Unsupervised Disaggregation of Low Granularity Resource Consumption Time Series

Pantelis Chronis[1,2], Giorgos Giannopoulos[2], Spiros Athanasiou[2], Spiros Skiadopoulos[1]

[1] Department of Informatics and Telecommunications, University of Peloponnese, Tripoli, Greece, chronis@uop.gr .
[2] Athena Research Center, Athens, Greece.

Abstract. Resource consumption is typically monitored at a single point that aggregates all activities of the household in one time series. A key task in resource demand management is disaggregation; an operation that decomposes such a composite time series in the consumption parts that comprise it, thus, extracting detailed information about how and when resources were consumed. Current state-of-the-art disaggregation methods have two drawbacks: (a) they mostly work for frequently sampled time series and (b) they require supervision (that comes in terms of labelled data). In practice, though, sampling is not frequent and labelled data are often not available. With this problem in mind, in this paper, we present a method designed for unsupervised disaggregation of consumption time series of low granularity. Our method utilizes a stochastic model of resource consumption along with empirical findings on consumption types (e.g., average volume) to perform disaggregation. Experiments with real world resource consumption data demonstrate up to 85% Recall in identifying different consumption types.

1 Introduction

Resource conservation, concerning for instance water, energy or fuel, is an important challenge for modern societies. Monitoring and analysing the consumption of resources is a valuable tool in developing resource conservation policies. Analysis of consumption time series includes several tasks, one of which, disaggregation, is the focus of this paper.

Disaggregation is the process of analysing a composite time series into the individual components that it consists of. In the case of resource consumption, the composite time series consists of several discrete *consumption types*. As an example of resource disaggregation we consider a household's water consumption: A household's water consumption is measured at the main supply where the consumption of all the various consumption types (e.g., clothes-washing, showering) is aggregated. The goal of a disaggregation algorithm would be to identify when a shower was taken, when the washing-machine was being used, etc.

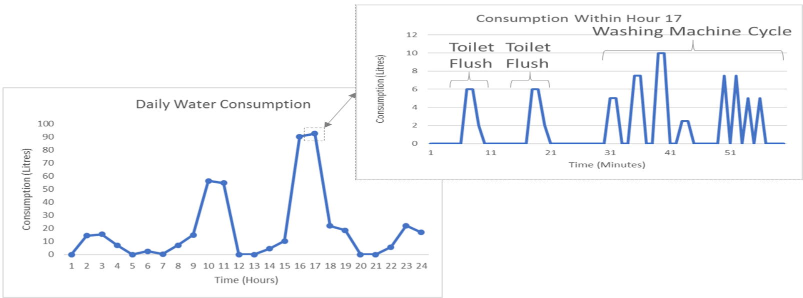


Fig. 1. An example of a group of patterns aggregated within an one hour measurement. In this example two toilet flushes and a washing machine cycle are aggregated within the measurement of 17:00 hours.

An important property that affects disaggregation performance is the relation between consumption type duration and measurement interval length. When the measurement interval is smaller than the expected consumption type duration, disaggregation can be effectively approached as a pattern recognition problem, since there are enough measurements for the pattern of each consumption type to be identified. However, if the measurement interval is equal or larger than the duration of a consumption type, an occurrence of a consumption type can start and finish inside the interval of a single measurement. This means that the pattern of the consumption type is essentially lost. This makes the disaggregation problem much more challenging. An example of this is presented in Figure **??**. Generally, in residential resource consumption, major consumption types have durations ranging from several minutes to 2-3 hours. Thus, in this setting, time series with measurement interval of 15 minutes or larger (e.g., 30 minutes, 1 hour) can be considered of *low granularity*. In practice, resources, especially water, are measured at low granularity. The reasons for this are limitations of the sensors, usually due to battery life, and increased infrastructure costs required for the transmission of high frequency measurements from a very large number of sensors. However, very few works have handled the problem of resource disaggregation in low granularity data.

Another shortcoming of most existing disaggregation algorithms is that they need to be trained on a labelled dataset [**?**,**?**,**?**]. This means they require a dataset with time series of each consumption type measured *separately* and labelled, so that the algorithm can learn to identify its pattern. However, gathering such datasets requires costly and intrusive measurement trials.

All existing algorithms have one or both of the above requirements, which makes them unsuitable for many real world applications. Motivated by this, we present a disaggregation method for low granularity time series that does not require a labelled dataset. Our method is based on a stochastic model of resource consumption that we have developed. Instead of labelled data, it requires approximate assumptions concerning the volume, frequency and usual time of occurrence of each consumption type. These assumptions are simple and intuitive and can be retrieved from the literature [?,?,?,?], provided by experts or be requested from the users. Utilizing those assumptions and the stochastic model, our method calculates the probability that each consumption type has occurred at each time.

In order to thoroughly evaluate our algorithm, we test it in two datasets: one consisting of residential water consumption data, and one consisting of residential energy consumption data, both of hourly granularity. Our algorithm achieves good performance on both datasets.

The rest of the paper is organized as follows: in Section ?? we present related work, in Section ?? we describe and formulate our method and, finally, in Section ?? we present the experimental evaluation.

2 Related Work

Reviewing the literature on time series disaggregation, we can discriminate existing work into two categories. The high granularity algorithms, that are designed for data with measurements intervals ranging from milliseconds up to one minute and the low granularity algorithms, that can be applied to data with measurement intervals from several minutes to several hours. Our distinction between low and high granularity is based on the relation of the measurement interval to the average consumption type duration, as we described in Section ??.

In the high granularity setting [?,?,?,?,?,?,?], existing works usually scan the time series to identify significant *step changes* in consumption that indicate the start or the end of a specific consumption event. Then, using a Machine Learning model and labelled data, they identify the consumption types each consumption event. Existing methods mainly vary in the adopted Machine Learning model. In [?], the authors use a Hidden Markov Model (HMM) to classify the events. [?] use a convex optimization approach, similar to Support Vector Machines (SVM), and [?] use a Neural Network. There also exist a few unsupervised approaches in the setting of high granularity data. In [?], the authors model consumption using an extension of the HMM. Instead of labelled data, they use detailed assumptions about each device's consumption pattern and usage. They evaluate their algorithm on data with granularity of 3 seconds. In the same line of work, [?] use a version of HMM to perform disaggregation on time series with 1 minute granularity. The algorithm does not require labelled data, however, it requires information concerning each device's exact consumption pattern. We note that HMM models are succesful in high granularity time series, where the transitions between different operating stages of an appliance are detectable in the time series pattern. On the contrary, those transitions are, generally, not detectable in low granularity data.

In the setting of low granularity data [?,?], the general approach is to model the aggregate time series as a sum of separate components, that represent the various consumption types, and apply an optimization algorithm in order to decompose the time series into those components. In [?], the authors apply Sparse Coding, a model that allows the combination of a large number of basis functions

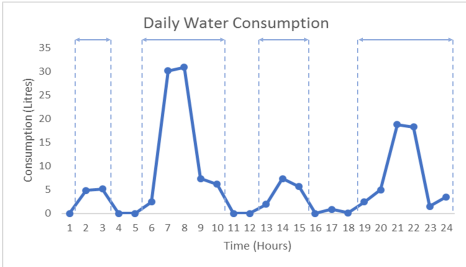


Fig. 2. A time series of the water consumption of a day, for a single household, divided in its major consumption events.

by imposing sparsity constraints, on low granularity (15 min) time series. [**?**] use the same idea as [**?**], but modify the algorithm to work iteratively, disaggregating only one consumption type in each iteration. These approaches also require a set of labelled time series for the various consumption types, which they use as basis functions.

3 Model Formulation and Disaggregation Algorithms

In this section, we present our method. We start by providing an intuitive overview and, then, we describe the details of the method.

We can obtain an intuitive understanding of the challenge by looking at an example. Given a consumption time series like the one depicted in Figure **??**, we aim at identifying the occurrences of each consumption type in time. There are two sources of information about the occurrence of a consumption type: (i) The *footprint* that the occurrence leaves on the aggregate time series. For example, in a water consumption time series, if a shower is taken at some time, we would expect to observe consumption of around 50 Litres or more at that time; (ii) The external information we have about each consumption type: how frequently it occurs and at which hours within a day.

Our disaggregation approach is based on developing a model for the consumption behaviour of the household that incorporates the aforementioned sources of information and which we can use to infer the occurence of the various consumption types. In order for disaggregation to be performed effectively, the model needs to capture the structure of the problem, use only available information and handle the variabillity of human behaviour. The model is based on the assumption that there exists a set of major *consumption types*, each of which is represented by an amount of consumption, an expected time and a frequency of occurrence. The model is stochastic in nature since: (i) These quantities may randomly vary between different occurences of the consumption type and between different households; (ii) There is inherent uncertainty in our estimates of these quantities for each household. The model and the disaggregation process are described in detail in the next subsection.

3.1 Method Description

Events identification. The first issue that arises is that an occurence of a consumption type may be divided in more than one measurements. For example, if an activity started at 09:50 and finished at 10:10, its consumption would be distributed in two consecutive measurements. However, if we select a part of the time series that starts and ends with (near) zero consumption, given the assumption that no consumption type can have a pause of one hour or greater, we can be certain that all consumption types that started inside this interval have also ended, i.e., this interval comprises only complete consumption types. We refer to those parts of the time series, that start and end at *near zero* consumption, as *consumption events* or just *events*. Figure **??** shows an example of a day's consumption events.

Thus, our first step is to identify the distinct consumption events. To achieve this, we sequentially scan the time series and isolate the sequences of all *consecutive* points whose value exceeds a threshold θ_e, above which the consumption is considered significant. Threshold θ_e is set using the assumptions about the volume of each consumption type, so that it is only exceeded if some consumption type is occuring. We denote as $t_{j_{start}}, t_{j_{end}}$ the times of start and end of consumption event j.

Model description. We consider a set of n major consumption types. For each consumption type $i, 1 \leq i \leq n$, we assume that its total consumption c_i is distributed according to a normal distribution with mean μ_i and standard deviation σ_i. We treat all minor consumption types as *background* noise and model them using variable b with mean μ_b and standard deviation σ_b. We denote the probability of consumption type i occurring at time h as τ_{ih}. In order to limit the computational complexity of the model, we make the assumption that each occurrence of a consumption type is only affected by other occurrences of the same consumption type within a specified *time period*. For most cases, this period would be a *day* or *a week* (e.g., people tend to shower once a day or use the washing machine two or three times a week). We denote as v_{ik} the probability that consumption type i occurs k times in the duration of the predefined time period. We also define K as the maximum value of k. For simplicity, we do not include in the model any dependencies between the total number of occurrences of different consumption types, i.e., v_{ik} is independent of $v_{lk}, l \neq i$.

Given a set of m consumption events, each event $j, 1 \leq j \leq m$ is represented by the following: the time it started $t_{j_{start}}$, the time it ended $t_{j_{end}}$ and its total consumption s_j. We denote as It_j a vector containing both $t_{j_{start}}$ and $t_{j_{end}}$, which defines the time interval of event j. The total consumption of event j is the result of the aggregation of each major consumption type i occurring x_{ij} times, plus the background noise:

$$s_j = \sum_{i=1}^{n} c_i x_{ij} + b \tag{1}$$

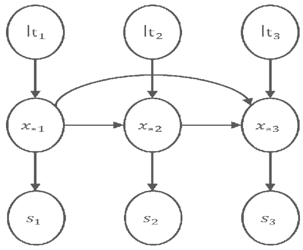


Fig. 3. The Bayesian Network that describes the dependencies within a period comprised of three consumption events. Each consumption type occurence depends on the observed consumption, the time of day and the previous occurences in the given period.

with $x_{ij} \in \mathbb{N}$. As x_{*j} we denote a vector $[x_{1j}, \cdots, x_{mj}]$, that contains the number of occurrences of every consumption types in the interval of event j. All notation is gathered in Table **??** for convenience.

The probability of occurrence of consumption types x_{*j} in an event j depends on the total consumption of the event s_j, the time of the event It_j and all other occurrences of consumptions of the same type x_{*l}, inside the time period. For further analysis, it is convenient to formulate these dependencies into a Bayesian Network (BN). For example, the BN of Figure **??** illustrates a period containing three events. More events may be handled in a similar fashion. We note that the directions of the arrows show the order of decomposition of the joint probability, however the dependencies between the variables are bidirectional [**?**].

The purpose of the disaggregation algorithm is to infer the values of $x_{*j}, 1 \leq j \leq m$. Since we consider each period to be independent from the others, we can treat each period separately. We assume that period d has m_d consumption events $j, 1 \leq j \leq m_d$. We denote as $\cap_j x_{*j}$ the joint possibility of all occurrences $x_{*j}, 1 \leq j \leq m_d$, i.e the possibility of occurrences $x_{*1},\ x_{*2}, \cdots,\ x_{*m_d}$ happening in the same period. $\cap_j x_{*j}$ contains the occurences of all consumption types in all the events of a period. $\cap_j s_j$ and $\cap_j It_j$ are defined in the same way.

The probability of $\cap_j x_{*j}$ can be written as:

$$p(\cap_j x_{*j} \mid \cap_j It_j,\ \cap_j s_j) = \frac{p(\bigcap_j It_j,\ \bigcap_j s_j \mid \bigcap_j x_{*j}) p(\bigcap_j x_{*j})}{p(\bigcap_j It_j,\ \bigcap_j s_j)} \tag{2}$$

Given the dependencies modelled by the BN, Equation (**??**) is transformed to:

$i, 1 \leq i \leq n$	index of consumption type	μ_i, σ_i	mean and stand. dev. of c_i
$j, 1 \leq m \leq m$	index of consumption event	b	volume of background noise
s_j	total consumption of event j	b_μ, b_σ	mean and stand. dev. of b
x_{ij}	occurences of type i in event j	It_j	interval $(t_{j_{start}}, t_{j_{end}})$ of event j
x_{*j}	all occurences within event j	τ_{ih}	prob. of cons. type i at hour h
c_i	consumption volume of type i	v_{ik}	prob. of type i occurring k times

Table 1. Notation

$$p(\cap_j x_{*j} \mid \cap_j It_j,\ \cap_j s_j) = \frac{p(\bigcap_j x_{*j} \mid \bigcap_j It_j) \prod_j p(s_j \mid x_{*j})}{p(\bigcap_j s_j \mid \bigcap_j It_j)} \tag{3}$$

From Equation (**??**) we have:

$$p(s_j \mid x_{*j}) = Normal(\sum_{i=1}^{n} \mu_i * x_{ij} + \mu_b, \sum_{i=1}^{n} \sigma_i^2 * x_{ij} + \sigma_b^2) \tag{4}$$

as s_j is a sum of normally distributed variables. The term $p(s_j|x_{*j})$ models the probability of observing consumption s_j, given that consumption types x_{*j} have occurred.

The term $p(\cap_j x_{*j}| \cap_j It_j)$ corresponds to the prior probability of the consumption types occurring at the specific times $\cap_j It_j$, irrespective of the observed $\cap_j s_j$. It depends on the probability that the activities defined by $\cap_j x_{*j}$ occur all in a single period and that the occurrences are distributed accordingly in the time of the observed consumption events. We model it using the assumptions about frequency and time of occurrence of each consumption type:

$$p(\cap_j x_{*j} \mid \cap_j It_j) = \prod_{i=1}^{n} v_{ik_i} * Multinomial(x_{ij}\ \forall j,\ \pi_{ij}\ \forall j) \tag{5}$$

As defined, the term v_{ik_i} is the probability of consumption type i occurring k times overall. The multinomial distribution models the probability for the consumption types to occur at the specific intervals of the consumption events j. We can break the joint probability into a product because we have assumed that the occurrence of the different consumption types are independent. In Equation (**??**), k_i is the total number of occurrences of consumption type i and π_{ij} is the probability of consumption type i occurring in the interval defined by It_j :

$$k_i = \sum_{j=1}^{m_d} x_{ij},\ \pi_{ij} = \sum_{h=t_{j_{start}}}^{t_{j_{end}}} \tau_{ih} \tag{6}$$

Finally, the denominator of Equation (**??**) is the sum of the probability of all possible joint events $\cap_j x_{*j}$ and is constant for all x_{*j}:

$$p(\cap_j s_j \mid \cap_j It_j) = \sum_{\cap_j x_{*j}} p(\cap_j x_{*j} \mid \cap_j It_j) \prod_j p(s_j \mid x_{*j}) \tag{7}$$

In order to perform the disaggregation, we need to find $\cap_j x_{*j}$ that maximizes the probability of Equation (**??**):

$$argmax_{\cap_j x_{*j}} p(\cap_j x_{*j} \mid \cap_j It_j,\ \cap_j s_j) \tag{8}$$

The above maximization problem is too complex to solve exhaustively, since the number of possible combinations grows exponentially. For example, in a case with five events, and five activites that can occur up to five times, there are more than 10^{17} combinations. In order to find a solution for Equation (**??**),

Algorithm: GreedyApproximation

Input : $It_j, s_j, 1 \leq j \leq m$
Output : $\cap_j x_{*j}$ *of maximum probability*

```
for j = 1 to m_d do
    x_{*j} = EventOfMaxProbability(It_j, s_j, x_{*1}, ..., x_{*j-1})
return ∩_j x_{*j}
```

we implement two different algorithms: a greedy approximation and a Markov Chain Monte Carlo simulation. Next we describe each method.

1. Greedy Approximation. We start by calculating the most probable occurrences for the first event, ignoring all next events. Then we incrementally calculate the most probable occurrences for following events, given the occurrences of all previously calculated ones. This process is described in Algorithm GreedyApproximation.

The probability of occurences given the previous events are:

$$p(x_{*j} \mid It_j,\ s_j,\ \cap_{l=1}^{j-1} x_{*l}) = \frac{p(s_j \mid x_{*j}) \cdot p(x_{*j} \mid It_j,\ \bigcap_{l=1}^{j-1} x_{*l})}{p(s_j \mid It_j,\ \bigcap_{l=1}^{j-1} x_{*l})} \tag{9}$$

where $p(s_j|x_{*j})$ is calculated as in Equation (**??**). The term $p(x_{*j}|It_j, \cap_{l=1}^{j-1} x_{*l})$, is only conditioned on the occurrences of the previous events $\cap_{l=1}^{j-1} x_{*l}$. Since only one event is examined at each step, the binomial distribution is used instead of the multinomial, to calculate the probability of occurrence of each consumption type in the specific time of the event:

$$p(x_{*j} \mid It_j,\ \cap_{l=1}^{j-1} x_{*l}) = \prod_{i=1}^{n} \sum_{k=k_i}^{K} v_{ik} * Binomial(x_{ij},\ k,\ \pi_{ij}), \quad k_i = \sum_{l=1}^{j} x_{il} \tag{10}$$

For each x_{*j}, we exhaustively search all its possible values and select the one with the maximum probabillity (Algorithm EventOfMaxProbabillity).

2. Markov Chain Monte Carlo (MCMC). MCMC is a set of algorithms used to sample from the joint probability distribution described by a Bayesian Network. Given a set of observed variables ($\cap_j s_j, \cap_j It_j$ in our case), we want to sample from the distribution of the unobserved variables ($\cap_j x_{*j}$), in order to find their most probable values. To achieve this we apply the Gibbs sampling algorithm, which sets the observed variables to their observed values, randomly initialises the unobserved variables and sequentially updates the value of each unobserved variable with a value sampled from its conditional distribution, conditioned on all other variables. The conditional distribution of x_{*j} is:

Algorithm: EventOfMaxProbabillity

Input : $It_j, s_j, x_{*1}, \ldots, x_{*j-1}$
Output : x_{*j} *of maximum probability*

$x_{max} = \emptyset$, $p_{max} = 0$
for *every possible value of* x_{*j} **do**
  $p = p(x_{*j} | It_j, s_j, \cap_{l=1}^{j-1} x_{*l})$
  if $p > p_{max}$ **then**
    $p_{max} = p, \ \ x_{max} = x_{*j}$
return x_{max}

$$p(x_{*j} | \cap_j It_j, \cap_j s_j, \cap_{l=1}^{m_d, l \neq j} x_{*l}) = \frac{p(s_j | x_{*j}) p(x_{*j} | \bigcap_j It_j, \bigcap_{l=1}^{m_d, l \neq j} x_{*l})}{p(\bigcap_j s_j | \bigcap_j It_j, \bigcap_{l=1}^{m_d, l \neq j} x_{*l})} \tag{11}$$

The terms of Equation (**??**) can be derived in a straightforward way from Equations (**??**),(**??**). Due to lack of space we skip those derivations. After many iterations of this process, the sampled values follow the joint probability of the Bayesian Network. Based on the obtained samples we find the most probable value for $\cap_j x_{*j}$.

4 Evaluation

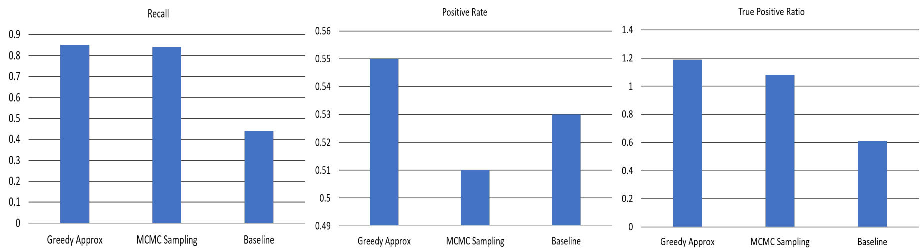
4.1 Baseline

To the best of our knowledge our work is the first to handle the problem of unsupervised disaggregation in time series of low granularity (>1 minute). To obtain some comparative results, we device a baseline method that uses clustering similarly to [**?**]. Each consumption event is represented as a triple, containing the starting time, the total consumption and the total length of the event. The k-means algorithm is used to cluster the consumption events and the known instances of all consumption types are assigned to the clusters accordingly. Then, in order to perform disaggregation of an event, we find its closest cluster and take the most probable consumption types of the cluster. For the water consumption dataset, where, as we explain in Section **??**, the negative events are not known with certainty, we select as representing of showering behaviour a set of clusters that has a total number of events close to the known total number of showers.

4.2 Water Consumption Dataset

The water cconsumption dataset was gathered from a real world trial performed in the context of DAIAD[3] project, that addresses the issue of water sustainability through the use of Information Technology. The dataset consists of water

[3] `http://daiad.eu/`



(a) (b) (c)

Fig. 4. The performance of all algorithms in terms of (a) Recall, (b) Positive Rate and (c) True Positive Ratio, on the water dataset.

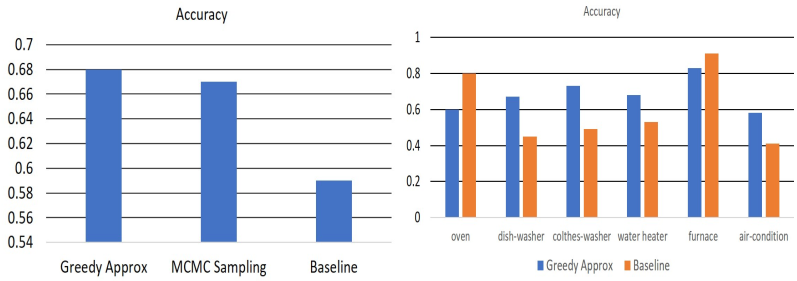
consumption data for 17 households, measured hourly. Also, for each household, there are measurements that contain starting, ending time and total consumption of numerous shower occurrences. Due to the real-world conditions of the experiment, the time of occurrence of a significant portion of the showers is actually unknown. Thus, we cannot state with certainty that at a given day and time a shower was not taken. This means that, while we can directly measure the recall of identifying the showers, we are unable to directly measure the accuracy of the algorithm. However, we have knowledge of the total number of showers, which we use to compensate for the latter. We achieve that by using appropriate evaluation metrics, which we describe next.

The first metric we use is *Recall* (RC), which measures how many of the known occurrences are retrieved by the algorithm. In order to evaluate if the algorithm is overly biased towards positive classification, we use the following two metrics: *Total Positive Ratio* (TPR) and *Positive Rate* (PR). Let A be the total number of showers that the algorithm predicts, B the total number of showers that have actually occurred and C the total number of consumption events. Then TPR and PR are defined as:

$$TPR = \frac{A}{B}, \quad PR = \frac{A}{C} \tag{12}$$

Finally, we use the *Average Length of an Event* (AEL), in hours, to evaluate the precision of the disggregation in time. The results are presented next.

As we can see in Figure **??**, our proposed methods achieve very good RC, with acceptable values of TPR and PR. We note that the optimal value for TPR is 1.0. For PR, values significantly different than 0 and 1 indicate a non-trivial behaviour. The greedy approximation algorithm achieves the highest RC (0.85). From the PR metric, we can see that such high RC is achieved without classifying excessively many instances as positive. The TPR metric shows that the greedy approximation overestimates the total number of showers by an acceptable factor of 20%. The MCMC algorithm is balanced in both metrics (0.84 RC, 1.08 TPR). The baseline method does not perform well in terms of RC, while it also severely



(a) (b)

Fig. 5. The performance of each algorithm on the energy consumption dataset: (a) The average Accuracy (b) The average Accuracy per consumption type.

underestimates the total number of showers. The fact that the baseline's PR is similar to that of the other methods, while its TPR is much lower, is because the algorithm does not predict multiple showers in the same event. Due to that, we experimented with a modified version of the baseline, which predicted two showers in each event that it classifies as positive. This did not improve RC while it severely increased TPR. Finally, the AEL value for all algorithms is 2.71 hours, which means that each occurrence of the consumption type is specified within a window of 2.71 hours, by average. It is the same for all algorithms because they share a common consumption events identification step.

4.3 Enery Consumption Dataset

For energy consumption, we use the Reference Energy Disaggregation Dataset (REDD) [**?**]. REDD contains separate consumption time series for several appliances, as well as the aggregate power consumption, for 6 households. The interval of measurement is 1 second. Since we are intereset in lower granularity datasets, we downsample the data to 1 hour. In this dataset, both positive and negative labels are available, thus we can use the Accuracy (ACC) measure. We calculate the AEL measure for this dataset as well.

In Figure **??**, we see the performance of the algorithms on the energy consumption dataset. We can see that our algorithm achieves relatively high ACC (0.68) and outperforms the baseline (0.59). Figure **??** presents the performance on each consumption type separately. We can see that our algorithm achieves high ACC and outperforms the baseline in four consumption types (clothes-washer, dish-washer and air-conditioner,). On the other hand, the baseline performs better in two consumption types (oven and furnace). The most likely explanation for this is that our assumptions were not sufficiently accurate for those particular consumption types, thus comprising a subject for further investigation, in future work.

5 Conclusion

In this paper, we presented a novel method for resource consumption disaggregation, that works effectively on low granularity data (e.g., 1 hour). Our method does not have the demanding requirement for labelled observations, which are hard to obtain. To our knowledge, our algorithm is the first that addresses the disaggregation problem under those constraints. This is particularly important in real world settings, especially for water consumption, where high frequency and labelled data are rarely available. We evaluated our algorithm in two residential consumption datasets and showed that it achieves high performance (up to 85% Recall) in identifying consumption types. Thus, our algorithm constitutes an effective solution for analysing resource consumption in real world settings.

References

1. J. Zico Kolter, Siddarth Batra, Andrew Y. Ng. *Energy Disaggregation via Discriminative Sparse Coding.* NIPS, 2010
2. O. Parson, S. Ghosh, M. Weal, A. Rogers. *Non-intrusive Load Monitoring using Prior Models of General Appliance Types.* AAAI, 2012.
3. H. Kim, M. Marwah, M. Arlitt, G. Lyon, J. Han. *Unsupervised Disaggregation of Low Frequency Power Measurements.* SDM, 2011.
4. Bradley Ellert, Stephen Makonin, Fred Popowich. *Appliance Water Disaggregation via Non-Intrusive Load Monitoring (NILM).* BigDASC, 2015.
5. Graham Cole, Rodney A. Stewart. *Smart meter enabled disaggregation of urban peak water demand: precursor to effective urban water planning.* UWJ, 2013.
6. Haili Dong, Bingsheng Wang, Chang-Tien Lu. *Deep Sparse Coding Based Recursive Disaggregation Model for Water Conservation.* JCAI, 2013
7. Ilya Kozlovskiy. *Non-intrusive disaggregation of water consumption data in a residential household.* IIWHEC Workshop, 2016.
8. Mario Vasak, Goran Banjac, Hrvoje Novak. *Water use disaggregation based on classification of feature vectors extracted from smart meter data.* CCWI, 2015.
9. K.A. Nguyen, R.A. Stewart, H. Zhang. *An intelligent pattern recognition model to automate the categorisation of residential water end-use events.* EMS, 2013.
10. William B. DeOreo, James P. Heaney, Peter W. Mayer. *Flow trace analysis to assess water use.* AWWA, 1996.
11. Jack Kelly, William Knottenbelt. *Neural NILM: Deep Neural Networks Applied to Energy Disaggregation.* ICESEBE, 2015.
12. Hunter Water Utility, Australia. *Water Calculator.* Retrieved from: http://www.hunterwater.com.au/Save-Water/Water-Usage-Calculator.aspx
13. P. Mayer et. al. *Residential End Uses of Water.* AWWA , 1999.
14. Jean-Paul Zimmermann et. al. *Household Electricity Survey A study of domestic electrical product usage.*
15. Energy.gov. *Energy Calculator.* Retrieved from: https://www.energy.gov /energysaver/estimating-appliance-and-home-electronic-energy-use
16. J. Froehlich et. al. *Disaggregated End-Use Energy Sensing for the Smart Grid.* IEEE Pervasive Computing, 2011.
17. C. M. Bishop *Pattern Recognition and Machine Learning. Chapter 8: Graphical Models.* Springer, 2006.
18. J. Zico Kolter, Matthew J. Johnson. *REDD: A public data set for energy disaggregation research.* SustKDD workshop, 2011.