2), 155–181. doi:10.1007/s00357-007-0004-5.

Galindo Garre F, Vermunt JK (2006). "Avoiding Boundary Estimates in Latent Class Analysis by Bayesian Posterior Mode Estimation." *Behaviormetrika*, **33**(1), 43–59. doi:10.2333/bhmk.33.43.

Ghosal A, Nandy A, Das AK, Goswami S, Panday M (2020). "A Short Review on Different Clustering Techniques and Their Applications." In JK Mandal, D Bhattacharya (eds.), *Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018*, pp. 69–83. Springer-Verlag. doi:10.1007/978-981-13-7403-6.

Gower JC (1971). "A General Coefficient of Similarity and Some of Its Properties." *Biometrics*, **27**(4), 857–871. doi:10.2307/2528823.

Grün B, Leisch F (2007). "Fitting Finite Mixtures of Generalized Linear Regressions in R." *Computational Statistics & Data Analysis*, **51**(11), 5247–5252. doi:10.1016/j.csda.2006.08.014.

Grün B, Leisch F (2008). "FlexMix Version 2: Finite Mixtures with Concomitant Variables and Varying and Constant Parameters." *Journal of Statistical Software*, **28**(4), 1–35. doi:10.18637/jss.v028.i04.

Grün B, Leisch F (2025). ***flexmix**: Flexible Mixture Modeling*. doi:10.32614/CRAN.package.flexmix. R package version 2.3-20.

Hennig C, Liao TF (2013). "How to Find an Appropriate Clustering for Mixed-Type Variables with Application to Socio-Economic Stratification." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **62**(3), 309–333. doi:10.1111/j.1467-9876.2012.01066.x.

Hubert L, Arabie P (1985). "Comparing Partitions." *Journal of Classification*, **2**(1), 193–218. doi:10.1007/BF01908075.

Jacques J, Biernacki C (2018). "Model-Based Co-Clustering for Ordinal Data." *Computational Statistics & Data Analysis*, **123**, 101–115. doi:10.1016/j.csda.2018.01.014.

Kaufman L, Rousseeuw P (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley Series in Probability and Statistics. John Wiley & Sons, New York. doi:10.1002/9780470316801.

Kondofersky I (2008). "Modellbasiertes Clustern mit der Beta-Binomialverteilung." Bachelor's thesis, Ludwig-Maximilians-University Munich.

Leisch F (2004). "FlexMix: A General Framework for Finite Mixture Models and Latent Class Regression in R." *Journal of Statistical Software*, **11**(8), 1–18. doi:10.18637/jss.v011.i08.

Leisch F (2006). "A Toolbox for K-Centroids Cluster Analysis." *Computational Statistics & Data Analysis*, **51**(2), 526–544. doi:10.1016/j.csda.2005.10.006.

Maechler M, Rousseeuw P, Struyf A, Hubert M, Hornik K (2024). ***cluster**: Cluster Analysis Basics and Extensions*. doi:10.32614/CRAN.package.cluster. R package version 2.1.8.

McParland D, Gormley IC (2016). "Model Based Clustering for Mixed Data: clustMD." *Advances in Data Analysis and Classification*, **10**(2), 155–169. doi:10.1007/s11634-016-0238-x.

Ortega Menjivar L, Ernst D (2025). ***flexord**: Flexible Clustering of Ordinal and Mixed-with-Ordinal Data*. doi:10.32614/CRAN.package.flexord. R package version 1.0.0.

Ortega Menjivar L, Ernst D, Scharl T, Grün B (2025). "zettlchen/AJS-flexord: AJS-flexord v1.0.0." doi:10.5281/zenodo.15074617.

Podani J (1999). "Extending Gower's General Coefficient of Similarity to Ordinal Characters." *Taxon*, **48**(2), 331–340. doi:10.2307/1224438.

Preedalikit K, Fernández D, Liu I, McMillan L, Nai Ruscone M, Costilla R (2024). "Row Mixture-Based Clustering with Covariates for Ordinal Responses." *Computational Statistics*, **39**(5), 2511–2555. doi:10.1007/s00180-023-01387-9.

R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Scrucca L, Fraley C, Murphy TB, Raftery AE (2023). *Model-Based Clustering, Classification, and Density Estimation Using **mclust** in R*. Chapman and Hall/CRC. doi:10.1201/9781003277965. URL https://mclust-org.github.io/book/.

Selosse M, Jacques J, Biernacki C (2020). "**ordinalClust**: An R Package to Analyze Ordinal Data." *The R Journal*, **12**(2), 173–188. doi:10.32614/rj-2021-011.

Smart K, Blake C, Staines A, Doody C (2010). "Clinical Indicators of 'Nociceptive', 'Peripheral Neuropathic' and 'Central' Mechanisms of Musculoskeletal Pain. A Delphi Survey of Expert Clinicians." *Manual Therapy*, **15**(1), 80–87. doi:10.1016/j.math.2009.07.005.

Smart K, Blake C, Staines A, Doody C (2011). "The Discriminative Validity of "Nociceptive","Peripheral Neuropathic", and "Central Sensitization" as Mechanisms-Based Classifications of Musculoskeletal Pain." *The Clinical Journal of Pain*, **27**(8), 655–663. doi:10.1097/AJP.0b013e318215f16a.

Szepannek G, Aschenbruck R, Wilhelm A (2024). "Clustering Large Mixed-Type Data with Ordinal Variables." *Advances in Data Analysis and Classification*, pp. 1–19. doi:10.1007/s11634-024-00595-5.

Taherdoost H (2019). "What is the Best Response Scale for Survey and Questionnaire Design; Review of Different Lengths of Rating Scale/Attitude Scale/Likert Scale." *International Journal of Academic Research in Management*, **8**(1), 1–10.

Walesiak M (1993). *Statystyczna Analiza Wielowymiarowa w Badaniach Marketingowych*. Wydawnictwo Akademii Ekonomicznej.

Walesiak M, Dudek A (2010). "Finding Groups in Ordinal Data: An Examination of Some Clustering Procedures." In H Locarek-Junge, C Weihs (eds.), *Classification as a Tool for Research*, pp. 185–192. Springer-Verlag, Berlin, Heidelberg. doi:10.1007/978-3-642-10745-0_19.

Walesiak M, Dudek A (2020). "The Choice of Variable Normalization Method in Cluster Analysis." In KS Soliman (ed.), *Education Excellence and Innovation Management: A 2025 Vision to Sustain Economic Development during Global Challenges*, pp. 325–340. International Business Information Management Association (IBIMA).

Weihs C, Ligges U, Luebke K, Raabe N (2005). "klaR Analyzing German Business Cycles." In D Baier, R Decker, L Schmidt-Thieme (eds.), *Data Analysis and Decision Support*, pp. 335–343. Springer-Verlag, Berlin. doi:10.1007/3-540-28397-8_36.

Zhang Y, Cheung YM (2020). "An Ordinal Data Clustering Algorithm with Automated Distance Learning." In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6869–6876. doi:10.1609/aaai.v34i04.6137.

Affiliation:

Dominik Ernst
Institute of Statistics
BOKU University Vienna
1180 Vienna, Austria
E-mail: dominik.ernst@boku.ac.at

Lena Ortega Menjivar
Institute of Statistics
BOKU University Vienna
1180 Vienna, Austria
E-mail: lena.ortega-menjivar@boku.ac.at
URL: https://boku.ac.at/personen/person/A8C1DFF065E2C50B

Austrian Journal of Statistics
published by the Austrian Society of Statistics
Volume 54
2025

http://www.ajs.or.at/
http://www.osg.or.at/

Submitted: 2024-11-04
Accepted: 2025-04-01