Learning to Predict with Highly Granular Temporal Data

Ushakova, Anastasia; Mikhaylov, Slava J.

Learning to Predict with Highly Granular Temporal Data: Estimating individual behavioral profiles with smart meter data*

Anastasia Ushakova
University College London
anastasia.ushakova.14@ucl.ac.uk

Slava J. Mikhaylov
University of Essex
s.mikhaylov@essex.ac.uk

November 27, 2017

Abstract

Big spatio-temporal datasets, available through both open and administrative data sources, offer significant potential for social science research. The magnitude of the data allows for increased resolution and analysis at individual level. While there are recent advances in forecasting techniques for highly granular temporal data, little attention is given to segmenting the time series and finding homogeneous patterns. In this paper, it is proposed to estimate behavioral profiles of individuals' activities over time using Gaussian Process-based models. In particular, the aim is to investigate how individuals or groups may be clustered according to the model parameters. Such a Bayesian non-parametric method is then tested by looking at the predictability of the segments using a combination of models to fit different parts of the temporal profiles. Model validity is then tested on a set of holdout data. The dataset consists of half hourly energy consumption records from smart meters from more than 100,000 households in the UK and covers the period from 2015 to 2016. The methodological approach developed in the paper may be easily applied to datasets of similar structure and granularity, for example social media data, and may lead to improved accuracy in the prediction of social dynamics and behavior.

Keywords: big data, time series models, consumer behavior, smart meters

*This work is funded by the UK ESRC Consumer Data Research Centre (CDRC) grant reference ES/L011840/1

Introduction

Social and political science research on time series data often focuses on prediction models that strive to understand the dynamics behind the outcome variables (Box-Steffensmeier et al., 2014). Time series classification models have been used for predictions of conflict and civil war (Muchlinski et al., 2015; Hegre et al., 2013), political instability (Goldstone et al., 2010) or design of aggregated indicators, such as political accountability (Chappell and Keech, 1985).

Fine-grained, complex non-stationary time series that characterize social dynamics have entered political science research via social media analysis (**?**). Prediction of individual behavior with such data is challenging. A mixture of methods to predict behavior with such data is posed as an alternative. The aim for this paper is to consider challenges and opportunities that may be associated with large highly-granular temporal datasets. In particular, the focus is on challenges relating to aggregation of time-series in the context of big data.

Sampling frequency (or equivalently aggregation level) of data affects the output of analysis. Intuitively, it is expected that the aggregation results in some information loss. The question posed is about the importance of this kind of information for inferences about the process. Aggregation is useful for data reduction, which, in turn, speeds up the computation. In general, the information lost in aggregation depends on properties of the process itself. For example, consider a signal that has very gradual variation over time, then increasing the sampling interval, or equivalently increasing aggregation over very granular samples, may not have much impact on inference about the process. Conversely, if activity varies rapidly, then oit can be expected that aggregation will have a large impact and significantly limit any insights obtained from the analysis. In the event-data application, Shellman (2004) demonstrates the varying effect of aggregation on both segmentation and prediction. We expect the effect to be even more pronounced for highly granular time-series big data.

As an illustrative example, we analyze smart meter data that records household energy consumption at half-hourly intervals. Such data may be used as a proxy for individual household behavior and activities (Anderson et al., 2017). In order to make the approach more generalizable, we propose a two stage procedure: first, segment the population with unsupervised

clustering methods to categorize behavior; second, predict cluster allocation using only time series features.

The paper proceeds as follows. First, we introduce our case study based on a sample of smart meter data available for the UK throughout 2014 and 2015. We assesses several methods for clustering time-series with the aim of segregating consumer behavior. This is followed by prediction of behavioral classes from the individual time-series records. The paper concludes with a discussion on how aggregation affects the analysis in our case study.

Smart Meter Time-Series Data

Time series data constitutes an ordered sequence indexed by time alongside values of the variables of interest at each point in time. For smart meter data there are various ways to represent such a time series sequence. For instance, one sequence could represent the total consumption per day, while another could track hourly energy consumption. We can then model different data generation processes depending on our level of aggregation.

Independence Assumption

In the case of smart meter data, data may be analyzed in either a univariate or multivariate setting. Traditional analysis is mainly univariate in nature and would impose an independence assumption across energy consumption levels if we are interested in fitting a parametric model. If we are concerned with the prediction of average (or aggregated) energy demand that is composed of consumption by individual users, then correlation or independence between streams may affect the aggregated processes. In the univariate case, each customer's time series is taken separately and it is attempted to predict their consumption using only their historical behavior.

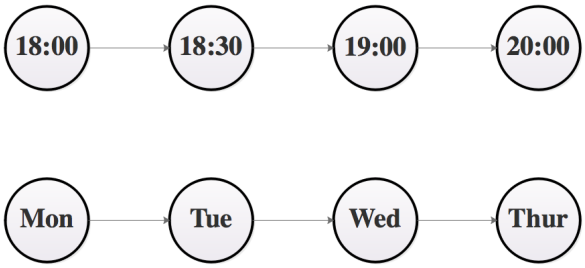

Figure 1: *Chains based on half-hourly readings and total daily readings.*

Energy consumption may be viewed as a first order chain of preceding readings. Figure 1 presents a schematic illustration. We assume that there are no interdependencies among the nodes in the chain other than on the previous time-step. Each time period is conditioned on the previous one. However, as an extension a second-order chain may be considered where **t3** may be dependent on **t1** and **t4** dependent on **t2**. Such models may be generalized to higher-orders at the expense of increasing model complexity and a drop in interpretability.

Description of Data

A summary of the dataset is given in Table 1. From the total set of 8.5 billion observations, we study two sub-samples of smart meter data streams. Aggregated patterns are calculated by taking the average half-hourly consumption across the whole year for each unique consumer at each postcode sector. This significantly reduces the volume of data. It is associated with certain levels of variability across the units of analysis, but the variability within the individual customer records is collapsed. To assess how much we can learn about the true dynamics from this aggregated level, we compare the aggregated results to those obtained on a disaggregated raw sample.

The overall dataset is sufficiently large and this may present computational limitations for some analyses. One approach, implemented in the disaggregated sample, is a random draw of 1,100 individuals. For privacy and security reasons, the overall sample is not used in the analysis. We present summary figures below to illustrate underlying data volumes. Computationally, datasets of such sizes can be problematic, especially for methods that extensively use matrix transformations.

Data	Overall	Aggregated Sample	Disaggregated Sample
Unique identifiers	489,000	8,171	1,100
Days	365	365	365
Daily readings	48	48	48
Total observations	8,567,280,000	143,155,920	19,272,000

Table 1: *Data structure.* The structure of our smart meter database and the samples. Note: Aggregated sample is average consumption at each geographical reference level (post-code sector).

Figure 2 presents an example of the average daily consumption pattern and variation around this average for a sample of consumers randomly taken from the overall dataset. As may be expected, the shape of consumption behavior aligns with morning and evening peaks. At the same time, if we are to differentiate among the patterns, the variation around the mean and median consumption may generate additional insights about consumer behavior.

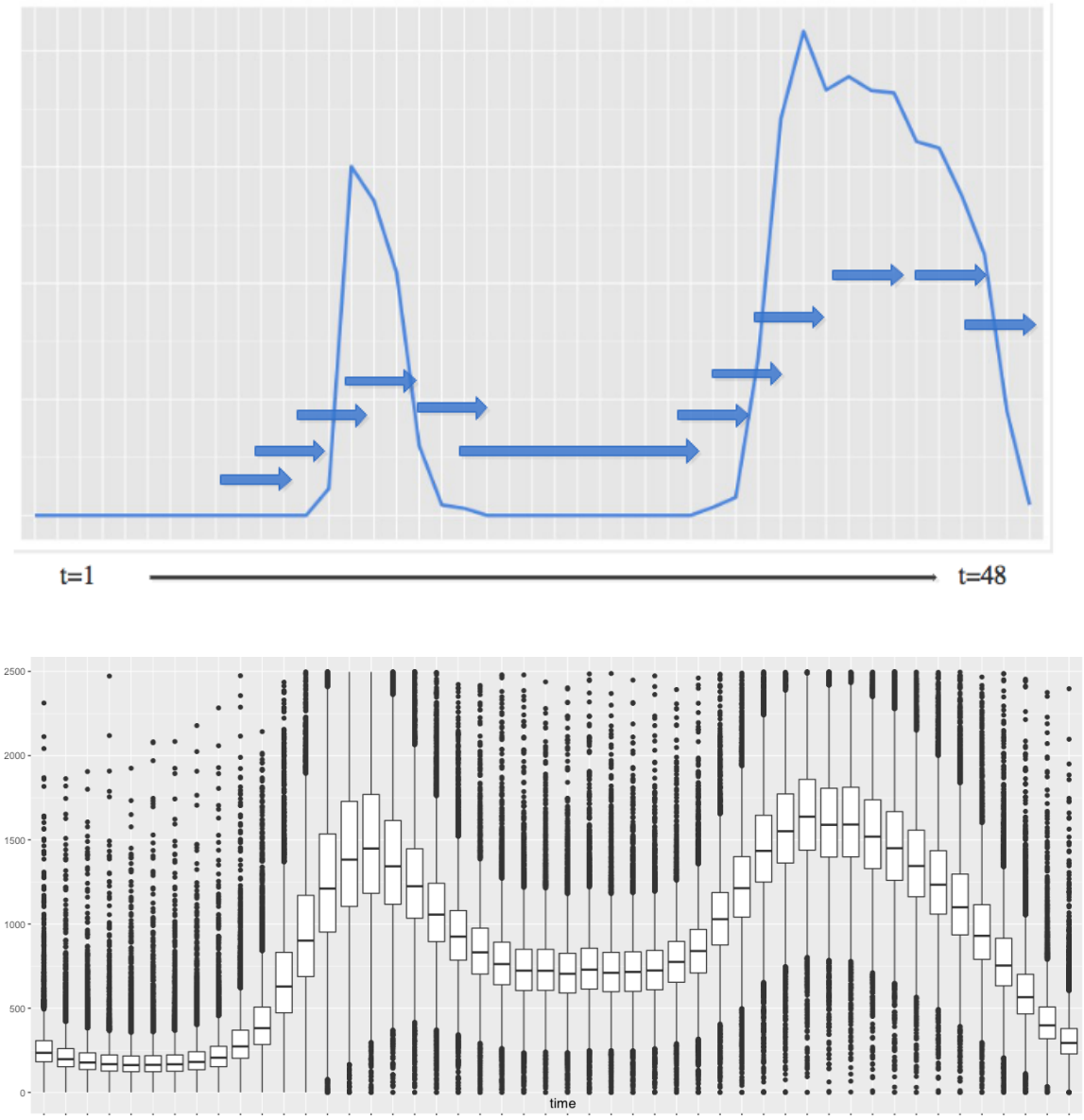

Figure 2: *Decisions on consumption can be made as granular as at each* ***t*** *to more aggregated structures such as evening, morning peak hour and the time intervals in between.*

Methodology

This section presents the workflow of the analysis. It is worth noting that a number of other solutions may be considered at the pre-stage of the method (e.g. feature transformations). However, here variable transformations are avoided in order to preserve the interpretability of the analysis as well, as in this case it is important to ensure that the analysis may be replicated using the raw smart meter data without any modifications applied. The significance of this is

driven by the applicability of the research method within the industry, for instance.

Our approach is based on a combination of both unsupervised and supervised machine learning techniques. First, the process that may have generated the patterns within the data is studied in order to find a way to group these based on the similarity of that process – a process often referred to as clustering. Since the data is unlabelled *a priori*, this step is also useful for segmenting large data sets into groups that can then be studied separately. It is often the case that these clusters may be associated with real-world segmentations in the data.

In the final step we predict assignment to clusters based only on time-series features. This models a setting wherein the researcher may be interested in individual behavior (clusters) without access to any additional information apart from past behavior. We perform this analysis both on the aggregated and disaggregated data streams.

Segmentation and Labelling of Time-Series

As discussed, the dataset represents solely the readings from smart meters and contains no information on individual characteristics of the users and properties. We use unsupervised machine learning techniques to segment the data and create artificial labeling. The next section will assess how well such labels can be predicted from the time-series data. The main goal of this section is to develop a method to read new unseen data and allocate it to a group of already known segments. Clustering is being accessed as a feasible strategy for segmenting large, granular data. Related work has been undertaken in energy classification using smaller and more aggregated samples (Albert and Rajagopal, 2013; McLoughlin, Duffy and Conlon, 2015; Haben, Singleton and Grindrod, 2016).

Clustering

Clustering is an unsupervised machine learning method that is used primarily to associate a simplified underlying structure with unlabelled data. For example, in the smart meter case, having solely energy consumption recordings, little is known as to whether the consumption patterns may be aggregated into similarity groups. For instance, people who work full time may

be grouped together, while those who are at home throughout the day may also be clustered together. The objective is thus to find an algorithm which ensures that similarity between individuals within each cluster is maximized, while also maximizing dissimilarity between clusters.

To date, a number of methods have been developed for clustering data. While many of these give a reliable performance on static data, they often disregard the dynamic structures of clusters. This poses further challenges if we are to consider the spatial and temporal dimensions in the analysis. One of the immediate solutions could be to transform dynamic data into the static format. For example, we may calculate the mean for each of the individuals and create a numerical indicator that represents an estimate of average consumption for the individuals in our sample. This can also be done for geographical references, reducing the dimensionality of the data and allowing for greater generalization. According to Liao (2005), the decision on which clustering method is appropriate for time series also depends on the data type. The characteristics can include: discrete vs real valued, uniformity of the sample, univariate vs multivariate series, and lengths of time series considered for the analysis.

Most clustering algorithms are designed to maximize dissimilarity among the groups using various distance measures (e.g., k-means, hierarchical), while others may consider the underlying data generation process (e.g., Gaussian Mixture Models, Bayesian clustering by dynamics). An important issue for these algorithms is how to treat outliers. For instance, whether outliers are weakly assigned to clusters (with some probability) or are associated strictly with a specific cluster (absolute/hard clustering).

K-means clustering is the most popular approach due to its simplicity and fast minimization of the similarities among the objects within each class centre. It is well suited for data sets with static features. For highly variable temporal variables, the assignment of the cluster may be highly unstable as individuals are likely to be assigned to a different cluster subject to the day and time. As an alternative, we consider a Gaussian Mixture Model (GMM) based on a probabilistic model (**?**). Such a setting brings about the ability to handle diverse types of data, including dealing with missing or unobservable data that may have contributed to variation differences among segmented groups. This is achieved by assigning a probability to a segmented

group membership. Under greater uncertainty about the assignment, additional variables may be introduced or the individual may be treated as an outlier or belonging to an uncertain group. Unlike k-means, it produces stable results and selects the number of clusters using the probability density fit. Clustering results are also replicable and remain the same regardless of how many times we run the algorithm.

Gaussian Mixture Models

Gaussian Mixture Models constitute a probabilistic method for clustering that handles diverse types of data, including dealing with missing data and hierarchical structures. The probabilities for each data point to be in a particular cluster are first assigned and then a cluster is allocated to each point using those probabilistic measures. The mixture is formed using the probabilities obtained from the standard Gaussian representation:

$$P[x|\mu,\Sigma] = \frac{1}{\sqrt{2\pi\det(\Sigma)}} \exp\left\{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right\}, \tag{1}$$

with μ representing the mean vector and Σ being a covariance matrix. A mixture of Gaussians is then represented as the following:

$$P[x] = \sum_{i=1}^{H} P[x|\mu_i,\Sigma_i]P[x \in \text{cluster } i]\,. \tag{2}$$

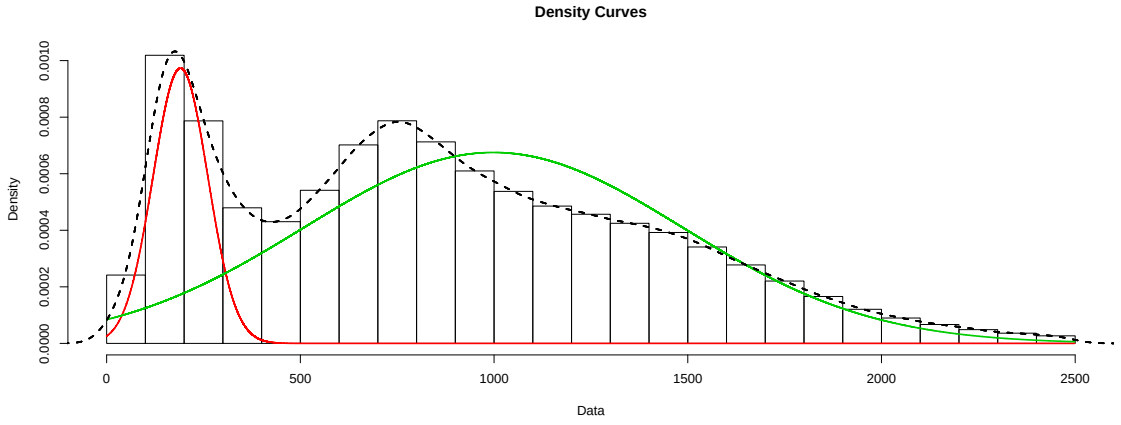

Figure 3: *An example of how the energy consumption density can be represented with the mixture of Gaussian distributions*

As an example, Figure 3 demonstrates how consumption variability can be represented as a mixture of densities. As can be seen, we may represent this data with a mixture of Gaussians, although they may differ in size or shape.[1]

Clustering Results

The results of GMM clustering analysis are presented in Figure 4 and Table 2.

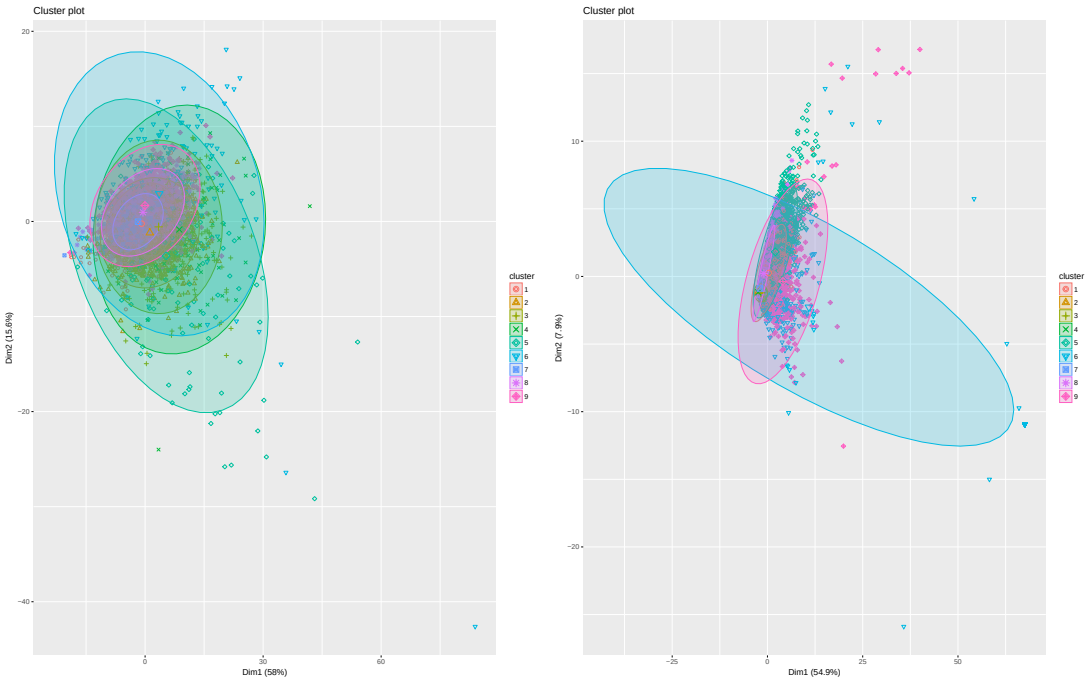

Figure 4: *Resulting clusters in high dimensional space*

As may be observed from Figure 4 and Table 2, while we are dealing with different samples we obtained the same number of clustered groups. However, the key differentiator between the two cluster models is the shape of the Gaussian models used to fit the patterns. While the aggregated sample presents smoother shapes, we see more variation in the disaggregated case (for resulting temporal profiles, see Appendix B).

In terms of sample allocation to each of the clusters, we are presented with an unbalanced allocation. This is caused by the fact that on average, as we saw in Figure 2, energy customers may be alike in their temporal behavior, particularly characterized by morning and evening peaks. In the case of clustering, the less represented groups of patterns are indeed those with

[1]The GMM algorithm is implemented in R in 'mclust' package (Scrucca et al., 2016). For the mixture models we utilize a likelihood based estimation procedure.

lower expected energy consumption, profiles that vary from very low to very high and persistent usage during the day.

Segment	% of total sample (Aggregated patterns)	% of total sample (Disaggregated patterns)
1	24.0%	15.7%
2	10.6%	14.2%
3	5.3%	1.4%
4	0.9%	5.9%
5	1.9%	20.0%
6	21.9%	3.4%
7	15.5%	13.6%
8	14.4%	22.5%
9	5.5%	3.4%

Table 2: *Results of consumption pattern segmentation using GMM.*

Behavior Prediction

A number of approaches can be used for time series prediction and classification. Initially, it was attempted to forecast the next unit of energy consumption in our data using the standard parametric family of models such as ARIMA, AR, and MA. However, performance was extremely poor and for readability it was decided to omit the details of this analysis here. Instead, given the greater variability in big data it is proposed that ability to predict the next half-hour or day of activity may appear troublesome, but as an initial stage the consumer may rather be associated with a class of known or similar users. In the case presented, we use the labels previously obtained from segmentation of the data. Once again the performance for aggregated and disaggregated samples is then compared. The choice of models was based on their popularity in past research, specifically in the multi-class setting.

K-Nearest Neighbor

K-Nearest Neighbor (KNN) is considered one of the simplest classification methods for both binary and multi-class problems. It is particularly useful for problems where the conditional distribution of the outcome variable on the independent variables is unknown (James et al.,

2013). KNN works by taking an input point, x, and K points that are in some sense close to it. The points nearby in the feature space can then be used to select an appropriate label. The estimator can be written mathematically as

$$\widehat{Y}(x) = \frac{1}{K} \sum_{x_i \in N_K(x)} y_i \,, \tag{3}$$

where y_i represents the labels of the points in the neighborhood $N_K(x)$ of input point x.

Tree-based methods

The other methods we assess, Random Forest (RF) and Gradient Boosting Trees (GBM), are based solely on decision tree mechanisms. They are differentiated by the approach use to select the best combination of trees and how samples of data are incorporated in the learning process. These methods are especially valuable due to their simplicity in interpretation compared to other machine learning algorithms. They can easily be used for regression and classification type problems and, additionally, to model non-linear relationships.

The Random Forest algorithm is based on building decision tress on bootstrapped (randomly sub-sampled) data with a smaller subset of randomly sampled predictors at each decision node. A large number of trees is grown until a stopping rule is achieved (e.g. minimum 5 observations in the terminal nodes) and then aggregated for final prediction. An example of the successful use of Random Forest in civil war onset prediction can be found in Muchlinski et al. (2015) and Strobl et al. (2008).

Our implementation of the model is as follows. The input variables are represented by the sequence $\{b...B\}$ which is a combination of half-hourly readings. The model draws bootstrap samples, Z, from the training set, and random forest trees are built using a combination of predictors that are responsible for the split of these trees. Once a number of tree classifiers have been generated, we take the average among all and form a single classifier. Output is represented by $\{T_b\}_1^B$ The class is then predicted for the unseen data (test set) through the majority vote that selects the best performing trees :

$$C_r\widehat{f^B}(x) = majority\ vote\left\{\widehat{C_b(x)}\right\}_1^B \tag{4}$$

An alternative tree algorithm known as Gradient Boosting, first used to tackle classification problems, is now widely used for regression as well Friedman, Hastie and Tibshirani (2001). Like Random Forest, the gradient boosting algorithm takes advantage of both weak and strong classifiers. By weak, we mean classifiers that bring a prediction which is slightly better or just the same as a random guess. Unlike Random Forest where at each iteration we are training a different solution, in the Gradient Boosting model we are updating the solution of the already trained model as more samples are taken. The trees are, therefore, updated at each iteration to obtain more powerful classifiers.

In boosting models, we first assign the weights $w_i = \frac{1}{N}$ to each of our training observations that include both input and output variables, with N being the total number of observations. We then iterate the process F times during which we fit the classifier $G_f(x)$ using the observation weights. The observations which were misclassified at the previous stage are assigned greater weights, so at each iteration we give more importance to those observations that were harder to classify initially. We calculate the error associated with each model fit as

$$e_f = \frac{\sum_{i\varepsilon N_i} w_i I(y_i \neq G_f(x_i))}{\sum_{i\varepsilon N_i} w_i} \tag{5}$$

Those with the highest error are assigned an increase to their weights using the factor of $\exp\gamma_f$. The final output $G(x)$ is based on continuous iterations of model fit using re-weighted observations until the error rate is minimized.

Results

The tables below report overall accuracy and kappa values for each of the models were used to predict the data segment. Entries for 'Accuracy' report the overall prediction power of the model including both true positives and true negatives over total of true and false positives and negatives. The Kappa statistic is used for the evaluation of classifiers by comparing the observed accuracy of prediction with that of a random chance. The optimal parameters were

obtained using ten-fold cross-validation. The results are followed by confusion tables that represent the ratio of observed versus predicted class.

Aggregated Results

Model	Accuracy	Kappa
K-Nearest Neighbor	23%	0.14
Gradient Boosting Trees	37%	0.29
Random Forest	40%	0.29

Table 3: *Results of multi-class prediction on aggregated sample.*

KNN		*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
	1	**13.79%**	3.42%	1.67%	0.00%	0.00%	0.00%	26.85%	5.08%	2.04%
	2	15.17%	**23.29%**	0.00%	0.00%	0.00%	0.00%	9.34%	11.02%	0.00%
	3	11.03%	21.23%	**6.67%**	9.52%	3.45%	10.53%	6.23%	13.56%	10.20%
	4	4.83%	13.01%	40.00%	**47.62%**	34.48%	21.05%	0.00%	7.63%	16.33%
	5	11.03%	14.38%	11.67%	23.81%	**41.38%**	21.05%	5.06%	8.47%	10.20%
	6	6.21%	7.53%	26.67%	19.05%	20.69%	**47.37%**	1.56%	11.02%	42.86%
	7	8.28%	1.37%	0.00%	0.00%	0.00%	0.00%	**30.35%**	0.85%	0.00%
	8	16.55%	5.48%	1.67%	0.00%	0.00%	0.00%	14.79%	**17.80%**	2.04%
	9	13.10%	10.27%	11.67%	0.00%	0.00%	0.00%	5.84%	24.58%	**16.33%**
GBM		*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
	1	**27.03%**	11.54%	1.52%	1.39%	1.32%	1.19%	24.79%	15.38%	2.15%
	2	**12.61%**	**36.54%**	21.21%	4.17%	5.26%	0.00%	5.98%	8.65%	4.30%
	3	**9.01%**	21.15%	**30.30%**	6.94%	10.53%	3.57%	3.42%	9.62%	11.83%
	4	**0.90%**	2.88%	12.12%	**52.78%**	22.37%	19.05%	0.00%	2.88%	7.53%
	5	**1.80%**	10.58%	1.52%	16.67%	**44.74%**	13.10%	2.56%	5.77%	3.23%
	6	**0.00%**	0.00%	13.64%	18.06%	10.53%	**41.67%**	0.00%	2.88%	26.88%
	7	**23.42%**	1.92%	0.00%	0.00%	0.00%	0.00%	**48.72%**	5.77%	2.15%
	8	**18.92%**	6.73%	6.06%	0.00%	0.00%	2.38%	13.68%	**29.81%**	12.90%
	9	**6.31%**	8.65%	13.64%	0.00%	5.26%	19.05%	0.85%	19.23%	**29.03%**
RF		*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
	1	**26.13%**	10.53%	3.17%	0.00%	0.00%	1.33%	27.73%	13.33%	4.12%
	2	11.71%	**43.42%**	22.22%	1.41%	0.06%	0.00%	9.24%	9.17%	5.15%
	3	9.91%	26.32%	**22.22%**	8.45%	12.64%	0.00%	0.84%	12.50%	15.46%
	4	0.90%	2.63%	14.29%	**46.48%**	27.59%	17.33%	0.00%	1.67%	9.28%
	5	4.50%	13.16%	14.29%	21.13%	**44.83%**	0.07%	0.84%	4.17%	4.12%
	6	0.00%	5.26%	7.94%	19.72%	9.20%	**50.67%**	0.00%	5.83%	17.53%
	7	21.62%	1.32%	1.59%	0.00%	0.00%	0.00%	**49.58%**	5.83%	1.03%
	8	18.02%	7.89%	3.17%	0.00%	0.00%	2.67%	10.92%	**30.83%**	13.40%
	9	7.21%	13.16%	11.11%	0.00%	2.30%	21.33%	0.84%	16.67%	**29.90%**

Figure 5: *Confusion matrix reporting the correspondence between predicted (rows) vs observed class (columns).*

Results on disaggregated sample

Model	Accuracy	Kappa
K-Nearest Neighbor	65%	0.58
Gradient Boosting Trees	80%	0.73
Random Forest	79%	0.75

Table 4: *Results of multi class prediction on disaggregated sample*

KNN		1	2	3	4	5	6	7	8	9
	1	**73.00%**	2.00%	15.00%	0.00%	3.00%	1.00%	6.00%	10.00%	0.00%
	2	0.00%	**67.00%**	0.00%	3.00%	0.00%	0.00%	1.00%	0.00%	0.00%
	3	0.00%	2.00%	**60.00%**	0.00%	0.00%	0.00%	2.00%	0.00%	2.00%
	4	0.00%	0.00%	0.00%	**97.00%**	0.00%	0.00%	0.00%	0.00%	0.00%
	5	20.00%	0.00%	42.00%	0.00%	**87.00%**	4.00%	4.00%	15.00%	1.00%
	6	2.00%	0.00%	13.00%	0.00%	2.00%	**64.00%**	2.00%	2.00%	8.00%
	7	0.00%	18.00%	17.00%	0.00%	0.00%	0.00%	**67.00%**	1.00%	0.00%
	8	1.00%	10.00%	29.00%	0.00%	1.00%	0.00%	19.00%	**70.00%**	1.00%
	9	3.00%	0.00%	29.00%	0.00%	6.00%	31.00%	0.00%	2.00%	**88.00%**
GBM		1	2	3	4	5	6	7	8	9
	1	**88.49%**	1.23%	2.24%	1.22%	4.36%	0.00%	3.43%	4.14%	0.24%
	2	0.59%	**84.06%**	0.00%	6.97%	0.04%	0.00%	6.63%	1.56%	0.00%
	3	0.32%	1.28%	**81.34%**	1.51%	0.21%	0.00%	1.11%	0.28%	0.96%
	4	0.00%	0.09%	0.00%	**83.52%**	0.00%	0.00%	0.00%	0.00%	0.00%
	5	6.82%	0.32%	5.22%	0.75%	**90.38%**	4.00%	2.18%	5.28%	0.00%
	6	1.26%	0.00%	5.97%	0.00%	0.88%	**92.22%**	0.05%	0.51%	3.37%
	7	0.09%	8.70%	0.75%	4.05%	0.18%	0.00%	**78.35%**	3.15%	0.00%
	8	1.22%	4.33%	2.24%	1.98%	2.14%	0.22%	8.25%	**84.82%**	0.00%
	9	1.22%	0.00%	2.24%	0.00%	1.83%	3.56%	0.00%	0.26%	**95.42%**
RF		1	2	3	4	5	6	7	8	9
	1	**82.73%**	0.96%	9.62%	0.00%	8.27%	0.00%	4.29%	5.00%	0.35%
	2	0.60%	**84.50%**	0.00%	2.64%	0.00%	0.00%	6.92%	2.24%	0.00%
	3	0.60%	1.33%	**75.00%**	0.00%	0.31%	0.27%	2.54%	1.34%	0.70%
	4	0.00%	0.23%	0.00%	**97.14%**	0.00%	0.00%	0.00%	0.00%	0.00%
	5	8.01%	0.23%	3.85%	0.00%	**83.67%**	6.04%	3.66%	7.67%	1.40%
	6	2.26%	0.23%	5.77%	0.00%	2.27%	**70.60%**	0.79%	1.22%	23.16%
	7	0.64%	9.77%	1.92%	0.22%	0.21%	0.00%	**67.97%**	2.76%	0.00%
	8	3.22%	2.75%	3.85%	0.00%	0.31%	0.27%	13.63%	**79.23%**	0.70%
	9	1.93%	0.00%	0.00%	0.00%	4.96%	22.80%	0.20%	0.54%	**73.68%**

Figure 6: *Confusion matrix reporting the correspondence between predicted (rows) vs observed class (columns).*

As observed from the confusion tables (Figures 5 and 6), the prediction methods show differential performance across clusters. One of the immediate observations is the difference

in performance when considering aggregated versus disaggregated analysis (Tables 3 and 4). Aggregated models are associated with higher misclassification rates, suggesting that by aggregating we have lost essential dynamics that contribute to identifiable patterns.

While RF and GBM tend to perform better on average, KNN showed higher accuracy in some classes. This is possibly related to different 'bias-variance' trade off for each of the tree models. While boosting aims to reduce the bias by taking the average of predictive performance among the estimated models, Random Forest fundamentally searches for a solution that reduces variance by imposing a strict structure of reducing the number of predictors at each split of the tree.

Often, the classes that are better represented in the data may be associated with better performance as there is more data available for the training. In our case, this had no implication on performance. Classes with smaller number of observations were more easily differentiated, while the bigger ones showed higher levels of misclassification.

Discussion and Conclusions

In this paper the analysis that can be performed on time series associated with substantial levels of variability across a large number of data-streams was presented. It was demonstrated that such data can be meaningfully clustered using Gaussian Mixture Models. The paper suggests a possible strategy for prediction and characterization of temporal profiles. One of the arising challenges is the effect of aggregation on prediction performance.

It was shown that both segmentation and predictive algorithms tend to work differently depending on whether we looked at aggregate or disaggregate samples. For prediction in particular, we show that using aggregated data records leads to much higher rates of misclassification, while the most granular data can be classified and predicted with more certainty.

Compared to Random Forest, in practice some classifications may be better performed using Gradient Boosting trees (Friedman, Hastie and Tibshirani, 2001). However, the performance may be at the cost of over-fitting the data. Nevertheless, what is observed is rather a mixture of performances with each method winning or losing for different prediction class.

This may be related to the essential 'bias-variance' trade-off that is worked differently by each model. While boosting aims to reduce the bias by taking the average of predictive performance among the estimated models, Random Forest fundamentally searches for the solution that reduces the variance by imposing a strict structure of reducing the number of predictors at each split of the tree.

In previous research, model stacking has shown an improvement in performance (Rokach, 2010; Dietterich et al., 2000). In future work we will evaluate a mix of the models as an ensemble to improve overall predictive performance.

References

Albert, Adrian and Ram Rajagopal. 2013. "Smart meter driven segmentation: What your consumption says about you." *IEEE Transactions on Power Systems* 28(4):4019–4030.

Anderson, Ben, Sharon Lin, Andy Newing, AbuBakr Bahaj and Patrick James. 2017. "Electricity consumption and household characteristics: Implications for census-taking in a smart metered future." *Computers, Environment and Urban Systems* 63:58 – 67. Spatial analysis with census data: emerging issues and innovative approaches.

Box-Steffensmeier, Janet M., John R. Freeman, Matthew P. Hitt and Jon C. W. Pevehouse. 2014. *Time Series Analysis for the Social Sciences*. Analytical Methods for Social Research Cambridge University Press.

Chappell, Henry W and William R Keech. 1985. "A new view of political accountability for economic performance." *American Political Science Review* 79(1):10–27.

Dietterich, Thomas G et al. 2000. "Ensemble methods in machine learning." *Multiple classifier systems* 1857:1–15.

Friedman, Jerome, Trevor Hastie and Robert Tibshirani. 2001. *The elements of statistical learning*. Vol. 1 Springer series in statistics New York.

Goldstone, Jack A, Robert H Bates, David L Epstein, Ted Robert Gurr, Michael B Lustik, Monty G Marshall, Jay Ulfelder and Mark Woodward. 2010. "A global model for forecasting political instability." *American Journal of Political Science* 54(1):190–208.

Haben, Stephen, Colin Singleton and Peter Grindrod. 2016. "Analysis and clustering of residential customers energy behavioral demand using smart meter data." *IEEE transactions on smart grid* 7(1):136–144.

Hamilton, James Douglas. 1994. *Time series analysis*. Vol. 2 Princeton university press Princeton.

Hegre, Håvard, Joakim Karlsen, Håvard Mokleiv Nygård, Håvard Strand and Henrik Urdal. 2013. "Predicting armed conflict, 2010–2050." *International Studies Quarterly* 57(2):250–270.

James, Gareth, Daniela Witten, Trevor Hastie and Robert Tibshirani. 2013. *An introduction to statistical learning*. Vol. 112 Springer.

Liao, T Warren. 2005. "Clustering of time series data-a survey." *Pattern recognition* 38(11):1857–1874.

McLoughlin, Fintan, Aidan Duffy and Michael Conlon. 2015. "A clustering approach to domestic electricity load profile characterisation using smart metering data." *Applied energy* 141:190–199.

Muchlinski, David, David Siroky, Jingrui He and Matthew Kocher. 2015. "Comparing random forest with logistic regression for predicting class-imbalanced civil war onset data." *Political Analysis* 24(1):87–103.

Rokach, Lior. 2010. "Ensemble-based classifiers." *Artificial Intelligence Review* 33(1):1–39.

Scrucca, Luca, Michael Fop, T Brendan Murphy and Adrian E Raftery. 2016. "mclust 5: Clustering, classification and density estimation using gaussian finite mixture models." *The R Journal* 8(1):289.

Shellman, Stephen M. 2004. "Time series intervals and statistical inference: The effects of temporal aggregation on event data analysis." *Political Analysis* 12(1):97–104.

Strobl, Carolin, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin and Achim Zeileis. 2008. "Conditional variable importance for random forests." *BMC bioinformatics* 9(1):307.

Wilkerson, John and Andreu Casas. 2017. "Large-Scale Computerized Text Analysis in Political Science: Opportunities and Challenges." *Annual Review of Political Science* 20:529–544.

Appendix

Appendix (A) Figure 7 presents the description of the data used for the analysis: aggregated and disaggregated sample.

	Aggregated			Disaggregated		
Time Interval	**Mean**	**Median**	**St.Dev**	**Mean**	**Median**	**St.Dev**
1	259.80	234.92	140.08	181.96	0.00	1178.16
2	220.58	197.30	124.38	164.12	0.00	1145.76
3	199.61	177.38	118.64	159.73	0.00	1140.11
4	188.91	167.45	116.58	156.93	0.00	1143.15
5	183.96	162.23	116.29	160.52	0.00	1143.68
6	184.54	163.47	117.05	158.71	0.00	1145.71
7	187.95	166.85	119.18	166.91	0.00	1182.66
8	203.57	180.67	126.72	180.73	0.00	1184.06
9	229.44	206.03	136.46	214.08	0.00	1238.64
10	306.70	273.04	182.71	314.94	0.00	1400.71
11	421.46	381.55	226.57	467.85	0.00	1626.99
12	686.83	629.86	343.77	694.64	0.00	1913.96
13	976.60	904.71	439.61	953.85	20.00	2175.56
14	1306.30	1226.27	531.64	1146.52	70.00	2286.96
15	1510.45	1419.06	592.26	1216.66	90.00	2271.78
16	1558.24	1481.12	555.39	1474.98	90.00	2150.01
17	1421.02	1358.17	473.94	1337.45	120.00	2020.68
18	1276.81	1227.62	399.74	880.61	90.00	1857.05
19	1093.03	1057.07	328.46	766.19	22.00	1759.55
20	951.90	925.39	277.99	656.50	0.12	1696.31
21	854.64	831.74	252.07	583.86	0.00	1625.30
22	781.08	762.16	233.69	549.54	0.00	1594.56
23	741.01	723.04	224.81	552.25	0.00	1598.88
24	741.73	722.33	227.28	551.11	0.00	1608.77
25	722.07	704.34	218.00	563.18	0.00	1634.49
26	750.96	728.97	230.48	553.34	0.00	1622.94
27	728.22	710.34	217.77	559.58	0.00	1644.10
28	220.58	715.41	212.91	164.12	0.00	1639.50
29	737.52	177.38	210.52	602.01	0.00	1681.72
30	787.08	774.86	116.58	643.80	0.00	1143.15
31	850.69	839.90	232.32	764.25	10.00	1828.98
32	1046.35	1029.30	289.77	930.65	67.00	2006.49
33	1234.72	1214.22	330.11	1171.51	100.00	2204.06
34	1472.50	1441.87	394.22	1285.21	150.00	2252.11
35	1603.22	1563.55	415.40	1369.46	200.00	2299.62
36	1698.90	1654.65	436.31	1314.84	191.00	2195.42
37	1644.06	1603.94	409.87	1316.56	177.00	2204.19
38	1644.44	1603.13	407.36	1241.10	138.50	2147.75
39	1565.43	1526.71	391.76	1176.25	112.00	2079.76
40	1492.84	1455.79	383.25	1083.67	90.00	2007.02
41	1390.42	1349.09	369.52	994.66	89.00	1952.78
42	1278.60	1236.79	353.70	1197.23	67.00	1893.60
43	1144.64	1100.02	334.37	1018.55	90.00	1829.42
44	977.39	929.89	304.52	630.75	23.00	1562.56
45	797.32	753.50	267.70	483.32	0.00	1403.52
46	605.45	565.37	228.64	350.38	0.00	1536.12
47	431.50	398.29	187.90	278.56	0.00	1482.01
48	320.94	293.57	157.03	245.98	0.00	1502.46

Figure 7: *Descriptive statistics for the samples we used for the experiments*

Appendix (B)

Figures 8 and 9 present the shapes and variation in the resulting clusters using the GMM model. As can be seen the number of clusters was defined as identical, however, the shape of aggregated clusters is far more smoother then those of the disaggregated sample. This fundamental difference may have had a direct implication for the predictability of aggregated clusters as the differentiation on the aggregated level may be more challenging as essential dynamics that distinguish the patterns were collapsed during the averaging of energy consumption.

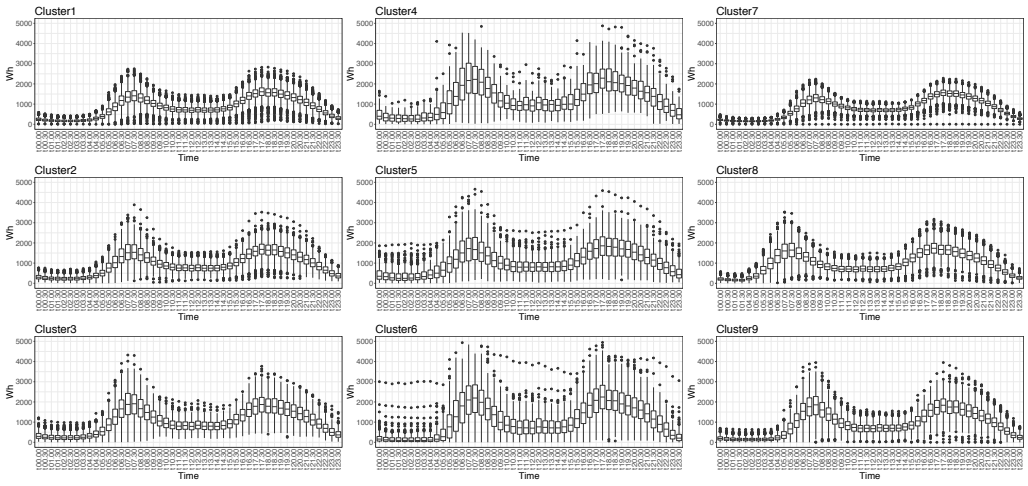

Figure 8: *Clusters observed on aggregated sample*

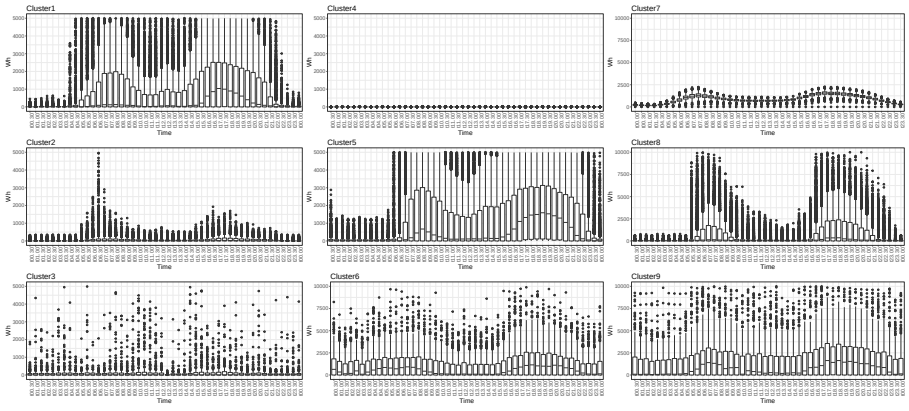

Figure 9: *Clusters observed on disaggregated sample*