Calibrating covariate informed product partition models

Garritt L. Page[1] · **Fernando A. Quintana**[2]

Received: 23 November 2016 / Accepted: 22 September 2017 / Published online: 6 October 2017

Abstract Covariate informed product partition models incorporate the intuitively appealing notion that individuals or units with similar covariate values a priori have a higher probability of co-clustering than those with dissimilar covariate values. These methods have been shown to perform well if the number of covariates is relatively small. However, as the number of covariates increase, their influence on partition probabilities overwhelm any information the response may provide in clustering and often encourage partitions with either a large number of singleton clusters or one large cluster resulting in poor model fit and poor out-of-sample prediction. This same phenomenon is observed in Bayesian nonparametric regression methods that induce a conditional distribution for the response given covariates through a joint model. In light of this, we propose two methods that calibrate the covariate-dependent partition model by capping the influence that covariates have on partition probabilities. We demonstrate the new methods' utility using simulation and two publicly available datasets.

Keywords High-dimensional covariate space · Prediction · Covariate-based clustering · Mixture of experts · Random partition models

Electronic supplementary material The online version of this article (doi:10.1007/s11222-017-9777-z) contains supplementary material, which is available to authorized users.

✉ Garritt L. Page
page@stat.byu.edu

Fernando A. Quintana
quintana@mat.uc.cl

[1] Department of Statistics, Brigham Young University, Provo, UT, USA

[2] Departamento de Estadística, Pontificia Universidad Católica de Chile, Santiago, Chile

1 Introduction

Considering global and local structure when developing statistical methodology has become more common as data structures have increased in complexity. One method of identifying both types of structure is by grouping observations or individuals into smaller subpopulations. These subpopulations can be identified by partitioning n individuals into k subgroups. A family of probability distributions whose support is the space of all possible partitions of n individuals into k groups and that has been employed in a variety of modeling settings is the family of product partition distributions (Hartigan 1990). This family of distributions assigns probabilities to partitions by way of a cohesion function that is often a function of cluster size and measures the tightness of cluster members.

In many applications, covariates are also measured on each individual or unit, and it would be natural to include this information when identifying subpopulations. One reasonable way to achieve this is to include covariate information when defining a prior distribution on partitions. As a result, individuals with similar covariate values would have higher probability of co-clustering a priori. This idea is carried out in Blei and Frazier (2011) who consider pair-wise distances between covariates in their distance-dependent Chinese Restaurant process. However, the procedure they develop does not directly model partitions; rather, a probability distribution on graphs is constructed, which induces a probability model on partitions. Along these same lines, work has been dedicated to making discrete nonparametric Bayes prior distributions covariate dependent, all of which induce a partition model that depends on covariates. Many of these methods are based on the dependent Dirichlet process (DDP) of MacEachern (2000) (see also Barrientos et al. 2012) which makes the weights and/or atoms found in the stick-breaking representation of a random probability measure covariate dependent (see for example Iorio et al. 2004; Griffin and Steel 2006; Dunson and Park 2008; Gelfand et al. 2005). An alternative approach to the DDP developed by Müller et al. (1996) is to model covariate and response jointly using a Dirichlet process mixture model (DPM) and employ the induced conditional. This approach was used in Rodriguez et al. (2009), Hannah et al. (2011) and Antoniano-Villalobos and Walker (2016) among others. Wade et al. (2014) provides a review of both methods.

We focus on methods that model partitions directly so that the covariates explicitly influence clustering. For example, similar to the distance-dependent Chinese Restaurant process, Dahl et al. (2016) propose a procedure that incorporates covariate information via pair-wise distances, but also explicitly define a probability distribution over partitions. They penalize partitions whose clusters contain members with covariate values that have large pair-wise distances. Park and Dunson (2010) extend the product partition distribution by constructing a cohesion function that is covariate dependent, but in the process treat covariates as random quantities and are somewhat computationally restricted in how sparse covariate vectors are penalized. We focus on the covariate-dependent product partition distribution of Müller et al. (2011) (here after PPMx) who instead of altering the cohesion, introduce a so-called similarity function that measures compactness of cluster-specific covariates and adjusts the partition probabilities specified by the product partition distribution to favor partitions with clusters that have compact covariates.

The PPMx has been successfully employed in a variety of settings when a relatively small number of covariates are available. See, for example, Page and Quintana (2015) and Quintana et al. (2015). However, as the number of covariates grows (but not necessarily the number of observations), their influence on clustering tends to overwhelm information from the response, and as a result, partitions with either a large number of singleton clusters or one large cluster are assigned high posterior probability. As expected, this negatively impacts inference and predictions. This phenomenon has also been observed when using a DPM to model covariate and response jointly. In fact, one of the motivations for developing the enriched Dirichlet process in Wade et al. (2014) was to down weight a covariate's influence on clustering.

Variable selection is an alternative approach that attempts to accommodate a large number of covariates. In particular, Quintana et al. (2015) propose a variable selection technique based on the PPMx. Other Bayesian nonparametric variable selection methods are Barcella et al. (2016), Papathomas et al. (2012), and Chung and Dunson (2009). A very nice review of these methods is provided in Barcella et al. (2016). Bayesian profile regression Molitor et al. (2010) is another approach that is related to variable selection. They employ a two-step procedure that flexibly models covariates that identifies "important" cluster-specific covariates and then connects them to a data model.

Yet another approach to accommodating a large number of covariates is to reduce the dimensionality of the covariate space by estimating the sufficient dimension reduction or principal subspace (Cook and Weisberg 1991). The coordinates associated with projections into the principal subspace are then modeled rather than the covariates themselves (e.g., Page et al. 2013; Wang and Xia 2008; Guhaniyogi and Dunson 2015).

Our approach to capping covariate influence on clustering is from a completely different perspective. All the works previously mentioned attempt to reduce the covariate's influence on clustering either through the likelihood or by selecting "important" variables through the likelihood or by dimension reduction in the covariate space. In contrast, our approach is to temper covariate influence on clustering only through the partition prior distribution. We carry this out by calibrating the similarity functions. We also introduce an alternative partition distribution that has connections to a type of mixture of experts model that naturally caps covariate influence on partition probabilities.

In addition to proposing an extension to the PPMx that is able to accommodate an increasing p, we also aim to provide guidance regarding the employ of the PPMx in a variety of settings. This is done via an extensive simulation study.

The remainder of the paper is organized as follows. In Sect. 2, we give requisite background on product partition distributions and detail data models that will be employed. In Sect. 3, we detail the novel methods we develop that cap covariate influence on partition probabilities. Section 4 contains results from a simulation study and two applications, and Sect. 5 contains some concluding remarks.

2 Background and preliminaries

We briefly introduce the notation that will be used throughout. Let $i = 1, \dots, m$ index the m experimental units in a designed experiment or m subjects in an observational study. Further, let $\rho_m = \{S_1, \dots, S_{k_m}\}$ denote a partitioning (or clustering) of the m units into k_m subsets such that $i \in S_j$ implies that unit i belongs to cluster j. A common alternative notation that specifies a partitioning of the m units into k_m clusters is to introduce m cluster labels $s_1, \dots, s_m$ such that $s_i = j$ implies $i \in S_j$. We will use y_i to denote the ith subject's response variable with $\boldsymbol{y} = (y_1, \dots, y_m)$ denoting an m dimensional response vector. For the jth cluster's response vector, we use $\boldsymbol{y}_j^\star = \{y_i : i \in S_j\}$ with $n_j = |S_j|$, i.e., the cardinality of S_j. For notational convenience, we focus on univariate y_i. However, nothing precludes employing the methods that we develop when multivariate $\boldsymbol{y}_i$ are available. Now similar to the response, let

$\boldsymbol{x} = (x_1, \ldots, x_m)$ denote a covariate vector and $\boldsymbol{x}_j^\star = \{x_i : i \in S_j\}$ a partitioned covariate vector. When p covariates are measured on each individual, $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})$ will denote the ith individual's p-dimensional covariate vector and $\boldsymbol{x}_j^\star = (\boldsymbol{x}_{j1}^\star, \ldots, \boldsymbol{x}_{jp}^\star)$. One possible way of conceptualizing $\boldsymbol{x}_j^\star$ when p covariates are available is to connect it with a $n_j \times p$ covariate matrix. The convention of identifying cluster-specific quantities using a super script "$\star$" will be followed throughout.

2.1 Product partition distribution with covariates

As mentioned in Introduction, a product partition distribution is a discrete probability distribution for ρ_m that is comprised of a so-called cohesion function $c(S) \geq 0$ for $S \subset \{1, \ldots, m\}$. The cohesion measures the compactness of the elements in S and is used to produce the following unnormalized partition probabilities

$$P(\rho_m) \propto \prod_{j=1}^{k_m} c(S_j). \tag{1}$$

A commonly used cohesion function that connects (1) with the marginal partition model available from a Dirichlet process (DP) is $c(S_j) = M \times (n_j - 1)!$ for some positive M. This cohesion function will be employed throughout. For more details regarding other possible cohesion functions and properties of the partition product distribution, see chapter 8 of Müller et al. (2015).

When covariates are available, Müller et al. (2011) incorporate them by introducing a nonnegative function $g(\boldsymbol{x}_j^\star)$ into (1) resulting in

$$P(\rho_m|\boldsymbol{x}) \propto \prod_{j=1}^{k_m} c(S_j)g(\boldsymbol{x}_j^\star). \tag{2}$$

$g(\boldsymbol{x}_j^\star)$ is called a similarity function and measures the homogeneity of the $x_i \in \boldsymbol{x}_j^\star$ by producing larger values for x's that are more similar. Müller et al. (2011) provide a bit of guidance on selecting a similarity function for different types of covariates (e.g., continuous, ordinal, or categorical), but in theory any nonnegative function that produces larger values for more similar covariate vectors could be used.

One of the purposes of this paper is to investigate the similarity function's influence on inference and partition probabilities. Therefore, we consider a number of possible $g(\cdot)$ functions beyond suggestions made by Müller et al. (2011) in the simulations and applications of Sect. 4. We introduce all of them below, but before doing so, we briefly mention that when p-dimensional covariate vectors are available we adopt $g(\boldsymbol{x}_j^\star) = \prod_{\ell=1}^{p} g(\boldsymbol{x}_{j\ell}^\star)$. The motivation behind employing a product form of the similarity when p covariates are available is to study the PPMx's ability to detect interactions in a data-driven fashion (see Müller et al. 2013). This ability to detect interactions is explored in applications of Sect. 4.2.

The product form of the similarity function does not carry an implicit assumption of independence among covariates (indeed the covariates are not assumed to be random). That said, since it is plausible that using a multivariate similarity function could more accurately measure "distance" among the members of $\boldsymbol{x}_j^\star$, we provide some possibilities of multivariate similarities. However, for the sake of being concise we only consider one type of multivariate similarity in the simulations and application. This similarity is based on the Gower dissimilarity metric (details follow). For an example of a method that employs multivariate similarity functions based on suggestions found in Müller et al. (2011), see Page and Quintana (2016). We now list the similarity functions that will be considered.

– *Auxiliary similarity function* The original similarity function suggested by Müller et al. (2011) has the following form

$$g(\boldsymbol{x}_j^\star) = \int \prod_{i \in S_j} q(x_i|\boldsymbol{\xi}_j^\star)q(\boldsymbol{\xi}_j^\star)d\boldsymbol{\xi}_j^\star, \tag{3}$$

where $q(\cdot|\cdot)$ and $q(\cdot)$ are density functions, the selection of which depends on the covariate type. This structure is not necessarily used for its probabilistic properties, but rather as a means to measure the similarity of the covariates in cluster S_j. Notice that if $\prod_{i \in S_j} q(x_i|\boldsymbol{\xi}_j^\star)$ is thought of as $\boldsymbol{x}_j^\star$'s "likelihood" and $q(\boldsymbol{\xi}_j^\star)$ as $\boldsymbol{\xi}_j^\star$'s "prior," then $g(\boldsymbol{x}_j^\star)$ is sometimes referred to as the marginal likelihood or prior predictive. For continuous x, Müller et al. (2011) suggest using

$$q(\cdot|\boldsymbol{\xi}_j^\star) = N(\cdot|m_j^\star, v_j^\star), \tag{4}$$

where $N(\cdot|m, v)$ is a Gaussian density with mean m and variance v. Further, they suggest using (4) in two ways. The first is to fix $v_j^\star = v = \kappa_1 \hat{S}$ where $\hat{S}$ is the empirical variance of the covariate and κ_1 a user-supplied value, thus making $\boldsymbol{\xi}_j^\star = m_j^\star$ and resulting in $q(\boldsymbol{\xi}_j^\star) = q(m_j^\star) = N(m_j^\star|m_0, s_0^2)$. Alternatively, they suggest maintaining $v_j^\star$ unknown. This results in $\boldsymbol{\xi}_j^\star = (m_j^\star, v_j^\star)$ and $q(\boldsymbol{\xi}_j^\star) = q(m_j^\star, v_j^\star) = N\text{-}IG(m_j^\star, v_j^\star|m_0, k_0, v_0, n_0)$, the Normal–Inverse-Gamma density function which is parametrized so that m_0, v_0 are a priori "guesses" for $m_j^\star$ and $v_j^\star$ and k_0, n_0 the corresponding a priori "sample sizes." The second method is obviously more flexible as the covariate clusters are not forced to have the same variance. In

Sect. 4, we will consider both similarities and will refer to the first as the "Auxiliary N–N" and the second as the "Auxiliary N–NIG." For categorical x, it is natural to use a Multinomial–Dirichlet conjugate pairing resulting in $\boldsymbol{\xi}_j^\star = \boldsymbol{\pi}_j^\star$ and $q(\cdot|\boldsymbol{\xi}_j^\star) = q(\cdot|\boldsymbol{\pi}_j^\star) = \text{Multinomial}(\cdot|\boldsymbol{\pi}_j^\star)$ and $q(\boldsymbol{\xi}_j^\star) = q(\boldsymbol{\pi}_j^\star) = \text{Dirichlet}(\boldsymbol{\pi}_j^\star|\boldsymbol{a}_j)$. We follow suggestions of Müller et al. (2011) and set $\boldsymbol{a}_j$ to a C-dimensional vector of 0.1 s where C is the number of categories. A multivariate Auxiliary similarity is very straightforward as $q(\cdot|\boldsymbol{\xi}_j^\star)$ extends naturally to this case. See chapter 8 of Müller et al. (2015).

- *Double Dipper similarity function* Quintana et al. (2015) introduce a similarity function that retains the same form as (3), but puts more weight on local covariate structure. This is done by weighting $\boldsymbol{x}_j^\star$'s "likelihood" with the "posterior" of $\boldsymbol{\xi}_j^\star$ instead of its "prior." Thus, changing (3) to

$$g(\boldsymbol{x}_j^\star) = \int \prod_{i \in S_j} q(x_i|\boldsymbol{\xi}_j^\star) q(\boldsymbol{\xi}_j^\star|\boldsymbol{x}_j^\star) \mathrm{d}\boldsymbol{\xi}_j^\star. \tag{5}$$

  Notice that the Double Dipper similarity function corresponds to $\boldsymbol{x}_j^\star$'s "posterior predictive." Since the covariates are used twice (but not the response), Quintana et al. (2015) called this similarity function the "Double Dipper." As with the auxiliary similarity, when x is continuous we treat $v_j^\star$ of $\boldsymbol{\xi}_j^\star = (m_j^\star, v_j^\star)$ in two different ways. The first is setting $v_j^\star$ to a known value, and the second is assuming it unknown. The former will be referred to as the "Double Dipper N–N," whereas the latter will be referred to as "Double Dipper N–NIG." For a categorical covariate, the same Multinomial–Dirichlet conjugate pair employed for the auxiliary similarity will be used. Just as with the Auxiliary similarity, a multivariate form of the Double Dipper similarity is readily available.

- *Cluster variance/entropy similarity function* This similarity uses the empirical cluster variance for continuous covariates and entropy for categorical covariates to measure tightness of $x_i \in \boldsymbol{x}_j^\star$, and has the following form

$$g(\boldsymbol{x}_j^\star) = \exp\{-\alpha H(\boldsymbol{x}_j^\star)\}. \tag{6}$$

  For continuous covariates, $H(\boldsymbol{x}_j^\star) = (1/n_j) \sum_{\ell \in S_j} (x_\ell - \bar{x}_j)^2$, and for the categorical case, we use the entropy of the empirical relative frequencies, $H(\boldsymbol{x}_j^\star) = -\sum_{c=1}^{C} \hat{p}_{cj} \log(\hat{p}_{cj})$, where $\hat{p}_{cj}$ denotes the proportion of observations in the jth cluster that belong to the cth category, with the convention that $0\log(0) = 0$. Here, α is a penalty parameter that provides user input on how much to penalize dissimilar covariate values, akin to the temperature parameter of Dahl et al. (2016)'s method. As α increases, clusters with disperse $\boldsymbol{x}_j^\star$ are penalized more heavily, resulting in clusters with more homogenous $\boldsymbol{x}_j^\star$. The appeal of this similarity function is simplicity. A multivariate similarity in this case can be constructed by considering the determinant of cluster-specific covariance matrices or multivariate entropy.

- *Gower dissimilarity similarity function* This similarity function employs pair-wise Gower's dissimilarity (Gower 1971) values to measure the closeness of the individual covariate vectors. Therefore, it is similar in spirit to ideas employed in Dahl et al. (2016). The appeal of using Gower's dissimilarity metric is that all covariate types are included in its calculation. As a result, this similarity may be considered multivariate. Letting $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ be defined as the Gower dissimilarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ (specific details regarding its calculation are provided in the "Appendix"), one possibility of employing Gower's dissimilarity is summing all cluster-specific pair-wise Gower dissimilarities, producing the following

  - *Total Gower dissimilarity*

$$g(\boldsymbol{x}_j^\star) = \exp\left\{-\alpha \sum_{\substack{\ell,k \in S_j \\ \ell \neq k}} d(\boldsymbol{x}_\ell, \boldsymbol{x}_k)\right\}. \tag{7}$$

    Since a property of Gower's dissimilarity is $0 \leq d(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq 1$ for all i, j, the total cluster-specific pair-wise Gower dissimilarity is a strictly increasing function of cluster size, and thus, (7) will favor a large number of small clusters. An alternative would be to use the average pair-wise dissimilarity resulting in

  - *Average Gower dissimilarity*

$$g(\boldsymbol{x}_j^\star) = \exp\left\{-\frac{2\alpha}{n_j(n_j-1)} \sum_{\substack{\ell,k \in S_j \\ \ell \neq k}} d(\boldsymbol{x}_\ell, \boldsymbol{x}_k)\right\}. \tag{8}$$

    For both cases, α is introduced to facilitate user input on the penalty assigned to partitions with clusters that contain covariates that are not similar. When $n_j = 1$, $g(\boldsymbol{x}_j^\star) = 1$ by definition for both cases.

- *PPM* In order to compare similarity functions to a baseline, we include the PPM model for which covariates have no influence on prior partition probabilities. The similarity function here is simply the following constant function,

$$g(\boldsymbol{x}_j^\star) = 1. \tag{9}$$

Müller et al. (2011) provide theoretically appealing justifications for considering similarity functions (3) and (5). Specifically, cluster labels are exchangeable and the partition distribution is sample size consistent. That is, the partition distribution for the $(n-1)$st individual can be derived by marginalizing over the nth individual's cluster label. Even though similarity functions (6), (7), and (8) do not guarantee sample size consistency or cluster label exchangeability, they still provide completely valid probability models on partitions.

2.2 Data model

Once a prior for ρ_{m} is specified, a data model for $\boldsymbol{y}$ must be constructed. A commonly specified data model, and one that we will employ, is $f(\boldsymbol{y}|\rho) = \prod_{j=1}^{k_m} f_j(\boldsymbol{y}^\star_j)$, where $f_j(\boldsymbol{y}^\star_j) = \int \prod_{i\in S_j} f_j(y_i|\boldsymbol{\theta}^\star_j)\mathrm{d}G_0(\boldsymbol{\theta}^\star_j)$. Here, $f(\cdot|\boldsymbol{\theta}^\star_j)$ denotes the likelihood for $\boldsymbol{y}$ and G_0 is a prior for cluster-specific parameters $\boldsymbol{\theta}^\star_j$. (Notice that independence across clusters and conditional independence within clusters is being assumed.) It is possible to write the complete model hierarchically, once cluster labels $s_1, \ldots, s_m$ are introduced, in the following way

$$\begin{aligned} y_i \mid \boldsymbol{\theta}^\star, s_i &\overset{\text{ind}}{\sim} f(\boldsymbol{\theta}^\star_{s_i}), \quad \text{for } i = 1, \ldots, m,\\ \boldsymbol{\theta}^\star_j &\overset{\text{iid}}{\sim} G_0, \text{ for } j = 1, \ldots, k_m,\\ Pr(\rho_m|\boldsymbol{x}) &\propto \prod_{j=1}^{k_m} c(S_j)g(\boldsymbol{x}^\star_j), \end{aligned} \tag{10}$$

with $\boldsymbol{\theta}^\star_1, \ldots, \boldsymbol{\theta}^\star_{k_m}$ denoting cluster-specific parameters so that $\boldsymbol{\theta}_i = \boldsymbol{\theta}^\star_{s_i}$. Notice that we only include covariate information in the prior distribution of ρ_m and not in the likelihood. This is done to more easily determine how covariates influence partition probabilities. Including covariates in the likelihood is a straightforward extension of model (10).

The computation associated with fitting (10) is based on Neal (2000)'s algorithm eight. For more details, see the computation sections of Müller et al. (2011), Page and Quintana (2015), or chapter 8 of Müller et al. (2015).

3 Controlling covariate influence on partition probabilities

As discussed earlier, a large p may result in an undesirably strong effect of the similarity function in the formation of clusters. We propose two procedures that are able to better balance the information that covariates and response have on clustering as the number of covariates increases. The first approach is to calibrate the similarity function of the PPMx. The second can be thought of as a special case of the PPMx and paired with a data model is structurally similar to a mixture of experts model. We begin by detailing similarity function calibration.

3.1 Calibrating similarity functions

Within the PPMx framework, one method of counteracting the overpowering effect of a large p on clusters is by specifying a "calibrated" similarity function that caps the influence that a p-dimensional covariate vector has on clustering. A possible way of doing this is to standardize the similarity values in the following way

– *Calibrated similarity function*

$$\tilde{g}(\boldsymbol{x}^\star_j) = \frac{g(\boldsymbol{x}^\star_j)}{\sum_{\ell=1}^{k_m} g(\boldsymbol{x}^\star_\ell)}, \tag{11}$$

where $g(\boldsymbol{x}^\star_j)$ is one of the previously detailed similarity functions, except for (7) and (8) which are essentially multivariate.

Note that the calibrated similarity function projects similarity values to the unit interval. This has the desired effect of putting an upper bound on the influence that $g(\cdot)$ has on (2). When employing (11) as a similarity function, a bit of computational care is required when updating ρ within a MCMC algorithm. For example, $\tilde{g}(\boldsymbol{x}^\star_j)$ needs to be defined when $n_j = 1$. We provide details of our approach in the "Appendix".

A second approach is to coarsen the similarity function similar to how Miller et al. (2015) coarsen the likelihood in a Bayes model as a means to robustify posterior inference. The coarsened similarity function has the following form.

– *Coarsened similarity function*

$$\tilde{g}(\boldsymbol{x}^\star_j) = g(\boldsymbol{x}^\star_j)^{1/p}, \tag{12}$$

where $g(\boldsymbol{x}^\star_j)$ is one of the previously detailed similarity functions, except for (7) and (8) which are essentially multivariate.

The calibrated and coarsened similarities temper covariate influence on the partition prior in very different ways. The calibrated similarity favors partitions with a small number of clusters. To see this, notice that by construction $\tilde{g}(\boldsymbol{x}^\star_j) = 1$ when $k_m = 1$ (partition with one cluster) and $\tilde{g}(\boldsymbol{x}^\star_j) < 1$ otherwise. Alternatively, the coarsened similarity tends to evenly distribute prior mass across partitions. A clear advantage of the coarsened similarity is that it is much easier to implement computationally.

3.2 Tempered mixture of experts partition model

3.2.1 Model definition

An alternative approach to tempering the covariates influence on partition probabilities is to develop a distribution on partitions that caps the influence of $\boldsymbol{x}_j^\star$ by standardizing a "density" rather than similarity (an idea developed through personal communications with Peter Müller). The exact form of this standardized partition model is

$$
\begin{aligned}
Pr\left(\rho_m|\boldsymbol{x},\boldsymbol{\xi}^\star\right) &\propto \prod_{j=1}^{k_m}\prod_{i\in S_j}\frac{q(\boldsymbol{x}_i|\boldsymbol{\xi}_j^\star)}{\sum_\ell q\left(\boldsymbol{x}_i|\boldsymbol{\xi}_\ell^\star\right)}\\
&= \prod_{j=1}^{k_m}\prod_{i\in S_j} w(\boldsymbol{x}_i;\boldsymbol{\xi}_j^\star). \qquad (13)
\end{aligned}
$$

Here $w(\boldsymbol{x}_i;\boldsymbol{\xi}_j^\star)$ is the covariate dependent probability that the ith subject belongs to the jth cluster, with $\sum_{j=1}^{k_m} w(\boldsymbol{x}_i;\boldsymbol{\xi}_j^\star) = 1$ for all $i = 1,\ldots,m$. $q(\cdot|\boldsymbol{\xi}_j^\star)$ plays a similar role as the "likelihood" found in (3) and (5) and measures the compactness of $\boldsymbol{x}_j^\star$. The cluster membership probabilities increase as $\boldsymbol{x}_j^\star$ becomes more homogeneous, but is capped by standardizing the contribution that each individual makes to the cluster-specific "likelihood" value. For a p-dimensional $\boldsymbol{x}_i$, $q(\boldsymbol{x}_i|\boldsymbol{\xi}_j^\star) = \prod_{\ell=1}^{p} q(x_{i\ell}|\boldsymbol{\xi}_j^\star)$ with the exact form of $q(\cdot|\boldsymbol{\xi}_j^\star)$ depending on the covariate type.

A difficulty in employing (13) is how to treat $\boldsymbol{\xi}_j^\star$. If $\boldsymbol{\xi}_j^\star$ is considered an unknown and assigned a prior distribution, then the MCMC algorithm employed to fit a model like that in (10) would be rendered doubly intractable as the normalizing constant of (13) would depend on $\boldsymbol{\xi}_j^\star$. This issue can be avoided by fixing $k_{\text{m}} = J$ where J represents an upper bound on the number of clusters and then noticing that upon marginalizing over ρ_{m} the desired conditional distribution has a workable form. Specifically, the full model under consideration (i.e., no covariates in the likelihood) can be written as

$$
\begin{aligned}
p\left(\boldsymbol{y},\rho_m,\boldsymbol{\xi}^\star,\boldsymbol{\theta}^\star|\boldsymbol{x}\right) \propto \prod_{j=1}^{J} &p(\boldsymbol{y}_j^\star|\boldsymbol{\theta}_j^\star)p(\boldsymbol{\theta}_j^\star)\\
&Pr\left(\rho_m|\boldsymbol{x},\boldsymbol{\xi}^\star\right)p\left(\boldsymbol{\xi}^\star\right),
\end{aligned}
$$

where $p(\boldsymbol{y}_j^\star|\boldsymbol{\theta}_j^\star)$ and $p(\boldsymbol{\theta}_j^\star)$ are the same likelihood/prior tandem as detailed in Sect. 2.2, and $p(\boldsymbol{\xi}^\star)$ is a prior on $\boldsymbol{\xi}^\star = (\boldsymbol{\xi}_1^\star,\ldots,\boldsymbol{\xi}_J^\star)$. Marginalizing over ρ_{m} (recall that J is fixed) produces the following conditional model

$$
p\left(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}^\star,\boldsymbol{\xi}^\star\right) \propto \prod_{i=1}^{m}\left\{\sum_{j=1}^{J} w(x_i;\boldsymbol{\xi}_j^\star)p(y_i|\boldsymbol{\theta}_j^\star)\right\}, \qquad (14)
$$

which is structurally very similar to the so-called mixtures of experts model (Jacobs et al. 1991). For this reason, we call (13) the tempered mixtures of experts (TME) partition model. Since (14) is essentially a finite mixture, it is possible to assign a prior to $\boldsymbol{\xi}_j^\star$ and use computational techniques that are commonly employed to fit Bayesian finite mixture models. Details are provided in the next section and in "Appendix".

We also considered a version of the TME model that employs a cluster-specific plugin estimate of $\boldsymbol{\xi}^\star$. In this case, the normalizing constant would not be a function of unknowns, and therefore, model fitting could be carried out with a straightforward variation in algorithm 8 of Neal (2000). However, the cluster-specific estimates of $\boldsymbol{\xi}^\star$ produced partition models that favored many small clusters resulting in very poor prediction, and because of this we consider the method no further. Finally, as mentioned, (13) can be considered a special case of (2). This can be seen by setting $c(S_j) = 1$ and $g(\boldsymbol{x}_j^\star) = \prod_{i\in S_j} w(x_i;\boldsymbol{\xi}_j^\star)$.

3.2.2 Computational details for the tempered mixture of experts model

When $\boldsymbol{\xi}_j^\star$ is considered unknown, the normalizing constant of $Pr(\rho_m)$ depends on $\boldsymbol{\xi}_j^\star$ and Algorithm 8 of Neal (2000) cannot be employed. In this case, we instead appeal to the finite mixture structure in (14) by introducing latent component labels for each individual and employ a Gibbs sampler–Metropolis–Hastings hybrid algorithm. The Gibbs steps associated with updating cluster labels and cluster-specific parameters are now well known (see McLachlan and Peel 2000). The Metropolis–Hastings steps associated with updating $\boldsymbol{\xi}_j^\star$ are provided in "Appendix". If $\boldsymbol{x}$ is a mix of continuous and discrete covariates, we employ the following priors for $\boldsymbol{\xi}^\star = (\eta_j^\star, v_j^\star, \boldsymbol{\pi}_j^\star)$: $\eta_j^\star \sim N(m_0, s_0^2)$, $\sqrt{v^\star} \sim UN(0,10)$, and $\boldsymbol{\pi}_j^\star \sim \text{Dirichlet}(\boldsymbol{a})$ with $\boldsymbol{a}$ a vector filled with 0.1.

4 Simulation studies and data examples

In this section, we provide numerical results from a simulation study conducted to assess how the calibrated similarity functions and the TME model are able to balance the influence that the response and covariates have on clustering and, ultimately, model fit and prediction. We also explore different structures among the covariates (dependence and interactions), along with sensitivity to prior parameter selection. Lastly, we discuss results from two publicly available real-world data sets.

4.1 Simulation study

To compare model fit and predictive performance of the methods described in Sects. 2 and 3, we conduct a simulation study that consists of generating data from four distinct data-generating mechanisms. Each generated data set is comprised of $m = 200$ observations with 100 being classified as testing observations and 100 as training observations. In all the simulations and applications that follow, to reduce the influence that the scale on which the covariates are measured has on inference, we standardize each covariate to have mean zero and standard deviation one.

We include data sets with a small number of covariates to compare performance (in terms of model fit, out-of-sample prediction and partition recovery) to when a relatively large number of covariates are available. The smaller (in terms of p) datasets also have interesting structure (dependence, interactions, nonlinear association with the response) which illustrates benefits of one procedure over the other. We now detail each of the four data-generating mechanisms.

- *DG1* The first method of generating datasets employs the synthetic data created for the simulation study found in section 5.2 of Müller et al. (2011). These data contain $p = 3$ covariates, two of which are binary. The covariates are generated independently, but an interaction is explicitly included in the sense that the density of y_i depends on specific combinations of the covariates. For these data, the y_i are not explicitly a function of x and no "true" clustering exists. These data are used to explore how complex interactions influence performance. From the $N = 1000$ observations that comprise the synthetic data, we create data sets used in the simulation, by randomly selecting a subset of 200 observations.
- *DG2* The second method of generating data produces data sets that contain $p = 4$ independent covariates where $x_1 \sim N(0, 1)$, $x_2 \sim UN(0, 10)$, $x_3 \sim \text{Ber}(0.5)$, and $x_4 \sim \text{Discrete}(1/3, 1/3, 1/3)$. To form clusters, x_3 and x_4 were crossed creating six groups to which unique slope (using x_1) and intercept combinations are assigned (creating an interaction and making clusters covariate dependent). When generating a response, x_2 is not included, rendering it noise in terms of clustering and the response. Here, y_i is explicitly a function of $\boldsymbol{x}_i = (x_{1i}, x_{2i}, x_{3i}, x_{4i})$ and clustering explicitly depends on covariates. We consider DG2 to explore how covariate-dependent clusters and interactions influence procedures when p is small.
- *DG3* The third method of generating data also consists of $p = 4$ covariates, and they are generated under two scenarios. The first follows DG2 such that $x_1 \sim N(0, 1)$, $x_2 \sim UN(0, 10)$, $x_3 \sim \text{Ber}(0.5)$, and $x_4 \sim \text{Discrete}(1/3, 1/3, 1/3)$ all independently. Here, we also consider dependent covariates by setting $x_1 = 2x_3 + \epsilon_1$ with $\epsilon_1 \sim N(0, 1)$, and $x_2 = -2x_4 + \epsilon_2$ with $\epsilon_2 \sim UN(0, 1)$ with x_3 and x_4 being generated as before. Clusters are not explicitly covariate dependent for this scenario in that regardless of how covariates are generated there are four clusters. Each cluster has an equal number of observations, and they are generated using a Gaussian distribution with means (− 5.0, − 2.5, 2.5, 5.0) and standard deviation 0.75. Therefore, these data contain no interaction, y_i is not a direct function of covariates, and clusters do not explicitly depend on covariates. However, the covariates are potentially dependent. We consider DG3 to study how correlated covariates influence procedures when p is small and clustering is not explicitly covariate dependent.
- *DG4* The fourth and final data-generating method is included to explore the impact that an increasingly larger p has on prediction and model fit, with $p \in \{10, 20, 50, 100, 150, 200\}$ together with clusters that depend explicitly on covariates. The p covariates are independently generated from the following five distributions: 20% of the total number of covariates (i.e., 2, 4, 10, 20, or 40) come from $N(0, 1)$, the next 20% from $UN(0, 10)$, another 20% from t_4, (t distribution with 4 degrees of freedom), 20% more from $SN(10, 1, 10)$, (a skew-normal with a, b, c denoting the mean, scale and skewness parameter, respectively) and the last 20% from a two component mixture of the form $0.4N(0, 1) + 0.6N(10, 2)$. Both correlated and uncorrelated covariates were considered. Correlated covariates were created via Gaussian copulas. The response values are generated in the same way as in DG3, and as a result there are four clusters. Here we considered the case when the clusters were covariate dependent and the case when the were not. We create the former by making covariate values for each cluster more similar relative to the other clusters. For example, for individuals that belong to the first cluster we employ $N(3, 1)$ and $UN(-5, 0)$ to generate x_1 and x_2 rather than $N(0, 1)$ and $UN(0, 1)$. More details are provided in the online supplementary material.

Table 1 summarizes the characteristics of the four types of data-generating mechanisms employed in the simulation study.

In addition to the different data-generating mechanisms, we considered two hyperparameter value combinations. For the Auxiliary N–N similarity, Müller et al. (2011) suggest setting $v = \kappa_1\hat{S} = 0.5\hat{S}$ which results in $v = 0.5$ since we are standardizing the x's. For s_0^2, they suggest using $10\hat{S}$ which results in $s_0^2 = 10$. Therefore, the first set of prior values we employ for the Auxiliary N–N and Double Dipper N–N similarities is ($m_0 = 0, s_0^2 = 10, v = 0.5$). Since we standardize x's prior to analysis, informal preliminary

Table 1 Summary of covariate structures contained in the four data-generating mechanisms that are employed in the simulation study

Dataset	Interaction	Correlated $\boldsymbol{x}$	Asymmetry $\boldsymbol{x}$	High dimension	Continuous and categorical	Inform clusters
DG1	Yes	No	No	No	Yes	No
DG2	Yes	No	No	No	Yes	Yes
DG3	No	Yes	No	No	Yes	No
DG4	No	Yes	Yes	Yes	No	Yes/no

investigations suggested that s_0^2 is the prior parameter that is most influential among m_0, s_0^2, and v (assuming reasonable values are selected for each). Thus, to explore sensitivity to prior specification we vary this parameter by considering $(m_0 = 0, s_0^2 = 1, v = 0.5)$ as a second set of prior values which favors partitions with more clusters a priori. For the Auxiliary and Double Dipper N–NIG similarities, no suggestions are given by Müller et al. (2011). Therefore, for these similarity functions the first set of priors are $(m_0 = 0, k_0 = 1.0, v_0 = 10.0, n_0 = 2)$ and the second $(m_0 = 0, k_0 = 1.0, v_0 = 1.0, n_0 = 2)$. For similarities found in (6), (7) and (8), we consider $\alpha \in \{1, 2\}$ to explore how this user-supplied value influences inference and predictions. For the TME partition model, we set prior values to $(m_0 = 0, s_0^2 = 1.0)$ and $(m_0 = 0, s_0^2 = 10.0)$ as an attempt to match priors used for the Auxiliary and Double Dipper similarities. We set the upper bound on the number of clusters in the TME to $J \in \{5, 10, 20\}$. Lastly, we did not consider multiple values for $\boldsymbol{a}$ as results are robust to its specification.

The following model was fit to each synthetic data set:

$$\begin{aligned}
&y_i|\mu_j^{\star}, \sigma_j^{2\star}, s_i \overset{\text{iid}}{\sim} N(\mu_{s_i}^{\star}, \sigma_{s_i}^{2\star}) \quad \text{with} \quad \sigma_j^{\star} \overset{\text{iid}}{\sim} UN(0, 5),\\
&\mu_j^{\star} \mid \mu_0, \sigma_0^2 \overset{\text{iid}}{\sim} N(\mu_0, \sigma_0^2) \;\; \text{with} \quad \sigma_0 \sim UN(0, 5), \qquad (15)\\
&\quad \mu_0 \sim N(0, 10^2),
\end{aligned}$$

with the added assumption that $p(\mu_0, \sigma_0^2) = p(\mu_0)p(\sigma_0^2)$ a priori. Thus, cluster label probabilities are all that change from one procedure to the next. This permits us to make clean comparisons on how each procedure assimilates covariate information when making predictions and fitting models to data. Each procedure was fit to data by collecting 1000 draws from the joint posterior distribution after discarding the first 1000 as burn-in. Maximum likelihood type estimates were used for starting values to ensure quick convergence.

We present simulation results for each data-generating scenario separately and then make some overall conclusions. Results are presented in graphical and tabular form. We use the following metrics as a means of assessing model fit, prediction and cluster production.

– *Number of clusters (#Clus)* The posterior mean number of clusters produced by the PPMx type models and the posterior mean number of occupied (non-empty) mixture components for the TME model.
– *Mean squared prediction error (MSPE)* represents the mean squared prediction error defined as $\frac{1}{100}\sum_{i=1}^{100}(Y_{pi} - \hat{Y}_{pi})^2$ where i indexes the 100 testing observations (Y_p) and $\hat{Y}_{pi} = E(Y_{pi}|\boldsymbol{Y}_o)$ (where Y_o denotes the 100 training observations). This quantity measures the predictive performance of the models.
– *Mean squared error (MSE)* represents the mean squared error defined as $\frac{1}{100}\sum_{i=1}^{100}(Y_{oi} - \hat{Y}_{oi})^2$ where i indexes the 100 training observations and $\hat{Y}_{oi}$ is the fitted value for the ith observation. This quantity measures goodness of fit.
– *Log pseudomarginal likelihood (LPML)* represents the log pseudo marginal likelihood which is a goodness-of-fit metric (see Geisser and Eddy 1979; Christensen et al. 2011) that takes into account model complexity.
– *Adjusted Rand Index (ARI)* measures the similarity between the estimated partition and that which generated the data (excluding DG1 for which one does not exist). Values range between [0, 1] with 1 indicating a perfect match (see Rand 1971).

All values listed in the tables and displayed in graphs are averages over the 100 synthetic data sets generated for each scenario. Results for the two prior specifications were very similar so we only present those that correspond to prior values suggested in Müller et al. (2011).

Results for DG1 are provided in Table 2. In terms of cluster production, the results suggest that Total Gower and the Double Dipper tend to produce the most clusters. This results in these similarities having very competitive MSE values with the Total Gower producing the smallest. However, the Total Gower dissimilarity does not perform well in terms of MSPE. It appears that the Gower dissimilarity being between zero and one prevents it from discriminating well between covariates when making predictions. Conversely, the Double Dipper maintains very good performance for out-of-sample prediction with the simplest model (N–N) performing best. MSE/LPML values for the TME models improve as the

Table 2 Simulation study results for DG1

Procedure	# Clus	MSPE	MSE	LPML
TME 5 components	5.00	615.44	24.64	– 383.39
TME 10 components	9.99	558.41	19.63	– 289.44
TME 20 components	17.72	569.83	16.63	– **135.04**
Auxiliary N–N	17.12	**543.93**	4.50	– 336.22
Auxiliary N–N calibrated	18.24	546.12	4.41	– 310.80
Auxiliary N–N coarsened	17.70	571.22	4.40	– 351.25
Auxiliary N–NIG	17.60	552.32	4.38	– 351.81
Auxiliary N–NIG calibrated	17.95	552.00	4.42	– 317.75
Auxiliary N–NIG coarsened	17.82	577.03	4.42	– 355.93
Double Dipper N–N	18.38	544.67	4.12	– 342.87
Double Dipper N–N calibrated	18.11	547.55	4.40	– 329.06
Double Dipper N–N coarsened	18.06	574.65	4.26	– 339.96
Double Dipper N–NIG	18.39	552.53	4.17	– 336.59
Double Dipper N–NIG calibrated	18.04	553.31	4.46	– 334.13
Double Dipper N–NIG coarsened	18.25	579.41	4.35	– 340.49
Cluster variance $\alpha = 1$	17.38	600.53	4.50	– 318.11
Cluster variance $\alpha = 1$ calibrated	17.66	599.46	4.57	– 323.28
Cluster variance $\alpha = 1$ coarsened	17.78	613.20	4.54	– 336.80
Cluster variance $\alpha = 2$	16.43	586.62	4.80	– 371.06
Cluster variance $\alpha = 2$ calibrated	17.84	583.45	4.57	– 329.16
Cluster variance $\alpha = 2$ coarsened	17.49	606.33	4.60	– 351.61
Mean Gower dissimilarity $\alpha = 1$	17.64	617.64	4.71	– 339.94
Mean Gower dissimilarity $\alpha = 2$	17.44	613.71	4.44	– 357.32
Total Gower dissimilarity $\alpha = 1$	19.76	591.58	3.56	– 268.45
Total Gower dissimilarity $\alpha = 2$	20.97	592.77	**2.98**	– 234.90
PPM	18.18	620.63	4.44	– 326.43

Values in columns correspond to averages over all 100 synthetic data sets
Bold numbers correspond to procedure with the best performance for associated metric

Table 3 Simulation study results for DG2

Procedure	# Clus	MSPE	MSE	LPML	ARI
TME 5 components	5.00	47.09	5.92	– 294.74	0.45
TME 10 components	9.94	5.61	0.79	– 152.29	0.69
TME 20 components	18.39	7.06	0.40	– 144.76	0.47
Auxiliary N–N	11.46	1.94	0.41	– 137.96	0.68
Auxiliary N–N calibrated	11.17	2.21	0.32	– 171.14	0.65
Auxiliary N–N coarsened	9.67	20.09	0.41	– 157.65	0.70
Auxiliary N–NIG	11.36	**1.43**	0.36	– 133.30	**0.72**
Auxiliary N–NIG calibrated	11.11	1.77	0.32	– 169.63	0.66
Auxiliary N–NIG coarsened	9.67	20.26	0.41	– 158.72	0.70
Double Dipper N–N	18.34	6.14	0.20	– 128.63	0.52
Double Dipper N–N calibrated	11.03	3.60	0.33	– 170.70	0.67
Double Dipper N–N coarsened	11.86	25.33	0.31	– 156.02	0.67
Double Dipper N–NIG	17.75	4.98	0.18	– **116.48**	0.52
Double Dipper N–NIG calibrated	11.03	2.83	0.33	– 169.78	0.65
Double Dipper N–NIG coarsened	11.54	25.28	0.32	– 152.87	0.67
Cluster variance $\alpha = 1$	8.51	37.22	0.55	– 173.74	0.70
Cluster variance $\alpha = 1$ calibrated	9.43	36.40	0.39	– 172.30	0.68
Cluster variance $\alpha = 1$ coarsened	9.33	42.76	0.46	– 174.16	0.69
Cluster variance $\alpha = 2$	8.06	31.28	0.64	– 172.30	0.70
Cluster variance $\alpha = 2$ calibrated	10.21	28.53	0.36	– 172.01	0.67
Cluster variance $\alpha = 2$ coarsened	8.96	40.72	0.50	– 174.37	0.69
Mean Gower dissimilarity $\alpha = 1$	9.57	44.07	0.44	– 175.84	0.69
Mean Gower dissimilarity $\alpha = 2$	9.30	43.12	0.46	– 174.53	0.69
Total Gower dissimilarity $\alpha = 1$	20.29	20.93	0.14	– 140.79	0.37
Total Gower dissimilarity $\alpha = 2$	24.36	15.25	**0.10**	– 134.76	0.30
PPM	9.91	44.88	0.42	– 175.28	0.69

Values in columns correspond to averages over all 100 synthetic data sets
Bold numbers correspond to procedure with the best performance for associated metric

number of components increases, but MSPE gets worse indicating an overfit. It seems that with a small p increasing the number of components for the TME leads to more cluster creations and overfitting. The Auxiliary similarity produces MSPE values that are comparable to those of the Double Dipper, but clearly is inferior when considering MSE/LPML. As expected for these data (p is small), calibrating or coarsening the similarity has little effect of reducing the number of clusters. It is interesting to note that the average number of clusters for Auxiliary, Double Dipper and cluster variance is similar to the PPM. This indicates that for small p if the covariates do not influence clustering, then the PPMx is not adversely influenced. Over all, it appears that the simple Auxiliary N–N and Double Dipper N–N do the best at balancing goodness-of-fit and out-of-sample prediction for these data

Results for DG2 are provided in Table 3. Recall that in this scenario data were generated using six clusters and that the covariates inform clustering. As a result, MSPE values are generally much better for the PPMx procedure (regardless of similarity) relative to the PPM. For these data, the Gower dissimilarity and cluster variance clearly perform the worst in terms of prediction, but do well in terms of MSE/LPML (an indication of overfit or creating too many clusters). It appears therefore that interacting covariates adversely affect prediction for these two similarities more than the other procedures. The Double Dipper produces more clusters relative to the other procedures save the Total Gower dissimilarity (which we expected). Also it appears that coarsening similarities for these data negatively impacts MSPE in a drastic way, while calibrating tends to better balance MSE/LPML and MSPE. The TME generally performed poorly. Overall, the Auxiliary similarity performs best for these data, producing very competitive MSE/LPML metrics and the best out-of-sample predictions (which is a result of producing the best ARI values).

Table 4 Simulation study results for DG3

Procedure	Independent covariates					Correlated covariates				
	# Clus	MSPE	MSE	LPML	ARI	# Clus	MSPE	MSE	LPML	ARI
TME 5 components	5.00	15.95	0.93	− 200.72	0.60	5.00	16.26	4.39	− 228.94	0.35
TME 10 components	9.99	16.62	1.08	− 216.11	0.29	7.32	17.01	2.85	− 238.80	0.22
TME 20 components	19.58	16.65	1.34	− 230.54	0.14	8.65	16.99	3.01	− 236.12	0.21
Auxiliary N–N	9.00	17.64	0.51	− 179.55	0.51	11.59	18.17	0.76	− 189.91	0.25
Auxiliary N–N calibrated	8.19	16.98	0.19	− 153.90	0.73	8.11	17.48	0.19	− 149.57	0.78
Auxiliary N–N coarsened	5.56	16.11	0.31	− 161.77	0.80	5.79	16.37	0.28	− 159.37	**0.84**
Auxiliary N–NIG	8.14	17.12	0.51	− 175.55	0.58	11.48	18.20	0.88	− 193.43	0.24
Auxiliary N–NIG calibrated	8.15	16.81	0.19	− 152.70	0.73	8.13	17.19	0.19	− 150.79	0.77
Auxiliary N–NIG coarsened	5.66	16.08	0.31	− 161.89	0.80	5.76	16.32	0.29	− 158.15	**0.84**
Double Dipper N–N	24.08	16.92	0.15	− 163.03	0.17	21.26	17.28	0.25	− 169.68	0.20
Double Dipper N–N calibrated	8.16	16.87	0.19	− 152.93	0.74	8.12	17.39	0.19	− 150.78	0.77
Double Dipper N–N coarsened	8.39	16.09	0.22	− 157.26	0.77	8.74	16.33	0.21	− 154.75	0.79
Double Dipper N–NIG	23.51	17.03	0.14	− 161.22	0.17	17.58	18.00	0.47	− 183.06	0.22
Double Dipper N–NIG calibrated	8.18	16.74	0.19	− 153.30	0.73	8.09	17.02	0.19	− 151.24	0.78
Double Dipper N–NIG coarsened	7.88	16.09	0.23	− 158.27	0.79	8.11	16.31	0.22	− 155.00	0.81
Cluster variance $\alpha = 1$	4.76	16.05	0.38	− 163.71	0.80	4.82	**16.24**	0.34	− 160.30	0.83
Cluster variance $\alpha = 1$ calibrated	7.82	16.03	0.20	− 153.56	0.75	7.73	16.27	0.20	− 151.12	0.79
Cluster variance $\alpha = 1$ coarsened	5.53	**16.01**	0.30	− 160.79	**0.82**	5.53	16.27	0.28	− 156.96	**0.84**
Cluster variance $\alpha = 2$	4.27	16.04	0.71	− 170.41	0.71	4.45	16.26	0.58	− 166.66	0.78
Cluster variance $\alpha = 2$ calibrated	8.10	16.07	0.20	− 154.00	0.73	7.99	16.26	0.19	− 151.72	0.76
Cluster variance $\alpha = 2$ coarsened	5.14	16.03	0.32	− 161.86	0.81	5.14	16.27	0.30	− 159.45	0.84
Mean Gower dissimilarity $\alpha = 1$	5.87	16.02	0.28	− 158.85	**0.82**	5.83	16.26	0.27	− 156.46	**0.84**
Mean Gower dissimilarity $\alpha = 2$	5.52	16.03	0.29	− 160.83	**0.82**	5.47	16.25	0.28	− 156.79	**0.84**
Total Gower dissimilarity $\alpha = 1$	26.98	16.08	0.05	**− 138.28**	0.13	27.00	16.35	**0.05**	**− 137.06**	0.13
Total Gower dissimilarity $\alpha = 2$	33.65	16.07	**0.04**	− 140.95	0.09	31.78	16.51	0.06	− 144.31	0.11
PPM	6.40	16.04	0.26	− 157.93	0.81	6.42	**16.24**	0.25	− 154.85	**0.84**

Values in columns correspond to averages over all 100 synthetic data sets
Bold numbers correspond to procedure with the best performance for associated metric

Results for DG3 are provided in Table 4. Perhaps, the first thing to notice is that the PPM is very competitive to the PPMx procedures in terms of prediction. Therefore, it appears that if clusters are not directly informed by the covariates, the PPMx provides little predictive benefit. This is to be expected. However, the PPMx still provides benefits relative the PPM in terms of MSE and LPML. Calibrating and coarsening the similarity function have the desired effect of reducing the number of clusters and in most cases improve MSE and LPML and prediction. In fact, it appears that calibrating or coarsening similarities counteracts negative impact of correlated covariates. The Total Gower dissimilarity is very competitive regarding MSE (winner for $\alpha = 2$), but does not predict as well. The Auxiliary and Double Dipper similarities produce very close results with Double Dipper doing slightly better both in prediction and MSE/LPML. The TME is competitive in both MSE/LPML and predictions, with five components producing the best predictions. In addition, since clustering does not depend on covariate values, we can see how calibrating the prior improves partition estimation as it should be similar to that obtained by the PPM.

Results for DG4 are provided in Figs. 1, 2, 3, 4 and 5. For ease of displaying results, these figures do not include Total Gower dissimilarity and the y-axis is unique to each row. Therefore, care must be taken when making comparisons between rows. Finally, tables of values that produced these figures are provided in a supplementary file for closer examination.

Figure 1 corresponds to the average number of clusters. The Double Dipper similarities produce many more clusters than what was used to generate data (recall four clusters were used to generate the data) as do the Auxiliary similarities, but to a lesser extent. This is true even if covariates inform clustering and is exasperated when covariates are uncorrelated and p grows. Calibrating/coarsening these two similarities has the desired effect of better regulating the number of clusters

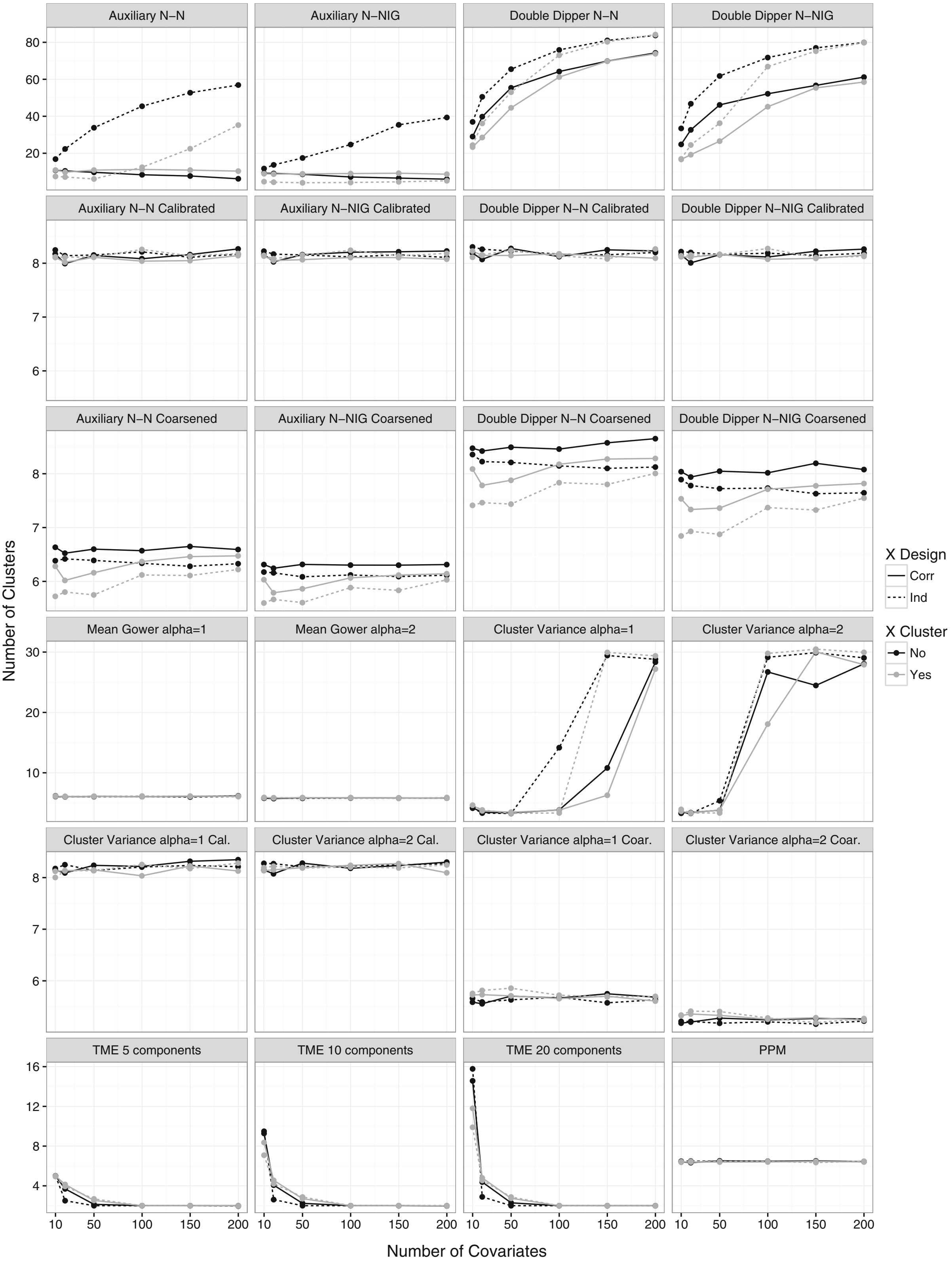

Fig. 1 Estimated number of average clusters for the DG4 data. The label "X Design" identifies if there is correlation present in the covariates while "X Cluster" indicates if covariates inform clustering. Note that the y-axis scale changes with row. Thus, comparisons between rows must be done with care

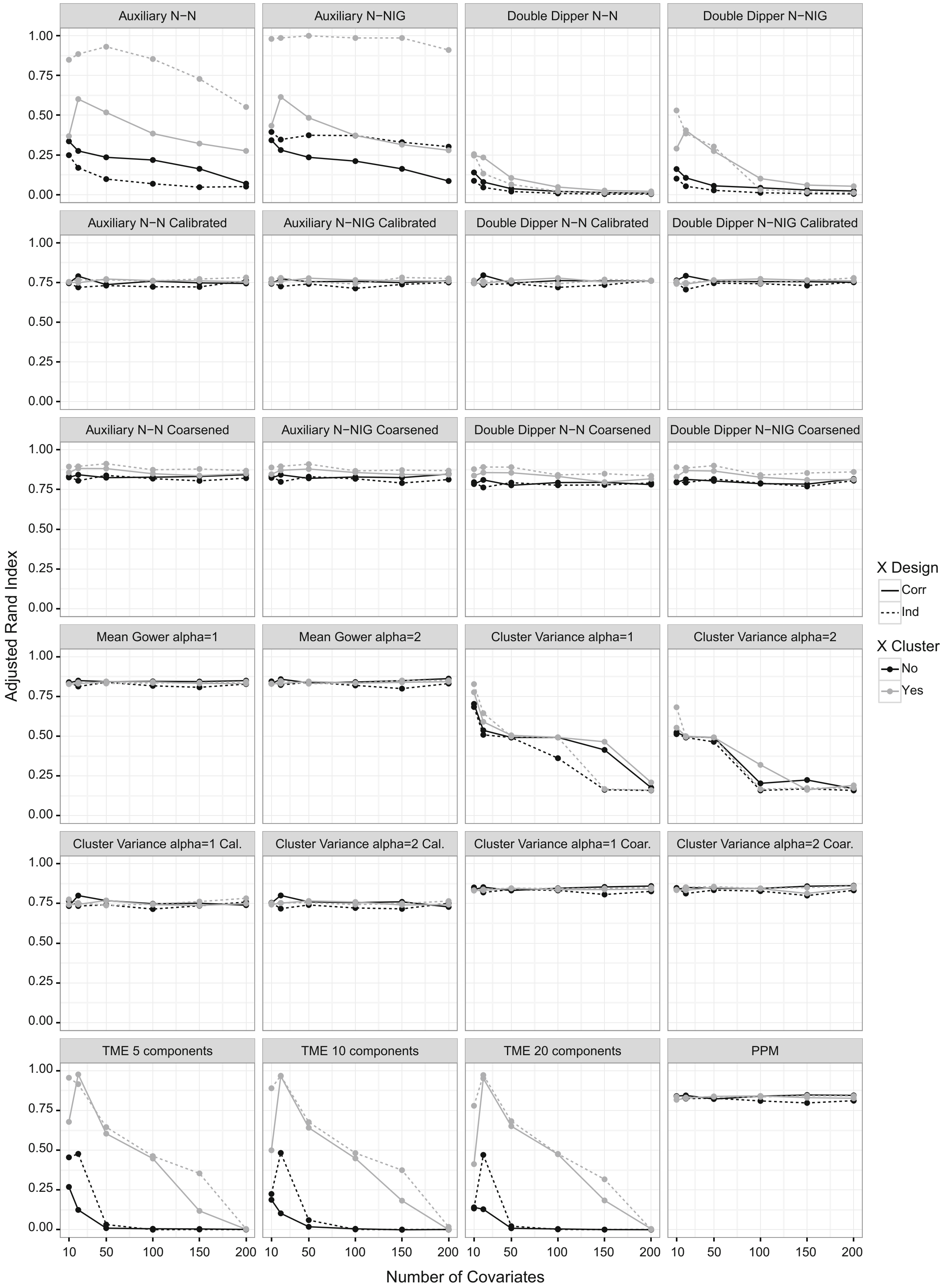

Fig. 2 ARI for the DG4 data. The label "X Design" identifies if there is correlation present in the covariates while "X Cluster" indicates if covariates inform clustering

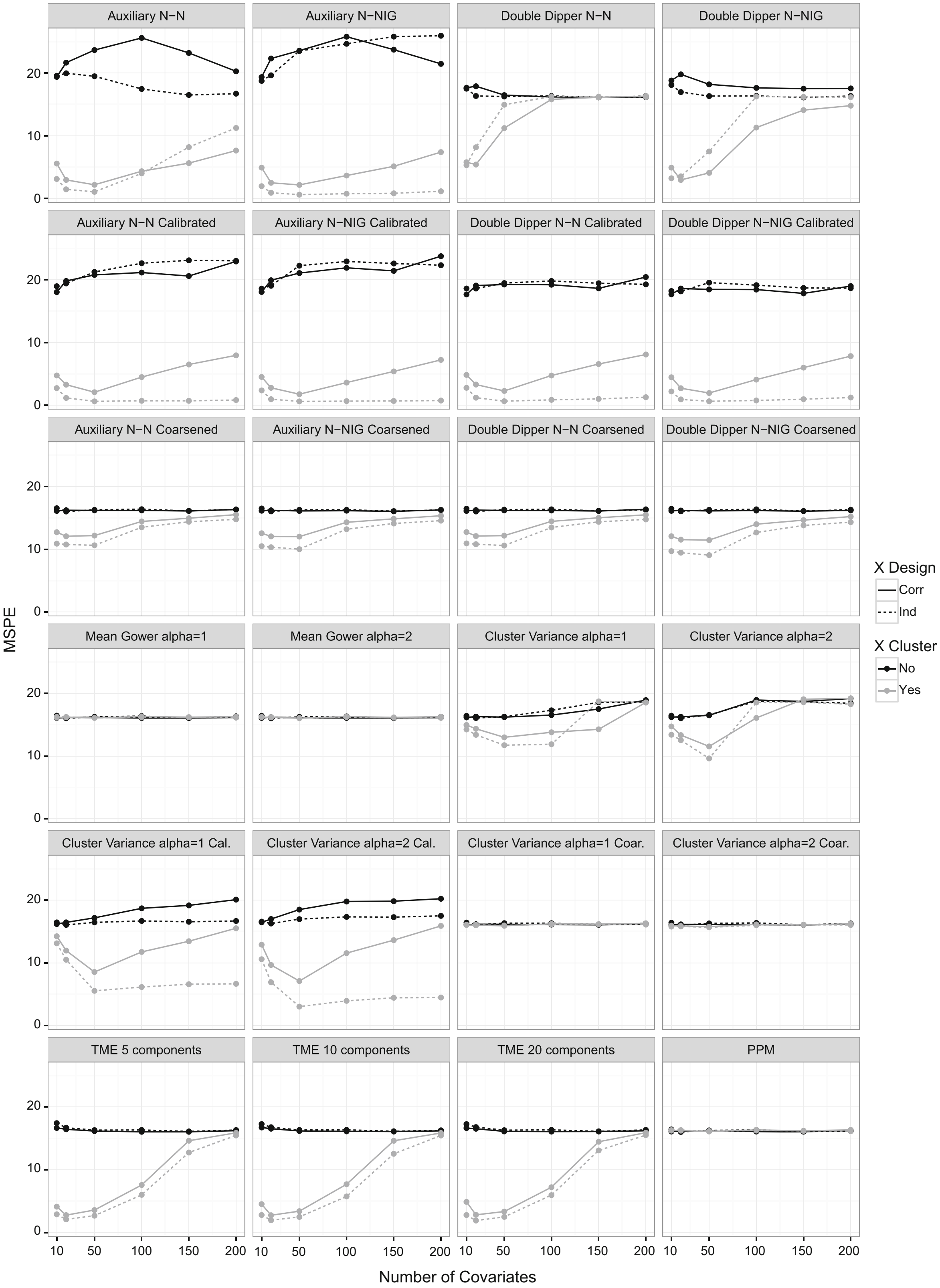

Fig. 3 MSPE for the DG4 data. The label "X Design" identifies if there is correlation present in the covariates while "X Cluster" indicates if covariates inform clustering

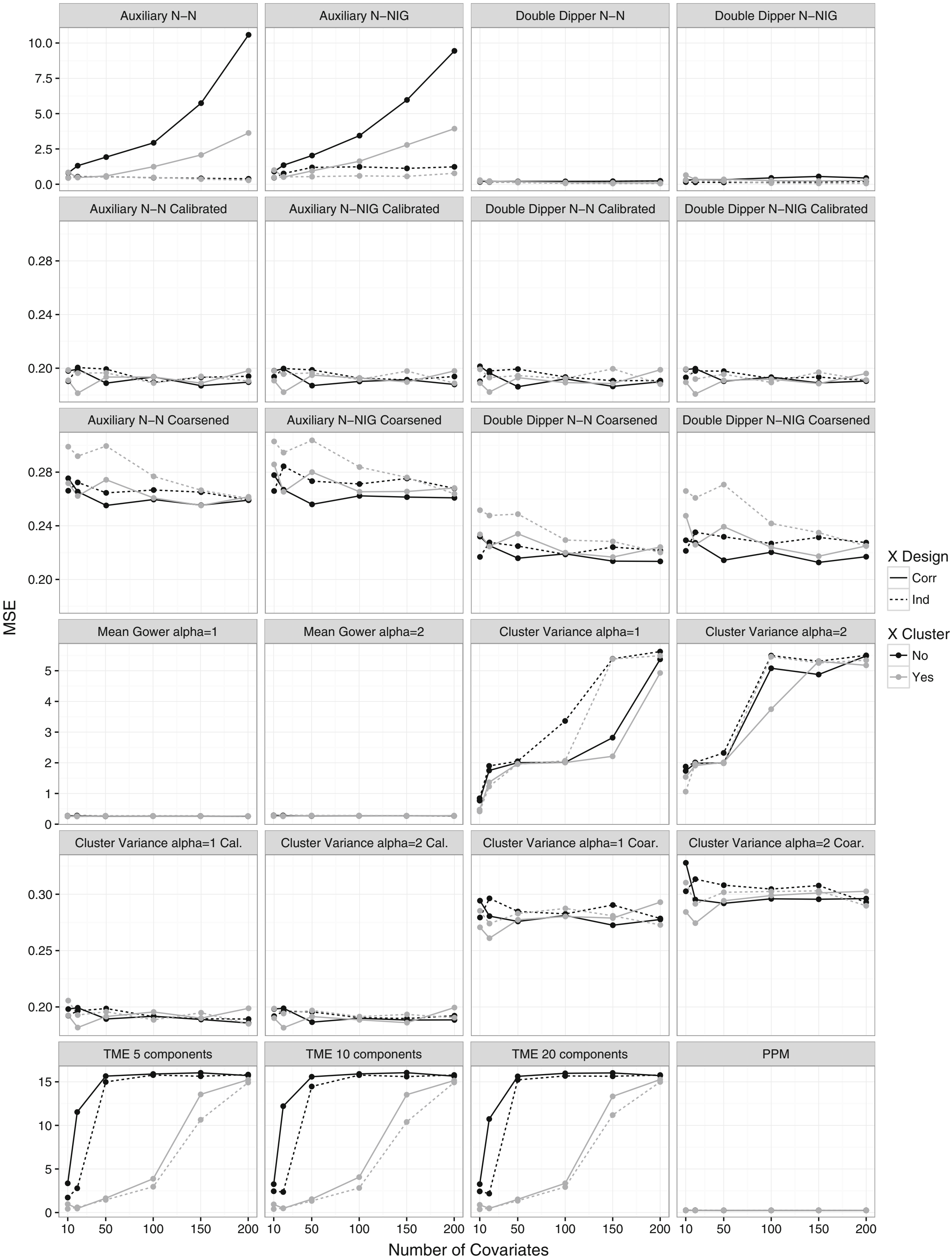

Fig. 4 MSE for the DG4 data. The label "X Design" identifies if there is correlation present in the covariates while "X Cluster" indicates if covariates inform clustering. Note that the y-axis scale changes with row. Thus, comparisons between rows must be done with care

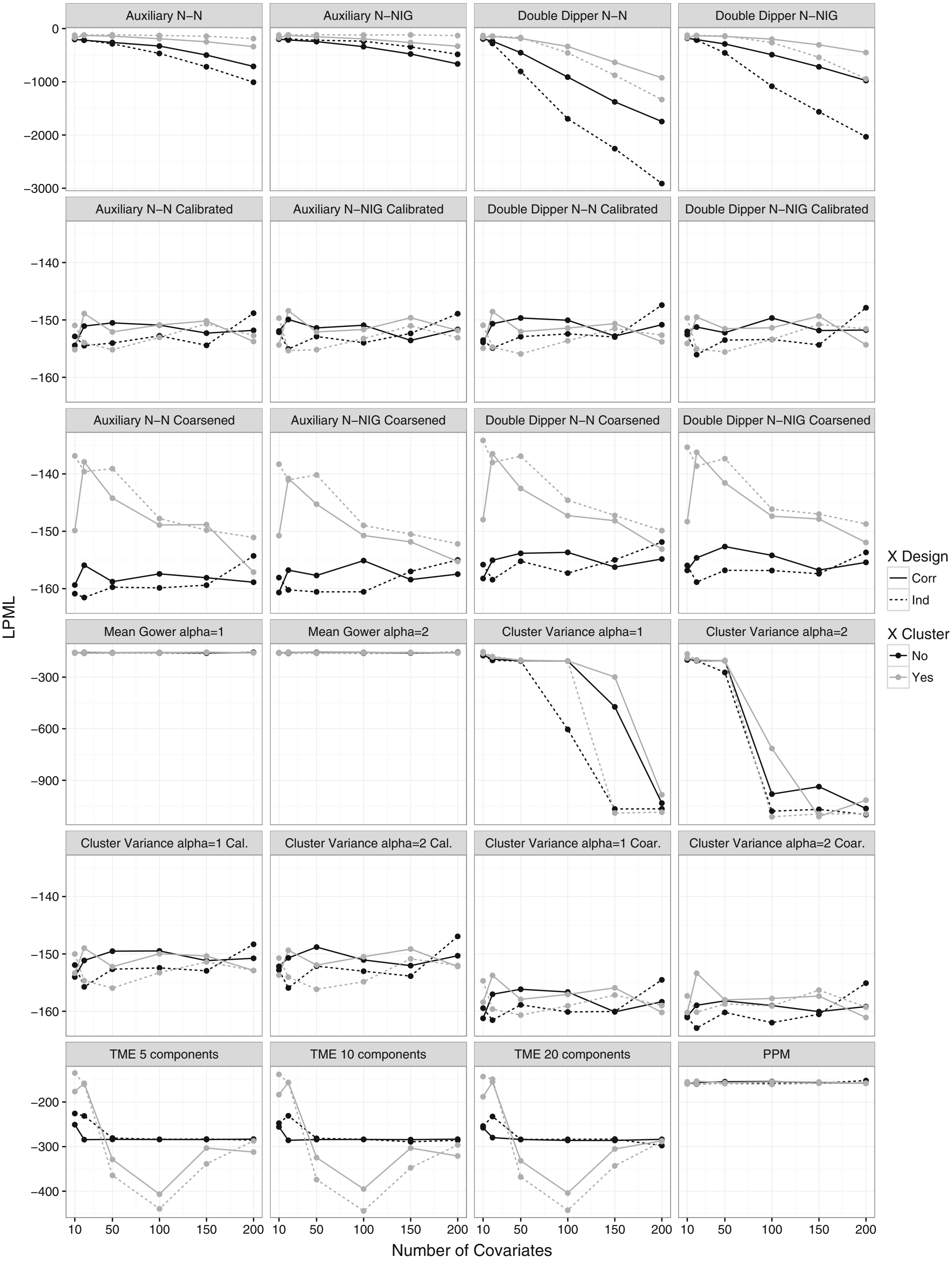

Fig. 5 LPML for the DG4 data. The label "X Design" identifies if there is correlation present in the covariates while "X Cluster" indicates if covariates inform clustering. Note that the y-axis scale changes with row. Thus, comparisons between rows must be done with care

and are not influenced by growing p. Cluster variance similarity does well even with out calibrating/coarsening though doing so seems to reduce influence of growing p. The Mean Gower Dissimilarity appears to perform well in identifying clusters (the Total Gower dissimilarity produces many more clusters). The TME procedure strangely reduces the number of clusters as p grows. Figure 2 displays the resulting ARI values which are in a sense connected to the number of clusters. Here the coarsened/calibrated Auxiliary and Double Dipper are the most consistent (in particular, not influenced by $p > m$) across scenarios with the coarsened similarities doing slightly better. TME did very well in estimating the partition when clusters were covariate dependent and for p moderate sized, but very poorly when $p > m$.

Figure 3 displays out-of-sample predictive performance. As expected, when covariates inform clustering, employing the PPMx greatly improves prediction relative to the PPM. This holds even when $p > m$. In addition, it appears that the PPM and PPMx predict similarly when covariates do not inform clustering. Thus, if prediction is of interest and covariates are available, then employing them regardless of their influence on clusters appears to be a good strategy. The remainder of our discussion on prediction focuses on the scenario when clusters are covariate dependent. Generally speaking, both Mean and Total Gower dissimilarity did not predict well (a common conclusion for these similarity functions). Calibrating the Auxiliary and Double Dipper leads to improvements in prediction (particularly as p grows), but coarsening results in much worse prediction (something that was unexpected). This trend appears to hold across all p. In fact, large p does not appear to negatively impact prediction. The TME model is very competitive regarding MSPE, but $p > m$ negatively impacts performance.

Figures 4 and 5 tell similar stories regarding model fit. Mainly that calibration/coarsening provides huge benefits in terms of MSE and LPML for all similarities. The Total Gower dissimilarity performs very well regarding MSE, but suffers from overfit. The TME model has good model MSE and LPML when the covariates inform the clustering (which is to be expected), but has the worst MSE and LPML values when covariates do not inform clustering.

The simulation suggests that for a small number of covariates the uncalibrated/coarsened Double Dipper and Auxiliary similarities are good choices. However, as the number of covariates grows, calibrating/coarsening these similarities provides great benefit to MSE/LPML, and ARI values, but marginal benefit for prediction. A very interesting result is that $p > m$ does not negatively impact calibrating these similarities. The TME performs well for a moderate number of covariates and when $p < m$. Between the three, the TME model and coarsened similarities are the most attractive computationally with the calibrated Double Dipper being the most expensive.

4.2 Data examples

We now consider two data examples both of which are publicly available. The first is available in R R Core Team (2016) and is often referred to as the Boston Housing data set. It is comprised of 506 observations and 13 covariates, one of which is categorical. Each observation corresponds to a house purchase with the response variable being the selling

Table 5 Results from the Boston housing data

Procedure	# Clus	MSPE	MSE	LPML
TME 5 components	4.99	0.08	0.07	−57.37
TME 10 components	8.27	0.06	0.05	−28.04
TME 20 components	8.75	0.06	0.04	−27.41
TME 40 components	9.56	0.06	0.04	−26.35
Auxiliary N–N	9.27	0.06	0.03	63.13
Auxiliary N–N calibrated	13.43	0.12	0.14	−113.25
Auxiliary N–N coarsened	6.37	0.11	0.01	60.18
Auxiliary N–NIG	5.00	0.09	0.09	−46.34
Auxiliary N–NIG calibrated	13.22	0.06	0.06	−50.11
Auxiliary N–NIG coarsened	5.87	0.08	0.02	81.54
Double Dipper N–N	19.23	0.05	0.02	114.13
Double Dipper N–N calibrated	13.35	0.06	0.05	−64.75
Double Dipper N–N coarsened	8.07	0.11	0.01	42.82
Double Dipper N–NIG	12.41	0.08	0.08	7.81
Double Dipper N–NIG calibrated	13.31	0.09	0.11	−102.39
Double Dipper N–NIG coarsened	9.30	0.08	0.02	89.56
PPM	6.40	0.16	0.09	−57.46

Table 6 Results for the crime Data

Procedure	# Clus	MSPE	MSE	LPML
TME 5 components	5.00	0.59	0.55	−1254.66
TME 10 components	6.68	0.56	0.50	−1346.02
TME 20 components	6.01	0.61	0.54	−1253.38
TME 40 components	7.11	0.55	0.46	−1270.83
Auxiliary N–N	32.90	0.58	0.49	−1121.63
Auxiliary N–N calibrated	26.25	0.58	0.16	−1348.66
Auxiliary N–N coarsened	7.01	1.04	0.12	−1299.07
Auxiliary N–NIG	22.00	0.62	0.55	−1122.13
Auxiliary N–NIG calibrated	26.32	0.52	0.17	−1358.49
Auxiliary N–NIG coarsened	7.37	0.69	0.06	−995.22
Double Dipper N–N	112.67	0.53	0.35	−1104.52
Double Dipper N–N calibrated	25.39	0.56	0.15	−1313.20
Double Dipper N–N coarsened	10.80	1.03	0.10	−1237.17
Double Dipper N–NIG	106.00	0.57	0.44	−1082.93
Double Dipper N–NIG calibrated	25.06	0.55	0.16	−1338.73
Double Dipper N–NIG coarsened	18.59	0.64	0.05	−837.58
PPM	18.51	0.66	0.05	−905.48

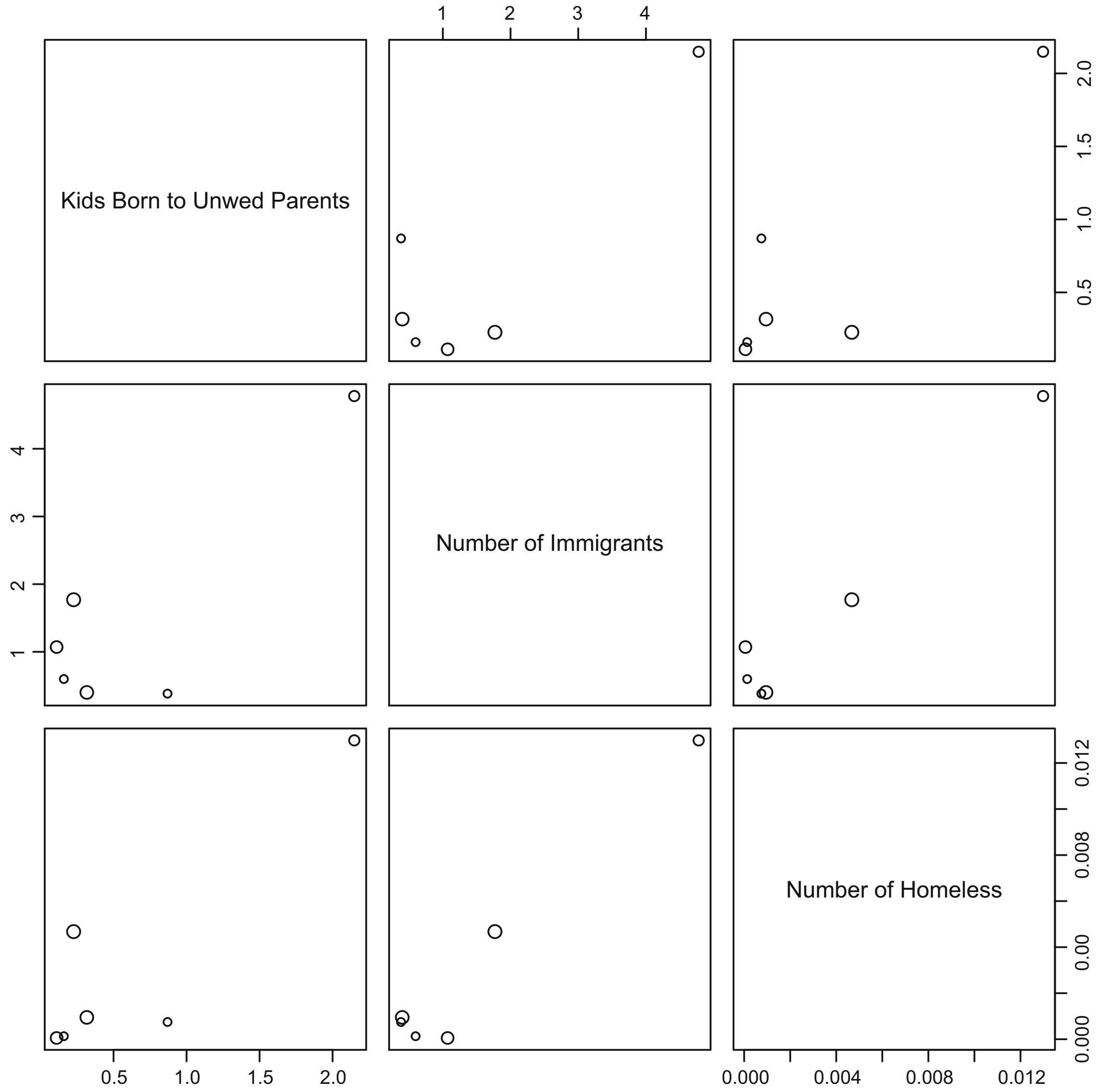

Fig. 6 Cluster-specific means for the three covariates "The Number of Kids Born to Unmarried Parents," "Number of Immigrants" and "Number of Homeless" associated with six clusters that comprise 80% of observations for the TME 40 Component fit. The circumference of dot is proportional to cluster size indicating fairly uniform sized clusters

price and the covariates providing additional house information (e.g., square footage, year built, etc.). These data were randomly partitioned into testing and training data with 206 observations in the former and 300 in the latter.

The second data set which will be referred to as the "Crime Data" is a data set that is available at the UCI machine learning repository (Lichman 2013)[1] The Crime Data is comprised of a total of 2215 observations with 125 covariates, but after removing observations that contain missing values, 1993 observations and 102 covariates remained. The response corresponds to the number of violent crimes per capita while the covariates measure a variety of other city characteristics (e.g., population, race percentages, housing information). These data were also randomly partitioned into testing and training data with 993 observations in the former and 1000 in the later.

For both datasets, model (15) with prior values suggested by Müller et al. (2011) was fit by collecting 1000 MCMC draws after a burn-in of 2000 and thinning of 3. Convergence was monitored using history plots of the MCMC iterates. Results for the Boston housing data are provided in Table 5 while those for the Crime Data can be found in Table 6. Note that for these data we no longer include the Gower dis-

[1] https://archive.ics.uci.edu/ml/machine-learning-databases/00211/CommViolPredUnnormalizedData.txt.

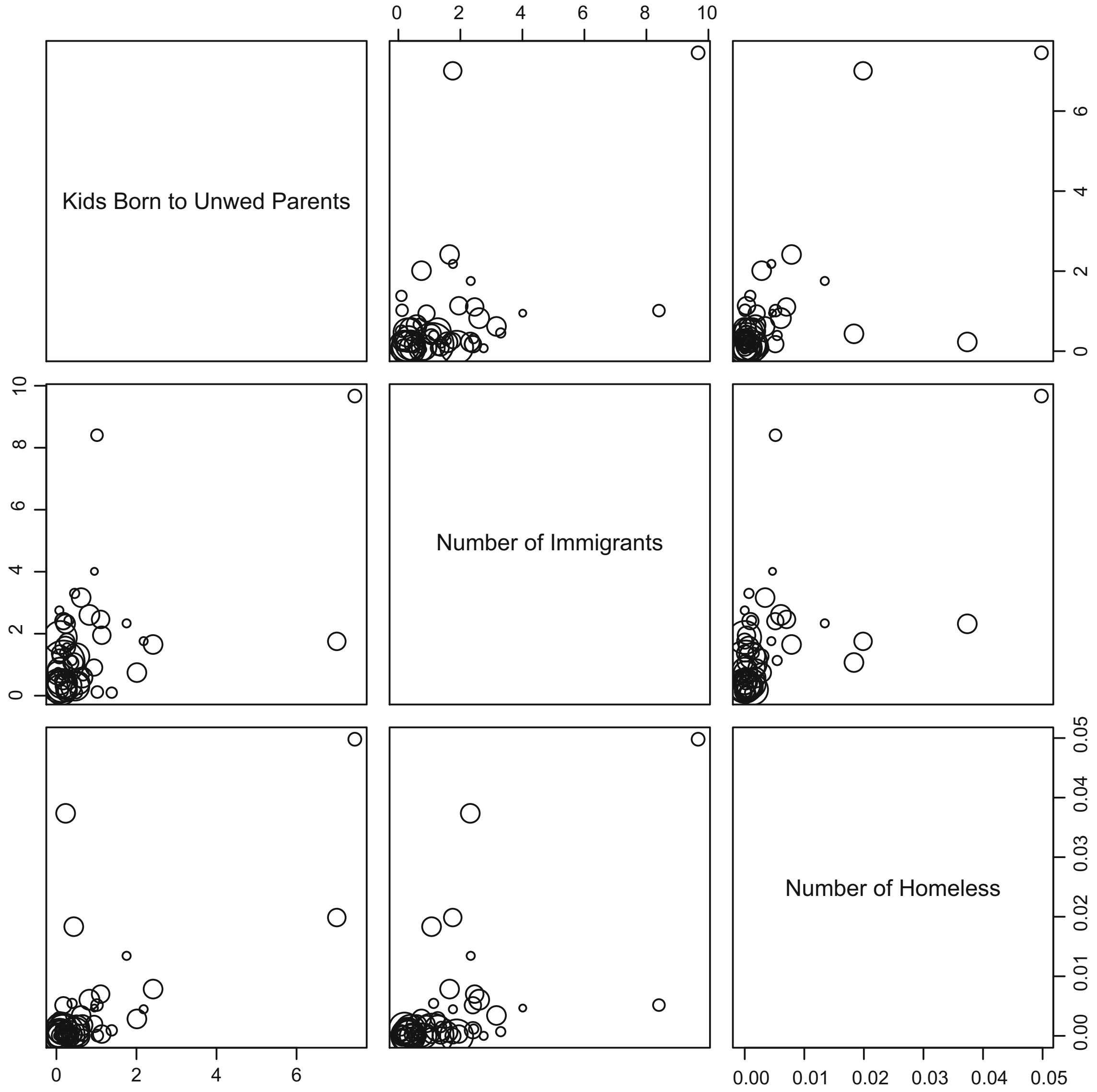

Fig. 7 Cluster-specific means for the three covariates "The Number of Kids Born to Unmarried Parents," "Number of Immigrants" and "Number of Homeless" associated with clusters that comprise 80% of observations for the Double Dipper N–N fit. The circumference of dot is proportional to cluster size

similarity and cluster variance similarity functions as their performance was shown in the simulation studies to be somewhat inferior.

For the Boston Housing data, Table 5 shows that the procedure that performed the best both in terms of out-of-sample prediction (MSPE) and model fit (MSE, LPML) was the Double Dipper N–N. This is somewhat expected given the simulation study results and the relatively small number of covariates. TME's performance seems to remain constant even as the number of components increased up to 40. The results for TME do seem to suggest that at least for these data overfitting is not a problem as including 40 components gives very similar results to 20. Coarsening for these data did seem to produce much less clusters, but at a predictive ability cost. Overall, the Double Dipper performed best.

For the Crime Data, from Table 6 it appears that the TME procedure with 40 components and calibrated Auxiliary and Double Dipper N–NIG performed best overall. Since there are a large number of covariates, this corroborates findings from the simulation study. The Double Dipper performed well and calibrating this similarity seems to not only reduce the number of clusters but also improve model fit and prediction. The same can be said about the Auxiliary similarity.

To further explore the differences in covariate-dependent partitions from the TME and calibrated similarities and their ability to detect data-driven interactions, we use Dahl

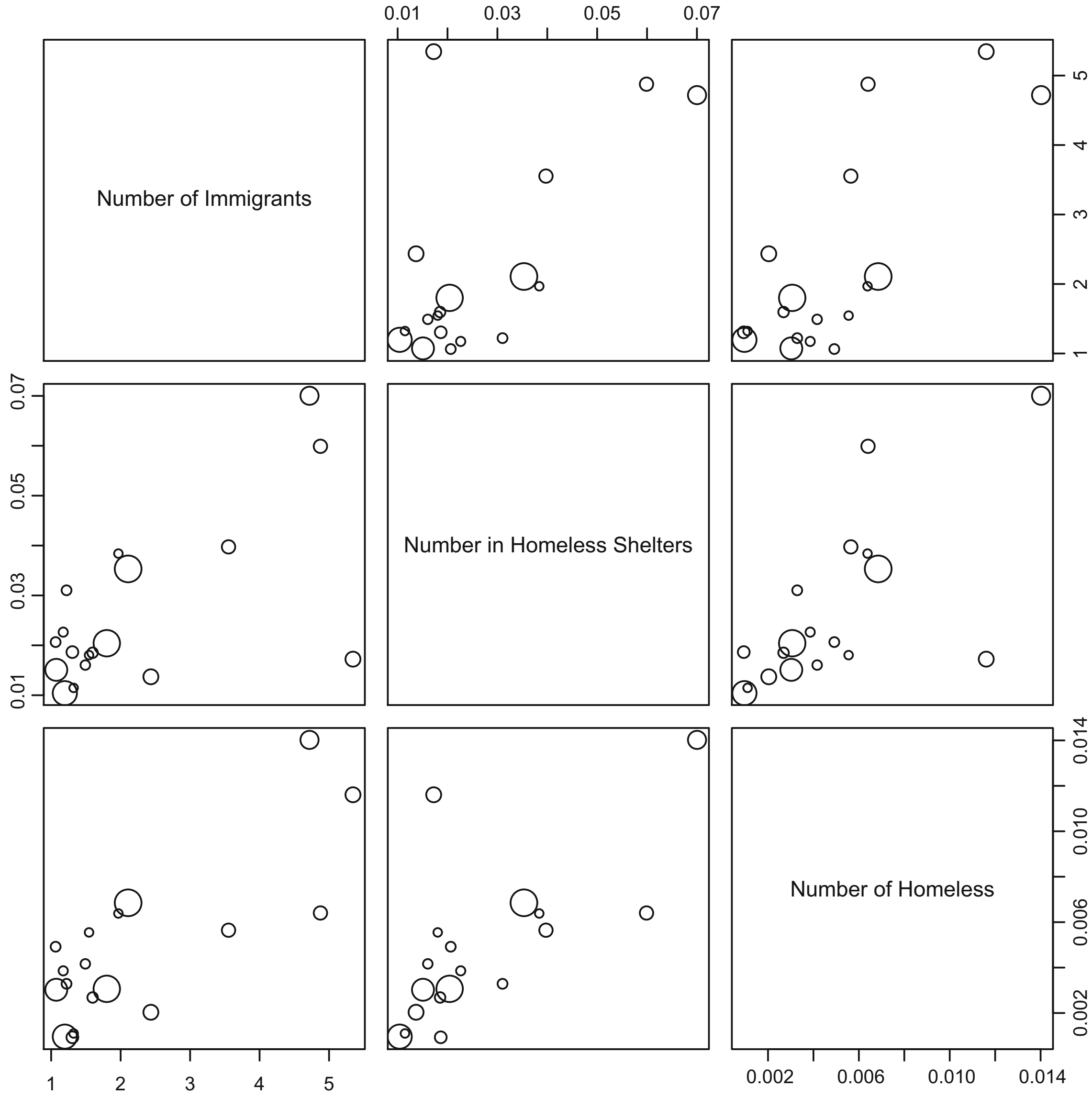

Fig. 8 Cluster-specific means for the three covariates “Number of Immigrants,” “Number in Homeless Shelters,” and “Number of Homeless” associated with clusters that comprise 80% of observations for the Double Dipper N–N calibrated fit. The circumference of dot is proportional to cluster size

(2006)’s least squares method to estimate the partition for the TME 40 Component, Double Dipper N–N and Double Dipper N–N calibrated and Double Dipper N–N coarsened methods. For each procedure, we identify some covariates related to cluster formation by computing covariate-specific variances within each cluster and retain the three covariates that display the least amount of within cluster variability for the most amount of clusters. Results are provided in the scatter plot matrices found in Figs. 6, 7, 8 and 9. The circle circumference in the plots reflects cluster size. You will notice that the same covariates were selected for all procedures save Double Dipper N–N calibrated. This leads one to hypothesize that although the partitions between the procedures are very different, the “important” covariates related to the partition subsets are similar. However, the relationships between the three covariates are slightly different. The association between Number of Homeless and Number of Immigrants was much more linear for the TME 40 and coarsened Double Dipper N–N than for the Double Dipper N–N and the calibrated Double Dipper N–N. The associations that appear to exist between the covariates in terms of cluster means are an indicator that these covariates interact.

We provide a bit more information associated with the estimated partitions for each procedure. The estimated partition of the TME 40 was comprised of seven clusters that were very homogeneous in terms of cluster size save for one that

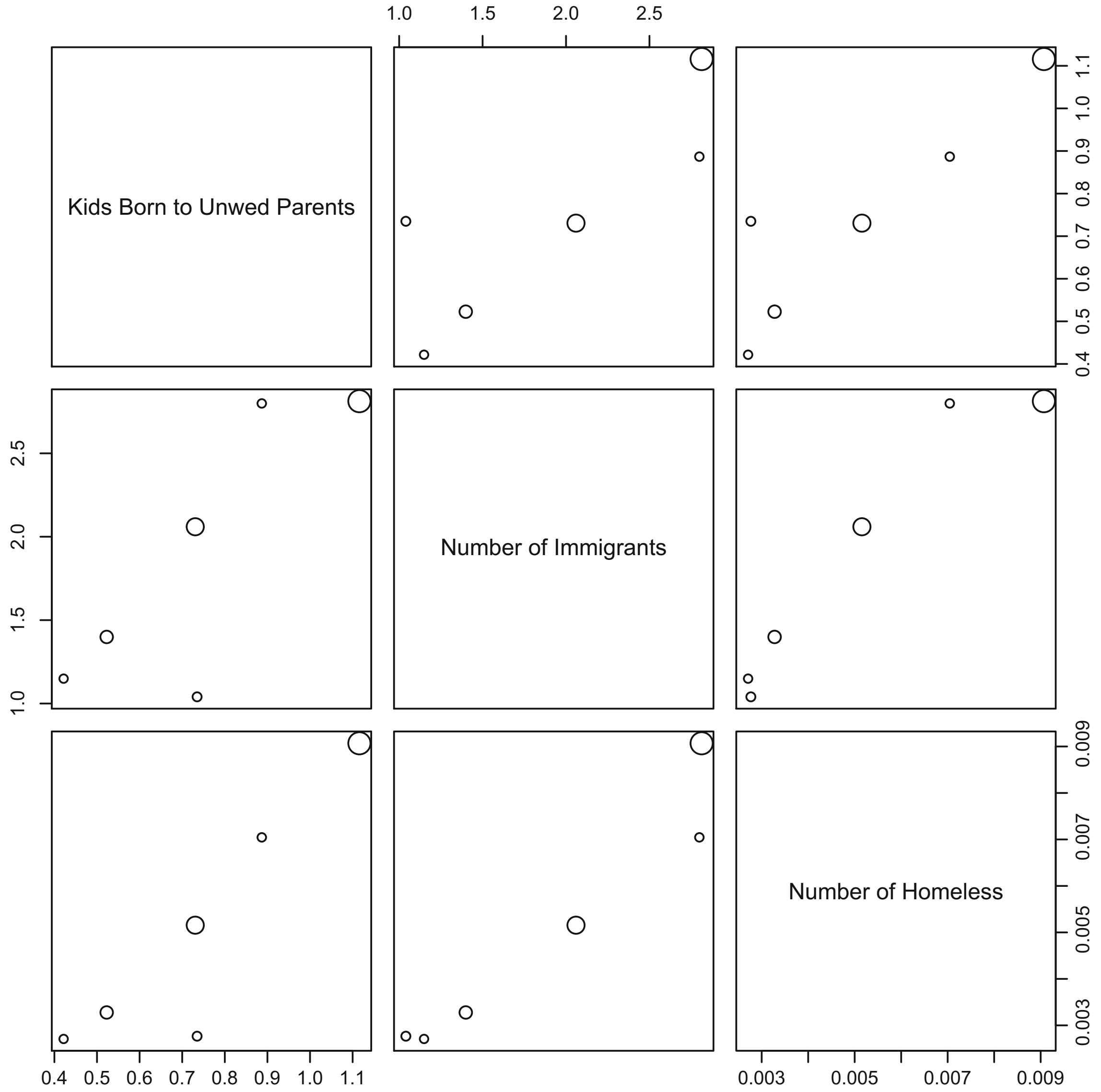

Fig. 9 Cluster-specific means for the three covariates "The Number of Kids Born to Unmarried Parents," "Number of Immigrants" and "Number of Homeless" associated with clusters that comprise 80% of observations for the Double Dipper N–N coarsened fit. The circumference of dot is proportional to cluster size

contained 12 cities all of which are high population areas such as Los Angeles and New York. For the Double Dipper N–N similarity, the estimated partition consisted of 113 clusters, nine of which were singletons. The largest cluster contained 36 cities. In total, 54 clusters comprised 80% of all observations. The partition estimated for calibrated Double Dipper N–N contained 25 clusters with the largest containing 191 observations with 18 clusters making up 80% of the observations (there were two singleton clusters). For the coarsened Double Dipper N–N, a partition with 18 clusters was estimated. The largest contained 248 and 6 clusters made up 80% of the observations (there was one singleton).

5 Conclusions

Including covariate information in partition models certainly is appealing. However, care must be taken regarding how much influence covariates have on partition probabilities. For instance, it is undesirable that they completely dominate the response when forming clusters. In this paper, we proposed two methods that are able to cap covariate influence when there are a moderate to large number of covariates. The methods are able to nicely balance the influence that covariates and the response might have when forming partition probabilities. We showed that calibrating similarity functions in a PPMx prior does regularize the influence that covariates have on clustering and reduces the number of clusters. We also showed that, as the number of covariates grows, this regularization results in improved model fit and prediction even when $p > m$. Additionally, as the number of covariates grows, the TME procedure places more weight on the response when forming clusters producing improved predictions. However, the TME is adversely impacted by $p > m$.

Our findings suggest that when the number of covariates is small, employing either the Double Dipper or the Auxiliary similarity is viable options as both performed well. For a moderate number of covariates, the TME or calibrated Double Dipper similarity is good options. When p is large and/or $p > m$, then the calibrated/coarsened Auxiliary or Double Dipper should be employed. If computational speed is an issue, then the coarsened Auxiliary should be favored over the other procedures.

An interesting extension of the work proposed here that is a topic of future research is to combine the variable selection methodology developed in Quintana et al. (2015) with the calibrating/coarsened similarities developed here. Such procedure would allow one to select the relevant covariates that are specific for each cluster, for any of the possible prior forms studied here, thus producing a better characterization of the covariates that drive the clustering. The benefits of this approach are highlighted in the simulation studies with p small and the coarsening similarity still providing benefit in terms of out-of-sample prediction.

Acknowledgements The authors would like to thank Peter Müller for helpful comments. The authors also thank all the reviewers for their valuable suggestions that substantially improved presentation. Garritt L. Page gratefully acknowledges the financial support of FONDECYT Grant 11121131 and Fernando A. Quintana was partially funded by Grant FONDECYT 1141057.

A Appendix: MCMC algorithm for the calibrated similarity and tempered mixture of experts

Here we provide pertinent computation details for the MCMC algorithm used to fit the TME model and PPMx with calibrated similarity. We focus primarily on the updating of cluster labels, as conditional on these updating the remaining model parameters is straightforward employing a Gibbs sampler or Metropolis–Hastings steps.

A.1 Calibrated similarity

To update the cluster membership of subject i for the calibrated similarity, cluster weights are created by comparing the unnormalized posterior for the jth cluster when subject i is excluded from that when subject i is included. In addition to weights for existing clusters, algorithm 8 of Neal (2000) requires calculating weights for p empty clusters whose cluster-specific parameters are auxiliary variables generated from the prior. To make this more concrete, let S_j^{-i} denote the jth cluster and k^{-i} the number of clusters when subject i is not considered. Similarly $\boldsymbol{x}_j^{\star-i}$ will denote the vector of covariates corresponding to cluster h when subject i has been removed. Then the multinomial weights associated with the k^{-i} existing clusters and one empty cluster are

$$Pr(s_i = j|-) \propto \tag{16}$$

$$\begin{cases} N(y_i; \mu_j^{\star}, \sigma_j^{2\star}) \dfrac{c(S_j^{-i} \cup \{i\})\tilde{g}(\boldsymbol{x}_j^{\star-i} \cup \{\boldsymbol{x}_i\})}{c(S_j^{-i})\tilde{g}(\boldsymbol{x}_j^{\star-i})} & \text{for } j = 1, \ldots, k^{-i} \\ N(y_i; \mu_{\text{new},j}^{\star}, \sigma_{\text{new},j}^{2\star}) c(\{i\})\tilde{g}(\{\boldsymbol{x}_i\}) p^{-1} & \text{for } j = k^{-i} + 1. \end{cases} \tag{17}$$

where as mentioned $\mu_{\text{new},j}^{\star}$ and $\sigma_{\text{new},j}^{2\star}$ are auxiliary variables that are drawn from their respective prior distributions. Since $\tilde{g}(\{\boldsymbol{x}_i\})$ needs to be accounted for when standardizing the multinomial weights, we employ the following ratios in the MCMC algorithm

$$\tilde{g}(\boldsymbol{x}_j^{\star-i} \cup \boldsymbol{x}_i) = \frac{g(\boldsymbol{x}_j^{\star-i} \cup \boldsymbol{x}_i)}{\sum_\ell g(\boldsymbol{x}_\ell^{\star-i} \cup \boldsymbol{x}_i)}$$

$$\tilde{g}(\boldsymbol{x}_j^{\star-i}) = \frac{g(\boldsymbol{x}_j^{\star-i})}{\sum_\ell g(\boldsymbol{x}_\ell^{\star-i}) + g(\{\boldsymbol{x}_i\})}$$

$$\tilde{g}(\{\boldsymbol{x}_i\}) = \frac{g(\{\boldsymbol{x}_i\})}{\sum_\ell g(\boldsymbol{x}_\ell^{\star-i}) + g(\{\boldsymbol{x}_i\})}.$$

When $\boldsymbol{x}_i$ is included in the j cluster then it is not able to form its own singleton. However, when it is excluded from the jth cluster, then it is completely plausible that it forms its own singleton cluster. For these reasons the similarity value $g(\{\boldsymbol{x}_i\})$ is only included in $\tilde{g}(\boldsymbol{x}_j^{\star-i})$ and $\tilde{g}(\{\boldsymbol{x}_i\})$.

A.2 TME with unknown $\boldsymbol{\xi}_j^\star$ and fixed J

Upon introducing latent component labels s_i such that $Pr(s_i = j) = w(\boldsymbol{x}_i; \boldsymbol{\xi}_j^\star)$, the data model (14) can be written hierarchically as

$$p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\mu}^\star, \boldsymbol{\sigma}^{2\star}, \boldsymbol{\xi}^\star, \boldsymbol{c}) = \prod_{i=1}^{m}\prod_{\ell=1}^{J} N(y_i|\mu_\ell^\star, \sigma_\ell^{2\star})^{I[s_i=\ell]} \tag{18}$$

$$s_i \sim \sum_{\ell=1}^{J} \delta_\ell w(\boldsymbol{x}_i; \boldsymbol{\xi}_\ell^\star) \tag{19}$$

where δ_ℓ is the dirac measure. With this hierarchical representation, a MCMC algorithm can be constructed by cycling through the following

- Update component labels using

$$Pr(s_i = h|-) \propto N(y_i|\mu^\star_h, \sigma^{2\star}_h)w(\boldsymbol{x}_i; \boldsymbol{\xi}^\star_h)$$

- If $\boldsymbol{x}_i$ is comprised of continuous and categorical variables, then without loss of generality let $\boldsymbol{x}_i = (x_{1i}, x_{2i})$ where x_{1i} is continuous and x_{2i} is categorical. Further, $\boldsymbol{\xi}^\star_j = (\eta^\star_j, \upsilon^{2\star}_j, \boldsymbol{\pi}^\star_j)$ with $\boldsymbol{\xi}^\star = (\boldsymbol{\xi}^\star_1, \ldots, \boldsymbol{\xi}^\star_J)$. Then $\boldsymbol{\xi}^\star_j = (\eta^\star_j, \upsilon^{2\star}_j, \boldsymbol{\pi}^\star_j)$ can be updated within the MCMC algorithm by way of a Metroplis–Hastings step employing

$$\begin{aligned}
[\boldsymbol{\xi}^\star_j|-] &\propto \prod_{i=1}^m Pr(s_i|\boldsymbol{\xi}^\star) \prod_{j=1}^J p(\boldsymbol{\xi}^\star_j)\\
&\propto \prod_{i=1}^m w(\boldsymbol{x}_i; \boldsymbol{\xi}^\star_1)^{I[s_i=1]} \times \ldots \times w(\boldsymbol{x}_i; \boldsymbol{\xi}^\star_1)^{I[s_i=J]} p(\boldsymbol{\xi}^\star_j)\\
&\propto \prod_{i:s_i=j} w(\boldsymbol{x}_i; \boldsymbol{\xi}^\star_j) p(\boldsymbol{\xi}^\star_j)\\
&= \prod_{i:s_i=j} \frac{q(x_{i1}|\eta^\star_j, \upsilon^{2\star}_j)q(x_{i2}|\boldsymbol{\pi}^\star_j)}{\sum_\ell^J q(x_{i1}|\eta^\star_\ell, \upsilon^{2\star}_\ell)q(x_{i2}|\boldsymbol{\pi}^\star_\ell)} p(\eta^\star_j, \upsilon^{2\star}_j, \boldsymbol{\pi}^\star_j)
\end{aligned}$$

where $q(x_{i1}|\eta^\star_j, \upsilon^{2\star}_j)$ is normal density and and $q(x_{i2}|\boldsymbol{\pi}^\star_j)$ a multinomial density. For $\boldsymbol{\pi}^\star_j$ an independent Metropolis–Hastings sampler with uniform (over the simplex) candidate density may be considered. This candidate density will cancel out in the Metropolis–Hastings ratio (though this may become more inefficient as the number of categories in x_{i2} increases). Updating $\eta^\star_j$ and $\upsilon^{2\star}_j$ can be accomplished using a random walk Metropolis step with normal candidate density for both.

- Updating the likelihood parameters $\mu^\star_j$ and $\sigma^{2\star}_j$ can be carried out using Gibbs steps as their full conditionals have well known closed forms.

B Computing Gower's dissimilarity

The daisy function found in the cluster package (Maechler et al. 2016) of the statistical software R was employed to calculate the Gower dissimilarity. The calculated dissimilarity is an "average" of the individual p dissimilarities

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{p}\sum_{\ell=1}^p d(x_{i\ell}, x_{j\ell}).$$

For numeric or continuous x's, $d(x_{i\ell}, x_{j\ell}) = |x_{i\ell} - x_{j\ell}|/R_\ell$ where $R_\ell = \max_h(x_{h\ell}) - \min_h(x_{h\ell})$. For nominal variables

$$d(x_{i\ell}, x_{j\ell}) = \begin{cases} 0 & \text{if } x_{i\ell} = x_{j\ell} \\ 1 & \text{otherwise.} \end{cases}$$

References

Antoniano-Villalobos, I., Walker, S.G.: A nonparametric model for stationary time series. J. Time Ser. Anal. **37**(1), 126–142 (2016)

Barcella, W., Iorio, M.D., Baio, G.: A comparative review of variable selection techniques for covariate dependent Dirichlet process mixture models (2016). https://arxiv.org/pdf/1508.00129.pdf

Barcella, W., Iorio, M.D., Baio, G., Malone-Lee, J.: Variable selection in covariate dependent random partition models: an application to urinary tract infection. Stat. Med. **35**, 1373–1389 (2016)

Barrientos, A.F., Jara, A., Quintana, F.A.: On the support of MacEachern's dependent Dirichlet processes and extensions. Bayes Anal. **7**, 277–310 (2012)

Blei, D.M., Frazier, P.I.: Distant dependent chinese restaurant processes. J. Mach. Learn. Res. **12**, 2461–2488 (2011)

Christensen, R., Johnson, W., Branscum, A.J., Hanson, T.: Bayesian Ideas and Data Analysis: An Introduction for Scientists and Statisticians. CRC Press, Boca Raton (2011). http://www.ics.uci.edu/~wjohnson/BIDA/BIDABook.html

Chung, Y., Dunson, D.B.: Nonparametric bayes conditional distribution modeling with variable selection. J. Am. Stat. Assoc. **104**, 1646–1660 (2009)

Cook, R.D., Weisberg, S.: Sliced inverse regression for dimension reduction: comment. J. Am. Stat. Assoc. **86**, 328–332 (1991)

Dahl, D.B.: Model-based clustering for expression data via a Dirichlet process mixture model. In: Vannucci, M., Do, K.A., Müller, P. (eds.) Bayesian Inference for Gene Expression and Proteomics, pp. 201–218. Cambridge University Press, Cambridge (2006)

Dahl, D.B., Day, R., Tsai, J.W.: Random partition distribution indexed by pairwise information. J. Am. Stat. Assoc. (2016). doi:10.1080/01621459.2016.1165103

De Iorio, M., Müller, P., Rosner, G., MacEachern, S.: An ANOVA model for dependent random measures. J. Am. Stat. Assoc. **99**, 205–215 (2004)

Dunson, D.B., Park, J.H.: Kernel stick-breaking processes. Biometrika **95**, 307–323 (2008)

Geisser, S., Eddy, W.F.: A predictive approach to model selection. J. Am. Stat. Assoc. **74**(365), 153–160 (1979)

Gelfand, A.E., Kottas, A., MacEachern, S.N.: Bayesian nonparametric spatial modeling with Dirichlet process mixing. J. Am. Stat. Assoc. **102**, 1021–1035 (2005)

Gower, J.C.: A general coefficient of similarity and some of its properties. Biometrics **27**, 857–871 (1971)

Griffin, J.E., Steel, M.F.J.: Order-based dependent Dirichlet processes. J. Am. Stat. Assoc. **101**, 179–194 (2006)

Guhaniyogi, R., Dunson, D.B.: Bayesian compressed regression. J. Am. Stat. Assoc. **110**, 1500–1514 (2015)

Hannah, L., Blei, D., Powell, W.: Dirichlet process mixtures of generalized linear models. J. Mach. Learn. Res. **12**, 1923–1953 (2011)

Hartigan, J.A.: Partition models. Commun. Stat. Theory Methods **19**, 2745–2756 (1990)

Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Comput. **3**, 79–87 (1991)

Lichman, M.: UCI machine learning repository (2013). http://archive.ics.uci.edu/ml

MacEachern, S.N.: Dependent Dirichlet processes. Ohio State University, Department of Statistics, Technical report (2000)

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K.: Cluster: Cluster Analysis Basics and Extensions (2016). R package version 2.0.4—For new features, see the 'Changelog' file (in the package source)

McLachlan, G., Peel, D.: Finite Mixture Models, 1st edn. Wiley Series in Probability and Statistics, New York (2000)

Miller, J.W., Dunson, D.B.: Robust Bayesian inference via coarsening (2015). http://arxiv.org/abs/arXiv:1506.06101

Molitor, J., Papathomas, M., Jerrett, M., Richardson, S.: Random partition models with regression on covariates. Biostatistics **11**, 484–498 (2010)

Müller, P., Erkanli, A., West, M.: Bayesian curve fitting using multivariate normal mixutres. Biometrika **83**, 67–79 (1996)

Müller, P., Quintana, F.A., Jara, A., Hanson, T.: Bayesian Nonparametric Data Analysis, 1st edn. Springer, Switzerland (2015)

Müller, P., Quintana, F.A., Rosner, G.L.: A product partition model with regression on covariates. J. Comput. Graph. Stat. **20**(1), 260–277 (2011)

Müller, P., Quintana, F.A., Rosner, G.L., Maitland, M.L.: Bayesian inference for longitudinal data with non-parametric treatment effects. Biostatistics **15**(2), 341–352 (2013)

Neal, R.M.: Markov chain sampling methods for Dirichlet process mixture models. J. Comput. Graph. Stat. **9**, 249–265 (2000)

Page, G.L., Bhattacharya, A., Dunson, D.B.: Classification via Bayesian nonparametric learning of affine subspaces. J. Am. Stat. Assoc. **108**, 187–201 (2013)

Page, G.L., Quintana, F.A.: Predictions based on the clustering of heterogeneous functions via shape and subject-specific covariates. Bayesian Anal. **10**, 379–410 (2015)

Page, G.L., Quintana, F.A.: Spatial product partition models. Bayesian Anal. **11**(1), 265–298 (2016)

Papathomas, M., Molitor, J., Hoggart, C., Hastie, D., Richardson, S.: Exploring data from genetic association studies using bayesian variable selection and the Dirichlet process: application to searchingfor gene × gene patterns. Genet. Epidemiol. **36**, 663–674 (2012)

Park, J.H., Dunson, D.B.: Bayesian generalized product partition model. Stat. Sin. **20**, 1203–1226 (2010)

Quintana, F.A., Müller, P., Papoila, A.L.: Cluster-specific variable selection for product partition models. Scand. J. Stat. **42**, 1065–1077 (2015). doi:10.1111/sjos.12151

R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2016). https://www.R-project.org/

Rand, W.M.: Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. **66**, 846–850 (1971)

Rodriguez, A., Dunson, D.B., Gelfand, A.E.: Bayesian nonparametric functional data analysis through density estimation. Biometrika **96**, 149–162 (2009)

Wade, S., Dunson, D.B., Petrone, S., Trippa, L.: Improving prediction from Dirichlet process mixtures via enrichment. J. Mach. Learn. Res. **15**, 1041–1071 (2014)

Wang, H., Xia, Y.: Sliced regression for dimension reduction. J. Am. Stat. Assoc. **103**, 811–821 (2008)