Clustering based on probability distributions with application on residential customers

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

October 5, 2021

Abstract

Clustering elements based on behavior across time granularities

Keywords: data visualization, statistical distributions, time granularities, calendar algebra, periodicties, grammar of graphics, R

*Email: Sayani.Gupta@monash.edu

1 Introduction

The Smart Grid, Smart City (SGSC) project, which rolled out Australia's first commercial-scale smart grid was implemented across eight local government areas in New South Wales (NSW). Data from more than 13,000 household electricity smart meters is obtained as part of that project. It provides half-hourly energy usage and demographic data for Australia, as well as detailed information on appliance use, climate, retail and distributor product offers, and other related factors. The trials were based in Newcastle, New South Wales, but also covered areas in Sydney CBD, Newington, Ku-Ring-Gai, and the rural township of Scone. The load time series is asynchronous as it is observed for these households for unequal time lengths and consists of missing observations.

The massive amount of data generated in such projects could be overwhelming for analysis. Electricity utilities can utilize the consumption patterns of customers to develop targeted tariffs for individual groups and alleviate the problem of volatility in production by capitalizing on the flexibility of consumers. Beyea (2010) has pointed out, there has been little discussion or exploration of the full potential of these data bases and their benefits can reach beyond the original intentions for collecting these data. Thus, there is a scope to investigate and analyze these data in various ways for a greater understanding of consumption patterns and how they correlate with other economic, physical or geographical factors. In this work, we are interested to see how we can utilize this dataset to group different customers with similar periodic behavior. Towards this goal, this chapter aims to: (a) describe the contents of the data set in SGSC database that we can utilize, and (b) propose a clustering algorithm to group customers with similar periodic behaviors. The distance metric introduced in Chapter 2 will be the inputs for this cluster analysis. One of the advantages of using our approach is that the technique is based on probability distributions instead of raw data. Many clustering approaches are limited by the type of noisy, patchy, and unequal time-series common in residential data sets. Since the distance measure considered is based on differences in probability distribution of time series, it is likely to be less sensitive to missing or noisy data.

Themes

- Dimension reduction: If each $P_{i,j,k}$ be considered to be a point in the space, key i

would have mp dimensions as opposed to n_i dimensions in case of considering raw data. Hence for a large number of observations ($n_i >> mp$), this approach benefits by transitioning to a lower dimension.

- Avoid loss of information due to aggregation: This approach ensures key characteristic information of the data is not lost due to averaging or aggregation measures in an attempt to transition to a lower dimension. Hence, this approach could be thought to somehow balance the drawback of considering raw data or aggregated data.

- Robustness to outliers: This approach could be adapted to be robust to outliers and extreme behaviors by trimming the tails of the probability distributions.

- Non-synchronized observed time periods: Considering probability distribution would imply the clustering process can handle keys that are observed over periods of time that are overlapping but don't necessarily coincide.

- Similar periodic behavior: Since cyclic granularities are considered instead of linear granularities, clustering would group keys that have similar behavior across these cyclic granularities. This implies they will be grouped according to their periodic behavior and not on the linear stretch of time over which they are observed.

Common load clustering techniques of smart meter data

The foundation for this study is Tureczek2017-pb, which conducts a systematic review of the current state of the art in smart meter data analytics, which evaluates approximately 2100 peer-reviewed papers and summarizes the main findings. None of the 34 selected papers which focus on clustering consumption are based on Australian smart meter data. The clustering is frequently applied directly to the raw data without scrutinizing for auto correlation and periodicity. The algorithm most ubiquitously employed is K-Means. But the omission of the time series structure or correlation in the analysis while employing K-Means leads to inefficient clusters. Principal Component Analysis or Self-Organizing Maps removes correlation structures and transitions the data to a reduced feature space, but it comes at a cost of interpretability of the final results. **?** has shown that a transformation of data to incorporate autocorrelation before K-Means clustering can improve performance

and enable K-Means to deliver smaller clusters with less within-cluster variance. However, it does not explain the cluster composition by combining it with external data. Some papers present pre-processing of the smart-meter data before clustering through principal component analysis or factor analysis for dimensionality reduction or self-organizing maps for 2-Dimensional representation of the data (**?**). Other algorithms used in the literature include k-means variations, hierarchical methods and k-medoids based on a greedy algorithm have been designed to select typical periods in the time series. As the methods are often situation specific, it makes sense to compare them on the performance rather than any standard performance metric. A type of clustering based on information theory such as Shannon or Renyi entropy and their variants are addressed in , which differs from typical methods adopted for electricity consumer classification, based on the Euclidean distance notion. **?** presents strategy to address the problems on patchy, and unequal time-series common in residential data sets by converting load time series into map models. Most time-series clustering models are limited to handling time domain with same start and end date and time. Most of the solutions to handle this like longest common subsequence, dynamic time warping are prone to computational limit with increased length of the series.

The following contributions are made through the following chapter:

- Present a cluster analysis of SGSC dataset to group households with similar periodic behavior
- Cluster validation by relating to external data

2 Clustering methodology

The data set solely contains readings from smart meters and no information about the consumers' specific physical, geographical, or behavioural attributes. As a result, no attempt is made to explain why consumption varies. Instead, this work investigates how much energy usage heterogeneity can be found in smart meter data and what some of the most common electricity use patterns are. It is worth noting that when studying these dynamics, a variety of objectives may be pursued. One objective could be to group consumers with similar shapes over all relevant cyclic granularities. In this scenario, the variation in customers

within each group is in magnitude rather than shape, while the variation between groups is only in shape. Most clustering algorithms offer only daily energy profiles throughout the hours of the day, but we suggest a broader approach to the problem, aiming to group consumers with similar shapes across all significant cyclic granularities. Another purpose of clustering could be to group customers that have similar differences in patterns across all major cyclic granularities, capturing similar jumps across categories regardless of the overall shape. For example, in the first goal, similar shapes across hours of the day will be grouped together, resulting in customers with similar behaviour across all hours of the day, whereas in the second goal, any similar big-enough jumps across hours of the day will be clubbed together, regardless of which hour of the day it is. Both of these objectives may be useful in a practical context and, depending on the data set, may or may not propose the same customer classification.

The proposed methodology aim to leverage the intrinsic temporal data structure hidden in time series data. The foundation of our method is unsupervised clustering algorithms based exclusively on time-series features. First, we study the underlying distributions that may have resulted in different patterns across temporal granularities in order to identify a mechanism to classify them based on the similarity of those distributions. Depending on the goal of clustering, the distance metric for defining similarity would be different. These distance metrics could be fed into a clustering algorithm to break large data sets into subgroups that can then be analyzed separately. These clusters may be commonly associated with real-world data segmentation. However, since the data is unlabeled a priori, more information is required to corroborate this. This section presents the work flow of the methodology:

- *Data preparation*

? introduced the tidy "tsibble" data structure to support exploration and modeling of temporal data. A tsibble comprises an index, optional key(s), and measured variables. An index is a variable with inherent ordering from past to present and a key is a set of variables that define observational units over time. A linear granularity is a mapping of the index set to subsets of the time domain. For example, if the index of a tsibble is days, then a linear granularity might be weeks, months or years. For each key variable, the raw

smart meter data is a sequence that is indexed by time and comprises values of several measurement variables at each time point. This sequence, though, could be depicted in a variety of ways. A shuffling of the raw sequence could reflect the distribution of hourly consumption over a single day, while another could indicate consumption over a week or a year. These temporal deconstructions of a time period into units such as hour-of-day, work-day/weekend are called cyclic temporal granularities. All cyclic granularities can be expressed in terms of the index set and could be augmented with the initial tsibble structure (index, key, measurements). It is worthwhile to note that the data structure changes while transporting from linear to cyclic scale of time as multiple observations of the measured variable would correspond to each category of the cyclic granularities. In this paper, quantiles are chosen to characterize the distributions for each category of the cyclic granularity. So, each category of a cyclic granularity corresponds to a list of numbers which is essentially few chosen quantiles of the multiple observations.

- *Finding significant cyclic granularities or harmonies*

These cyclic granularities are useful for exploring repetitive patterns in time series data that get lost in the linear representation of time. It is advantageous to consider only those cyclic granularities across which there is a significant repetitive pattern for the majority of customers or noteworthy in an electricity-behavior context. In that case, when the customers are grouped, we can expect to observe some interesting patterns across the categories of the cyclic granularities considered. [XXX reference 2nd chapter] proposes a way to select significant cyclic granularities and harmonies which is used for this paper.

- *Individual or combined categories of cyclic granularities as DGP*

The existing work on clustering probability distributions assumes we have an iid sample $f_1(v), \ldots, f_n(v)$, 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 work, we are using i as denoting a customer and the underlying variable as the electricity demand. So $f_i(v)$ is the distribution of household i and v is electricity demand. In this work, instead of considering the probability distributions of the linear time series, we assume that the measured variables across different categories of any cyclic granularity are from

data generating processes. Hence, we want to be able to cluster distributions of the form $f_{i,A,B\ldots,N_C}(v)$, where A, B represent the cyclic granularities under consideration such that $A = \{a_j : j = 1, 2, \ldots J\}$, $B = \{b_k : k = 1, 2, \ldots K\}$ and so on. We consider individual each category of a cyclic granularity (A) or combination of categories for interaction of cyclic granularities (for e.g. $A * B$) to have a distribution. For example, let us consider we have two cyclic granularities of interest, $A = 0, 1, 2, \ldots, 23$ representing hour-of-day and $B = \{Mon, Tue, Wed, \ldots, Sun\}$ representing day-of-week. each customer i consist of a collection of probability distributions. In case individual granularities (A or B) 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. As a result, a distance between collections of these univariate probability distributions is required. Depending on the objective of the problem, there could be many approaches to considering such distances. This paper considers two approaches, which are explained in the next segment.

- *Distance metrics*

Considering each individual or combined categories of cyclic granularities as a data generating process lead to a collection of conditional distributions for each customer i. The (dis) similarity between each pair of observations should be obtained by combining the distances between these collections of conditional distributions such that the resulting metric is a distance metric, which could be fed into the clustering algorithm. Two types of distance metric is considered:

Inter-category distances

This distance matrix considers two objects to be similar if every category of an individual cyclic granularity or combination of categories for interacting cyclic granularities have similar distributions. In this study, the distribution for each category is characterized using deciles and the distances between distributions are computed by using the Jensen-Shannon distance, which is symmetric and hence could be used as a distance measure.
The total distance between two elements x and y is then defined as

$$S^A_{x,y} = \sum_j D_{x,y}(A)$$

(sum of distances between each category j of cyclic granularity A) or

$$S_{x,y}^{A*B} = \sum_j \sum_k D_{x,y}(A, B)$$

(sum of distances between each combination of categories (j, k) of the harmony (A, B). When combining distances from individual cyclic granularities A and B,

$$S_{x,y}^{A,B} = S_{x,y}^{A}/J + S_{x,y}^{B}/K$$

is used, which could also be shown to be a distance metric easily. This is shown for cyclic granularity A and B, but could be practically extended to more granularities.

Intra-category distances

Compute weighted pairwise distances (*wpd*) (XXX reference) for all considered granularities for all objects. *wpd* is designed to capture the maximum variation in the measured variable explained by an individual cyclic granularity or their interaction and is estimated by the maximum pairwise distances between consecutive between consecutive categories normalized by appropriate parameters. A higher value of *wpd* indicates that some interesting pattern is expected, whereas a lower value would indicate otherwise.
Distance between objects is then taken as the euclidean distances between them with the granularities being the variables and *wpd* being the value under each variable. Since Euclidean distance is chosen, the observations with high values of features (*wpd* values) will be clustered together. The same holds true for observations with low values of features. Thus this distance matrix would be useful to group customers that have similar significance of patterns across different granularities.

- *Pre-processing steps*

  Handling trend, seasonality, non-stationarity and auto-correlation: Trend and seasonality are fundamental characteristics of time series data, and it is reasonable to define a time series according to its degree of trend and seasonality. These characteristics of the time series are lost or handled independently by considering probability distributions (trend is lost) across categories of cyclic granularities (by independently modeling all seasonal fluctuations), and so there is no need to de-trend

or de-seasonalize the data before conducting the clustering method. There is no need to omit holiday or weekend patterns for similar reasons.

Data transformation: Robust scaling method is used before computing the Inter-category distances and NQT is built-in transformation used for computation of *wpd*, which forms the basis of Intra-category distances.

- *Clustering algorithm*

In the analysis of energy smart metre data, K-Means or hierarchical clustering are often employed. These are simple and effective techniques that work well in a range of scenarios. For clustering, both employ a distance measure, and the distance measure chosen has a major influence on the structure of the clusters. We employ agglomerative hierarchical clustering in conjunction with Ward's criteria (XXX reference). Individual entities with the highest similarity computed using the desired distance metrics are sequentially merged using agglomerative algorithms. We can possibly employ any clustering method that supports the given distance metric as input.

- *Characterization of clusters*

Depending on the distance measure utilized for the study, the cluster characterization technique will differ. Clusters that utilise intra-category distances are characterised using multi-dimensional scaling and parallel coordinate displays. For inter-category distances, the distribution across major granularities may be presented to ensure that the goal of similar shapes within clusters and distinct shapes across clusters is met. This technique may potentially make advantage of multi-dimensional scaling.
Multidimensional scaling (MDS) (XXX reference) refers to a family of methods that analyse a matrix of distances or dissimilarities to provide a representation of the data points in a reduced-dimension space. There are many kinds of MDS, but they all solve the same fundamental issue: Given a $n*n$ matrix of dissimilarities and a distance measure, identify a configuration of n points $x_1, x_2, \ldots, x_n$ in the reduced dimension space R^q $(q < p)$ where the distance between the points is near to the dissimilarity between the points. All techniques must determine the coordinates of the points as well as the space dimension, q. Metric

and nonmetric MDS are the two main kinds of MDS. Metric MDS methods presume a functional connection between the interpoint distances and the supplied dissimilarities and assume that the data are quantitative. We use metric MDS.
Parallel coordinate plots (XXX reference) Parallel coordinates have been extensively used to display high-dimensional and multivariate data, allowing for the detection of patterns within the data via visual grouping.

3 Validation

To evaluate clustering approaches, we create data designs that replicate prototype behaviours that might be seen in electricity data contexts. We spiked several features in the data to see where one method works better than the other and where they might give us the same outcome or the effect of missing data and trends on the proposed methods. A continuous measured variable y of length T indexed by $0, 1, \ldots T-1$. Three circular granularities $g1$, $g2$ and $g3$ are considered with 2, 3 and 5 levels respectively. Categories of $g1$, $g2$ and $g3$ are represented by $g10, g11$, $g20, g21, g22$ and $g30, g31, g32, g33, g34$. 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 $Mon, Tue, \ldots, Sun$. Each categories of $g1$, $g2$ and $g3$ could be assumed to be from same distribution, few from one and others from separate distributions, or all from different distributions, resulting in distinct data designs. We created independent replications ($R = \{50, 200, 500\}$) of all data designs to see if our proposed clustering approaches can detect distinct designs in various groups for small, medium and large number of series. A sample size of $T = \{300, 500, 2000\}$ is used in all designs to test small, medium and large sized series. For all data designs, the data type is set to "continuous," and a gaussian setup for errors is assumed. The code for creating these designs can be found in the Supplementary section (link to github repo). The results for $T = 300$ and $R = 50$. The rest of the results could be found in the supplementary paper.

3.1 Data generating processes

An ARMA (p,q) process is used to generate series, where p and q are selected at random such that the series is stationary. The various designs on $g1$, $g2$, and $g3$ are introduced by adding matching designs to this series' innovations. The innovations are considered to have a normal distribution, although they follow the same pattern as the designs. To eliminate the effect of starting values, the first 500 observations in each series are discarded.

3.2 Data designs

Three signifiant granularities

Consider the scenario when all three granularities $g1$, $g2$, and $g3$ are responsible for distinguishing the designs. That means that for at least one among the to-be-grouped designs, the pattern for each of $g1$, $g2$, and $g3$ will change. We consider null instances to represent no variation in distribution across categories, i.e., all categories have the same distribution N(0,1).

granularity	Alternate design
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)

A combination of these null and alternate distributions are considered when building the designs as in the table **??**. Figure **??** shows the linear and cyclic representation of y under these five designs.

design	g1	g2	g3
design-1	null	null	null
design-2	vary	null	null
design-3	null	vary	null
design-4	null	null	vary
design5	vary	vary	vary

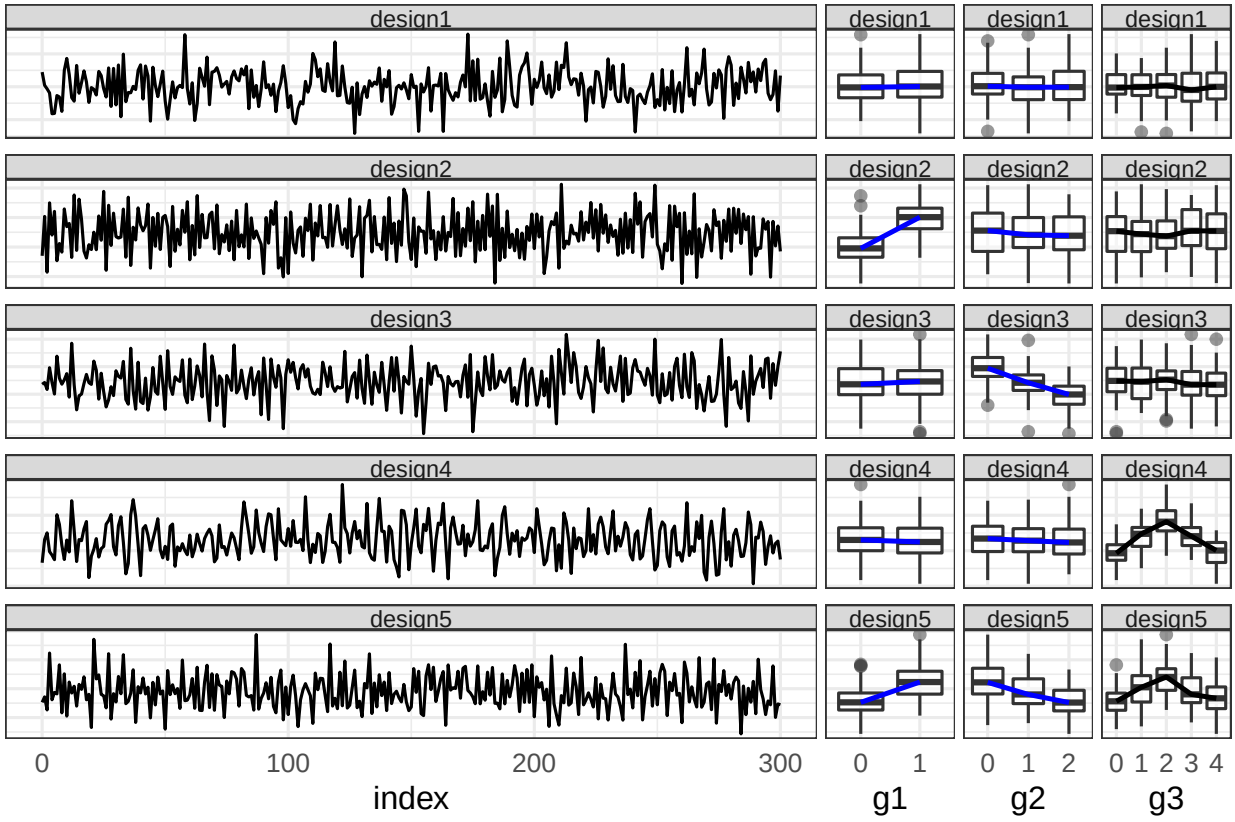


Figure 1: The linear and cyclic representation of the time series variable y. It is not possible to comprehend these patterns across cyclic granularities g1, g2 and g3 or group similar series just by looking at the linear plots.

(#fig:plot-linear-change)

Two significant granularities

Consider a case where distribution of y would vary across levels of $g2$ for all designs, across levels of $g1$ for few designs and $g3$ does not change across designs. So $g3$ is not responsible for distinguishing across designs. Figure **??** shows the linear and cyclic representation of y. The first panel shows raw plot of y in a linear scale and the second panel shows distribution of y across cyclic granularities namely $g1$, $g2$ and $g3$. As could be seen from the plots, it is impossible to decipher from the raw time plot that the time series variable shows such pattern across different granularities.

One signifiant granularity

Consider a case where distribution of y would vary across levels of $g2$ for all designs, across levels of $g1$ for few designs and $g3$ does not change across designs. So $g3$ is not responsible for distinguishing across designs.

Missing data

Trend

Outliers

observations:

designs:

A subset of many possible designs are shown in Figure **??**. For the parameter space (XXX unique combinations shown in table YYY), 100, 500 independent replications of all possible combination of simulation parameters were generated. The clustering methodologies were run all these unique combinations and subsets of these to verify if the methodologies work as expected.

Granularity type	# Significant	# Replications
Individual # obs: 300, 500, 2000 # clusters: 6/7	1/2/3	25, 100, 200
Interaction # obs: 500, 2000 # clusters: 4	1/2	25, 100, 200

3.3 Results

All the methods were fitted to each data designs and results are reported through confusion matrices. With increasing difference between categories, it gets easier for the methods to

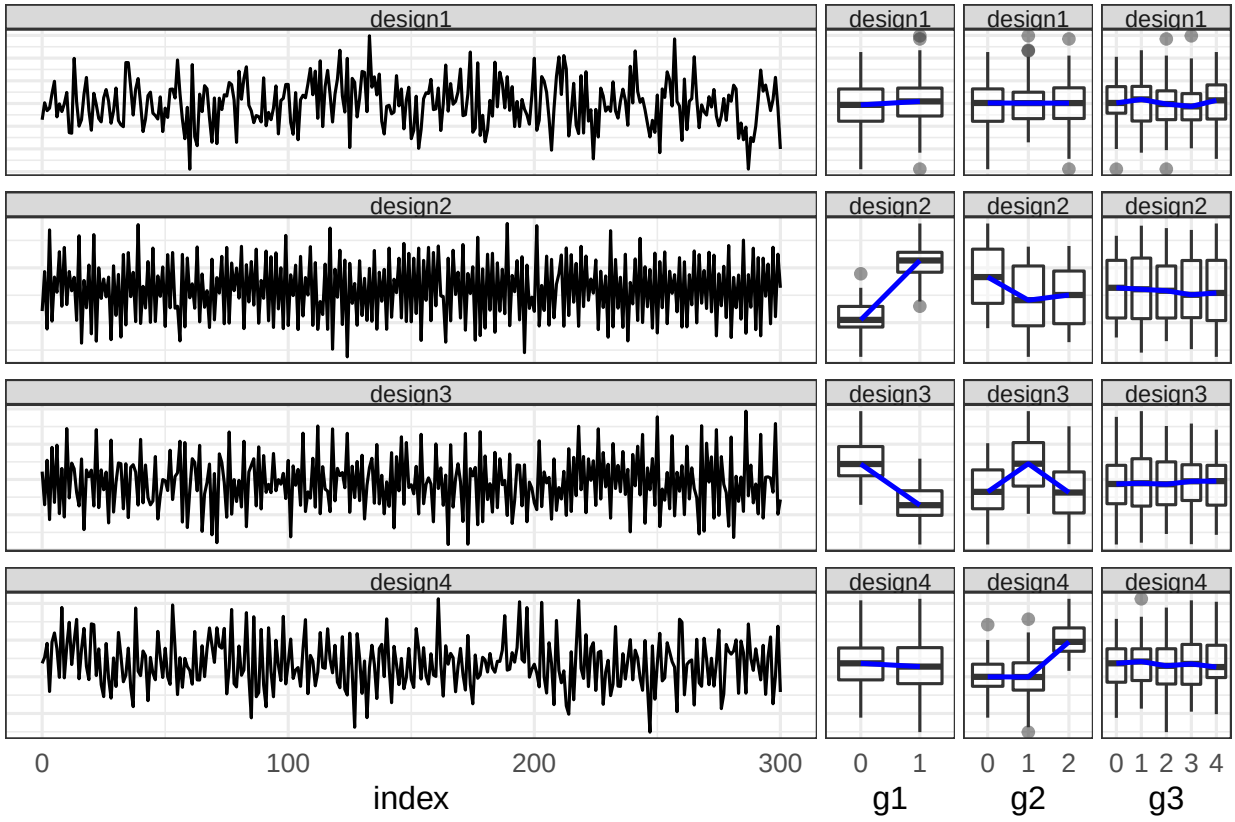


Figure 2: The linear and cyclic representation of the time series variable y. It is not possible to comprehend these patterns across cyclic granularities g1, g2 and g3 or group similar series just by looking at the linear plots.

(#fig:plot-linear)

correctly distinguish the designs. For difference=1, the performances are pretty bad for js-robust methods and wpd method. The performance starts getting better with increasing difference and get worse with increasing number of replications. Length of series do not show to have any effect on the performance of the methods.

4 Application

4.1 Data source

The entire data is procured from CSIRO. A subset of this data is also available from SGSC consumer trial data is available through Department of the Environment and Energy. It consists of the following data sets. *1. CustomerData:* 78720 customers with 62 variables about them *2. EUDMData:* 300 billion half-hourly consumption level data
3. OffersData: Method of contact to customer to join SGSC customer trial, either door-to-door (D2D) or via Telesales
4. PEResponseData: Peak Events response customer wise
5. PETimesData: Peak Events time stamps

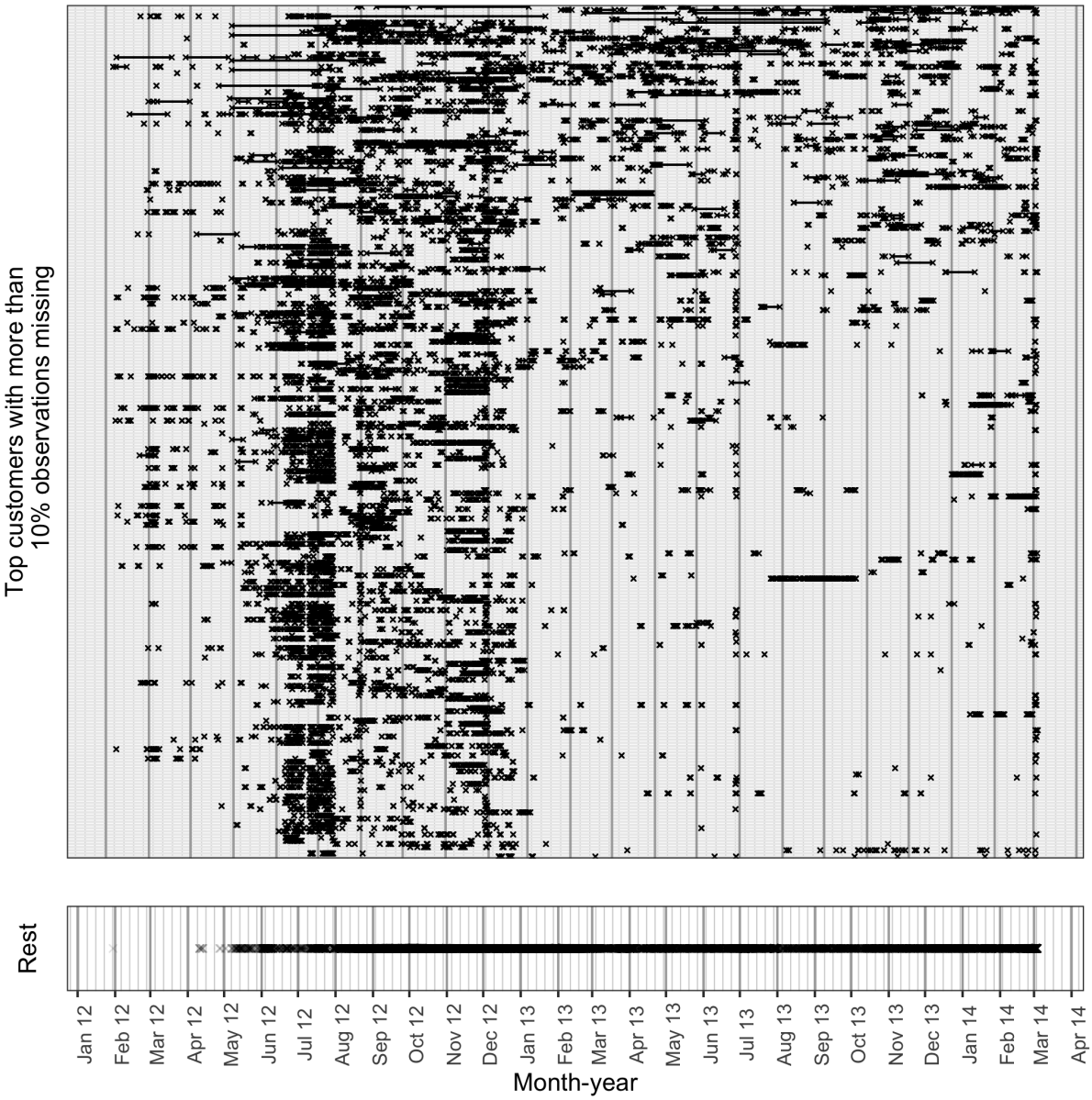
Only *CustomerData* and *EUDMData* are relevant for the clustering goals of this paper. *EUDMData* contains half-hourly general supply in KwH for 13,735 customers, resulting in 344,518,791 observations in total. `CustomerData` provides demographic data for 78,720 customers most of which are missing and not utilized for the purpose of this paper. To meet the requirements for anonymity preservation, the energy patterns could not be identified at the individual level, but rather by the geographical location of their residence information about their Local Government Area.

4.1.1 Raw data

4.1.2 Missing Data

Electricity usage for some customers may become unavailable due to power outage or not recording their usage properly, thus resulting in implicit missing values in the database. It is interesting to explore where missing-ness occurs or if there is a relationship between the

underlying missing patterns. We use the R package `tsibble` to do this.



- if there is any systematic missing patterns in the data
- this missing plot can go in the supplementary
- how to add missing values (should be added in data pre-processing)
- instance learning
- types of summary techniques to use (generally multivariate means and sd are used, I can't use that in a time series context, you can show across different granularity? within-group sum of squares and between-group sum of squares

)

13735 customers in elec_ts 8685 customers in elec_nogap 5050 customers in count_na_df

Then is the graph of missing observations even interesting. You can show two graphs, one to show that missingness do not have a pattern another to show even if no missing, they start and end at different times. (A sample of 50 customers).

A dataset of 100 SGSC homes has been used to lay out the structure to be used for analyzing the big dataset. The smaller dataset contains half-hourly kwh values form 2012 to 2014 and has asynchronous time series distributed evenly over the observation period (Figure **??**), similar to the bigger data set. Figure **??** can be used to interpret missingness in the data, where the customers are arranged from maximum to minimum missing. It looks like data is most missing before 2013 and for a particular date in 2014.

4.1.3 Cluster validation

Internal cluster validation uses the internal information of the clustering process to evaluate the goodness of a clustering structure without reference to external information. It can be also used for estimating the number of clusters and the appropriate clustering algorithm without any external data.

Generally most of the indices used for internal clustering validation combine compactness and separation measures.

We will also determine if any identified clusters or patterns are indeed statistically meaningful in the sense that they actually exist and are not a random allocation. Hence, the robustness of this methodology is tested through simulations. This section will contain the data structure and detailed methodology to be employed for the cluster analysis. The cluster validation indexes like average silhouette width (ASW) is to be employed here to check how homogeneous these clusters are.

External cluster validation consists in comparing the results of a cluster analysis to an externally known result, such as externally provided class labels. It measures the extent to which cluster labels match externally supplied class labels. Since we know the "true" cluster number in advance, this approach is mainly used for selecting the right clustering algorithm for a specific data set.

Relative cluster validation evaluates the clustering structure by varying different parameter values for the same algorithm (e.g.,: varying the number of clusters k). It's generally used for determining the optimal number of clusters.

Interval validation includes *Compactness or cluster cohesion:* Measures how close are the objects within the same cluster. A lower within-cluster variation is an indicator of a good compactness (i.e., a good clustering). The different indices for evaluating the compactness of clusters are base on distance measures such as the cluster-wise within average/median distances between observations.

Separation: Measures how well-separated a cluster is from other clusters. The indices used as separation measures include: distances between cluster centers the pairwise minimum distances between objects in different clusters

Connectivity: corresponds to what extent items are placed in the same cluster as their nearest neighbors in the data space. The connectivity has a value between 0 and infinity and should be minimized.

4.2 Clustering results for 100 customers

4.3 Clustering results for 5K customers

4.4 Software implementation

The implementation for our framework is available in the R package `gracsr` for ease of use in other applications.

4.5 Discussion

This section will cover some drawback of this clustering method and potential extensions of this work.

–>

–>

–>

–>

–>

References