A Bayesian Dual-Network Clustering Approach for Selecting Data and Parameter Granularities

Mingyung Kim, Eric T. Bradlow, Raghuram Iyengar*

ABSTRACT

While there are well-established methods for model selection (e.g., BIC, marginal likelihood), they generally condition on an a priori selected data (e.g., SKU-level data) and parameter granularity (e.g., brand-level parameters). That is, researchers think they are doing model selection, but what they are really doing is model selection *conditional* on their choices of data and parameter granularities. In this research, we propose a *Bayesian dual-network clustering* method as a novel way to make these two decisions simultaneously. To accomplish this, the method represents data and parameters as two separate networks with nodes being the unit of analysis (e.g., SKUs). The method then (a) clusters the two networks using a covariate-driven distance function which allows for a high degree of interpretability and (b) infers the data and parameter granularities that offer the best in-sample fit, akin to standard model selection methods. We apply our method to SKU-level demand analysis. The results show that the choices of data and parameter granularities based on our method as compared to those from extant approaches (e.g., latent class analysis) impact the demand elasticities and the optimal pricing of SKUs. We conclude by highlighting the generalizability of our framework to a broad array of marketing problems.

Keywords: Data granularity, parameter granularity, network clustering, Bayesian non-parametrics

* Mingyung Kim is a PhD Candidate in Marketing at the Wharton School of the University of Pennsylvania (mingyung@wharton.upenn.edu). Eric T. Bradlow is the K.P. Chao Professor, Professor of Marketing, Economics, Education, Statistics and Data Science at the Wharton School of the University of Pennsylvania (ebradlow@wharton.upenn.edu). Raghuram Iyengar is Miers-Busch, W'1885 Professor, Professor of Marketing at the Wharton School of the University of Pennsylvania (riyengar@wharton.upenn.edu). Researcher(s) own analyses calculated (or derived) based in part on data from Nielsen Consumer LLC and marketing databases provided through the NielsenIQ Datasets at the Kilts Center for Marketing Data Center at The University of Chicago Booth School of Business. The conclusions drawn from the NielsenIQ data are those of the researcher(s) and do not reflect the views of NielsenIQ. NielsenIQ is not responsible for, had no role in, and was not involved in analyzing and preparing the results reported herein.

1. Introduction

One of the fundamental, and academically well researched, decisions that a marketing manager routinely makes is how to model the relationship between firm-controlled (e.g., price) and exogenous (e.g., macroeconomic) drivers and a specific outcome of interest (e.g., sales). This problem is commonly denoted as *model selection*, for which there have been many solutions proposed that compare model fit (e.g., BIC, marginal likelihood) across models to "select the winner". While model selection, as described above, is a laudable goal, we suggest that this is *not* the problem that researchers have been solving; rather they are performing *conditional* model selection, based on a priori chosen levels of data and parameter granularities. The former refers to decisions about whether to employ the most granular data (e.g., weekly) or aggregate it to a coarser level (e.g., monthly) for model estimation. Similarly, the latter refers to decisions about whether to vary parameters across the most granular units (e.g., across individuals) or at a more aggregate level (e.g., across zip codes) in a demand model. In this paper, it is the simultaneous selection, using Bayesian inference and posterior sampling, of the "big three" (the aforementioned data aggregation[1] and parameter granularity, our research contribution, and the standard model and parameter inference) that we seek to address.

We propose a *Bayesian dual-network clustering* method that allows researchers to select both data and parameter granularities. The proposed method, as the name suggests, represents data and parameters as two networks with each unit being a node. Then, it probabilistically clusters the nodes in the networks to infer the levels of data and parameter granularities. One notable contribution is that by representing both data and parameters in a generalized manner (i.e., using a network), our method is *flexible* and can accommodate differing units of analysis with little

[1] We utilize the terms data aggregation (which is standard) and data granularity (to make its choice symmetric with that of the parameter granularity) interchangeably.

modification (e.g., time, space, person, SKU). Equally important is that our method offers a high degree of *interpretability* as to the underlying drivers of data and parameter clusters by relating distances between nodes in the respective networks to observed attributes of the unit of analysis (e.g., brand or package size for an SKU-level demand analysis).

We make the above-mentioned contributions by developing a novel extension of the Bayesian non-parametric clustering method, the distance dependent Chinese Restaurant Processes (ddCRP; Blei and Frazier 2011), originally used to cluster documents and images (e.g., Arfa, Yusof, and Shabanzadeh 2019; Ghosh et al. 2011). Notably, extant methods such as latent class analysis (hereafter, LCA) and unsupervised clustering (e.g., K-means) select *either* parameter or data granularity but not both. LCA determines the level of parameter granularity while fixing the level of data aggregation (typically, at the most granular level). For instance, Smith, Rossi, and Allenby (2019) fixed sales data at the brand level and applied LCA to this data to select the level of parameter granularity (e.g., the level of price elasticity). In contrast, unsupervised clustering (e.g., K-means) clusters data points that have similar features without accounting for a downstream demand model. For instance, Morwitz and Schmittlein (1992) used K-means to segment households with respect to their characteristics (e.g., demographics and past purchase behaviors) and aggregated the household-level data accordingly. Note also that while many past studies have demonstrated that empirical results (e.g., price elasticity) and the corresponding marketing decisions (e.g., optimal price) vary based on the chosen levels of data and/or parameter granularities (e.g., Christen et al. 1997), how to select their levels jointly is an important gap in the literature that our research fills.

To assess the performance of our method in comparison to extant methods, we conduct a simulation study wherein we vary the signal-to-noise ratio (SNR) in the data generation process.

We find that the bias in the coefficients of the estimated demand model (e.g., price elasticity) is smaller under our proposed method than under extant methods (e.g., LCA). An in-depth analysis reveals that a key driver of these results is that our proposed method performs better in recovering the underlying true levels of data and parameter granularities than extant approaches do. For example, LCA selects overly granular parameters (too much heterogeneity) when the data is fixed at the most granular level.

Following the simulation study, we apply our method to a Nielsen scanner data set containing SKU-week-level sales of orange juice and marketing actions (price and advertising) and compare its performance with that from extant methods. We provide three key findings. First, our method provides a significantly better (in-sample and out-of-sample) model fit than the extant methods do. This result implies that it is important to select the levels of *both* data and parameter granularities along with the model itself. Second, the levels of data and parameter granularities chosen by our method differ from those chosen by the extant methods. It is because unlike our method, the extant methods select either data or parameter granularity while *conditioning* on the other. Lastly, the inference of demand parameters (e.g., price elasticity) and optimal marketing decisions (e.g., optimal price) based on our method differ substantially from those obtained by employing alternative methods.

Although we apply the proposed method to SKU-level demand analysis, researchers make decisions regarding data and parameter granularities in many other contexts. One example is a temporal or spatial analysis of demand, for which researchers select the temporal or spatial unit of analysis and build a model conditional on their choices. In most cases, they decide whether to use the most granular data (e.g., daily data) or aggregate it to a coarser level (e.g., weekly level) for analysis. Another example is customer (or group) level analysis, for which researchers often cluster customers and then aggregate data (and/or parameters) based on the chosen clustering. Our framework is flexible (as explained above) and can be easily applied to such contexts.

The remainder of the paper is as follows. In Section 2, we propose our Bayesian dual-network clustering method as a tool for data and parameter granularities. Section 3 describes a simulation study that assesses the performance of our method in selecting data and parameter granularities as compared to other extant approaches. In Section 4, we present an application of our method to a data set containing SKU-level purchases. Section 5 concludes with limitations and future research directions, both from a methodological and applied perspective.

2. Methodological Framework

We propose a *Bayesian dual-network clustering* method to provide researchers a method to select the levels of data and parameter granularity in their chosen model. The former refers to decisions regarding the granularity of the data employed for model estimation (e.g., daily versus weekly sales data). In a similar vein, the latter refers to decisions about whether parameters in the model should be allowed to vary across the most granular units (e.g., across individuals) or at a more aggregate level (e.g., across zip codes). In what follows, we first lay out a general overview of our methodological framework (2.1) and the prior distribution chosen (a flexible non-parametric distribution) for selecting data and parameter granularities (2.2). We use a modified version of the distance dependent Chinese restaurant processes (ddCRP) introduced by Blei and Frazier (2011). We then discuss the proposed framework in detail (2.3) and how it compares to, and nests, other data and parameter clustering methods (e.g., unsupervised learning and latent class analysis, respectively) commonly employed in the statistics and marketing literatures (2.4).

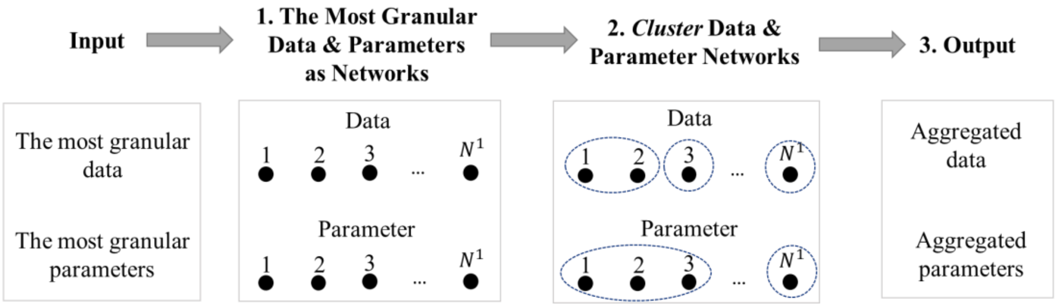
2.1. General Overview of the Methodological Framework

Figure 1 provides an overview of our framework. There are three key steps. First, for a given context, we assemble the most granular level of data available (e.g., individual-level purchase data). Let N^1 indicate the number of observations in this dataset with the superscript "1" denoting the most granular data. Similarly, suppose that the model employed in the analysis is specified with parameters at the most granular level (e.g., observation-specific parameters corresponding to the most granular data with N^1 observations). One notable contribution of our research is to represent the most granular data and parameters as two networks – a data network and a parameter network, respectively – with each unit being a node.

Second, we probabilistically cluster nodes in both the data and parameter networks. We employ Bayesian inference and sample the posterior distribution of the data clustering (denoted as D) and parameter clustering (denoted as M) which recognizes and utilizes their uncertainty. By doing so, we extend typical statistical modeling, which conditions on the choice of D and M. We accomplish this dual-network clustering (as per the title of the paper) by building on extant ddCRP-based clustering methods.

As a last step, we aggregate data and parameters based on the chosen clusters. Specifically, we aggregate the data for nodes in the same data cluster and set the parameters equal for nodes in the same parameter cluster. We restrict our data clustering D to be at least as granular as the parameter clustering M as in Figure 1. That is, we do not consider problems of demand estimation with aggregated data (e.g., Chen and Yang 2007; Musalem, Bradlow, and Raju 2008) where the parameters are more granular than the data.

Figure 1: Methodological Framework



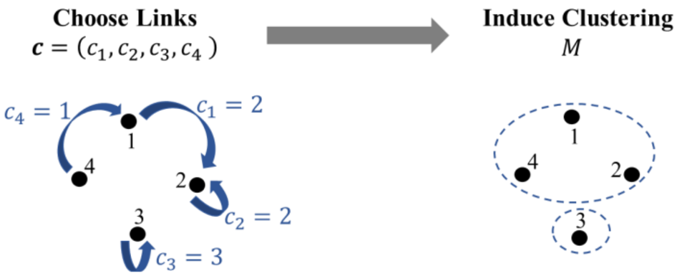
2.2. Distance Dependent Chinese Restaurant Processes (ddCRP)

The ddCRP is a non-parametric probability distribution over clusters. It represents the input of interest (e.g., an image) as a network with each unit (e.g., a pixel from the image) being a node. Then, it clusters the network by iterating across nodes and, for each node, chooses a node to link itself to (see Figure 2). Specifically, the ddCRP links node i to itself (denoted as $c_i = i$) with a probability proportional to a self-link parameter α, or to another node i' (denoted as $c_i = i'$) with a probability proportional to a non-increasing function of their distance: $f(dist_{ii'})$. These assumptions lead to the following multinomial distribution conditional on the self-link parameter α, the pairwise distance $dist_{ii'}$, and a decay function $f(.)$

$$p(c_i = i' | \, dist_{ii'}, \alpha, f) \propto \begin{cases} \alpha & \text{if } i = i' \\ f(dist_{ii'}) & \text{if } i \neq i' \end{cases} \qquad (1)$$

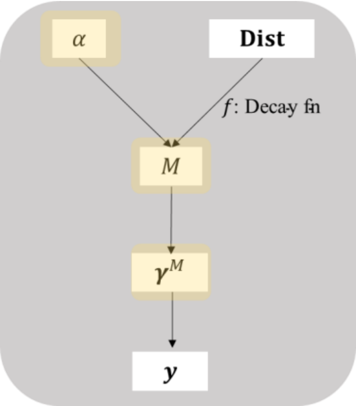
The decay function is non-negative, non-increasing, and $f(\infty) = 0$. Note that several decay functions (e.g., exponential decay, logistic decay) meet these conditions.

Figure 2: **Illustration of ddCRP for Parameter Clustering *M***



Extant studies have employed the ddCRP for latent class analysis (hereafter, LCA). They impose a non-parametric prior on the number of parameter clusters while fixing the level of data granularity (e.g., Arfa, Yusof, and Shabanzadeh 2019; Blei and Frazier 2011; Ghosh et al. 2011). The ddCRP-based LCA is composed of three steps (see Figure 3). First, given a sampled self-link parameter (α) and a set of pairwise distances ($\mathbf{Dist}$), we sample parameter clustering (M) based on Equation (1). Next, we aggregate the parameters given the sampled parameter clustering. Specifically, we set the parameters equal for nodes in the same parameter cluster and denote the aggregated parameters as $\boldsymbol{\gamma}^M$. Finally, we apply a model with the aggregated parameters to the observed data ($\boldsymbol{y}$) at the most granular level.

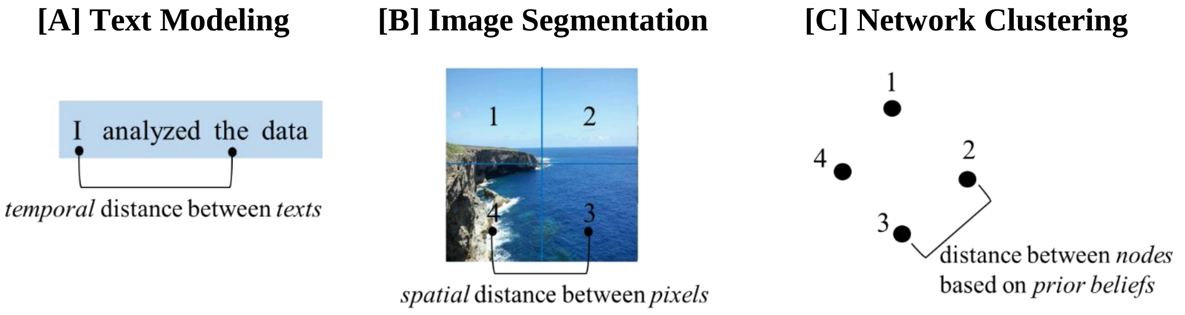
Figure 3: Directed Acyclic Graph for the ddCRP-Based LCA



Note. We represent in yellow the parameters – α (self-link parameter), M (parameter clustering), and $\boldsymbol{\gamma}^M$ (model parameters) – that are sampled in the ddCRP-based LCA.

Note that the ddCRP-based LCA has mainly been applied for segmenting documents and images. A notable feature of the method is the ease with which prior beliefs relevant for segmentation can be accommodated via a distance function. For instance, extant studies use the distance between texts (pixels) as an input in a model for determining which texts (pixels) should be grouped together (Arfa, Yusof, and Shabanzadeh 2019; Blei and Frazier 2011; see Figures 4[A] and 4[B]). Similarly, we employ the ddCRP to capture prior beliefs for how the nodes of a network (e.g., parameter network) would be aggregated based on their inter-node distance (see Figure 4[C]).

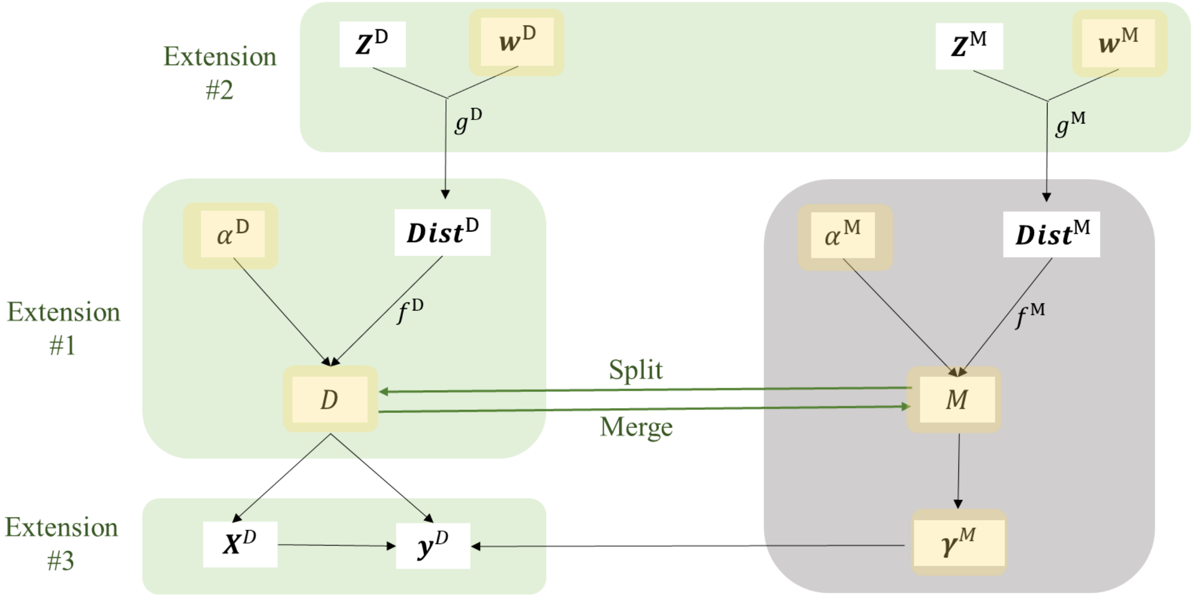
Figure 4: Distances in Text Modeling and Image Segmentation Versus Network Clustering


[A] Text Modeling
[B] Image Segmentation
[C] Network Clustering
I analyzed the data
temporal distance between *texts*
1
2
3
4
spatial distance between *pixels*
1
2
3
4
distance between *nodes* based on *prior beliefs*

2.3. Dual-Network Clustering Method

Our proposed dual-network clustering method extends the extant ddCRP-based LCA in three ways. Figure 5 illustrates our key extensions (shaded green) as compared to the extant ddCRP-based LCA (shaded grey). We will explain each extension (and the corresponding vertices and edges in Figure 5) in the remaining part of this section.

Figure 5: Directed Acyclic Graph for Our Proposed Method



Notes. Figure 5 illustrates key extensions (in green) as compared to the extant ddCRP-based LCA (in grey). We represent parameters sampled in our proposed method – e.g., *M* (parameter clustering) – in yellow. We describe the role of each parameter in the main text of this section.

(1) Allowing for dual (parameter and data) networks

Extant applications of the ddCRP cluster only a single network, the parameter network. To apply the ddCRP to two (and if need be, more) networks, as illustrated in Figure 5, we allow for two sets of ddCRPs – one for parameter clustering (denoted as M) and the other for data clustering (denoted as D). Note, as before, that for model identification, we do not allow M to be more granular than D. We accommodate this relationship (restriction) by linking D and M via a novel *split-merge* MCMC sampler as follows, which is an additional computational contribution of our research:

Split sampler. We sample D by *splitting* each parameter cluster in a previously sampled M. The split sampler is composed of three steps as illustrated in Figure 6[A]. First, we treat each parameter cluster as an independent data sub-network. Then, within each data sub-network, we induce data clustering (D) by iterating over nodes with each node being sampled to form a link (denoted as c_i^{D}) from the following ddCRP distribution:

$$p\left(c_i^{\mathrm{D}} | dist_{ii'}^{\mathrm{D}}, f^{\mathrm{D}}, \alpha^{\mathrm{D}}\right) \propto \begin{cases} \alpha^{\mathrm{D}} & \text{if } i = i' \\ f^{\mathrm{D}}\left(dist_{ii'}^{\mathrm{D}}\right) & \text{if } i \neq i' \end{cases} \tag{2}$$

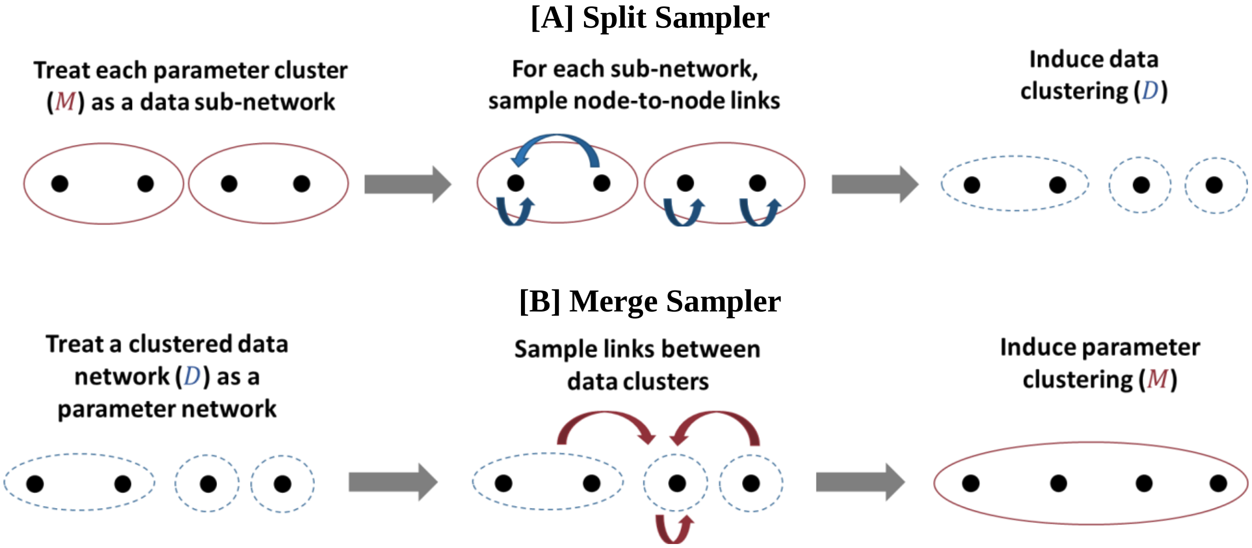
Note that Equation (2) is equivalent to the ddCRP prior introduced in Equation (1) except that it has the superscript D to denote that the ddCRP prior is imposed for data clustering.

Merge sampler. We sample M by *merging* data clusters in a previously sampled D. The merge sampler is composed of three steps as illustrated in Figure 6[B]. First, we treat the clustered data network (D) as a parameter network and then treat each data cluster as a node in the parameter network. For instance, in Figure 6[B], there are three data clusters, and so we treat these three clusters as nodes in the parameter network. Then, we induce parameter clustering (M) by iterating each data cluster d and sample a data cluster to link itself to (denoted as c_d^{M}) from the following ddCRP distribution:

$$p\left(c_d^{\mathrm{M}} \middle| dist_{dd'}^{\mathrm{M}}, f^{\mathrm{M}}, \alpha^{\mathrm{M}}\right) \propto \begin{cases} \alpha^{\mathrm{M}} & \text{if } d = d' \\ f^{\mathrm{M}}\left(dist_{dd'}^{\mathrm{M}}\right) & \text{if } d \neq d' \end{cases} \tag{3}$$

Note that Equation (3) is equivalent to the ddCRP prior introduced in Equation (1) with two key differences. First, it has the superscript M to denote that the ddCRP prior is imposed for parameter clustering. Second, since Equation (3) clusters data clusters instead of individual nodes, it uses $dist_{dd'}^{\mathrm{M}}$ (distance between data clusters d and d') instead. Note that $dist_{dd'}^{\mathrm{M}}$ is a summary (e.g., average, median) of the distances between all the pairs of nodes in the clusters d and d'.

Figure 6: Illustration of the Proposed Split-Merge Sampler


[A] Split Sampler
Treat each parameter cluster (M) as a data sub-network
For each sub-network, sample node-to-node links
Induce data clustering (D)
[B] Merge Sampler
Treat a clustered data network (D) as a parameter network
Sample links between data clusters
Induce parameter clustering (M)

(2) Modeling distances between nodes and its interpretability

Extant studies on the ddCRP assume that the pairwise distance between nodes is known a priori (see Section 2.1). This assumption is reasonable in applications of text and image segmentation, where the *actual distance* between words and between pixels, respectively, is known. However, in other problems relevant for marketers this assumption may not capture all the nuances of the context. For instance, consider the problem of how best to segment SKUs. In this context, there is no literal distance between two SKUs. Thus, we define the distance between nodes i and i'

($dist_{ii'}^{\mathrm{D}}$) as a function of the weighted average of the differences in their observed attributes (denoted as $\mathrm{diff}(\boldsymbol{z}_i^{\mathrm{D}}, \boldsymbol{z}_{i'}^{\mathrm{D}})$) where a vector of weight parameters ($\boldsymbol{w}^{\mathrm{D}}$) is estimated in conjunction with the clusterings.

$$dist_{ii'}^{\mathrm{D}} = g^{\mathrm{D}}\big(\boldsymbol{w}^{\mathrm{D}} \cdot \mathrm{diff}(\boldsymbol{z}_i^{\mathrm{D}}, \boldsymbol{z}_{i'}^{\mathrm{D}})\big) \tag{4}$$

Here, a link function $g^{\mathrm{D}}(.)$ translates the weighted differences into distances; hence, it is a non-negative and increasing function. Several types of link functions (e.g., exponential) meet these conditions. Note that we define a distance between nodes i and i' in the parameter network in a similar way:

$$dist_{ii'}^{\mathrm{M}} = g^{\mathrm{M}}\big(\boldsymbol{w}^{\mathrm{M}} \cdot \mathrm{diff}(\boldsymbol{z}_i^{\mathrm{M}}, \boldsymbol{z}_{i'}^{\mathrm{M}})\big) \tag{5}$$

A notable contribution of our method is that the latent weight parameters in the distance functions enhance the interpretability of the results as it helps to explain *why* certain levels of data and parameter granularities are chosen.

(3) Making the likelihood comparable across the levels of data aggregation

The extant method, the ddCRP-based LCA, fixes the data (e.g., image) at the most granular level and clusters parameters in a way that fits the data well. Hence, the posterior probability of choosing M (and aggregating the most granular parameters accordingly) is denoted as:

$$p(M|\boldsymbol{y}^1, \boldsymbol{X}^1) \propto \pi(M) \cdot p(\boldsymbol{y}^1|\boldsymbol{X}^1, M) \tag{6}$$

where $\pi(M)$ is the ddCRP prior for M, and $p(\boldsymbol{y}^1|\boldsymbol{X}^1, M)$ is the marginal likelihood given M and $(\boldsymbol{y}^1, \boldsymbol{X}^1)$, the most granular data (with superscript 1, as before).

In our work, we propose to cluster and aggregate both model parameters and the data. The latter allows for the possibility that the most granular data may not be the one best fitted for the focal analysis conditional on the parameter granularity (M). Hence, the posterior probability of choosing D and M together (and aggregating the most granular data and parameters, respectively) is denoted as:

$$p(D, M|\boldsymbol{y}^1, \boldsymbol{X}^1) \propto \pi(D, M) \cdot p(\boldsymbol{y}^1|\boldsymbol{X}^1, D, M) \tag{7}$$

where $\pi(D, M)$ is the ddCRP prior for (D, M), and $p(\boldsymbol{y}^1|\boldsymbol{X}^1, D, M)$ is the marginal likelihood given (D, M) and $(\boldsymbol{y}^1, \boldsymbol{X}^1)$.

It is important to note that the likelihood (and so the posterior probability) in Equation (7) is *not comparable* across different levels of data aggregation (D) and hence cannot be used directly in our approach. In particular, the total likelihood multiplies individual likelihood terms over the *number* of observations. Hence, the marginal likelihood (and so the posterior probability) is higher for coarser data as it has fewer likelihood terms. To solve this issue, as originally addressed in Kim, Bradlow, and Iyengar (2022), we note that $p(\boldsymbol{y}^1|\boldsymbol{X}^1, D, M)$ in Equation (7) actually represents the marginal likelihood for aggregated data $(\boldsymbol{y}^D, \boldsymbol{X}^D)$, which aggregates $(\boldsymbol{y}^1, \boldsymbol{X}^1)$ based on D, and so can be denoted more formally as $p_D(\boldsymbol{y}^D|\boldsymbol{X}^D, M)$:

$$p(D, M|\boldsymbol{y}^1, \boldsymbol{X}^1) \propto \pi(D, M) \cdot p_D(\boldsymbol{y}^D|\boldsymbol{X}^D, M) \tag{8}$$

Then, to make the marginal likelihood (and so the posterior probability) comparable, we extend the ddCRP and scale the marginal likelihood $p_D(\boldsymbol{y}^D|\boldsymbol{X}^D, M)$ in Equation (8) to the same data granularity (particularly, to the most granular data):

$$p_{1(D)}(D, M|\boldsymbol{y}^1, \boldsymbol{X}^1) \propto \pi(D, M) \cdot p_{1(D)}(\boldsymbol{y}^D|\boldsymbol{X}^D, M) \tag{9}$$

where $p_{1(D)}(\boldsymbol{y}^D|\boldsymbol{X}^D, M)$ and $p_{1(D)}(D, M|\boldsymbol{y}^1, \boldsymbol{X}^1)$ indicate the scaled marginal likelihood and the scaled posterior probability, respectively. We further explain the scaling process with an example in Online Appendix A.

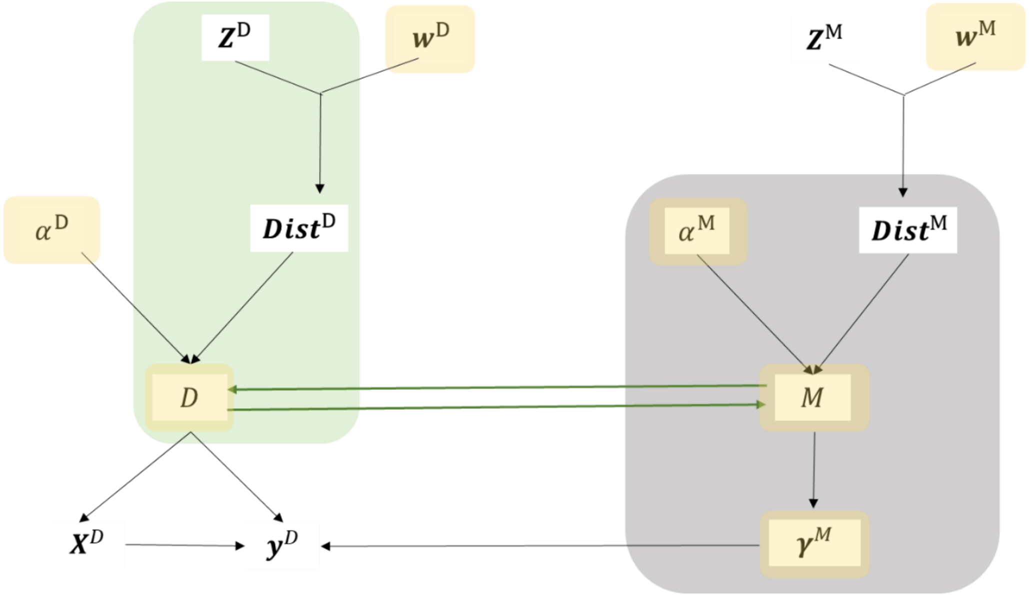
2.4. Related Methods

In Figure 7, we show that our proposed method nests two notable extant clustering methods namely, LCA (in grey) and unsupervised data clustering (in green).

LCA is typically used to cluster the parameters of a model. Specifically, LCA fixes the data at a certain level of data granularity (e.g., the most granular data) and determines the clustering of parameters. This objective is exactly what the extant ddCRP-based LCA (in grey in Figure 7) is designed for. Hence, our method nests a version of LCA that imposes the ddCRP prior on the parameter clustering.

Unsupervised data clustering methods (e.g., K-means) combine data points that have similar attributes without accounting for a downstream demand model. This objective is what the ddCRP-based data aggregation would attain (in green in Figure 7). Hence, our method nests (or more specifically, accomplishes the same goal but uses a different loss function as) unsupervised clustering methods. For instance, unlike our proposed method, K-means uses the sum of the squared distances from each node to its nearest cluster as a loss function.

Figure 7: Directed Acyclic Graph for the Proposed Framework Versus Related Methods



Notes. Our proposed framework nests LCA (in grey) and unsupervised clustering (in green). We represent in yellow the parameters sampled in our proposed method – e.g., D (data clustering) and M (parameter clustering). We explained the role of each parameter in Section 2.3.

3. Simulation Study

We use a simulation study to compare the performance of our proposed method with that of other alternatives on two key dimensions: (1) the recovery of data and parameter clusters and (2) bias in the parameters of interest. To align the simulation study with our real-data application, we consider the context of a researcher who has access to SKU-level sales data for a CPG category (e.g., orange juice) and wishes to assess the relationship between sales and prices.

3.1.Simulation Design

We begin by laying out the data generating process and then comparing our method with other alternatives.

(1) Data generating process

We generate a data set with 100 SKUs of different brands and package sizes in a category. Note that ‘brand’ and ‘package size’ are factors that extant studies tend to consider when aggregating data and parameters (e.g., Dubé and Gupta 2008; Hoch et al. 1995; Wedel and Zhang 2004). To assess the ability (and generalizability) of our method in recovering the drivers of clustering, we assume that data and parameters are clustered based on *different* factors. Specifically, we cluster data and parameters based on brands and package sizes, respectively, and generate data based on these data and parameter clusterings. We explain the data generating process in four steps:

Step 1: Generate data clustering (D). We generate data clustering by constructing a data network in which each node is an SKU and, for each node, drawing a node to link itself to (denoted as c_i^{D}) from the following density:

$$p\left(c_i^{\mathrm{D}} = i' | dist_{ii'}^{\mathrm{D}}, f^{\mathrm{D}}, \alpha^{\mathrm{D}}\right) \propto \begin{cases} \alpha^{\mathrm{D}} & \text{if } i = i' \\ f^{\mathrm{D}}\left(dist_{ii'}^{\mathrm{D}}\right) & \text{if } i \neq i' \end{cases} \tag{10}$$

which is equivalent to Equation (2), the ddCRP distribution that we introduced for data clustering. The superscript D denotes that the ddCRP distribution is relevant for data clustering. We set the self-link probability parameter $\alpha^{\mathrm{D}} = 0.1$ and f^{D} as the exponential decay function.

To cluster SKUs in the same brand together, we set the distance between SKUs i and i' (denoted as $dist_{ii'}^{\mathrm{D}}$) as an increasing function of whether they are from different *brands* or not:

$$dist_{ii'}^{\mathrm{D}} = g^{\mathrm{D}}\left(w^{\mathrm{D}} \cdot \mathbf{1}\left[z_i^{\mathrm{D}} \neq z_{i'}^{\mathrm{D}}\right]\right) \tag{11}$$

where z_i^{D} is SKU i's *brand*. Specifically, we set g^{D} as the exponential function and the weight parameter w^{D} positive (here, $w^{\mathrm{D}} = 2$). Hence, Equations (10)–(11) indicate that the SKUs in the same brand are likely to be clustered and their data would be aggregated.

Step 2: Generate parameter clustering (M). To ensure that parameter clustering (M) is not more granular than data clustering (D) (as described in the split-and-merge sampler in Section 2), we generate M by constructing a parameter network in which each node is a data cluster and, for each node, drawing a node to link itself to (denoted as c_d^{M}) from the following density:

$$p\left(c_d^{\mathrm{M}} = d' | dist_{dd'}^{\mathrm{M}}, f^{\mathrm{M}}, \alpha^{\mathrm{M}}\right) \propto \begin{cases} \alpha^{\mathrm{M}} & \text{if } d = d' \\ f^{\mathrm{M}}\left(dist_{dd'}^{\mathrm{M}}\right) & \text{if } d \neq d' \end{cases} \tag{12}$$

which is equivalent to Equation (3), the ddCRP distribution that we introduced for parameter clustering. The superscript M denotes that the ddCRP distribution is relevant for parameter clustering. As in *Step 1*, we set the self-link probability parameter $\alpha^{\mathrm{M}} = 0.1$, and f^{M} as the exponential decay function. We set $dist_{dd'}^{\mathrm{M}}$ as the average (without loss of generality) of the true distances between all the pairs of SKUs in data clusters d and d'.

To cluster SKUs of the same package size together, we set the distance between SKUs i and i' (denoted as $dist_{ii'}^{\mathrm{M}}$) as an increasing function of whether they are of different *sizes* or not:

$$dist_{ii'}^{\mathrm{M}} = g^{\mathrm{M}}\left(w^{\mathrm{M}} \cdot \mathbf{1}\left[z_i^{\mathrm{M}} \neq z_{i'}^{\mathrm{M}}\right]\right) \tag{13}$$

where z_i^{M} is SKU i's *package size*. As in *Step 1*, we set g^{M} as the exponential function and the weight parameter w^{M} positive (here, $w^{\mathrm{M}} = 2$). Hence, Equations (12)–(13) indicate that the SKUs of the same package size are likely to be clustered and their parameters would be the same in the demand model.

Step 3: Generate data and parameters given the data clustering (D) and parameter clustering (M). We generate data (here, price and unit sales) and parameters (here, intercept and price elasticity) based on the data and parameter clustering. Specifically, for each cluster d in the data clustering D, we draw its price from a $UNIF(1,10)$ and denote it by $Price_d^D$. For each cluster m in the parameter clustering M, we draw its intercept and price elasticity from $UNIF(5,7)$ and $UNIF(-3,-1)$, respectively, and denote them by $\gamma_{0,m}^M$ and $\gamma_{1,m}^M$. Finally, we generate unit sales using the price ($Price_d^D$) and parameters ($\gamma_{0,m}^M$ and $\gamma_{1,m}^M$). Specifically, for each data cluster d, we generate unit sales from the following log-log demand model, which uses $Price_d^D$ as a covariate and $\gamma_{0,m}^M$ and $\gamma_{1,m}^M$ as the corresponding parameters, and denote the generated sales by y_d^D.

$$\begin{aligned} \log(y_d^D) &= \mu_d^D + \varepsilon_d^D \\ &= \gamma_{0,m}^M + \gamma_{1,m}^M \cdot log(Price_d^D) + \varepsilon_d^D \end{aligned} \tag{14}$$

where ε_d^D is normally distributed with mean 0 and variance $V[\varepsilon_d^D]$.

Step 4: Disaggregate the generated data to the most granular level. We generate the most granular data, which is an input of our method, by disaggregating the aggregated data. Specifically, for each data cluster d, we disaggregate its price ($Price_d^D$) and unit sales (y_d^D) to the most granular (here, SKU) level. We disaggregate $Price_d^D$ by assuming that the price stays the same across SKUs in the cluster. We disaggregate y_d^D by randomly distributing it across SKUs in the cluster:

$$y_{i(d)}^D = y_d^D \cdot share_{i(d)} \tag{15}$$

where $y_{i(d)}^D$ indicates unit sales for SKU i in data cluster d. By definition, $\sum_i y_{i(d)}^D = y_d^D$ and hence $\sum_i share_{i(d)} = 1$. We draw a vector of $share_{i(d)}$ from a symmetric Dirichlet distribution to add random noise to the disaggregation process. We set the concentration parameter of the Dirichlet distribution small (here, 1) to increase the noise in the process and so in the most granular data.

Finally, when generating the SKU-level sales data, we systematically vary the signal to noise ratio (SNR) = $V[\mu_d^D]/V[\varepsilon_d^D]$. We vary SNR at two levels – 9 (high) and 1/9 (low). This manipulation will help assess the impact of SNR on the performance of our method as well as alternatives.

(2) Methods

Table 1 shows four methods. For each method, we use the most granular data as an input and choose data clustering (D) and/or parameter clustering (M). Then, we estimate a log-log demand model in Equation (14) with parameters at aggregation M to data at aggregation D.

Table 1: Comparison of Our Dual-Network Clustering Method (DM-Scaled) with Alternatives

	Choose D	Choose M	Scale the likelihood to the finest data granularity
Unsupervised Clustering	V		
LCA		V	
DM-Unscaled	V	V	
DM-Scaled	V	V	V

Notes.

1) Unsupervised clustering chooses D only. It uses the SKU features (z_i^{D} and z_i^{M}) as inputs. Since the two features – brand and size – are categorical variables, we use K-modes that extends K-means to handle the categorical inputs (e.g., Huang 1998). We choose K (the number of clusters) using the elbow method.
2) Latent Class Analysis (LCA) chooses M only while fixing data at the most granular level.
3) DM-Unscaled chooses both M and D. It still uses the standard likelihood, which is not comparable across the levels of data aggregation (as explained in Section 2.3).

4) DM-Scaled is our proposed dual-network clustering method. It chooses both M and D. It also scales the likelihood in the sampler to the finest data granularity.
5) For the last three methods, we impose weakly informative priors on their parameters: $Beta(1,1)$ on the self-link probability parameters (α^{D} and α^{M}), $Normal(0,10)$ on the model parameters ($\gamma_{0,m}^{M}$ and $\gamma_{1,m}^{M}$), and $InvGamma(1, 1)$ on the error variance ($V[\varepsilon_d^D]$). We will use the same priors in our real-data application.

(3) Metrics for evaluating the recovery of clusters

We evaluate the performance of the four methods in recovering the true underlying clusters using four metrics – Rand Index (RI), Recall-Positive (RP), Recall-Negative (RN), and Normalized Mutual Information (NMI). The first three metrics assess the recovery of pairwise clustering while the last one is based on mutual information. All four metrics range from 0 (perfect disagreement) to 1 (perfect agreement). Online Appendix B contains more details on these metrics, but we note that it is important to consider all of them as they reflect true positives, true negatives, and a combination thereof.

3.2.Simulation Results

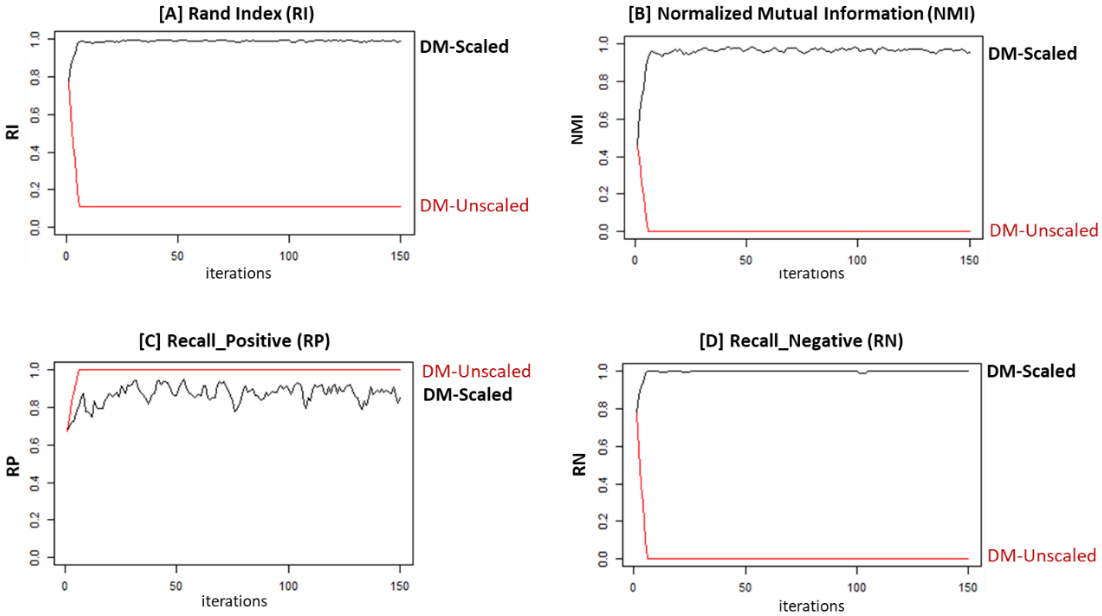
We compare the performance of our method with that of the three related methods on: (1) recovery of the underlying true data and parameter clusters and (2) the absolute bias in the parameters of interest.

(1) Recovery of data and parameter clusters

Figures 8 and 9 compare our method (DM-Scaled) and the other methods in recovering the true data and parameter clusters, respectively, based on the four metrics. Figure 8 shows that all four accuracy metrics (RI, NMI, RP, RN) for the data clusters (D) are quite high for our proposed method. In comparison, not all metrics are high under the DM-Unscaled. Specifically, its RP and RN reach values close to 1 and 0, respectively. On further inspection, this result indicates that the

DM-Unscaled chooses coarser data, because it uses the standard likelihood that increases with the decreasing number of data clusters (as explained in Section 2.3).

Figure 8: Our Method Versus the Alternatives in Recovering the True Data Clustering (D)

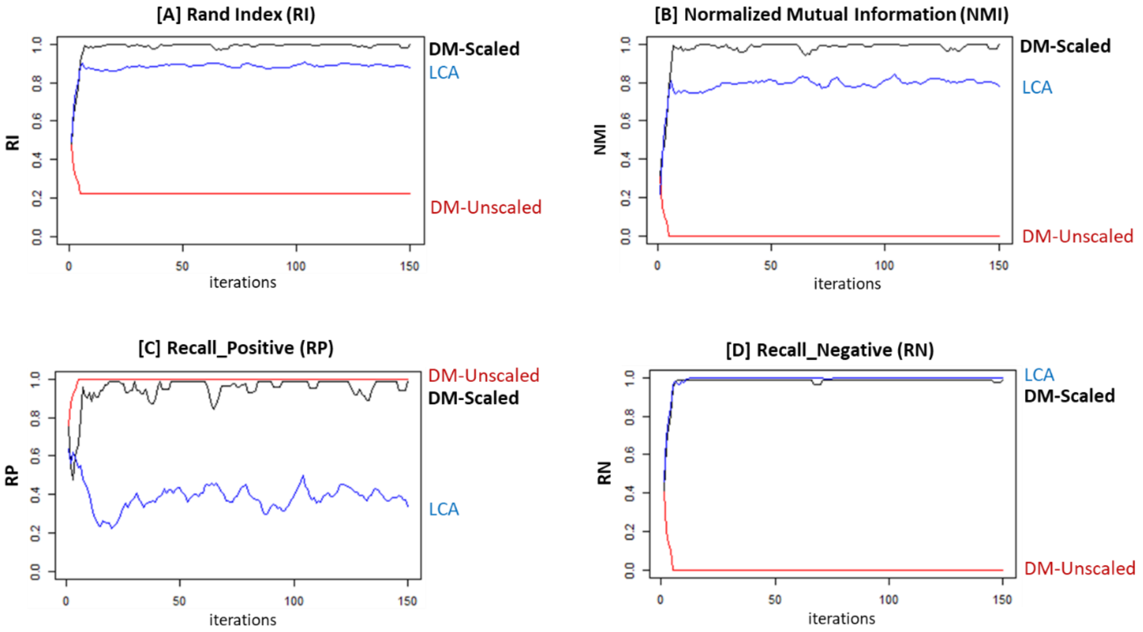


Notes.

1) All the clustering accuracy metrics under K-modes are lower than those under our method (DM-Scaled). Under K-modes, RI=0.82, NMI=0.49, RP=0.72, and RN=0.82.
2) All the plots and the accuracy metrics under K-modes are under high SNR (SNR=9).

Figure 9 shows that all the clustering accuracy metrics (RI, NMI, RP, RN) for the parameter clustering (M) are high under our proposed method. In contrast, not all the metrics are high under the other methods (DM-Unscaled, LCA). As for the DM-Unscaled, RP and RN reach values close to 1 and 0, respectively. This result implies that the DM-Unscaled chooses overly coarse parameters, as it chooses overly coarse data (as in Figures 8[C] and 8[D]) and chooses the level of parameter granularity conditional on the chosen coarse data. Next, as for the LCA, RP and RN reach values close to 0 and 1, respectively. This finding indicates that the LCA chooses overly granular parameters, as it fixes the data at the most granular level and chooses the level of parameter granularity conditional on the most granular data (as in Table 1).

Figure 9: **Our Method Versus the Alternatives in Recovering the True Parameter Clustering** $(\boldsymbol{M})$

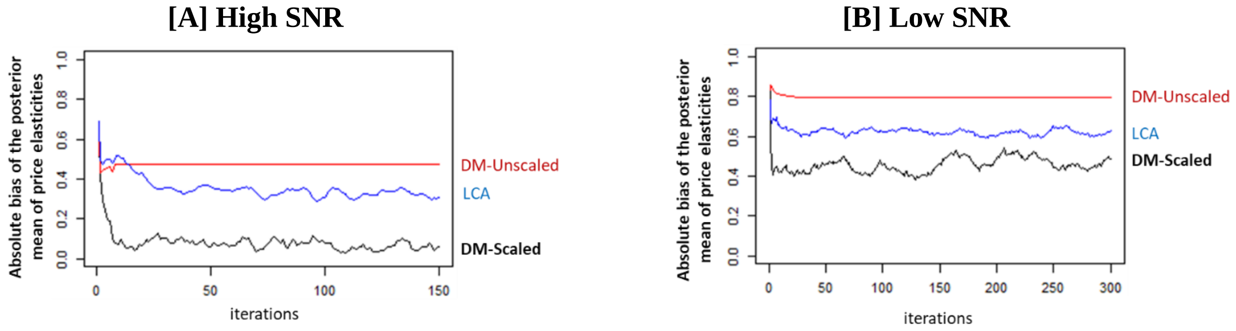


Note. All the plots are under high SNR (SNR=9).

(2) Bias in the parameters of interest

Figure 10 compares the absolute bias of the posterior mean of price elasticities across methods. The absolute bias is smaller for our proposed method than for other methods. It is because our method recovers D and M better than the alternatives (as shown in Figures 8 and 9). The bias for our model becomes negligible when SNR is high.

Figure 10: **Our Method Versus the Alternatives in Absolute Bias of the Posterior Mean of Price Elasticities**



4. Application

4.1. Data

Our data come from the Nielsen database at the Kilts Center at the University of Chicago. The original data contains unit sales, price, the incidence of in-store feature advertising, and the incidence of in-store display advertising at the stock keeping unit (SKU)-store-week level. We apply our method to the data for the refrigerated orange juice category. SKUs in this category can be categorized based on multiple features (i.e., brand, package size, package material, and pulp amount). Hence, this category is well suited to assess our proposed method. To remove other confounding effects, we make inferences using data from a single store in Philadelphia, Pennsylvania, for which in-store advertising information was available, and for a single year (2017).[2]

The SKU-weekly data from the chosen product category, store, and year contain 60 SKUs. Each week Nielsen collected price and in-store advertising only from SKUs that were sold in that week. Hence, we focus on 30 SKUs that were sold every week. Note that the chosen SKUs account for 80% of total unit sales in the orange juice category. Hence, the final most granular data is at the SKU-week level and consists of 1,560 observations (= 30 SKUs x 52 weeks).

Table 2 and Figure 11 summarize the distributions of the variables in the final SKU-weekly data. Figure 11 shows that prices differ across brands and package sizes but not so much across package materials and pulp amounts. Given this observed variation, a few extant studies on the orange juice category allow price elasticity to vary across brands (e.g., Tropicana Premium versus Private brand) and package sizes (e.g., 12 oz versus 59 oz) but not across other features (e.g., Wedel and Zhang 2004). Other extant studies also noted that they aggregated data to the brand-

[2] We use the data from the same store for the next year (2018) for out-of-sample prediction.

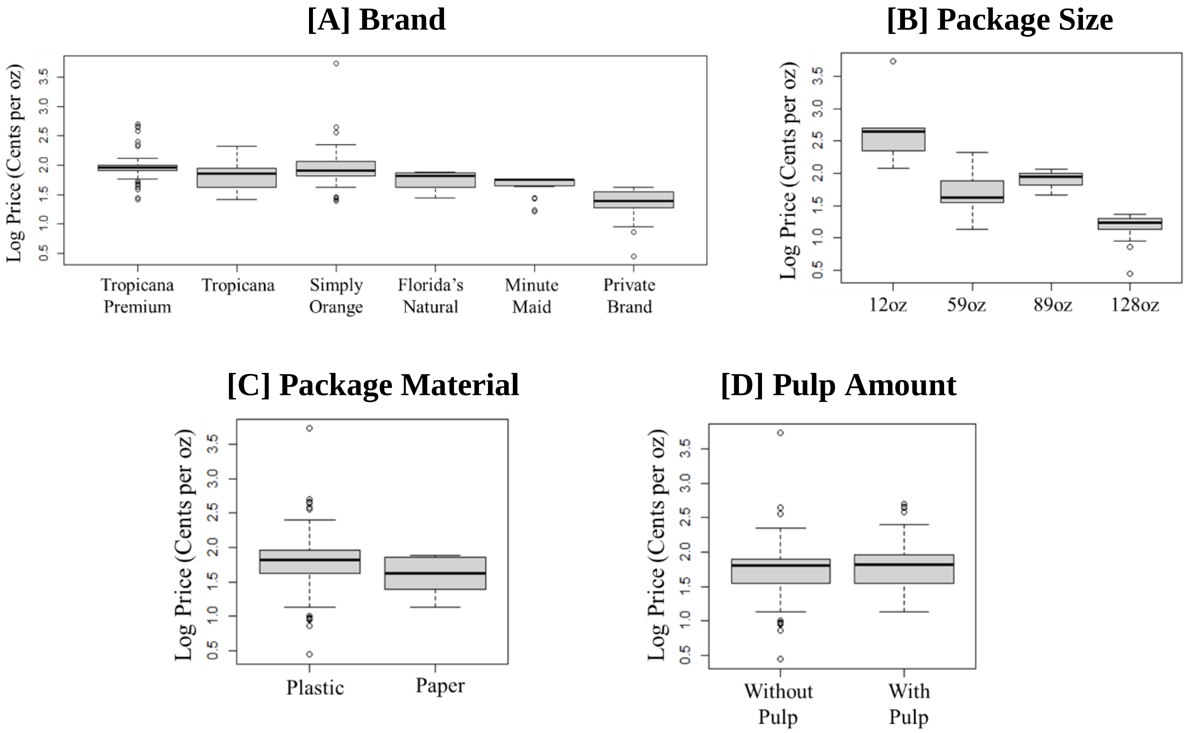
size level for the same reason (e.g., Dubé and Gupta 2008; Hoch et al. 1995). In Section 4.3, we assess the performance of this common practice – i.e., aggregating both data and parameters to the brand-size level.

Table 2: Descriptive Statistics of the Final SKU-Week-Level Orange Juice Data

	Mean	SD	Min	Max
Unit sales	21.71	27.05	1	298
Price (cents per oz)	6.11	2.42	1.55	41.67
Number of feature advertisements	0.33	0.47	0	1
Number of display advertisements	0.15	0.35	0	1

Note. The final SKU-week-level data consist of 1,560 observations (= 30 SKUs x 52 weeks).

Figure 11: Pricing Patterns of Orange Juice Across SKU Features



[A] Brand **[B] Package Size** **[C] Package Material** **[D] Pulp Amount**

4.2. Analysis

4.2.1. Data and parameter clustering methods. We choose data clustering (D) and/or parameter clustering (M) using three methods (see Table 1 in Section 3):

(1) Proposed method. Our dual-network clustering approach chooses both D and M. To estimate the effect of the four observed SKU features on the choice of data/parameter clustering, we set the distance between SKUs in the data/parameter network as a function of the weighted average of four feature-based indicator variables. The four variables are 1) whether these SKUs are from different brands or not, 2) whether their package sizes are different or not, 3) whether their package materials are different or not, and 4) whether they contain different amounts of pulp or not. The corresponding latent weights allow us to interpret the chosen data/parameter clustering with respect to the observed SKU features, as explained in Section 2.3.

(2) Unsupervised clustering. It chooses D only. In this context, we use the four SKU features – brand, package size, package material, and pulp amount – as inputs for data clustering. The four features are categorical variables (see Figure 11). Hence, we use K-modes, which extends K-means to handle categorical inputs (Huang 1998). We expect that D chosen by the unsupervised clustering method would differ from D chosen by the proposed method.

(3) Latent class analysis (LCA). It chooses M only while fixing the data at the most granular (here, SKU-weekly) level. Hence, we expect that M chosen by LCA would differ from M chosen by our method. Specifically, we expect that M based on LCA would be more granular than M chosen from our method, as previously discussed.

4.1.2. Data and parameter aggregation. We apply a demand model conditional on the chosen data and parameter clusterings (D and M). First, we aggregate the most granular data based on D. We denote the aggregated variables as sales_{dt}^{D}, feature_{dt}^{D}, display_{dt}^{D}, and price_{dt}^{D} with the superscript D indicating the sampled data clustering. Specifically, sales_{dt}^{D} is the total number of SKUs in data cluster d that were sold in week t. We use the total unit sales as a dependent variable to estimate price elasticity (i.e., the impact of price change on quantity demand) (e.g., Hoch et al. 1995; Smith, Rossi, and Allenby 2019). The variables feature_{dt}^{D} and display_{dt}^{D} refer to the total number of feature and display advertisements for SKUs in data cluster d in week t, respectively. The variable price_{dt}^{D} is the sales-weighted price averaged across SKUs in data cluster d in week t. We apply the following log-log demand model to the aggregated data.

$$\begin{aligned} &\log(\text{sales}_{dt}^{D}) \\ &= \gamma_{1,m}^{M}\log(\text{price}_{dt}^{D}) + \gamma_{2,m}^{M}\log(\text{feature}_{dt}^{D}+1) + \gamma_{3,m}^{M}\log(\text{display}_{dt}^{D}+1) + \gamma_{d}^{\mathrm{D}} + \varepsilon_{dt}^{D} \end{aligned} \tag{16}$$

In the model, we aggregate model parameters – price, feature advertising, and display advertising elasticities – based on M. We denote the aggregated parameters as $\gamma_{1,m}^{M}$, $\gamma_{2,m}^{M}$, and $\gamma_{3,m}^{M}$, respectively, with the superscript M indicating the sampled parameter clustering.

4.3.Results

We use MCMC sampling to fit the proposed method and the LCA. For each method, we run three parallel chains with different starting values for 300 iterations each.[3] For each chain, we discard the first 200 iterations as burn-in and use the last 100 iterations for analysis. We compute the Gelman-Rubin statistic (Gelman and Rubin 1992) on the post-burn-in iterations for each parameter. The statistic is always less than 1.2, suggesting that the model convergence is satisfactory.

4.3.1. Model fit. We use the log of the in-sample Bayes factor ($\log \mathrm{BF}_{\mathrm{in}}$) and the in-sample mean absolute errors (MAE) to compare the in-sample fit of our method with that of the LCA. First,

[3] Rapid mixing and convergence is one notable feature of the ddCRP sampler (Blei and Frazier 2011). This feature has been observed in many of its application papers (e.g., Arfa, Yusof, and Shabanzadeh 2019; Ghosh et al. 2011). For instance, Ghosh et al. (2011) clustered 1,000 image pixels using the ddCRP sampler, and the sampler was converged within 50 iterations. Similarly, Blei and Frazier (2011) clustered around 3,000 texts in articles using the ddCRP sampler, and the sampler was converged within 100 iterations.

$\log \mathrm{BF}_{\mathrm{in}}$ denotes the difference between the log of the in-sample *scaled* marginal likelihood of the proposed method ($\mathrm{SML}_{\mathrm{in,Proposed}}$) and that of the LCA ($\mathrm{SML}_{\mathrm{in,LCA}}$). We estimate the log of the scaled marginal likelihood by taking the harmonic mean of the scaled likelihood in post-burn-in MCMC iterations. $\log \mathrm{BF}_{\mathrm{in}}$ greater than 5 is considered as strong evidence for our method over LCA (Kass and Raftery 1995). $\log \mathrm{BF}_{\mathrm{in}} = \log \mathrm{SML}_{\mathrm{in,Proposed}} - \log \mathrm{SML}_{\mathrm{in,LCA}} = 33.0$, indicating that our method outperforms the LCA model in sample. We also find that the posterior mean of the in-sample MAE (in estimating the log of SKU-weekly unit sales) is lower under our method than that under the LCA (i.e., $0.827 < 0.903, \text{an } 8\% \text{ reduction}$).

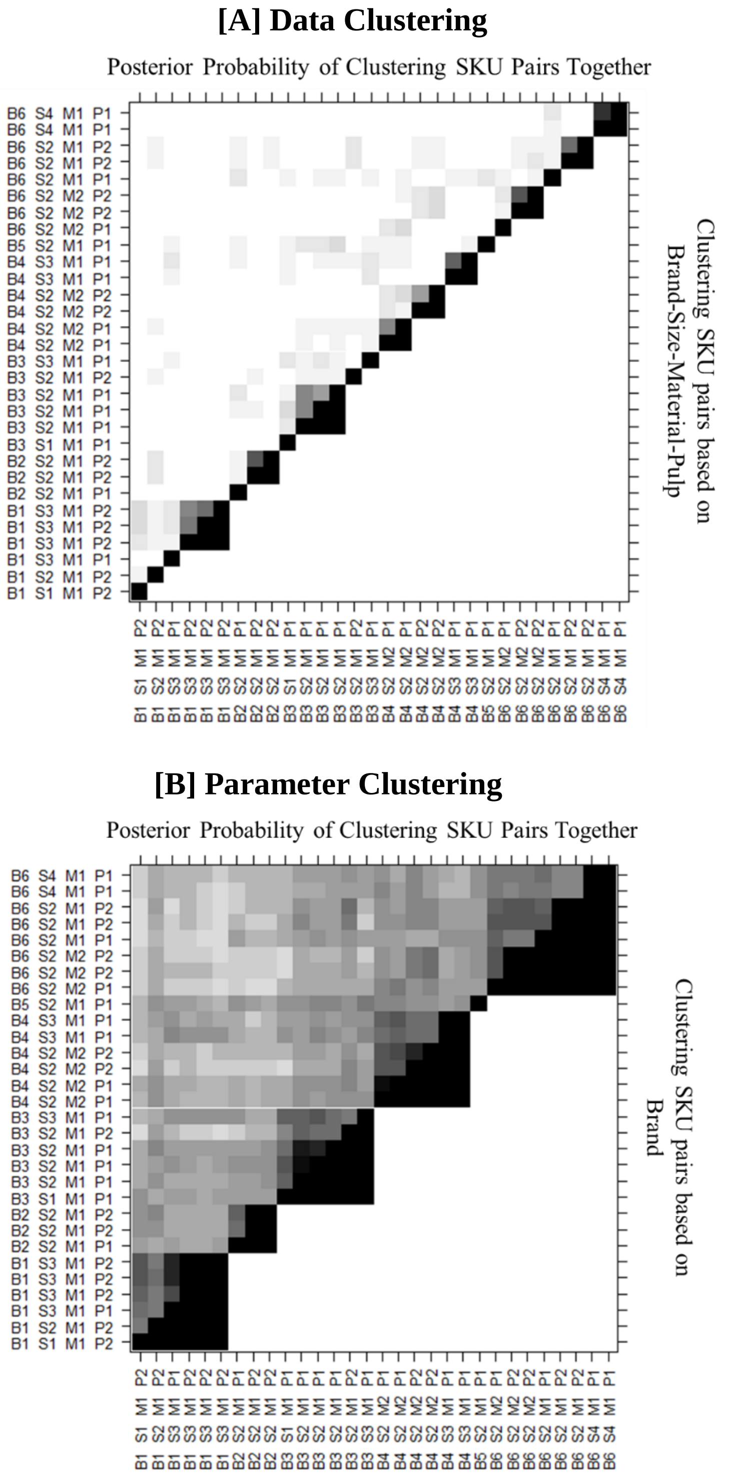
While the primary motivation for our approach is model selection using in-sample fit, researchers may wish to use out-of-sample prediction as their primary goal. To this end, we also assess the out-of-sample predictive validity. We use the log of the out-of-sample Bayes factor ($\log \mathrm{BF}_{\mathrm{out}}$) and the out-of-sample MAE to compare the out-of-sample performance of our method with that of LCA. We find that $\log \mathrm{BF}_{\mathrm{out}} = 47.5$, suggesting that our method performs significantly better than the LCA model in out of sample as well. The posterior mean of the out-of-sample MAE is also lower under our method than that under the LCA (i.e., $1.055 < 1.195, \text{a } 5\% \text{ reduction}$). Hence, the results for both the in-sample and out-of-sample model fit demonstrate the importance of selecting the levels of data and parameter granularities (D and M) simultaneously.

4.3.2. Inference for data and parameter clusterings. Figures 12[A] and 12[B] summarize the posterior distributions of data and parameter clusterings. Specifically, each cell in the upper left corner captures the proportion of the post-burn-in iterations that each SKU pair was in the same data/parameter clusters. The darker colors imply higher clustering probabilities.

In Figure 12[A], the bottom right corner represents the data clustering structure implied by using the *observed* brands, package sizes, package materials, and pulp amount. By comparing the top and bottom corner patterns, we find that the data clusterings with high posterior probability are consistent with the clusterings based on brands, package sizes, package materials, and pulp amount. Our finding suggests that while researchers tend to assume that data variance across brands and sizes are solely important in understanding the effects of marketing actions on sales, it is important to also include the data variance across package materials and pulp amounts.

In Figure 12[B], the bottom right corner represents the parameter clustering structure implied by the brands. Here, we find that the parameter clusterings with high posterior probability are fairly consistent with groupings based on brands. We find this result particularly interesting because it cautions against the aforementioned common practice of setting the parameter granularity at the brand-size level. For instance, while researchers tend to assume that price elasticities would differ across both brands and sizes, our result suggests that the price elasticities in this context differ across the brands and not so much across the sizes.

Figure 12: Heatmaps for Data and Parameter Clusterings

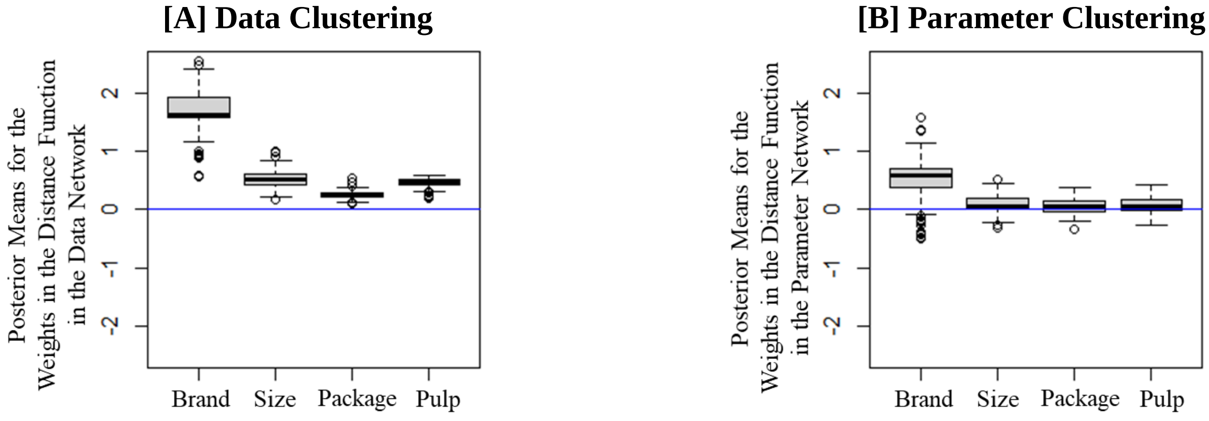

[A] Data Clustering

[B] Parameter Clustering

Notes. The x-axis and y-axis contain labels for the 30 SKUs. Each label consists of the four SKU features: brand, package size, package material, and pulp amount. The first feature includes six brands (B1:Tropicana Premium, B2:Tropicana, B3:Simply Orange, B4:Florida's Natural, B5:Minute Maid, B6:Private Brand). The second feature includes four package sizes (S1:12oz, S2:59oz, S3:89oz, S4:128oz). The third feature includes two package materials (M1:plastic, M2:paper). The last feature includes two pulp amounts (P1:without pulp, P2:with pulp).

While the above figures provide one way of interpreting the data and parameter clusterings, large-scale heatmaps are not always clear particularly when there are many SKUs. To address this concern, we compare the posterior means for the latent weights in the corresponding data and parameter distance functions. In Figure 13[A], the posterior means for the ‘brand,’ ‘size,’ ‘package,’ and ‘pulp’ weight parameters are significantly greater than 0, indicating that the data for SKUs within the same brand, size, package material, and pulp amount are likely to be aggregated together. In Figure 13[B], the posterior mean for the ‘brand’ weight parameter is significantly greater than 0, indicating that the parameters for SKUs in the same brand are likely to be aggregated together. The results in Figure 13 are in line with the interpretation that we gave for our chosen data and parameter clusterings.

Figure 13: The Posterior Means for the Latent Weights in the Distance Functions



[A] Data Clustering

[B] Parameter Clustering

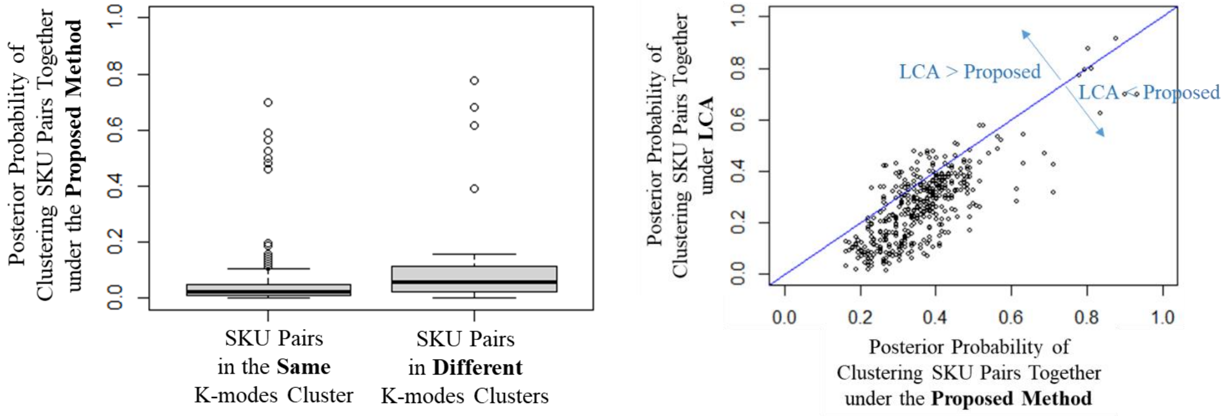
Finally, it is important to note that our chosen data and parameter clusterings (D and M) differ from those under the extant clustering approaches:

(1) Unsupervised clustering. The data clustering (D) under our method differs from D under unsupervised clustering. Specifically, in Figure 14[A], we assign each SKU pair to two groups based on whether the corresponding two SKUs are clustered together under K-modes or not. Figure 14[A] shows that the posterior data clustering probability under our method is not significantly different between these two groups. This result indicates that the D sampled under our method differs from D chosen under K-modes. It also reinforces the fact that, unlike our method, unsupervised clustering selects D only.

(2) LCA. The parameter clustering (M) under our method differs from M based on the LCA model. Specifically, in Figure 14[B], for each SKU pair, we compare the posterior probability of clustering the corresponding SKU pair together under the LCA with that under our method. It is worthwhile to note that for 88% of the SKU pairs, the posterior parameter clustering probability is greater under our method than under the LCA. That is, the LCA selects more granular parameters than our method does.

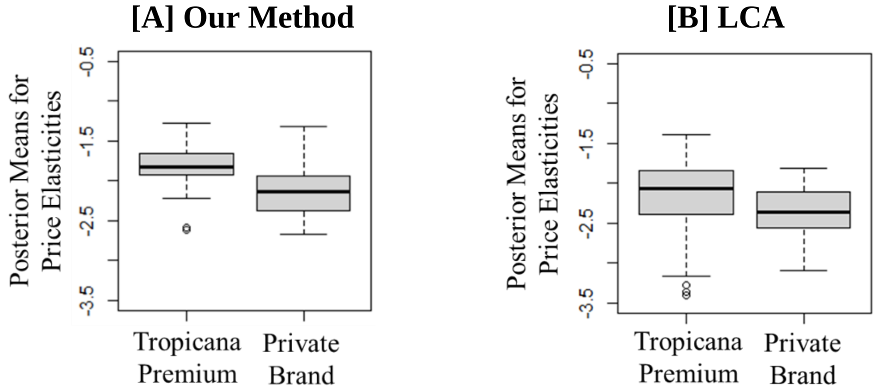
Figure 14: Clusterings from Our Proposed Method Versus the Extant Methods

[A] Unsupervised Clustering **[B] LCA**



4.3.3. Inference for price elasticities. To further highlight our findings, we next compare the price elasticities from our proposed method with those from the LCA. We focus on two brands: Tropicana Premium (which is in the highest price tier – i.e., 7.89 cents per oz on average) and the private brand (which is in the lowest price tier – i.e., 4.02 cents per oz on average). Figure 15[A] shows that price elasticities sampled from our method are more elastic for the private brand than for Tropicana Premium. Specifically, the price elasticity for the private brand is more elastic than that for Tropicana Premium in 81% of the post-burn-in iterations. Figure 15[B] shows that this cross-brand difference becomes less salient under the LCA. Specifically, the price elasticity for the private brand is more elastic than that for Tropicana Premium in 69% of the post-burn-in iterations. This result suggests that the LCA may lead to imprecise inference because the LCA fixes data at the most granular level, which can be noisier as compared to more aggregated data.

Figure 15: The Posterior Means for the Price Elasticities Across Brands



[A] Our Method

[B] LCA

To assess the implications of these findings on optimal pricing decisions, we compare the optimal prices proposed by our method and those by the LCA. Note that price is defined as “optimal” if it maximizes profit:

$$\max_{\text{price}_{dt}^{D}} \pi_{dt}^{D} = (\text{price}_{dt}^{D} - c_v) \cdot \text{sales}_{dt}^{D} - c_{\text{feature}} \cdot \text{feature}_{dt}^{D} - c_{\text{display}} \cdot \text{display}_{dt}^{D} \tag{17}$$

where c_v, c_{feature}, and c_{display} capture the variable cost (cents) per oz, the unit price of feature advertisement, and the unit price of display advertisement, respectively. We derive the optimal price by solving the following:

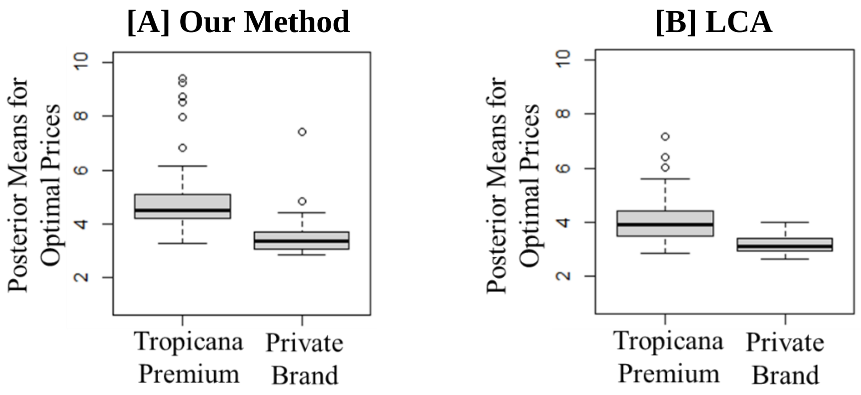
$$\frac{\partial \pi_{dt}^{D}}{\partial \text{price}_{dt}^{D}} = 0 \tag{18}$$

The resulting expression for optimal price is well-known and given by:

$$\text{price}_{dt}^{D,*} = \frac{c_v}{1+\frac{1}{\gamma_{1,m}^{M}}} \tag{19}$$

For the analysis, we assume $c_v = 2.02$ cents per oz for SKUs in the higher-price-tier brand and $c_v = 1.79$ cents per oz for SKUs in the lower-price-tier brand (e.g., Kadiyali, Chintagunta, and Vilcassim 2000). Note that the resulting optimal prices are within the data bounds – i.e., [1.55, 41.67] cents per oz. Figure 16[A] shows that the optimal price for Tropicana Premium is higher than that for the private brand under our method. Specifically, the optimal price for Tropicana Premium is higher than that for the private brand in 94% of the post-burn-in iterations. Figure 16[B] shows that this cross-brand difference becomes less salient under the LCA. Specifically, the optimal price for Tropicana Premium is higher than that for the private brand in 79% of the post-burn-in iterations. It is because the uncertainty in the price elasticity is higher under the LCA than under our method (see Figure 15). Thus, if we assume that our method selects the correct data and parameter clusterings, marketers who apply the LCA could (mistakenly) set the price for Tropicana Premium closer to the price for the private brand.

Figure 16: The Posterior Means for Optimal Prices Across Brands



[A] Our Method

[B] LCA

5. Conclusions and Future Research

Researchers often use well-known model selection tools (e.g., BIC, marginal likelihood) to select the best-fitted model. While researchers pay a lot of attention to their model specification, they unknowingly make two important modeling decisions – data and parameter granularities – and select a model *conditional* on these dual decisions. While extant research has applied various heuristics to justify the decisions – e.g., using the most granular data and parameters, or following a common practice (e.g., Dubé and Gupta 2008; see Section 4) – how to do so is not straightforward.

In this research, we propose a Bayesian dual-network clustering method as a novel way to select data and parameter granularities jointly. Formally, we represent each unit as a node in two networks – data and parameter networks – and cluster the networks to select the best-fitting levels of data and parameter granularities. To do so, we propose a novel extension of the Bayesian nonparametric clustering method, the distance dependent Chinese Restaurant Processes (ddCRP), originally used to cluster texts and images. By representing data and parameters in a generalized way (particularly, in a network structure), our proposed method is *flexible* and allows for many different types of units (e.g., SKU, person, time, space). A notable feature of our model is the *interpretability* of the results. By relating distances between nodes (e.g., SKUs) in the two networks to their observed attributes (e.g., brand, package size), the proposed method sheds light on why certain levels of data and parameter granularities are chosen.

We highlight the performance of our proposed method as compared to that from extant methods (e.g., LCA, unsupervised clustering) using a simulation study and a real data application. In the simulation study, we show that our method recovers the true levels of data and parameter granularities better than the extant methods do. It is because unlike our method, the extant methods select *either* data or parameter granularity while *conditioning* on the other. In particular, the LCA chooses overly granular parameters because it fixes the data at the most granular level and chooses the level of parameter granularity conditional on the most granular data. We apply our proposed method as well as the extant ones to the Nielsen scanner data and confirm several findings from the simulation study. Furthermore, the inference of demand parameters (e.g., price elasticity) and optimal marketing decisions obtained by our method differ substantially from those by the extant methods.

Our proposed framework is flexible (as explained above) and can be applied to a variety of contexts relevant to marketers. One example is a temporal or spatial analysis, for which researchers often decide whether to use data at the most granular level (e.g., daily) or aggregate it to a coarser level (e.g., weekly). This decision becomes particularly important when researchers can readily access granular data (e.g., data with a per-second frequency) which can be noisy and sparse. Another example is a customer (or group) level analysis where researchers often cluster customers and then aggregate data and/or parameters based on the chosen clustering.

There are methodological extensions to our modeling framework that are worthy of deeper investigation. First, whereas our method aggregates data and parameters each along a single dimension, there are contexts where researchers wish to aggregate each along multiple dimensions. This can be done by extending our dual-network method to accommodate multiple networks. For instance, consider a researcher who wishes to perform a demand analysis using customer-week-level panel data. The researcher can aggregate the data and parameters across both customers and weeks by clustering four (instead of two) networks – i.e., two networks for cross-customer and cross-week data aggregation and the other two for parameter aggregation. Second, whereas the sampler for estimating our proposed model is reasonable for small-to-moderate data, it can be computationally costly for a large data set (e.g., with millions of customers). It will be interesting to explore how our sampler can be made computationally more efficient by applying a mini-batching technique, which has been used to scale up other machine learning algorithms (e.g., De Sa, Chen, and Wong 2018; Smolyakov, Liu, and Fisher 2018).

In summary, we hope that the generality of our method makes it an important tool for marketing scholars who have to make critical decisions regarding data and parameter granularities across a wide variety of contexts.

References

Arora N, Allenby GM, Ginter JL (1998) A hierarchical Bayes model of primary and secondary demand. *Marketing Science.* 17(1):29-44.

Arfa R, Yusof R, Shabanzadeh P (2019) Novel trajectory clustering method based on distance dependent Chinese restaurant process. *Peer Journal of Computer Science*. 8:1-5.

Blei DM, Frazier PI (2011) Distance dependent Chinese restaurant processes. *Journal of Machine Learning Research.* 12:2461-2488.

Chen Y, Yang S (2007) Estimating disaggregate models using aggregate data through augmentation of individual choice. *Journal of Marketing Research.* 44(November):613-621.

Christen M, Gupta S, Porter JC, Staelin R, Wittink DR (1997) Using market-level data to understand promotion effects in a nonlinear model. *Journal of Marketing Research.* 34(3):322-334.

De Sa C, Chen V, Wong W (2018) Minibatch Gibbs sampling on large graphical models. *Proceedings of the 35th International Conference on Machine Learning*.

Dubé JP, Gupta S (2008) Cross-brand pass-through in supermarket pricing. *Marketing Science.* 27(3):324-333.

Gelman A, Rubin DB (1992) Inference from iterative simulation using multiple sequences. *Statistical Science*. 7(4): 457-472.

Ghosh S, Ungureanu AB, Sudderth EB, Blei DM (2011) Spatial distance dependent Chinese restaurant processes for image segmentation. *Advances in Neural Information Processing Systems*. 24.

Hoch SJ, Kim BO, Montgomery AL, Rossi PE (1995) Determinants of store-level price elasticity. *Journal of Marketing Research.* 32(1):17-29.

Huang Z (1998) Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*. 2:283-304.

Kadiyali V, Chintagunta P, Vilcassim N (2000) Manufacturer-retailer channel interactions and implications for channel power: An empirical investigation of pricing in a local market. *Marketing Science*. 19(2):127-148.

Kass RE, Raftery AE (1995) Bayes factor. *Journal of the American Statistical Association*. 90(430):773-795.

Kim M, Bradlow ET, Iyengar R (2022) Selecting data granularity and model specification using the scaled power likelihood with multiple weights. *Marketing Science*. 41(4):848-866.

Morwitz VG, Schmittelein D (1992) Using segmentation to improve sales forecasts based on purchase intent: Which "intenders" actually buy? *Journal of Marketing Research.* 29(4):391-405.

Musalem A, Bradlow ET, Raju JS (2008) Who's got the coupon? Estimating consumer preferences and coupon usage from aggregate information. *Journal of Marketing Research.* 45(6):715-730.

Smith AN, Rossi PE, Allenby GM (2019) Inference for product competition and separable demand. *Marketing Science*. 38(4):690-710.

Smolyakov V, Liu Q, Fisher JW (2018) Adaptive scan Gibbs sampler for large scale inference problems. *Proceedings of the 32nd International Conference on Machine Learning*.

Wedel M, Zhang J (2004) Analyzing brand competition across subcategories. *Journal of Marketing Research*. 41(4):448-456.

Online Appendix A. Likelihood Scaling

We illustrate how we scale the likelihood at data aggregation D to the most granular level using a simple example. Suppose that we have two observations and want to decide whether to aggregate these observations or not. Formally, we want to decide between two levels of data granularity – the most granular level (denoted as 1) and the aggregated level (denoted as D) – while fixing the level of parameter granularity at M. It is important to note that we cannot select data granularity by just comparing the standard likelihood at (D, M) with that at $(1, M)$ (as explained in Section 2.3).

We address this noncomparability issue by scaling the likelihood at (D, M) to $(1, M)$. The likelihood scaling that we propose is composed of two steps. First, we disaggregate the aggregated data to the most granular level. We denote the aggregated data and disaggregated observations by y^D and $(\tilde{y}_1^1, \tilde{y}_2^1)$, respectively, with the superscript indicating the corresponding data granularity. Next, we write the likelihood using the disaggregated observations and denote it by $p_1(\tilde{y}_1^1, \tilde{y}_2^1|M)$. If only a single candidate of $(\tilde{y}_1^1, \tilde{y}_2^1)$ exists, we can scale the likelihood at (D, M) – $p_D(y^D|M)$ – by writing the likelihood with this single disaggregated candidate:

$$p_{1(D)}(y^D|M) = p_1(\tilde{y}_1^1, \tilde{y}_2^1 \mid \tilde{y}_1^1 + \tilde{y}_2^1 = y^D, M) \tag{A1}$$

where $p_{1(D)}(y^D|M)$ is the scaled likelihood where subscript 1 indicates the most granular data. In comparison, if more than a single candidate exists (as in many real empirical cases), we can scale $p_D(y^D|M)$ by taking the probability-weighted average of Equation (A1) across all the possible candidates of $(\tilde{y}_1^1, \tilde{y}_2^1)$:

$$\begin{aligned} &p_{1(D)}(y^D|M) && \text{(A2)} \\ &= \int_{\min(\tilde{y}_2^1)}^{\max(\tilde{y}_2^1)} \int_{\min(\tilde{y}_1^1)}^{\max(\tilde{y}_1^1)} \Pr(\tilde{y}_1^1, \tilde{y}_2^1 | \tilde{y}_1^1 + \tilde{y}_2^1 = y^D, M) \cdot p_1(\tilde{y}_1^1, \tilde{y}_2^1 | \tilde{y}_1^1 + \tilde{y}_2^1 = y^D, M)\, d(\tilde{y}_1^1, \tilde{y}_2^1) \end{aligned}$$

where $\Pr(\tilde{y}_1^1, \tilde{y}_2^1 | \tilde{y}_1^1 + \tilde{y}_2^1 = y^D, M)$ is a weight term and indicates how likely each candidate $(\tilde{y}_1^1, \tilde{y}_2^1)$ is observed conditional on that $\tilde{y}_1^1$ and $\tilde{y}_2^1$ sum up to y^D. As $\tilde{y}_2^1 = y^D - \tilde{y}_1^1$, we can simplify Equation (A2) by expressing it with respect to $\tilde{y}_1^1$:

$$p_{1(D)}(y^D|M) = \int_{\min(\tilde{y}_1^1)}^{\max(\tilde{y}_1^1)} \Pr(\tilde{y}_1^1, y^D - \tilde{y}_1^1\ |M) \cdot p_1(\tilde{y}_1^1, y^D - \tilde{y}_1^1\ \ |M)\, d(\tilde{y}_1^1) \tag{A3}$$

The weight term, as it indicates how likely each candidate $(\tilde{y}_1^1, y^D - \tilde{y}_1^1)$ is observed, is proportional to each candidate's likelihood. Since all the weights sum up to 1, we can express the weight term as follows:

$$\Pr(\tilde{y}_1^1, y^D - \tilde{y}_1^1|M) = \frac{p_1(\tilde{y}_1^1, y^D - \tilde{y}_1^1|M)}{\int_{\min(\tilde{y}_1^1)}^{\max(\tilde{y}_1^1)} p_1(\tilde{y}_1^1, y^D - \tilde{y}_1^1|M)d(\tilde{y}_1^1)} \tag{A4}$$

By replacing $\Pr(\tilde{y}_1^1, y^D - \tilde{y}_1^1|M)$ in Equation (A3) with the right-hand side of Equation (A4):

$$p_{1(D)}(y^D|M) = \int_{\min(\tilde{y}_1^1)}^{\max(\tilde{y}_1^1)} \frac{p_1(\tilde{y}_1^1, y^D - \tilde{y}_1^1|M)}{\int_{\min(\tilde{y}_1^1)}^{\max(\tilde{y}_1^1)} p_1(\tilde{y}_1^1, y^D - \tilde{y}_1^1|M)d(\tilde{y}_1^1)} \cdot p_1(\tilde{y}_1^1, y^D - \tilde{y}_1^1|M)\, d(\tilde{y}_1^1) \tag{A5}$$

Since the denominator of Equation (A5) – $\int_{\min(\tilde{y}_1^1)}^{\max(\tilde{y}_1^1)} p_1(\tilde{y}_1^1, y^D - \tilde{y}_1^1|M)\, d(\tilde{y}_1^1)$ – does not depend on $\tilde{y}_1^1$, we take the denominator outside the integral:

$$p_{1(D)}(y^D|M) = \frac{\int_{\min(\tilde{y}_1^1)}^{\max(\tilde{y}_1^1)} p_1^2(\tilde{y}_1^1, y^D - \tilde{y}_1^1|M)d(\tilde{y}_1^1)}{\int_{\min(\tilde{y}_1^1)}^{\max(\tilde{y}_1^1)} p_1(\tilde{y}_1^1, y^D - \tilde{y}_1^1|M)d(\tilde{y}_1^1)} \tag{A6}$$

If the integrals in Equation (A6) cannot be evaluated in a closed form, we approximate them using a numerical integration technique:

$$p_{1(D)}(y^D|M) \cong \frac{\sum_{\tilde{y}_1^1 \in S} p_1^2(\tilde{y}_1^1, y^D - \tilde{y}_1^1|M)}{\sum_{\tilde{y}_1^1 \in S} p_1(\tilde{y}_1^1, y^D - \tilde{y}_1^1|M)} \tag{A7}$$

where S is a set of m (e.g., 1000) candidates for $\tilde{y}_1^1$. Since this example requires one-dimensional integration, we can choose S by using traditional integration methods such as a trapezoidal rule.

However, high-dimensional numerical integration is required in many empirical cases, and the traditional methods are not applicable. Consider a simple example that we have five observations at the most granular level and want to decide whether to aggregate these observations to a single observation or not. We can denote the aggregated data as $y^D = \sum_{i=1}^{5} \tilde{y}_i^1$ and the corresponding scaled likelihood as $f_{1(D)}(y^D|M)$. Even in this simple example, as shown in Equation (A8), we should integrate the likelihood functions over four dimensions to compute the scaled likelihood:

$$p_{1(D)}(y^D|M) = \frac{\int_{\min(\tilde{y}_4^1)}^{\max(\tilde{y}_4^1)} \int_{\min(\tilde{y}_3^1)}^{\max(\tilde{y}_3^1)} \int_{\min(\tilde{y}_2^1)}^{\max(\tilde{y}_2^1)} \int_{\min(\tilde{y}_1^1)}^{\max(\tilde{y}_1^1)} p_1^2\left(\tilde{y}_1^1, \tilde{y}_2^1, \tilde{y}_3^1, \tilde{y}_4^1, y^D - \sum_{i=1}^4 \tilde{y}_i^1 \,|M\right) d\left(\tilde{y}_1^1, \tilde{y}_2^1, \tilde{y}_3^1, \tilde{y}_4^1\right)}{\int_{\min(\tilde{y}_4^1)}^{\max(\tilde{y}_4^1)} \int_{\min(\tilde{y}_3^1)}^{\max(\tilde{y}_3^1)} \int_{\min(\tilde{y}_2^1)}^{\max(\tilde{y}_2^1)} \int_{\min(\tilde{y}_1^1)}^{\max(\tilde{y}_1^1)} p_1\left(\tilde{y}_1^1, \tilde{y}_2^1, \tilde{y}_3^1, \tilde{y}_4^1, y^D - \sum_{i=1}^4 \tilde{y}_i^1 \,|M\right) d\left(\tilde{y}_1^1, \tilde{y}_2^1, \tilde{y}_3^1, \tilde{y}_4^1\right)} \tag{A8}$$

If we use the traditional integration methods (e.g., trapezoidal rule), the number of function evaluations grows exponentially as the number of dimensions (d) increases.

To overcome this curse of dimensions, we use a standard Monte Carlo integration method in which the number of function evaluations is independent of the number of dimensions. In this example, we can approximate the integrals in Equation (A8) by randomly selecting just m (not m^d) candidates for $(\tilde{y}_1^1, \tilde{y}_2^1, \tilde{y}_3^1, \tilde{y}_4^1)$ and evaluating their likelihood function :

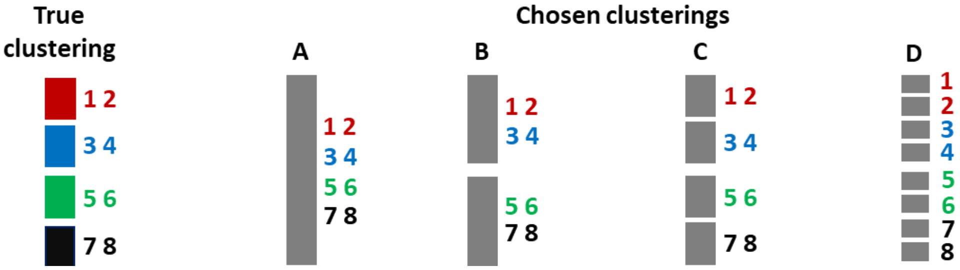
$$p_{1(D)}(y^D|M) \cong \frac{\sum_{(\tilde{y}_{1,1}, \tilde{y}_{1,2}, \tilde{y}_{1,3}, \tilde{y}_{1,4}) \in S} p_1^2(\tilde{y}_1^1, \tilde{y}_2^1, \tilde{y}_3^1, \tilde{y}_4^1, y^D - \sum_{i=1}^4 \tilde{y}_i^1 \,|M)}{\sum_{(\tilde{y}_{1,1}, \tilde{y}_{1,2}, \tilde{y}_{1,3}, \tilde{y}_{1,4}) \in S} p_1(\tilde{y}_1^1, \tilde{y}_2^1, \tilde{y}_3^1, \tilde{y}_4^1, y^D - \sum_{i=1}^4 \tilde{y}_i^1 \,|M)} \tag{A9}$$

where S is a set of m (e.g., 1000) candidates for $(\tilde{y}_1^1, \tilde{y}_2^1, \tilde{y}_3^1, \tilde{y}_4^1)$.

Online Appendix B. Clustering Evaluation Metrics

We evaluate the performance of recovering the true data and parameter clusterings (granularities) using four metrics (i.e., Rand index, Recall-Positive, Recall-Negative, and Normalized Mutual Information). The first three metrics are based on pairwise clustering comparisons, and the last one is based on mutual information. All four metrics range from 0 (perfect disagreement) to 1 (perfect agreement). We explain their underpinnings using an example (see Figure B1):

Figure B1: Examples of True Clustering Versus Chosen Clusterings



(1) **Rand index (RI)** is the proportion of true positives (i.e., node pairs that are correctly clustered to the same cluster) and true negatives (i.e., node pairs that are correctly clustered to different clusters) among all the node pairs:

$$\mathrm{RI} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}} \tag{B1}$$

where TP, TN, FP, FN are true positives, true negatives, false positives, false negatives, respectively. RI for clustering A, B, C, and D is 0.1, 0.6, 1.0, and 0.9, respectively. RI reaches 1 when a chosen clustering (here, clustering C) is equivalent to the true one and decreases toward 0 as a chosen clustering becomes dissimilar to the true one. Since RI considers both false positives and false negatives, we cannot infer from RI how much a chosen clustering suffers from false positives and how much from false negatives. For this reason, we also use the second and third metrics.

(2) **Recall_Positive (RP)** is the proportion of node pairs that are correctly clustered to the "same" cluster in the chosen clustering among node pairs that are clustered to the "same" cluster in the true clustering:

$$\mathrm{RP} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{B2}$$

If we revisit the example in Figure B1, RP for clustering A, B, C, and D is 1, 1, 1, and 0, respectively. RP reaches 1 when a chosen clustering (here, clustering A, B, or C) does not suffer from false negatives and decreases toward 0 as the proportion of false negatives increases.

(3) **Recall_Negative (RN)** is the proportion of node pairs that are correctly clustered to "different" clusters in the chosen clustering among node pairs that are clustered to "different" clusters in the true clustering:

$$\mathrm{RN} = \frac{\mathrm{TN}}{\mathrm{TN+FP}} \tag{B3}$$

RN for clustering A, B, C, and D is 0, 0.5, 1, and 1, respectively. RN reaches 1 when a chosen clustering (here, clustering C or D) does not suffer from false positives and decreases toward 0 as the proportion of false positives increases.

(4) **Normalized Mutual Information (NMI)** is an information-theoretic measure. The numerator, $\mathrm{I(Chosen, True)}$, indicates mutual information and so measures the amount of information shared by the chosen and true clusterings. We normalize the numerator with the average of marginal entropies:

$$\mathrm{NMI} = \frac{\mathrm{I(Chosen,True)}}{\frac{1}{2}\cdot\big(\mathrm{H(Chosen)} + \mathrm{H(True)}\big)} \tag{B4}$$

In Figure B1, NMI for clustering A, B, C, and D is 0, 0.7, 1.0, and 0.8, respectively. Like RI, NMI measures overall clustering accuracy and reaches 1 when a chosen clustering (here, clustering C) is equivalent to the true one and decreases as a chosen clustering becomes dissimilar to the true one.

It is important to note that the four metrics are monotonically related to the number of clusters (e.g., Vinh, Epps, and Bailey 2009). RI, RN, and NMI tend to increase with the increasing number of clusters, whereas RP tends to increase with the decreasing number of clusters. In other words, RI, RN, and NMI favor granular clustering, whereas RP favors coarse clustering. Hence, only using a subset of these metrics can lead to incorrect clustering evaluation. For example, suppose that we use RI, RN, and NMI to measure the clustering accuracy of our method and that these metrics are high under our method. This result is not sufficient to conclude that our method performs well. We need more evidence to reject the possibility that RI, RN, and NMI are high just because the chosen clustering is granular. Specifically, we can reject the above possibility if RP, which favors coarse clustering, is also high under our method. Therefore, in the simulation section, we use all four metrics to evaluate clustering performance accurately.

Reference

Vinh NX, Epps J, Bailey J (2009) Information theoretic measures for clusterings comparison: Is a correction for chance necessary? *Proceedings of the 26th Annual International Conference on Machine Learning*. 1073-1080.