Clustering time series based on probability distributions across temporal granularities

Sayani Gupta *
Department of Econometrics and Business Statistics, Monash University
and
Dianne Cook
Department of Econometrics and Business Statistics, Monash University
and
Rob J Hyndman
Department of Econometrics and Business Statistics, Monash University

November 18, 2021

Abstract

Clustering is a potential approach for organizing large collections of time series into small homogeneous groups, but a difficult step is determining an appropriate metric to measure similarity between time series. The similarity metric needs to be capable of accommodating long, noisy, and asynchronous time series and also capture cyclical patterns. In this paper, two approaches for measuring distances between time series are presented, based on probability distributions over cyclic temporal granularities. Both are compatible with a variety of clustering algorithms. Cyclic granularities like hour-of-the-day, work-day/weekend, and month-of-the-year, are useful for finding repeated patterns in the data. Measuring similarity based on probability distributions across cyclic granularities serves two purposes: (a) characterizing the inherent temporal data structure of long, unequal-length time series in a manner robust to missing or noisy data; (b) small pockets of similar repeated behaviors can be captured. This approach is capable of producing useful clusters, as demonstrated on validation data designs and a sample of residential smart meter records.

Keywords: clustering, time granularities, probability distributions, Jensen-Shannon distances, periodic data, smart meter, electricity consumption behavior, R

*Email: Sayani.Gupta@monash.edu

1 Introduction

Time series clustering is the process of unsupervised partitioning of n time series data into k ($k < n$) meaningful groups such that homogeneous time series are grouped together based on a certain similarity measure. The time series features, length of time series, representation technique, and, of course, the purpose of clustering time series all influence the suitable similarity measure or distance metric to a meaningful level. The three primary methods to time series clustering (Liao 2005) are algorithms that operate directly with distances or raw data points in the time or frequency domain (distance-based), with features derived from raw data (feature-based), or indirectly with models constructed from raw data (model-based). The efficacy of distance-based techniques is highly dependent on the distance measure utilized. Defining an appropriate distance measure for the raw time series may be a difficult task since it must take into account noise, variable lengths of time series, asynchronous time series, different scales, and missing data. Commonly used distance-based similarity measures as suggested by a decade review of time series clustering approaches (Aghabozorgi et al. 2015) are Euclidean, Pearson's correlation coefficient and related distances, Dynamic Time Warping (DTW), Autocorrelation, Short time series distance, Piecewise regularization, cross-correlation between time series, or a symmetric version of the Kullback–Liebler distances (Liao 2007) but on vector time series data. Among these alternatives, Euclidean distances have high performance but need the same length of data over the same period, resulting in information loss regardless of whether it is on raw data or a smaller collection of features. DTW works well with time series of different lengths (Corradini 2001), but it is incapable of handling missing observations. Surprisingly, probability distributions, which may reflect the inherent temporal structure of a time series, have not been considered in determining time series similarity.

This work is motivated by a need to cluster a large collection of residential smart meter data, so that customers can be grouped into similar energy usage patterns. These can be considered to be univariate time series of continuous values which are available at fine temporal scales. These time series data are long (with more and more data collected at finer resolutions), are asynchronous, with varying time lengths for different houses and sporadic missing values. Using probability distributions is a natural way to analyze these

types of data because they are robust to uneven length, missing data, or noise. This paper proposes two approaches for obtaining pairwise similarities based on Jensen-Shannon distances between probability distributions across a selection of cyclic granularities. Cyclic temporal granularities (Gupta et al. 2021), which are temporal deconstructions of a time period into units such as hour-of-the-day or work-day/weekend, can measure repetitive patterns in large univariate time series data. The resulting clusters are expected to group customers that have similar repetitive behaviors across cyclic granularities. The benefits of this approach are as follows.

- When using probability distributions, data does not have to be the same length or observed during the exact same time period (unless there is a structural pattern).

- Jensen-Shannon distances evaluate the distance between two distributions rather than raw data, which is less sensitive to missing observations and outliers than other conventional distance methods.

- While most clustering algorithms produce clusters similar across just one temporal granularity, this technique takes a broader approach to the problem, attempting to group observations with similar distributions across all interesting cyclic granularities.

- It is reasonable to define a time series based on its degree of trend and seasonality, and to take these characteristics into account while clustering it. The modification of the data structure by taking into account probability distributions across cyclic granularities assures that there is no trend and that seasonal variations are handled independently. As a result, there is no need to de-trend or de-seasonalize the data before applying the clustering method. For similar reasons, there is no need to exclude holiday or weekend routines.

The primary application of this work is data from the Smart Grid, Smart City (SGSC) project (2010–2014) available through the Department of the Environment and Energy. Half-hourly measurements of usage for more than 13,000 electricity smart meter customers is provided from October 2011 to March 2014. Customers vary in size, location, and amenities such as solar panels, central heating, and air conditioning. The behavioral patterns

differ amongst customers due to many temporal dependencies. Some customers use a dryer, while others dry their clothes on a line. Their weekly usage profile may reflect this. They may vary monthly, with some customers using more air conditioners or heaters than others, while having equivalent electrical equipment and weather circumstances. Some customers are night owls, while others are morning larks. Daily energy usage varies depending on whether customers stay home or work away from home. Age, lifestyle, family composition, building attributes, weather, availability of diverse electrical equipment, among other factors, make the task of properly segmenting customers into comparable energy behavior complex. The challenge is to be able to cluster consumers into these types of expected patterns, and other unexpected patterns, using only their energy usage history (Ushakova & Jankin Mikhaylov 2020). There is a growing need to have methods that can examine the energy usage heterogeneity observed in smart meter data and what are some of the most common power consumption patterns.

There is an extensive body of literature focused on time series clustering related to smart meter data. Tureczek & Nielsen (2017) conducted a systematic study of over 2100 peer-reviewed papers on smart meter data analytics. The most often used algorithm is k-means (Rhodes et al. 2014). k-means can be made to perform better by explicitly incorporating time series features such as correlation or cyclic patterns rather than performing it on raw data. To reduce dimensionality, several studies use principal component analysis (PCA) or factor analysis to pre-process smart-meter data before clustering (Ndiaye & Gabriel 2011). PCA eliminates correlation patterns and decreases feature space, but loses interpretability. Other algorithms utilized in the literature include k-means variants, hierarchical clustering, and greedy k-medoids. Time series data, such as smart meter data, are not well-suited to any of the techniques mentioned in Tureczek & Nielsen (2017). Only one study (Ozawa et al. 2016) identified time series characteristics by first conducting a Fourier transformation, to convert data from time to frequency domain, followed by k-means to cluster by greatest frequency. Motlagh et al. (2019) suggests that the time feature extraction is limited by the type of noisy, patchy, and unequal time series common in residential customers and addresses model-based clustering by transforming the series into other objects such as structure or set of parameters which can be more easily characterized and clustered. Chicco

& Akilimali (2010) addresses information theory-based clustering such as Shannon or Renyi entropy and its variations. Melnykov (2013) discusses how outliers, noisy observations and scattered observations can complicate estimating mixture model parameters and hence the partitions. None of these methods focuses on exploring heterogeneity in repetitive patterns based on the dynamics of multiple temporal dependencies using probability distributions, which forms the basis of the methodology reported here.

This paper is organized as follows. Section 2 provides the clustering methodology. Section 3 shows data designs to validate our methods. Section 4 discusses the application of the method to a subset of the real data. Finally, we summarize our results and discuss possible future directions in Section 5.

2 Clustering methodology

The existing work on clustering probability distributions assumes we have independent and identically distributed samples $f_1(v), \ldots, f_n(v)$, where $f_i(v)$ denotes the distribution from observation i over some random variable $v = \{v_t : t = 0, 1, 2, \ldots, T-1\}$ observed across T time points. In our approach, instead of considering the probability distributions of the linear time series, we compare them across different categories of a cyclic granularity. We can consider categories of an individual cyclic granularity (A) or combination of categories for two interacting granularities (A, B) to have a distribution, where A and B are defined as $A = \{a_j : j = 1, 2, \ldots J\}$ and $B = \{b_k : k = 1, 2, \ldots K\}$. For example, let us consider two cyclic granularities, A and B, representing hour-of-day and day-of-week, respectively. Then $A = \{0, 1, 2, \ldots, 23\}$ and $B = \{Mon, Tue, Wed, \ldots, Sun\}$. In case individual granularities are considered, there are $J = 24$ distributions of the form $f_{i,j}(v)$ or $K = 7$ distributions of the form $f_{i,k}(v)$ for each customer i. In case of interaction, $J * K = 168$ distributions of the form $f_{i,j,k}(v)$ could be conceived for each customer i. Hence clustering these customers is equivalent to clustering these collections of conditional univariate probability distributions. Towards this goal, the next step is to decide how to measure distances between collections of univariate probability distributions. Here, we describe two approaches for finding distances between time series. Both of these approaches may be useful in a practical context, and produce very different but equally useful customer groupings. The distances can be supplied

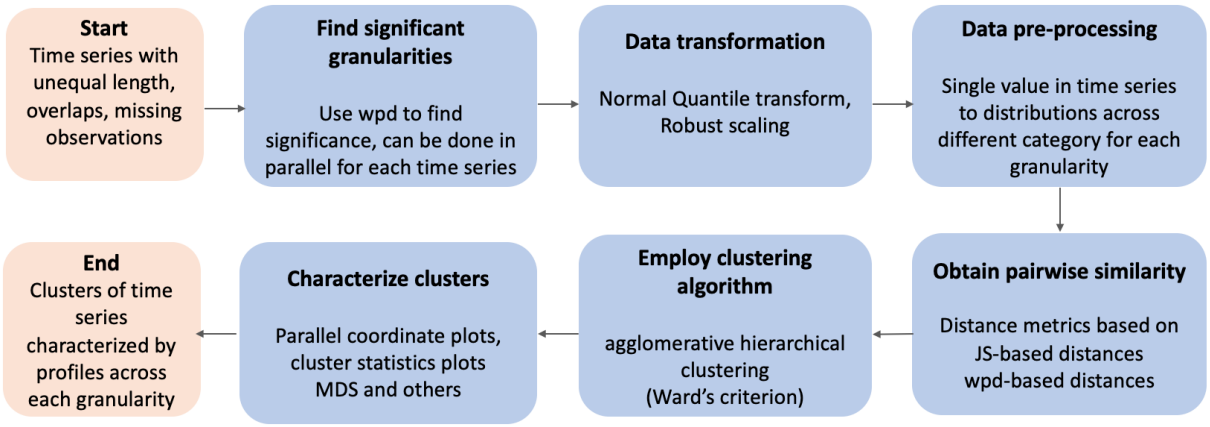

Figure 1: Flow chart illustrating the pipeline for out method for clustering time series.

to any usual clustering algorithm, including k-means or hierarchical clustering, to group observations into a smaller more homogeneous collection. The flow of the procedures is illustrated in Figure 1.

2.1 Selecting granularities

Gupta et al. (2021) provide a distance measure (*wpd*) for determining the significance of a cyclic granularity, and a ranking of multiple cyclic granularities. (This extends to harmonies, pairs of granularities that might interact with each other.) We define "significant" granularities as those with significant distributional differences across at least one category. The reason for subsetting granularities in this way, is that, clustering algorithms perform badly in the presence of nuisance variables. Granularities that do not have some difference between categories are likely to be nuisance variables. It should be noted that all of the time series in a collection may not have the same set of significant granularities. This is the approach for generating a subset (S_c) of significant granularities across a collection of time series:

(a) Remove granularities from the comprehensive list that are not significant for any time series.

(b) Select only the granularities that are significant for the majority of time series.

2.2 Data transformation

The shape and scale of the distribution of the measured variable (e.g. energy usage) affects distance calculations. Skewed distributions need to be symmetrized. Scales of individuals need to be standardized, because clustering is to select similar patterns, not magnitude of usage. (Organizing individuals based on magnitude can be achieved simply by sorting on a statistic like the average value across time.) For the JS-based approaches, two data transformation techniques are recommended, normal-quantile transform (NQT) and robust scaling (RS). While Gupta et al. (2021) already use NQT when computing *wpd*, it could be useful to standardize it for the selected set of significant granularities prior to computing the distances.

- RS: The normalized i^{th} observation is denoted by $v_{norm} = \frac{v_t - q_{0.5}}{q_{0.75} - q_{0.25}}$, where v_t is the actual value at the t^{th} time point and $q_{0.25}$, $q_{0.5}$ and $q_{0.75}$ are the 25^{th}, 50^{th} and 75^{th} percentiles of the time series for the i^{th} observation. Note that v_{norm} has zero mean and median, but otherwise the shape does not change.

- NQT: The raw data for all observations is individually transformed (Krzysztofowicz 1997), so that the transformed data follows a standard normal distribution. NQT will symmetrize skewed distributions. A drawback is that any multimodality will be concealed. This should be checked prior to applying NQT.

2.3 Data pre-preprocessing

Computationally in R, the data is assumed to be a "tsibble object" (Wang et al. 2020) equipped with an index variable representing inherent ordering from past to present and a key variable that defines observational units over time. The measured variable for an observation is a time-indexed sequence of values. This sequence, however, could be shown in several ways. A shuffle of the raw sequence may represent hourly consumption throughout a day, a week, or a year. Cyclic granularities can be expressed in terms of the index set in the tsibble data structure.

The data object will change when cyclic granularities are computed, as multiple observations will be categorized into levels of the granularity, thus inducing multiple probability

distributions. Directly computing Jensen-Shannon distances between the entire probability distributions can be computationally intensive. Thus it is recommended that quantiles are used to characterize the probability distributions. In the final data object, each category of a cyclic granularity corresponds to a list of numbers which is composed of a few quantiles.

2.4 Distance metrics

The total (dis) similarity between each pair of customers is obtained by combining the distances between the collections of conditional distributions. This needs to be done in a way such that the resulting metric is a distance metric, and could be fed into the clustering algorithm. Two types of distance metrics are considered:

2.4.1 JS-based distances

This distance metric considers two time series to be similar if the distributions of each category of an individual cyclic granularity or combination of categories for interacting cyclic granularities are similar. In this study, the distribution for each category is characterized using deciles (can potentially consider any list of quantiles), and the distances between distributions are calculated using the Jensen-Shannon distances (Menéndez et al. 1997), which are symmetric and thus could be used as a distance measure.

The sum of the distances between two observations x and y in terms of a cyclic granularity A is defined as

$$S_{x,y}^{A} = \sum_{j \in A} D(x_j, y_j)$$

where D is the Jensen-Shannon distances, x_j is the set of quantiles over the values filtered by j^{th} level of granularity A for observation x (similar for y).

The sum of the distances between two observations x and y in terms of a pair of cyclic granularities (A, B) is defined as

$$S_{x,y}^{A,B} = \sum_{(j,k) \in (A,B)} D(x_{jk}, y_{jk})$$

x_{jk} is the set of quantiles over the values filtered by the combination of j^{th} level of granularity A and k^{th} level of granularity B for observation x (similar for y).

After determining the distance between two series in terms of one granularity, we must combine them to produce a distance based on all significant granularities. When combining distances from individual L cyclic granularities C_l with n_l levels,

$$S_{x,y} = \sum_{l \in L} S_{x,y}^{C_l} / n_l$$

is employed, which is also a distance metric since it is the sum of JS distances. This approach is expected to yield groups, such that the variation in observations within each group is in magnitude rather than distributional pattern, while the variation between groups is only in distributional pattern across categories.

2.4.2 wpd-based distances

We compute weighted pairwise distances *wpd* (Gupta et al. 2021) for all considered granularities for all observations. *wpd* is designed to capture the maximum variation in the measured variable explained by an individual cyclic granularity or their interaction. It is estimated by the maximum pairwise distances between distributions across consecutive categories normalized by appropriate parameters. A higher value of *wpd* indicates that some interesting patterns are expected, whereas a lower value would indicate otherwise.

Once we have chosen *wpd* as a relevant feature for characterizing the distributions across one cyclic granularity, we have to decide how we combine differences between the multiple features (corresponding to multiple granularities) into a single number. The Euclidean distance between them is chosen, with the granularities acting as variables and *wpd* representing the value under each variable. With this approach, we should expect the observations with similar *wpd* values to be clustered together. Thus, this approach is useful for grouping observations that have a similar significance of patterns across different granularities. Similar significance does not imply a similar pattern, which is where this technique varies from JS-based distances, which detect differences in patterns across categories.

2.5 Clustering

2.5.1 Number of clusters

Determining the number of clusters is typically a difficult task. Many metrics have been defined for choosing clusters. Most metrics for choosing the optimal number of clusters are based on comparing distances between observations within a class to those distances between observations between classes, which makes the assumption that there are some separated clusters. Some common procedures include the gap statistic (Tibshirani et al. 2001), average silhouette width (Rousseeuw 1987), Dunn index (Dunn 1973) and the separation index (*sindex*) (Hennig 2019, 2014). These are constructed by balancing within-cluster homogeneity and between-cluster separation.

All of the common approaches can give contradictory suggestions for the optimal number of clusters, particularly when the data does not naturally break into groups, or in the presence of nuisance variables (no contribution to clustering) or nuisance observations (inlying and outlying observations falling between clusters). There is no one best metric, which is perhaps a reason why so many metrics exist.

In this work, we have chosen to use *sindex*. It is a very simple but effective metric. This is computed by averaging the smallest 10% of inter-cluster distances. It is relatively robust to nuisance observations. The value of *sindex* always decreases, and sharp drops in value indicate candidates for the optimal number of clusters. The number of clusters corresponding to the value **before the drop** is the recommendation.

.

2.5.2 Algorithm

With a way to obtain pairwise distances, any clustering algorithm can be employed that supports the given distance metric as input. A good comprehensive list of algorithms can be found in Xu & Tian (2015) based on traditional ways like partition, hierarchy, or more recent approaches like distribution, density, and others. We employ agglomerative hierarchical clustering in conjunction with Ward's linkage. Hierarchical cluster techniques fuse neighboring points sequentially to form bigger clusters, beginning with a full pairwise distance matrix. The distance between clusters is described using a "linkage technique".

This agglomerative approach successively merges the pair of clusters with the shortest between-cluster distance using Ward's linkage method.

2.5.3 Characterization of clusters

Cluster characterization is an important final stage of a cluster analysis. The primary purpose is to compare the homogeneity within a cluster to the heterogeneity of clusters. This can be done numerically, by tabulating cluster means and standard deviations (Dasu et al. 2005), and visually using methods for graphics multivariate data. Cook & Swayne (2007) provide visual examples using both tours (Asimov 1985) and parallel coordinate plots (Wegman 1990). Dimension reduction techniques like principal component analysis (Jolliffe & Cadima 2016), multidimensional scaling (MDS) (Borg & Groenen 2005), t-distributed stochastic neighbor embedding (t-SNE) (Van der Maaten & Hinton 2008) and linear discrimant analysis (LDA) (Fisher 1936) are also useful.

3 Validation

To validate our clustering methods, we have created several different data designs containing different granularity features. There are three circular granularities $g1$, $g2$ and $g3$ with categories denoted by $\{g10, g11\}$, $\{g20, g21, g22\}$ and $\{g30, g31, g32, g33, g34\}$ and levels $n_{g_1} = 2$, $n_{g_2} = 3$ and $n_{g_3} = 5$. These categories could be integers or some more meaningful labels. For example, the granularity "day-of-week" could be either represented by $\{0, 1, 2, \ldots, 6\}$ or $\{Mon, Tue, \ldots, Sun\}$. Here categories of $g1$, $g2$ and $g3$ are represented by $\{0, 1\}$, $\{0, 1, 2\}$ and $\{0, 1, 2, 3, 4\}$ respectively. A continuous measured variable v of length T indexed by $\{0, 1, \ldots T-1\}$ is simulated such that it follows the structure across $g1$, $g2$ and $g3$. We constructed independent replications of all data designs $R = \{25, 250, 500\}$ to investigate if our proposed clustering method can discover distinct designs in small, medium, and big numbers of series. All designs employ $T = \{300, 1000, 5000\}$ sample sizes to evaluate small, medium, and large-sized series. Variations in method performance may be due to different jumps between categories. So a mean difference of $\mu = \{1, 2, 5\}$ between categories is considered. The performance of the approaches varies with the number of

Table 1: The range of parameters used for the validation study, for the three different scenarios, number of simulations (R) for each design, differences between means (μ) across granularities and series lengths (T).

scenario	designs	R	μ	T
S1	5	25, 20, 500	1, 2, 5	300, 1000, 5000
S2	4			
S3	4			

granularities which has interesting patterns across its categories. So three scenarios are considered to accommodate that. Table 1 shows the range of parameters considered for each scenario.

3.1 Data generation

Each category or combination of categories from $g1$, $g2$ and $g3$ are assumed to come from the same distribution, a subset of them from the same distribution, a subset of them from separate distributions, or all from different distributions, resulting in various data designs. As the methods ignore the linear progression of time, there is little value in adding time dependency to the data generating process. The data type is set to be "continuous," and the setup is assumed to be Gaussian. When the distribution of a granularity is "fixed", it means distributions across categories do not vary and are considered to be from N (0,1). μ alters in the "varying" designs, leading to varying distributions across categories.

3.2 Data designs

3.2.1 Individual granularities

Scenario 1 (S1) - All granularities significant: Consider the instance where $g1$, $g2$, and $g3$ all contribute to design distinction. This means that each granularity will have significantly different patterns at least across one of the designs to be clustered. In Table 2 various distributions across categories are considered (top) which lead to different designs

(bottom). Figure 2 shows the simulated variable's linear (left) and cyclic (right) representations for each of these five designs. The structural difference in the time series variable is impossible to discern from the linear view, with all of them looking very similar. The shift in structure may be seen clearly in the distribution of cyclic granularities. The following scenarios use solely graphical displays across cyclic granularities to highlight distributional differences in categories.

Scenario 2 (S2) - Few significant granularities: This is the case where one granularity will remain the same across all designs. We consider the case where the distribution of v varies across $g2$ levels for all designs, across $g3$ levels for a few designs, and $g1$ does not vary across designs. The proposed design is shown in Figure 3(b).

Scenario 3 (S3) - Only one significant granularity: Only one granularity is responsible for identifying the designs in this case. This is depicted in Figure 3 (right) where only $g3$ affects the designs significantly.

Table 2: For S1, distributions of different categories when they vary (displayed on top). If distributions are fixed, they are set to N(0, 1). The various distributions across categories result in five designs (displayed below).

granularity	Varying distributions
g1	g10 ~ N(0, 1), g11 ~ N(2, 1)
g2	g21 ~ N(2, 1), g22 ~ N(1, 1), g23 ~ N(0, 1)
g3	g31 ~ N(0, 1), g32 ~ N(1, 1), g33 ~ N(2, 1), g34 ~ N(1, 1), g35 ~ N(0, 1)

design	g1	g2	g3
design-1	fixed	fixed	fixed
design-2	vary	fixed	fixed
design-3	fixed	vary	fixed
design-4	fixed	fixed	vary
design-5	vary	vary	vary

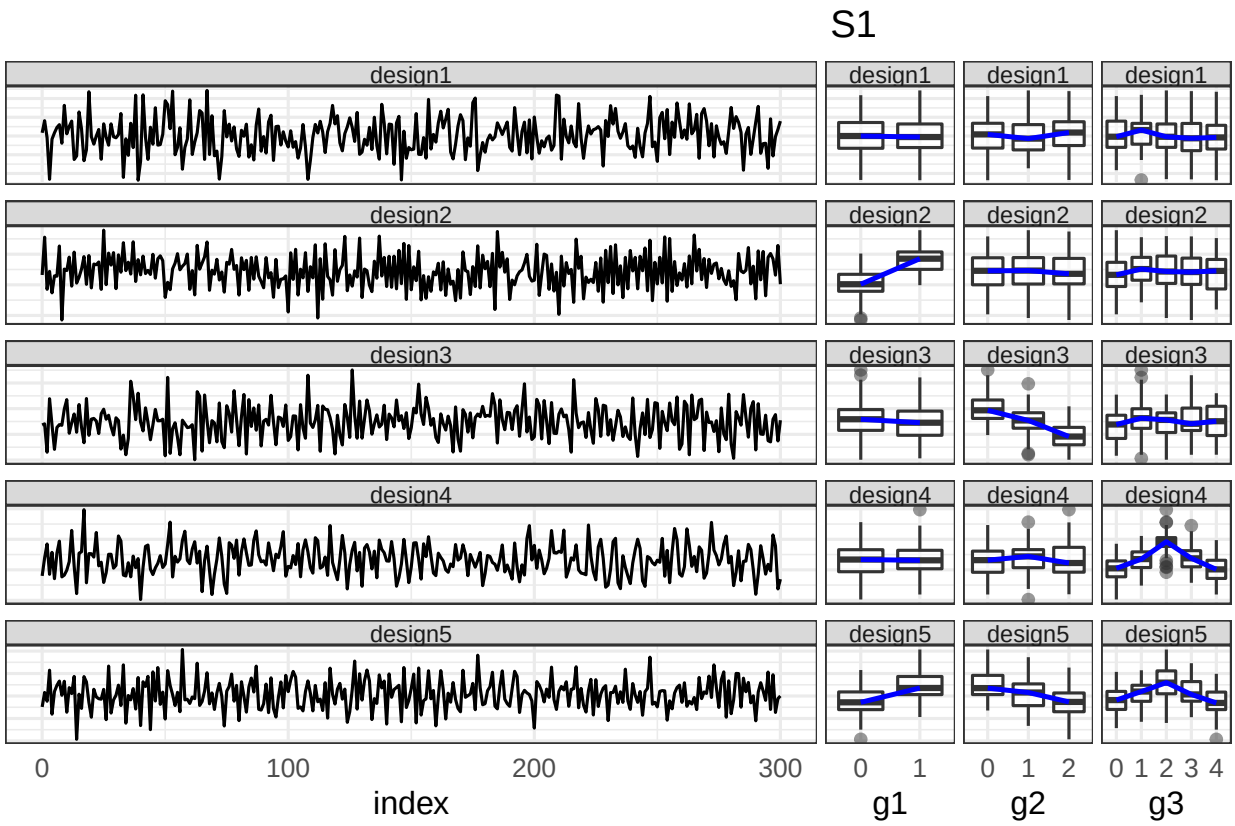

Figure 2: The linear (left) and cyclic (right) representation is shown under scenario S1 using line plots and boxplots respectively. Each row represents a design. Distributions of categories across $g1$, $g2$ and $g3$ change across at least one design as can be observed in the cyclic representation. It is not possible to comprehend these structural differences in patterns just by looking at or considering the linear representation.

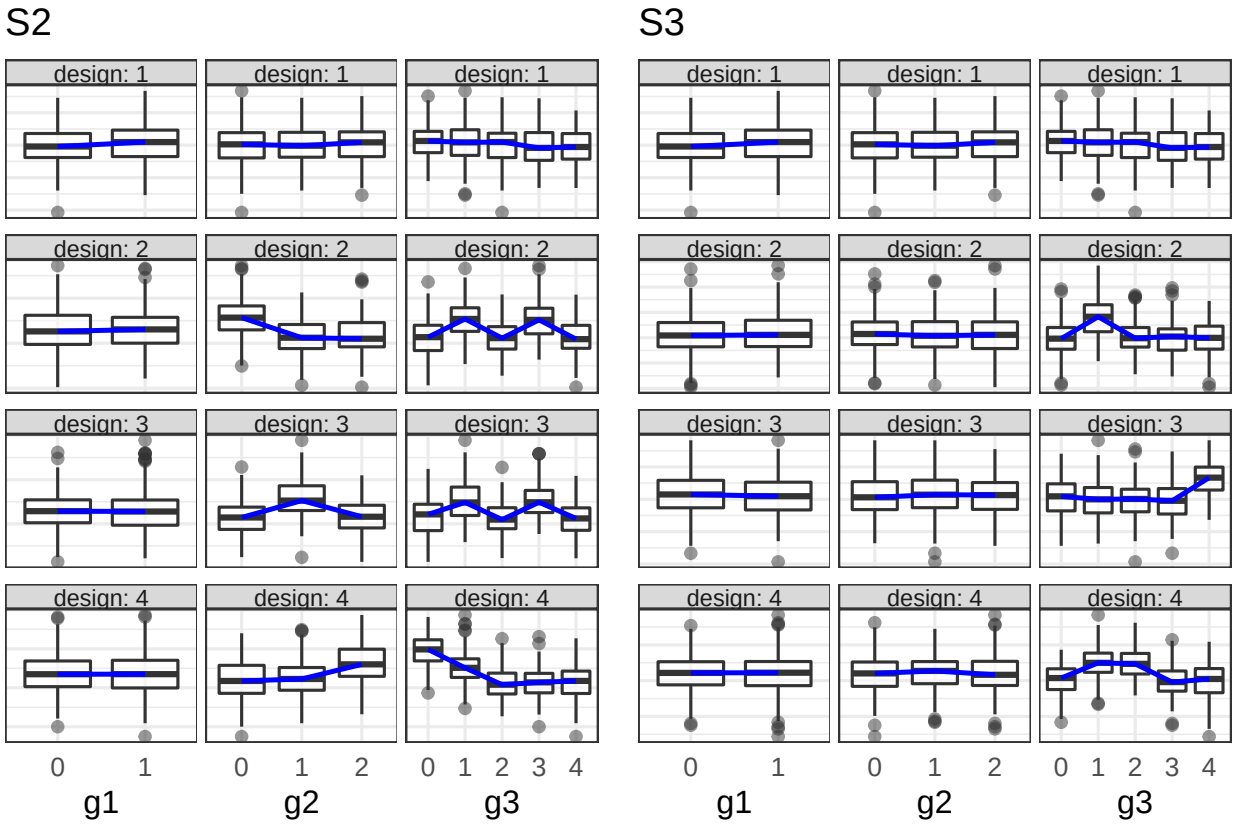

Figure 3: Boxplots showing distributions of categories across different designs (rows) and granularities (columns) for scenarios S2 and S3. In S2, $g2$, $g3$ change across at least one design but $g1$ remains constant. Only $g3$ changes across different designs in S3.

3.2.2 Interaction of granularities

The proposed methods could be extended when two granularities of interest interact and we want to group subjects based on the interaction of the two granularities. Consider a group that has a different weekday and weekend behavior in the summer but not in the winter. This type of combined behavior across granularities can be discovered by evaluating the distribution across combinations of categories for different interacting granularities (weekend/weekday and month-of-year in this example). As a result, in this scenario, we analyze a combination of categories generated from different distributions. Display of design and related results can be found in supplementary material.

3.3 Visual exploration of results

All of the approaches were fitted to each data design and to each combination of the considered parameters. The formed clusters have to match the design, be well separated, and have minimal intra-cluster variation. MDS and parallel coordinate graphs are used to demonstrate the findings, as well as an index value plot to provide direction on the number of clusters. JS-based approaches corresponding to NQT and RS are referred to as JS-NQT and JS-RS respectively. In the following plots, results for JS-NQT are reported, and results with JS-RS or wpd-based distances, which are similar are in the supplementary material.

Figure 4 shows *sindex* plotted against the number of clusters (k) for the range of mean differences (rows) under the different scenarios (columns). This can be used to determine the number of clusters for each scenario. When *sindex* for each scenario are examined, it appears that $k = \{5, 4, 4\}$ is justified for scenarios S1, S2, and S3, respectively, given the sharp decrease in *sindex* from that value of k. Thus, the number of clusters corresponds to the number of designs that were originally considered in each scenario.

Figure 5 shows separation of our clusters. It can be observed that in all scenarios and for different mean differences, clusters are separated. However, the separation increases with an increase in mean differences across scenarios. This is intuitive because, as the difference between categories increases, it gets easier for the methods to correctly distinguish the designs.

Figure 6 depicts a parallel coordinate plot with the vertical bar showing total inter-

cluster distances with regard to granularities $g1$, $g2$, and $g3$ for all simulation settings and scenarios. So one line in the figure shows the inter-cluster distances for one simulation setting and scenarios vary across facets. The lines are not colored by group since the purpose is to highlight the contribution of the factors to categorization rather than class separation. Panel S1 shows that no variable stands out in the clustering, but the following two panels show that {g1} and {g1, g2} have very low inter-cluster distances, meaning that they did not contribute to the clustering. It is worth noting that these facts correspond to our original assumptions when developing the scenarios, which incorporate distributional differences over three (S1), two (S2), and one (S3) significant granularities. Hence, Figure 6 (S1), (S2), and (S3) validate the construction of scenarios (S1), (S2), and (S3) respectively.

The JS-RS and wpd-based methods perform worse for $nT = 300$, then improve for higher nT evaluated in the study. However, a complete year of data is the minimum requirement to capture distributional differences in winter and summer profiles, for example. Even if the data is only available for a month, nT with half-hourly data is expected to be at least 1000. As a result, as long as the performance is promising for higher nT, this is not a challenge.

In our study sample, the method JS-NQT outperforms the method JS-RS for smaller differences between categories. More testing, however, would be needed to to be confident in this conclusion.

4 Application

Clustering with the new distances is illustrated on the smart meter energy usage for a sample of customers from Department of the Environment and Energy (2018). The full data contains half-hourly general supply in KwH for 13,735 customers, resulting in 344,518,791 observations in total. The raw data for these consumers is of unequal length, with varying starting and end dates. Additionally, there were missing values in many series. (The supplementary material contains details from checking for systematic missingness.) Because our proposed methods evaluate probability distributions rather than raw data, these data issues are not problematic, unless there is any systematic structure related to granularities.

Huge data sets present more complications for clustering. Clustering algorithms work

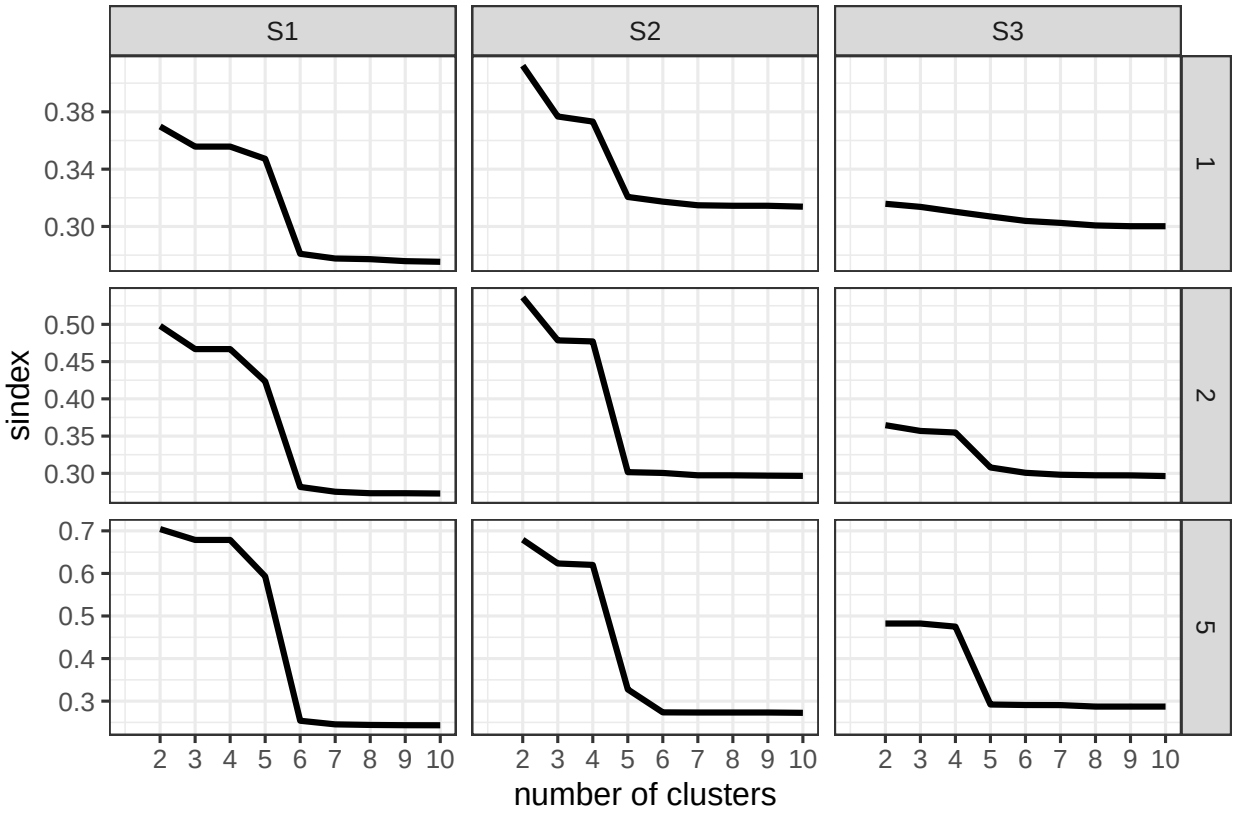

Figure 4: Choosing optimal cluster number across the range of scenarios and mean differences used in the validation study, using the cluster separation index (sindex) for the JS-NQT. S1 has a sharp decrease in sindex from 5 to 6, whereas S2 and S3 have a decrease from 4 to 5, especially when mean difference is large, providing the recommended number of clusters to be 5, 4, 4, respectively. This precisely reflects the structure in designs that we would hope the clustering could recover.

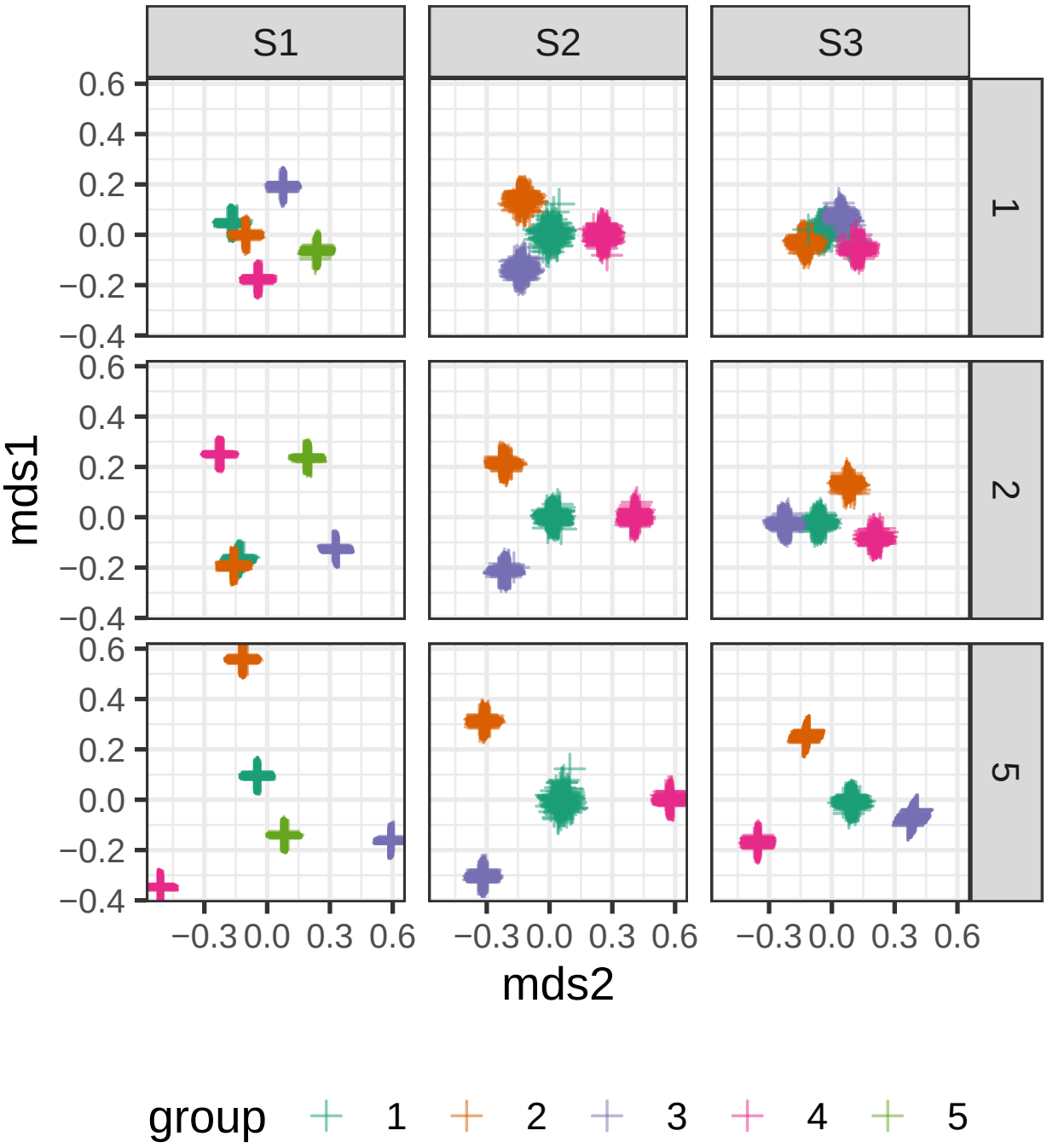

Figure 5: MDS summary plots to illustrate the cluster sparation for the range of mean differences (rows) under the different scenarios (columns). It can be observed that clusters become more compact and separated for higher mean differences between categories across all scenarios. Between scenarios, separation is least prominent corresponding to Scenario (S3) where only one granularity is responsible for distinguishing the clusters.

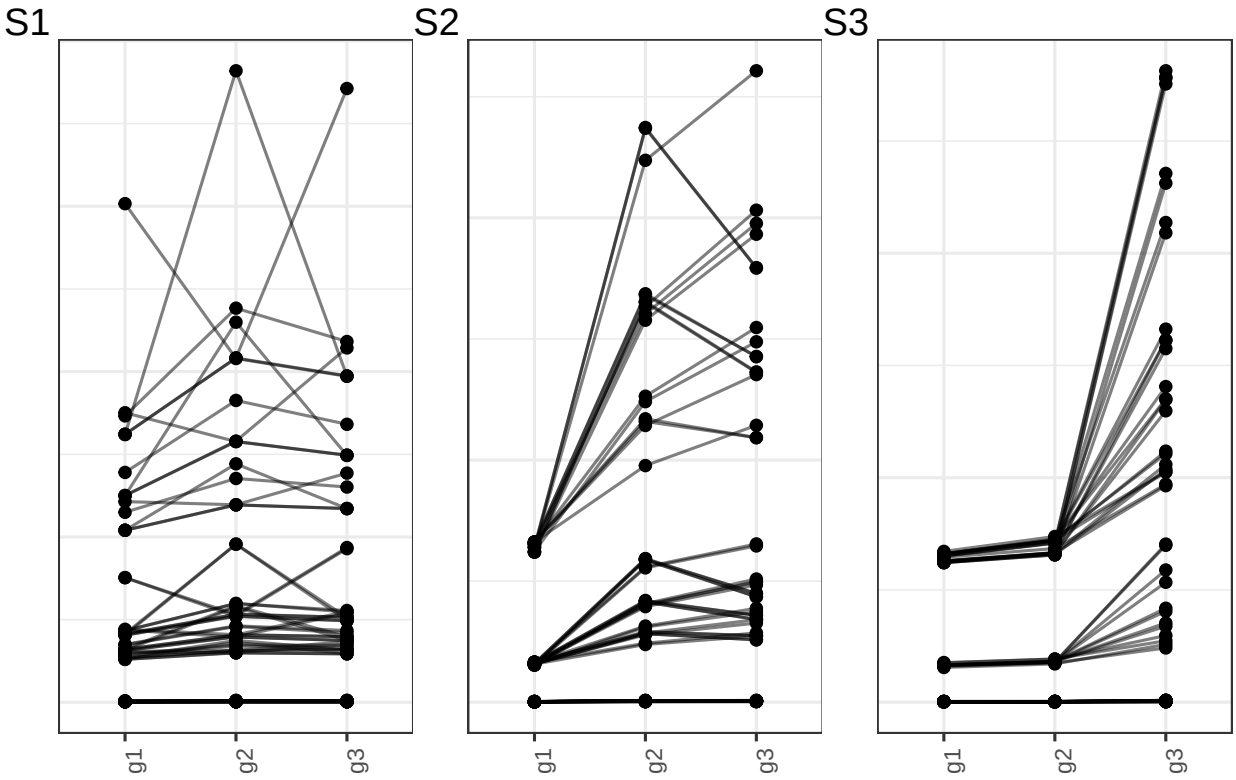

Figure 6: Exploring the contribution of granularities in the clustering for scenarios S1, S2, S3, using parallel coordinate plots. Inter-cluster distances are displayed vertically. All three granularities $g1$, $g2$, and $g3$ have high inter-cluster distances for S1, suggesting all are important. In S2 $g1$ and in S3 both $g1$ and $g2$ have smaller inter-cluster distances, indicating that they did not contribute to clustering.

well when there are well-separated clusters, with no nuisance variables or nuisance observations. When converting a series to granularities, many variables (each level of a granularity) are generated, possibly creating a slew of nuisance variables. Some customers may have a mix of energy use patterns, which could be considered nuisance observations located between major clusters. For this reason, we have chosen to select a small group of customers with relatively distinct and different patterns in order to illustrate the clustering more simply. Figure 7 shows the distribution across *hod*, *moy* and *wnwd* for the set of 24 customers used to illustrate clustering. The customers are displayed in two columns of 12 for space reasons. Each row, of each column, represents the profile of a single customer across different variables. This is often a good approach to tackling a big analysis task, to start with a simpler task. The approach, however, is applicable to all customers.

As a result, we dissect the larger problem and test our solutions on a small sample of prototype customers. To do this, data is first filtered to generate a small sample, and then significant cyclic granularities (variables) for them are chosen (as described in Section 4.1). The sample set is subsequently examined along all dimensions of interest, to ensure that they reveal some patterns across at least one specified variable (as described in Section 4.2). Because the data does not contain additional customer characteristics, we cannot explain why consumption varies, but can only identify how it varies.

4.1 Data filtering and variable selection

The steps for customer filtering and variable selection were:

1. Choose a smaller subset of randomly selected 600 customers with no implicit missing values for 2013.
2. Obtain *wpd* for all cyclic granularities considered for these customers. It was found that *hod* (hour-of-day), *moy* (month-of-year) and *wnwd* (weekend/weekday) are significant for most customers. We use these three granularities while clustering.
3. Remove customers whose data for an entire category of *hod*, *moy* or *wnwd* is empty. For example, a customer who does not have data for an entire month is excluded because their monthly behavior cannot be analyzed.

4. Remove customers whose energy consumption is 0 in all deciles. These are the clients whose consumption is likely to remain essentially flat and with no intriguing repeated patterns that we are interested in studying.

4.2 Selecting prototypes

It is common to filter data prior to fitting a supervised classification model using instance selection (Olvera-López et al. 2010) which removes observations that might impede the model building. For clustering, this is analogous to identifying and removing nuisance observations. Prototype selection is more severe than instance selection, because only a handful of cases is selected. Cutler & Breiman (1994) proposed a method called archetypal analysis which has inspired this approach but the procedure we have used follows Fan et al. (2021). First dimension reduction such as t-SNE, MDS or PCA is used to project the data into a 2D space. Second a few "anchor" customers far apart in 2D space are selected. Additional close neighbors to the anchors are selected. To check the selections relative to the full set of variables, we used a tour linked to a t-SNE layout using the R package `liminal` (Lee 2021). This ensured that the final sample of clustered customers were also far apart in the high-dimensional space. (See the supplementary materials for further details.)

4.3 Clustering results

Clustering of the 24 prototypes was conducted with all three distances, JS-NQT, JS-RS and WPD, and is summarised in Figures 8, 9 and 10. The t-SNE visualization suggests that there are four well-separated clusters. It is possible that because the representation is only 2D, that the fifth group from the original prototype selection is distinctly different in high dimensions. The *sindex* plots for the three methods indicate some disagreement: JS-NQT suggests 3, JS-RS suggests 2 or 5 and WPD suggests 3 or 5. JS-RS would appear to match the original prototypes with the five cluster solution, but it actually differs. Even though the *sindex* for JS-NQT suggests three clusters, the five cluster solution more closely matches the original prototypes. The WPD clustering provides a different grouping of the customers, and even though it disagrees with the original prototypes it is a useful grouping.

Figure 9 displays the summarized distributions across 4 and 5 clusters in (a) and (b)

Table 3: Summary table from WPD clusters showing median *wpd* values (*moy*, *hod*, *wnwd*), cluster size (*nobs*) and the list of the customer-prototype id for each cluster with 3 and 5 number of clusters (k).

k	group	nobs	moy	hod	wnwd	customer-prototype id
3	P-1	2	66.7	-2.7	39.4	18, 16
	P-2	9	129.0	-0.4	12.7	12, 9, 17, 2, 19, 13, 20, 10, 11
	P-3	13	14.9	24.5	4.4	8, 22, 23, 24, 14, 15, 3, 1, 4, 21, 5, 6, 7
5	Q-1	2	66.7	-2.7	39.4	18, 16
	Q-2	9	129.0	-0.4	12.7	12, 9, 17, 2, 19, 13, 20, 10, 11
	Q-3	4	88.2	29.4	2.6	22, 14, 4, 6
	Q-4	4	10.1	32.1	4.2	23, 21, 5, 7
	Q-5	5	14.9	11.9	4.6	8, 24, 15, 3, 1

respectively, and helps to characterize each cluster. In the quantile plots the line represents the median, and the region shows the area between the 25^{th} and 75^{th} percentiles. The only difference between the four and five cluster solution is that A-4 divides further into B-4 and B-5. This additional division makes a clearer clustering, because it resolves the heterogeneity in *moy* creating a group (B-5, customers 1-3) which has a winter peak in usage, and a group (B-4, customers 16-20) which has a start of the year peak in usage. B-2 (customers 4-9) and B-1 (customers 21-24) have distinctive *hod* patterns but are both heterogeneous in *moy* and *wnwd*. B-3 (customers 10-15) has peak usage at the end of the year, but is heterogeneous on *hod* and *wnwd*. This clustering almost agrees with the clusters visible in the t-SNE plot.

Figure 10 shows the *wpd* values *hod*, *moy*, and *wnwd* of the 24 customers, colored by 3 (a) and 5 (b) cluster solution as suggested by Figure 8 through a parallel coordinate plot. The variables (wpd's for different granularities) are standardized prior to clustering using WPD. In the display, the variables are sorted according to their separation across groups. This means that *wnwd* is the most important variable in distinguishing the groups, followed by *hod* and *moy* for both (a) and (b). Group P-1 and P-2 correspond to Q-1 and Q-2 respectively. Cluster P-3 splits into Q-3, Q-4 and Q-5. Customers 16 and 18 are

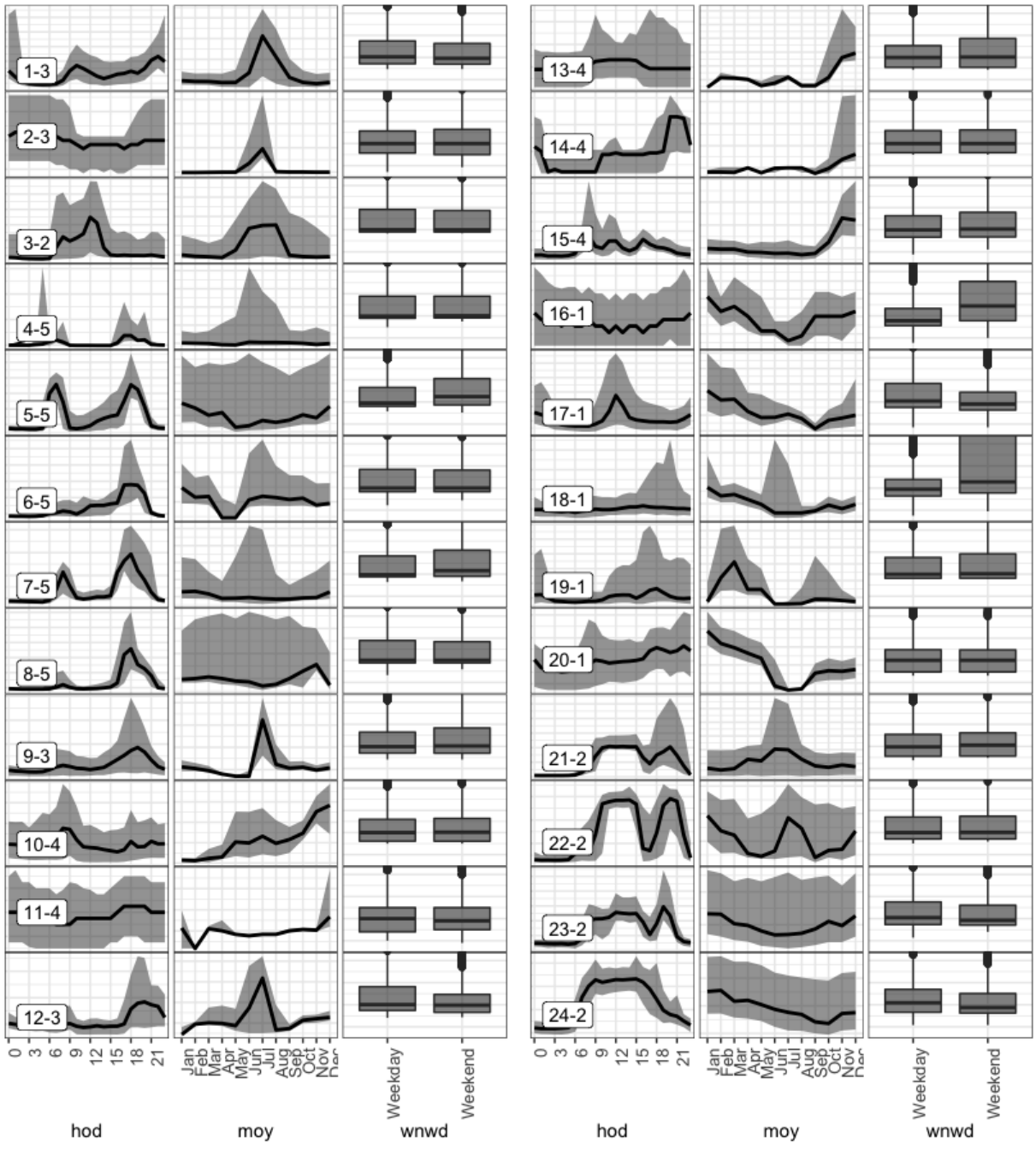

Figure 7: The distribution of electricity demand across individual customers over three granularities *hod*, *moy*, and *wnwd* are shown for the 24 selected customers using quantile and box plots. They are split into batches of 12 in (a) and (b), with each row in (a) or (b) representing a customer. The number indicates a unique customer id and a prototype id.

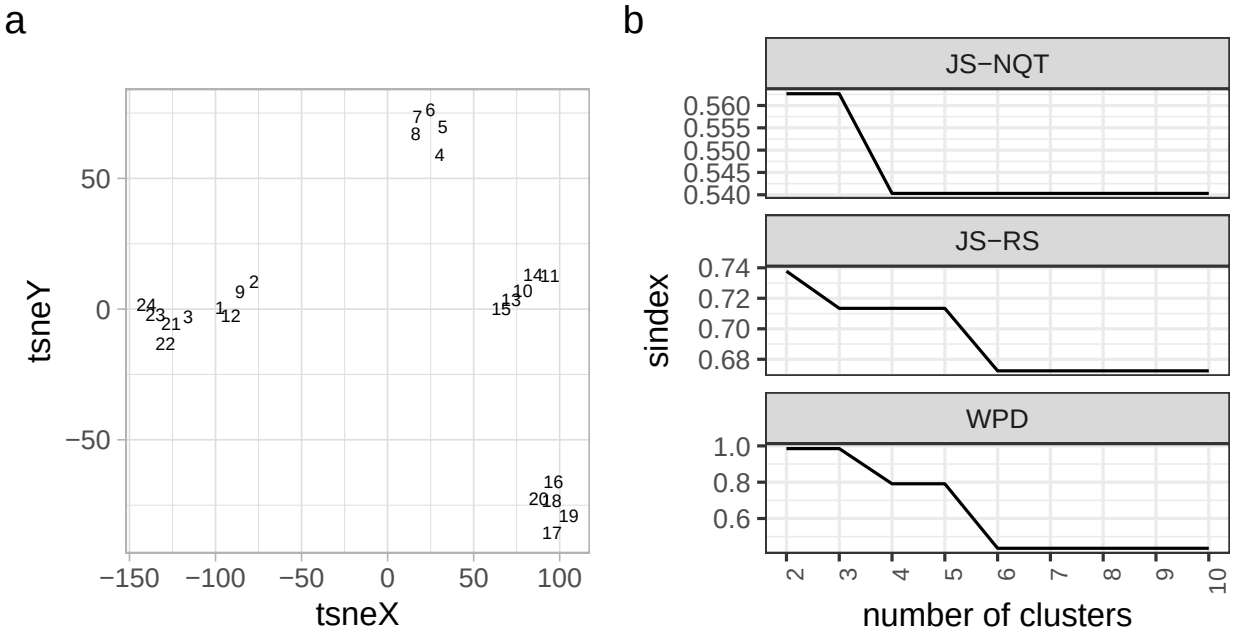

Figure 8: Clustering summaries: (a) t-SNE computed on the 24 selected customers, and (b) separation index (*sindex*) for 2-10 clusters using JS-NQT, JS-RS and WPD. Various choices in number of clusters would be recommended. Four clusters are visible in t-SNE, although it might hide a fifth cluster because dimension reduction to 2D may be insufficient to see the difference. JS-NQT suggests 3, JS-RS suggests 2 or 5 and WPD suggests 3 or 5.

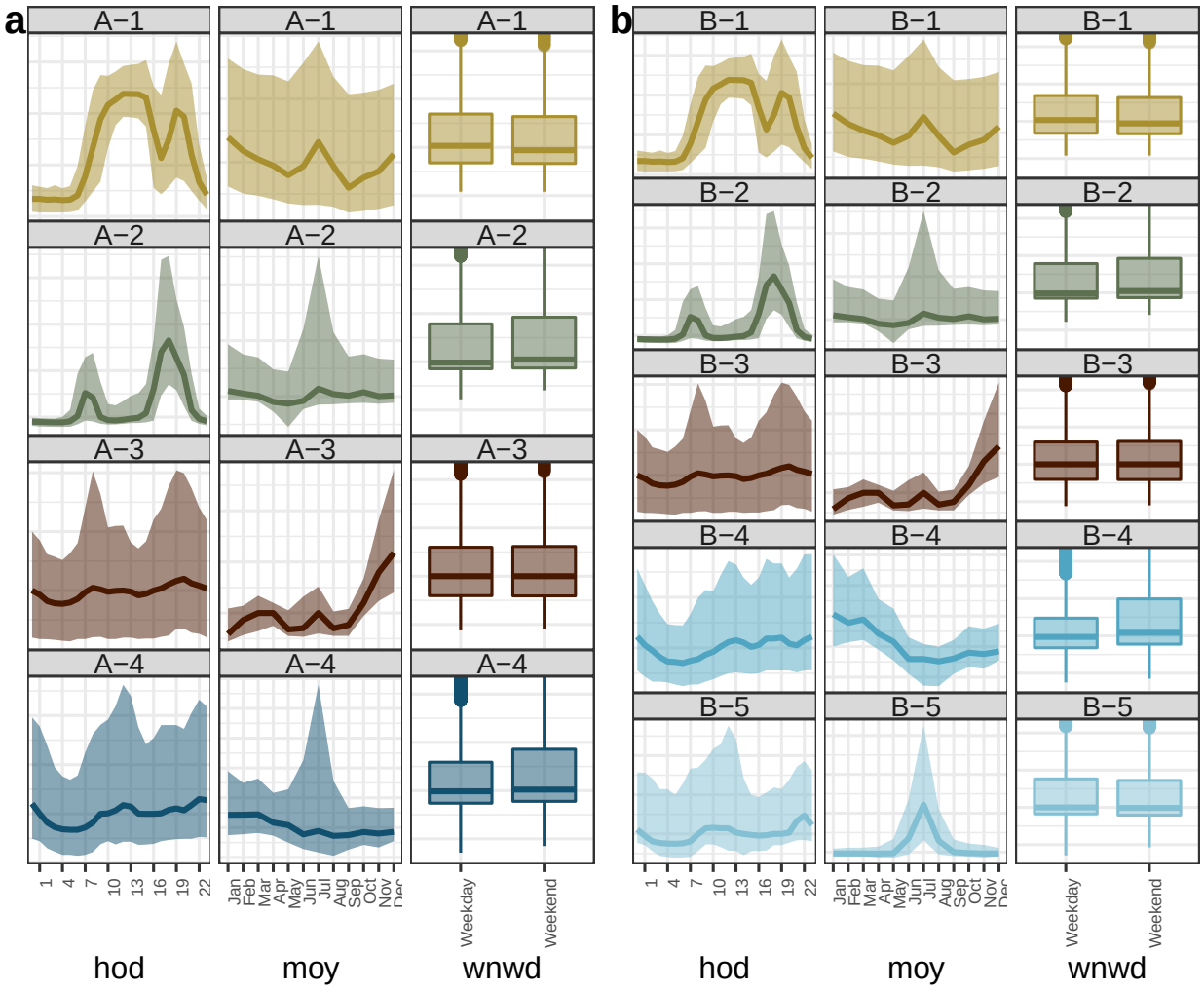

Figure 9: Summary plots for four and five clusters from JS-NQT showing the distribution of electricity demand combined for all members over *hod*, *moy*, and *wnwd*. Groups A-1 (customers 21-24), A-2 (customers 4-9), and A-3 (customers 10-15) profiles correspond to Groups B-1, B-2, and B-3, respectively. Cluster A-4 splits into B-4 (customers 16-20) and B-5 (customers 1-3) to produce the five clusters, which better resolves the *moy* distribution.

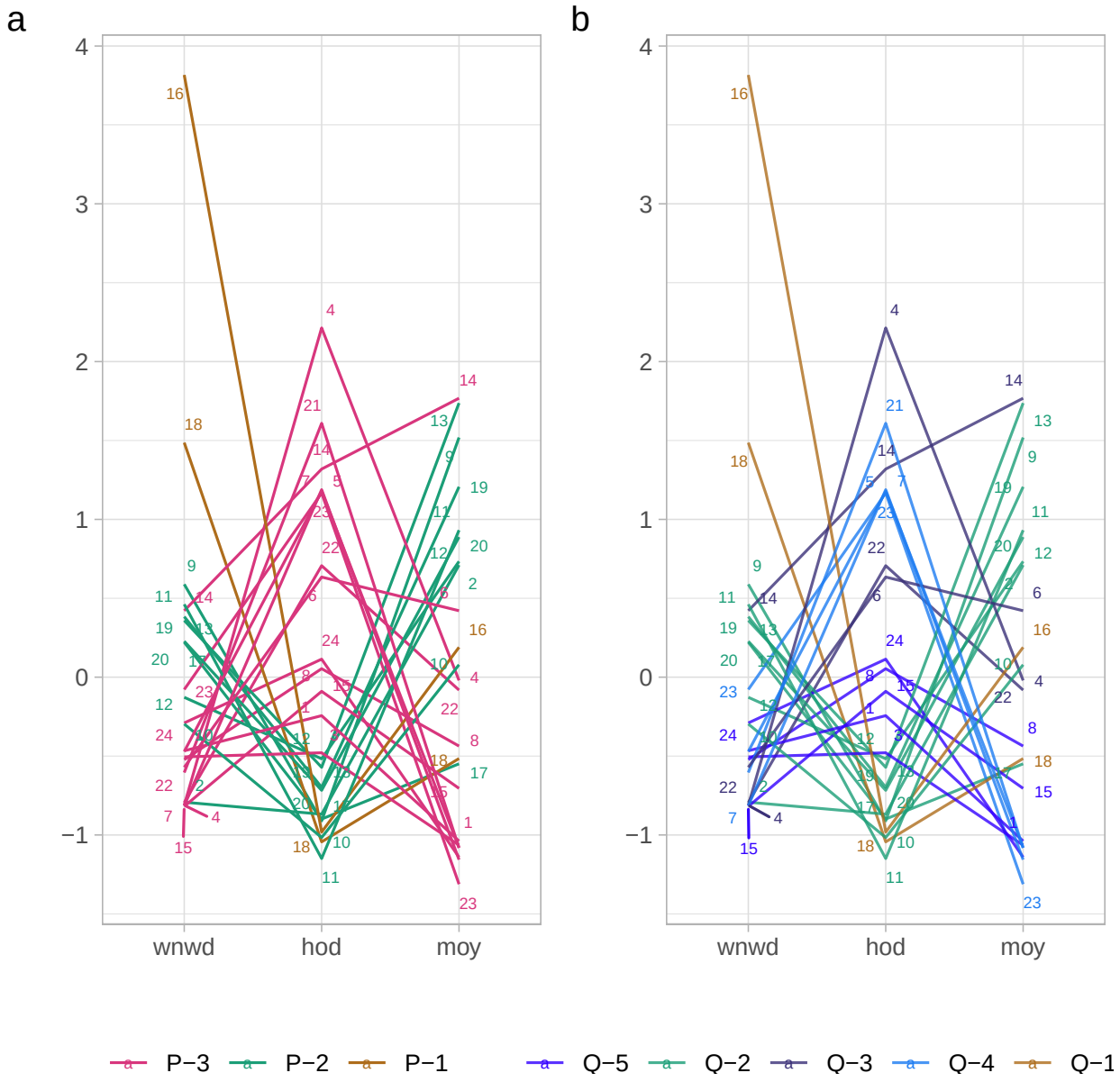

Figure 10: Summary plots for three (a) and five (b) clusters from WPD showing the *wpd* values of each customers across *hod*, *moy*, and *wnwd* through a parallel coordinate plot. Group P-1 and P-2 correspond to Q-1 and Q-2 respectively. Cluster P-3 splits into Q-3, Q-4 and Q-5. P-1 (customers 16, 18) is characterized by high values of *wpd* on *wnwd*. P-2 has lower *wpd* for *hod* than *moy* and *wnwd*. P-3 behaves exactly opposite to P-2 with higher *wpd* for *hod* compared to *moy* and *wnwd*. For 5 cluster solution, this groups gets split into Q3-Q5 characterizing different relative significance over *moy* and *wnwd*.

characterized by unusual high values of *wpd* on *wnwd* compared to the rest of the customers and hence form a group. This could again be verified from figure 7, where these were the only two customers with a difference in their *wnwd* behavior. They are represented by P-1. P-2 has lower *wpd* for *hod* than *moy* and *wnwd*. This can also be verified from Table 3 where *hod* has very low median values of *wpd* over the members in the cluster, implying, at least half of these customers do not have an interesting *hod* pattern. P-3 behaves exactly opposite to P-2 with higher *wpd* for *hod* compared to *moy* and *wnwd*. These are the customers who have some significant pattern across *hod* and this can again be validated by looking at 7. For 5 cluster solution, this groups gets split into Q3-Q5 characterizing different relative significance over *moy* and *wnwd*. For example, Q-4 and Q-5 have almost no pattern across *moy*, but Q-3 has a *moy* pattern and thus it is reasonable to split them. The patterns could be different, but they are significant. Q-4 and Q-5 are separated because of their significance across *hod*.

In summary, none of the methods captured the five original prototypes exactly. JS-NQT was almost identical, though, but WPD produced quite a different grouping. This is quite a reasonable result and illustrates both the difficulties of clustering to obtain a particularly expected grouping and the ability to learn unexpected patterns in the data. It is possible that the JS-based distances were distracted by the presence of nuisance variables, levels of the granularities, that do not contribute to clustering. This would also be supported by the results of the validation study, where clustering was less effective in S2 and S3, where only some granularities had differences between levels. Clustering using WPD is expected to produce quite different results because it will group only by overall value of a granularity, not a particular pattern. A cluster summary like Figure 9 is not possible because there may be different but equally interesting patterns (e.g. high evening *hod* and high day time *hod*) in the same cluster. Simply it provides information that across a collection of customers this specific cluster has interesting patterns in a granularity (eg *hod*). One would need to post-process the cluster to separate specific patterns.

5 Discussion

We offer two approaches for calculating pairwise distances between time series based on probability distributions over multiple cyclic granularities at once. Depending on the goal of the clustering, these distance metrics, when fed into a hierarchical clustering algorithm using Ward's linkage, yield meaningful clusters. Probability distributions provide an intuitive method to characterize noisy, patchy, long, and unequal-length time series data. Distributions over cyclic granularities help to characterize the formed clusters in terms of their repeating behavior over these cyclic granularities. Furthermore, unlike earlier efforts that group customers based on behavior across only one cyclic granularity (such as hour-of-day), our method is more comprehensive in detecting clusters with repeated patterns at all relevant granularities.

There are a few areas to extend this research. First, larger data sets with more uncertainty complicate matters, as is true for any clustering task. Characterizing clusters with varied or outlying customers can result in a shape that does not represent the group. Moreover, integrating heterogeneous consumers may result in visually identical end clusters, which are potentially not useful. Hence, a way of appropriately scaling it up to many customers such that anomalies are removed before clustering would be useful for bringing forth meaningful, compact and separated clusters. Secondly, we have assumed the time series to be stationary, and hence the distributions are assumed to remain constant for the observation period. In reality, however, it might change. For the smart meter example, the distribution for a customer moving to a different house or changing electrical equipment can change drastically. Our current approach can not detect these dynamic changes. Thirdly, it is possible that for a few customers, data for some categories from the list of considered significant granularities are missing. In our application, we have removed those customers and done the analysis but the metrics used should be able to incorporate those customers with such structured missingness. Finally, *wpd* is computationally heavy even under parallel computation. Future work can make the computations more efficient so that they are easily scalable to a large number of customers. Moreover, experiments can also be run with non-hierarchy based clustering algorithms to verify if these distances work better with other algorithms.

Acknowledgments

The authors thank the ARC Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS) for supporting this research. Sayani Gupta was partially funded by Data61 CSIRO during her PhD. The Monash eResearch Centre and eSolutions-Study Support Services supported this research in part through the resource usage of the MonARCH HPC Cluster. The Github repository, github.com/Sayani07/paper-gracsr, contains all materials required to reproduce this article and the code is also available online in the supplemental materials. This article was created with R (R Core Team 2021), `knitr` (Xie 2015, 2020) and `rmarkdown` (Xie et al. 2018, Allaire et al. 2020). Graphics are produced with `ggplot2` (Wickham 2016) and `GGally` (Schloerke et al. 2021).

6 Supplementary Materials

Data and scripts: Data sets and R code to reproduce all figures in this article (main.R).

Supplementary paper: Additional tables, graphics and R code to reproduce it (paper-supplementary.pdf, paper-supplementary.Rmd). The code for creating validation designs and running the methodologies is available at (`https://github.com/Sayani07/paper-gracsR/Validation`).

R-package: To implement the ideas provided in this research, the open-source R package 'gracsr' is available on Github (`https://github.com/Sayani07/gracsr`).

7 Bibliography

References

Aghabozorgi, S., Seyed Shirkhorshidi, A. & Ying Wah, T. (2015), 'Time-series clustering – a decade review', *Inf. Syst.* **53**, 16–38.

Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W. & Iannone, R. (2020), *rmarkdown: Dynamic Documents for R*. R

package version 2.1.
URL: *https://github.com/rstudio/rmarkdown*

Asimov, D. (1985), 'The grand tour: a tool for viewing multidimensional data', *SIAM journal on scientific and statistical computing* **6**(1), 128–143.

Borg, I. & Groenen, P. J. (2005), *Modern multidimensional scaling: Theory and applications*, Springer Science & Business Media.

Chicco, G. & Akilimali, J. S. (2010), 'Renyi entropy-based classification of daily electrical load patterns', *IET generation, transmission & distribution* **4**(6), 736–745.

Cook, D. & Swayne, D. F. (2007), *Interactive and Dynamic Graphics for Data Analysis: With R and Ggobi*, Springer, New York, NY.

Corradini, A. (2001), Dynamic time warping for off-line recognition of a small gesture vocabulary, *in* 'Proceedings IEEE ICCV workshop on recognition, analysis, and tracking of faces and gestures in real-time systems', IEEE, pp. 82–89.

Cutler, A. & Breiman, L. (1994), 'Archetypal analysis', *Technometrics* **36**(4), 338–347.

Dasu, T., Swayne, D. F. & Poole, D. (2005), Grouping multivariate time series: A case study, *in* 'Proceedings of the IEEE Workshop on Temporal Data Mining: Algorithms, Theory and Applications, in conjunction with the Conference on Data Mining, Houston', Citeseer, pp. 25–32.

Department of the Environment and Energy (2018), *Smart-Grid Smart-City Customer Trial Data*, Australian Government, Department of the Environment and Energy.
URL: *https://data.gov.au/dataset/4e21dea3-9b87-4610-94c7-15a8a77907ef*

Dunn, J. C. (1973), 'A fuzzy relative of the ISODATA process and its use in detecting compact Well-Separated clusters', *Journal of Cybernetics* **3**(3), 32–57.

Fan, H., Liu, P., Xu, M. & Yang, Y. (2021), 'Unsupervised visual representation learning via dual-level progressive similar instance selection', *IEEE Transactions on Cybernetics* pp. 1–11.

Fisher, R. A. (1936), 'The use of multiple measurements in taxonomic problems', *Annals of eugenics* **7**(2), 179–188.

Gupta, S., Hyndman, R. J. & Cook, D. (2021), Detecting distributional differences between temporal granularities for exploratory time series analysis, Working paper.

Hennig, C. (2014), How many bee species? a case study in determining the number of clusters, *in* 'Data Analysis, Machine Learning and Knowledge Discovery', Springer International Publishing, pp. 41–49.

Hennig, C. (2019), Cluster validation by measurement of clustering characteristics relevant to the user, *in* 'Data Analysis and Applications 1', John Wiley & Sons, Inc., Hoboken, NJ, USA, pp. 1–24.

Jolliffe, I. T. & Cadima, J. (2016), 'Principal component analysis: a review and recent developments', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **374**(2065), 20150202.

Krzysztofowicz, R. (1997), 'Transformation and normalization of variates with specified distributions', *J. Hydrol.* **197**(1-4), 286–292.

Lee, S. (2021), *liminal: Multivariate Data Visualization with Tours and Embeddings*. R package version 0.1.2.
URL: *https://CRAN.R-project.org/package=liminal*

Liao, T. W. (2005), 'Clustering of time series data—a survey', *Pattern recognition* **38**(11), 1857–1874.

Liao, T. W. (2007), 'A clustering procedure for exploratory mining of vector time series', *Pattern Recognition* **40**(9), 2550–2562.

Melnykov, V. (2013), 'Challenges in model-based clustering', *Wiley Interdiscip. Rev. Comput. Stat.* **5**(2), 135–148.

Menéndez, M., Pardo, J., Pardo, L. & Pardo, M. (1997), 'The jensen-shannon divergence', *Journal of the Franklin Institute* **334**(2), 307–318.

Motlagh, O., Berry, A. & O'Neil, L. (2019), 'Clustering of residential electricity customers using load time series', *Appl. Energy* **237**, 11–24.

Ndiaye, D. & Gabriel, K. (2011), 'Principal component analysis of the electricity consumption in residential dwellings', *Energy Build.* **43**(2), 446–453.

Olvera-López, J. A., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F. & Kittler, J. (2010), 'A review of instance selection methods', *Artificial Intelligence Review* **34**(2), 133–143.

Ozawa, A., Furusato, R. & Yoshida, Y. (2016), 'Determining the relationship between a household's lifestyle and its electricity consumption in japan by analyzing measured electric load profiles', *Energy and Buildings* **119**, 200–210.

R Core Team (2021), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *https://www.R-project.org/*

Rhodes, J. D., Cole, W. J., Upshaw, C. R., Edgar, T. F. & Webber, M. E. (2014), 'Clustering analysis of residential electricity demand profiles', *Appl. Energy* **135**, 461–471.

Rousseeuw, P. J. (1987), 'Silhouettes: a graphical aid to the interpretation and validation of cluster analysis', *Journal of computational and applied mathematics* **20**, 53–65.

Schloerke, B., Cook, D., Larmarange, J., Briatte, F., Marbach, M., Thoen, E., Elberg, A. & Crowley, J. (2021), *GGally: Extension to 'ggplot2'*. R package version 2.1.1.
URL: *https://CRAN.R-project.org/package=GGally*

Tibshirani, R., Walther, G. & Hastie, T. (2001), 'Estimating the number of clusters in a data set via the gap statistic', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63**(2), 411–423.

Tureczek, A. M. & Nielsen, P. S. (2017), 'Structured literature review of electricity consumption classification using smart meter data', *Energies* **10**(5), 584.

Ushakova, A. & Jankin Mikhaylov, S. (2020), 'Big data to the rescue? challenges in analysing granular household electricity consumption in the united kingdom', *Energy Research & Social Science* **64**, 101428.

Van der Maaten, L. & Hinton, G. (2008), 'Visualizing data using t-sne.', *Journal of machine learning research* **9**(11).

Wang, E., Cook, D. & Hyndman, R. J. (2020), 'A new tidy data structure to support exploration and modeling of temporal data', *Journal of Computational & Graphical Statistics* **29**(3), 466–478.

Wegman, E. J. (1990), 'Hyperdimensional data analysis using parallel coordinates', *Journal of the American Statistical Association* **85**(411), 664–675.

Wickham, H. (2016), *ggplot2: Elegant Graphics for Data Analysis*, Springer-Verlag New York.
URL: *http://ggplot2.org*

Xie, Y. (2015), *Dynamic Documents with R and knitr*, 2nd edn, Chapman and Hall/CRC, Boca Raton, Florida.
URL: *https://yihui.name/knitr/*

Xie, Y. (2020), *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.28.
URL: *https://yihui.org/knitr/*

Xie, Y., Allaire, J. J. & Grolemund, G. (2018), *R Markdown: The Definitive Guide*, Chapman and Hall/CRC, Boca Raton, Florida.
URL: *https://bookdown.org/yihui/rmarkdown*

Xu, D. & Tian, Y. (2015), 'A comprehensive survey of clustering algorithms', *Annals of Data Science* **2**(2), 165–193.