Appliance Water Disaggregation via Non-intrusive Load Monitoring (NILM)

Bradley Ellert(✉), Stephen Makonin, and Fred Popowich

Simon Fraser University, Burnaby, BC V5A 1S6, Canada
{bellert,smakonin,popowich}@sfu.ca

Abstract. The world's fresh water supply is rapidly dwindling. Informing homeowners of their water use patterns can help them reduce consumption. Today's smart meters only show a whole house's water consumption over time. People need to be able to see where they are using water most to be able to change their habits. We are the first to present work where appliance water consumption is non-intrusively disaggregated using the results from a non-intrusive load monitoring algorithm. Unlike previous works that require the installation of water sub-meters or water sensors, our method does not. Further, our method uses low-frequency data from standardized meters and does not rely on labelled data. We modify the Viterbi Algorithm to apply a supervised method to an unsupervised disaggregation problem. We are able to achieve very high accuracy results having mean squared errors of under $0.02\,\mathrm{L}^2/\mathrm{min}^2$.

Keywords: Water disaggregation · Water conservation · Non-intrusive load monitoring · NILM · Smart homes · Sustainability

1 Introduction

Globally we have become concerned with the cost of consuming energy and the lack of supply. Many studies have emerged that investigate ways to reduce consumption. Computational methods such as non-intrusive load monitoring (NILM) allow us to understand how appliances consume power [5]. Cities are increasingly becoming concerned with fresh water, where demand is also exceeding supply. Soon every house will have a water meter, if they do not already. We have the means to disaggregate appliance power usage using NILM and we demonstrate a computational method to take those disaggregation results to essentially disaggregate the water usage of appliances. This allows homeowners to understand how much water is consumed by appliance use and by human use. It is the human consumption (e.g. showering, bathing, lawn watering) that can be targeted for conservation and motivate behaviour change in our societies.

Household water consumption can be viewed as a hierarchy (Fig. 1). In the broadest sense, it can be broken down based on the agent causing the consumption: human use, appliance use, and leaks. Human use refers to fixtures that can be used for variable lengths of time with varying flow rates, such as showers and

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2016
A. Leon-Garcia et al. (Eds.): Smart City 2015, LNICST 166, pp. 455–467, 2016.
DOI: 10.1007/978-3-319-33681-7_38

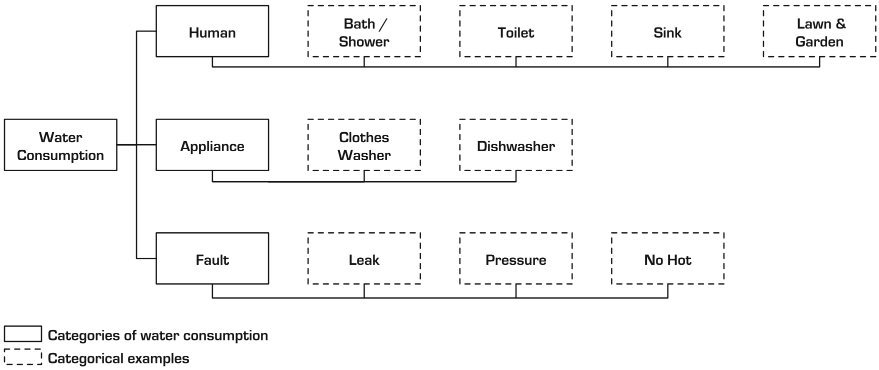

Fig. 1. Within the three sub-categories of household water consumption for households there are many examples of how water is consumed. We focus on appliances that consume water. By subtracting appliance consumption, we can inform homeowners of the amount of consumption that can be changed due to behaviour or habit.

sinks. Appliance use refers to machines that follow cycles with fixed patterns after being initiated, such as (clothes) washing machines and dishwashers. Our work focuses on finding events of this latter variety, as they can be correlated with energy consumption. Certain household water-users can be more difficult to classify. For example, toilets could be considered *appliances* as they are constrained to consuming fixed amounts of water. However based on our definition of correlating with electricity consumption, toilets fall under human use.

In this paper, we present an algorithm for disaggregating appliance water use from main water meter data given disaggregated electricity data. This allows low-frequency, unlabelled data to be used to build a model specific to each household. Domain knowledge about the general consumption patterns is not needed, as this information is learned from smart meter data. By not relying on generic ontological information, the system does not make incorrect assumptions about different makes and models of appliances and is able to be used on future appliances that may follow unforeseen patterns. This represents the first water disaggregator that has the potential of providing homeowners with detailed appliance water use information outside of a controlled setting without the need for tuning by a trained professional or the installation of non-standard metering equipment. By developing this technology alongside NILM, we open the door to a symbiotic relationship between these common related tasks, instead of relying on more obscure sensor data.

2 Previous Work

Household water disaggregation is still quite a young topic, and therefore relatively few papers have been published on the topic. Table 1 summarizes the previous work in the field of household water disaggregation. We discuss each in detail below.

Table 1. The major previous works in household water disaggregation.

Name	Group	Dates	Contribution	Measurement
Trace wizard	Aquacraft	1996 – 2004	Seminal papers	Flow rate
NAWMS	UCLA	2008	Flow rate estimate	Vibration
WaterNILM	MIT	2014	Single sensor	Vibration
HydroSense	Washington	2009 – 2014	Consumer feedback	Pressure
WaterSense	U.Va	2011 – 2013	Motion detectors	Flow rate
DSCRDM	Virginia Tech	2011 – 2013	Low sample rate	Flow rate

The first true method for water disaggregation was developed by Aquacraft, a water management company based in Boulder, Colorado [1] which performed flow trace analysis using a largely manual process. A flow trace is simply a plot of water flow rate over a period of time. This information was collected using a retrofit device that attaches to a water meter and logs flow rate every 10 s by measuring changes in magnetic field. Signature traces were first collected for each fixture and appliance in each house followed by technicians who manually labelled future examples by hand. Eventually a set of heuristics was developed to automatically categorize based on flow rate and duration. Dishwashers and washing machines were noted as difficult to disaggregate because they often co-occurred with miscellaneous faucet use. Flow trace analysis is limited to determining the fixture class (e.g. toilet, sink) of an instance of water use. Recently, the trend has been to focus on finer grained disaggregation down to the level of specific fixture (e.g. kitchen sink, bathroom sink).

NAWMS (Nonintrusive Autonomous Water Monitoring System) is the first method to demonstrate disaggregation down to the level of individual pipes [6]. The authors show that flow rate can be estimated based on pipe vibration readings collected at a frequency of 100 Hz using individual accelerometers. The flow rate of the water main was monitored to fit a model that used a cubic root curve. Although the error rate was just over 1 %, these results only serve as a proof of concept, as their approach was not able to scale up, requiring dozens of sensors in a realistic setting.

More recently, WaterNILM (Non-Intrusive Load Monitoring) applied a similar physical architecture to a real-world setting [12]. The authors developed a technique requiring only two accelerometers (12 kHz–16 kHz) for an entire house. One was installed downstream from the water meter and the other at the outlet of the hot water tank. These readings are down-sampled to 4 kHz and the stream is segmented into 0.75 s chunks. Models were built using labelled data to create clusters. In the best case, they were able to achieve a misclassification rate under 2 %. However, addressing simultaneous water use requires training examples of each possible combination in question. Further, true disaggregation is not performed, as only the combined labels were identified as the source.

HydroSense introduced a simple single point sensor for whole house water pressure that took readings at a rate of 1 kHz [4]. This sensor was attached to an unused outside tap to infer approximate flow rate. In each home hand-labelled data was collected. Baseline static water pressure was measured and pressure signatures were taken for each valve (hot and cold) on each fixture using their proprietary HydroSense unit. Valve events were classified as open or close based on the change in pressure or the average derivative if the pressure did not exceed a fixed threshold. Valve events were associated with individual fixtures by their similarity to other events in the same home via a trained classifier. Average home error rates were reported to be around 5 %. However, this method was very sensitive to the location where the pressure sensor was installed. This process required the installation of a pressure sensor and labelled training data to train the classifier – a very expensive procedure.

Other than the original flow trace analysis, the previously mentioned methods relied on sensors that monitored a house's plumbing. WaterSense instead utilizes data captured by motion sensors to help with water disaggregation [13]. Unlike the previous approaches where supervised learning was required, WaterSense claimed to be unsupervised. The house's water main is monitored at a frequency of 2 Hz and motion detectors were read once every 7 s. Motion sensors were installed in three rooms with water fixtures (two bathrooms and kitchen). Water meter samples discovered using edge detection were clustered into rooms based on the temporal proximity of motion sensor readings. Toilets are differentiated from sinks simply by having an average flow rate greater than 0.3 kL/hr and a duration greater than 30 s. Accuracies between 80 %–90 % were reported. This approach only focuses on the *human* side of our hierarchy defined in Fig. 1. The *appliance* side used flow trace analysis. The need for additional water monitoring devices is reduced but the installation of motion sensors is required.

The aforementioned methods rely on moderate- to high-frequency sampling to allow for the detection of rising and falling edges in water usage data. Corresponding to a real-world setting with commodity hardware and privacy concerns, it makes sense to restrict oneself to lower frequency data streams. Unfortunately, this makes the problem of water disaggregation much more difficult with the loss of event granularity.

DSCRDM (Deep Sparse Coding Based Recursive Disaggregation Model) formulates the problem by iteratively decomposing the aggregate water reading one device at a time [2]. After the first device has been separated, the second device is disaggregated from the residual, and so on until one device remains. Using the flow trace data from an Aquacraft study [10], they were able to achieve F-measures of: above 70 % for the shower, 35 % for the toilet, and 45 % for the washing machine. Sampling at such a low frequency means that the results are more useful for seeing longterm consumption rather than the consumption while a specific appliance/faucet is being used.

Water disaggregation has been dominated by studies that rely on high-frequency data, proprietary meters, and/or labelled datasets. The focus has been on disaggregating subtleties in human use before more fundamental parts of the problem have been solved. In contrast, we use low-frequency data (per minute) from standardized meters and do not rely on labelled data to disaggregate appliance water use from household water data. Note, smart meters report readings at $\frac{1}{8}$ Hz within the house.

3 Our Approach

We leverage data from electricity disaggregation to help with water disaggregation. Given the electrical state of a water-consuming appliance and the whole house water meter reading, the goal is to build a model that can predict the amount of water used by the appliance. Figure 2 shows an outline of our NILM system.

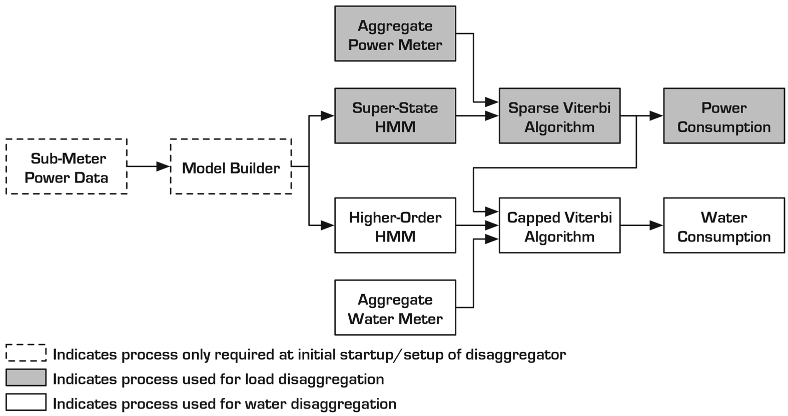

Fig. 2. A block diagram depicting how water disaggregation is a part of a whole NILM process. NILM appliance classification is used by the water disaggregator to disaggregate that appliance's water consumption.

3.1 Disaggregating Electricity

In [7], a method for non-intrusive load monitoring is provided. By quantizing the electricity readings of an appliance based on peaks in its probability mass function (PMF), the time series can be viewed as a list of discrete state transitions. These states may correlate to different functions of the appliance. For example, in the dishwasher, one may represent the electricity consumed when the water pump is on, while another may represent the electricity consumed when the heating element is on.

For our purposes, we assume we have a method for obtaining the series of states an appliance transitions through, given the series of whole house electricity readings. The method described in [7] determines the entire house's superstate (a combined state representing the state of each appliance simultaneously) at once. A Hidden Markov Model is built that takes the whole house's electricity reading as input and produces the house's superstate as output.

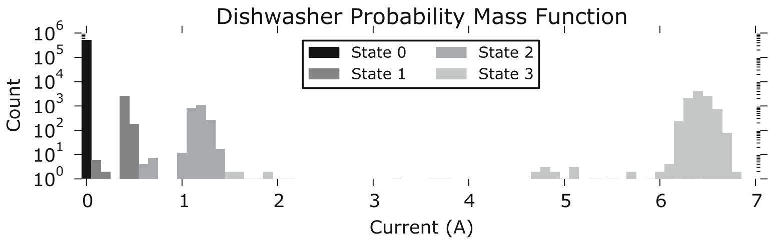

Fig. 3. Dishwasher PMF with unnormalized counts to depict the scale of the dataset used. Most of the time the dishwasher is in the OFF state. When it is ON, it is in one of three other distinct states.

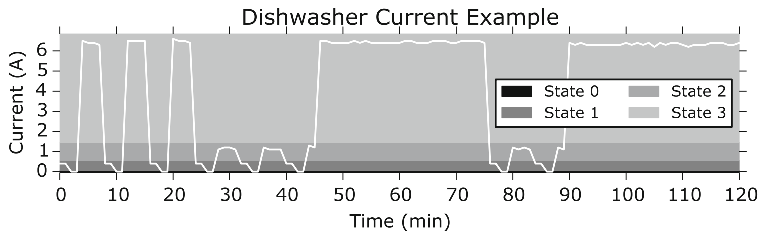

Fig. 4. Current (A) readings for one dishwasher example. States emerge from peaks within the histogram. The exact current reading does not matter, but rather the state the dishwasher is in.

To build and test our model we used The Almanac of Minutely Power Dataset (AMPds) [9]. AMPds is a standardized, low-frequency dataset capturing meter information from a household in the Greater Vancouver region of British Columbia, Canada. It is designed with disaggregation in mind, providing data for electricity, water, and natural gas. Compared to other datasets for this purpose, this data is of extremely high quality and has been cleaned to ensure that different researchers will be working from the same starting point. The dataset contains two years of data from the beginning of April 2012 to the end of March 2014. Meters read in at a rate of once per minute. This means there are over one million records for each meter. In addition to the main electricity meter there are 20 sub-meters (one of which is for the dishwasher), resulting in over 20 million total electricity records. The electricity readings are taken by two DENT PowerScout 18 units.

To test our model, we just use the sub-metered ground truth current to determine the appliance's state. States are assigned by using peaks in the appliance's PMF to discretize the raw current reading. Figure 3 shows the PMF for the dishwasher's current reading which are algorithmically determined [7]. Note that the logarithmic scale means that readings of 0 far outnumber other readings. This is the OFF state. When this appliance is ON, three additional distinct peaks form. Using this method on an example dishwasher run from AMPds, we obtain Fig. 4.

3.2 Disaggregating Water

For water there are only meters for the main and the instant hot water unit, resulting in over two million total water records. Elster/Kent V100 water meters are used to take the water readings. Each record includes a pulse counter (litres), the average rate (litres per minute), and the instantaneous rate (litres per minute). This means there are over three million data points for each water meter, resulting in over six million water data points total. For our purposes we only need the average flow rate, which can also be determined directly from the change in the pulse counter, since we are measuring in L/min and pulses are recorded once per minute. Water readings are collected in half-litre pulses. For the first few months of AMPds, the water meters were only set to pulse at every gallon (3.785 L). Due to this, we only consider the second year of AMPds.

Figure 5 shows the whole house water readings during the example from Fig. 4. Looking at the state changes alongside, a clear pattern emerges. In the case of the dishwasher, water is used in spurts of roughly 3 L over the course of two minutes. We can clearly see that only using an appliance's electrical state and the whole house's water consumption from a few points in time is necessary to provide a good indication of whether the water use is due to the appliance in question or from something else in the house.

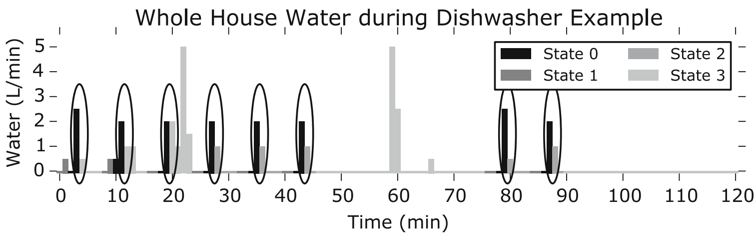

Fig. 5. Whole house water readings for one dishwasher example. Circled amounts are from the dishwasher consuming water – a repeating pattern of 2 L followed by 1 L.

Hidden Markov Models (HMMs) are designed to capture these kinds of relationships. Our solution requires some modifications to the traditional formulation. Like an unsupervised scenario, we do not have labelled data to train with. Unlike an unsupervised scenario, we have some prior knowledge about the output we are looking for (i.e. each appliance's water use is bound above by the whole house's water reading). Machine learning methods use HMMs for efficiently modelling and learning patterns in state transitions over time [11]. They rely on the Markov assumption that a state at a given point in time being dependent on all previous states can be simplified to only being a conditional probability on a single previous state. A standard HMM can be defined as HMM $= \{h, o, \mathbf{S}, \mathbf{T}, \mathbf{E}\}$, where h is the number of possible hidden states, o is the number of possible observed states, $\mathbf{S}$ is the start probability vector (of length h), $\mathbf{T}$ is the transition probability matrix ($h \times h$), and $\mathbf{E}$ is the emission probability matrix ($h \times o$). Where there exists a finite sequence of hidden states, $H = (h_0, h_1, h_2, \ldots, h_T), h_t \in 0..h-1$, and a finite sequence of observed states, $O = (o_0, o_1, o_2, \ldots, o_T), o_t \in 0..o-1$, both of length T. In a first-order HMM, the probability of being in a given hidden state at time t is only conditioned on the hidden state at time $t-1$. The probability of seeing a given observed state at time t is only conditioned on the hidden state at that time.

In the problem we are trying to solve, we know the appliance is in the OFF state (State 0) outside of the extracted samples regardless of the representation we choose. We prepend a 0 hidden state to H and a 0 observed state to O to simplify the start probabilities. With this $\mathbf{S} = [1]$, a simple one-element vector since the probability of starting in state 0 is 100 %. This simplifies things when we look at higher-order models later.

3.3 Capped Viterbi Algorithm

The Viterbi Algorithm is a dynamic programming solution to the problem of finding the most likely sequence of hidden states corresponding to a sequence of observed states [3,14]. Running through the steps of the algorithm can be visualized with a trellis diagram. At each point in time, only the most likely path leading there needs to be considered.

For our particular problem, we know the electrical state of the appliance and the whole house water reading at each point in time. We want to find the disaggregated water reading for the appliance in question. Since we do not have sub-metered water data, we cannot use a standard supervised learning method. Conversely, we do not want to use a standard unsupervised learning method, because we have prior knowledge about the disaggregated water reading. That is, we know that it must be in half-litre increments and is bound above by the whole house water reading.

To utilize this information, we formulate the model as we would if we were learning the whole house water reading given only the electrical state of one appliance. This means the hidden states are the range of possible whole house water readings in half-litre increments and the observed states are the appliance's electrical states. Since we already have both of these pieces of information, we can use a supervised training method. The intuition behind this is that we are purposely *under-fitting* the data in the hopes that we are left with only the disaggregated reading we are looking for when predicting. Instances where the appliance in question is the only thing consuming water (i.e. the whole house water reading is exactly the desired answer) are more consistent with each other and train the model to ignore noise from the rest of the house.

Not only do we have access to the true hidden labels at training time, but we also know them for the sequences we want to predict. Of course we do not want our model to outright see this information, as this would just leave us with the whole house water readings we already know. Instead, we provide this information as a *hint* to the Viterbi Algorithm.

We call this variant the Capped Viterbi Algorithm. In addition to the sequence of observed states O, our algorithm takes as input a sequence of upper bounds (or *caps*) on the hidden states $C = (c_0, c_1, c_2, \ldots, c_T), c_t \in 0..h-1$. For the problem at hand, this is the whole house water meter reading (which is used as H when training). Pseudocode for the Capped Viterbi Algorithm is as follows:

```
input:  C = (c_0, c_1, c_2, ..., c_T)  // the sequence of caps
        O = (0, o_0 , o_1 , o_2 , ..., o_T)
        S = [s_j]
        T = [t_(i,j)]
        E = [e_(j,n)]
Output: H = (h_0, h_1, h_2, ..., h_T)
for j in S do
  H[j] = ();  // the most likely sequence H ending in j
  P[j] = s_j*e_(j,0); // P[x] is the probability of H[x]
end
for t = 0 to T do
  for j = 0 to c_t do  // where c_t is the current cap
    newP[j] = max_(i in P) P[i]*t_(i,j)*e_(j,o_t);
    newH[j] = H[argmax_(i in P) P[i]*t_(i,j)*e_(j,o_t)] + j;
  end
  P = newP;
  H = newH;
end
return H[argmax_(i in P) P[i]]
```

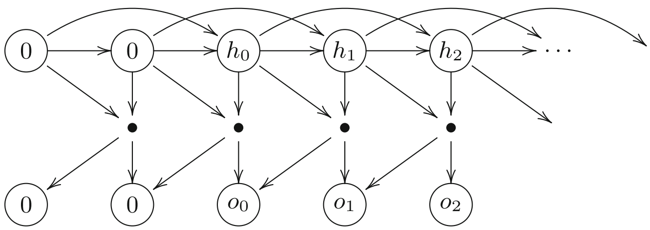

Fig. 6. Our modified second-order HMM. The relationship with the observed states is not a standard second-order HMM. Bullets line up with t where emission pairs are taken into account when predicting the optimal hidden state sequence.

A simple first-order model is not able to capture the relationships in our dataset. Using a second-order model variation, we get $\mathbf{T}$ being a $h^2 \times h$ matrix and $\mathbf{E}$ being an $h^2 \times o^2$ matrix where $t_{(i_2,i),j} = P(h_t = j | h_{t-2} = i_2, h_{t-1} = i)$ and $e_{(i,j),(m,n)} = P(o_{t-1} = m, o_t = n | h_{t-1} = i, h_t = j)$. Figure 6 shows the modified model. Compare this to a standard second-order model where the emission matrix is the same as in a standard first-order model. In our case, pairs of hidden states emit pairs of observed states.

This is easily generalized to higher-order models. There are h^T possible sequences of hidden states for a given sequence of observed states. The standard first-order Viterbi algorithm finds the optimal sequence by only looking at h possibilities for each hidden state at each point in time. This results in an $O(h^2T)$ running time. Generalizing to order-n models, it looks at h^n possibilities for each hidden state at each point in time, giving a running time of $O(h^{n+1}T)$.

The Capped Viterbi Algorithm reduces the search space by only looking at $\prod_{x=t-n}^{t-1} c_x \leq h^n$ possibilities for each hidden state up to $c_t \leq h$ at each point in time. In the worst case $C = (h, h, \ldots, h)$, keeping the $O(h^{n+1}T)$ running time. Empirically the running time is much less than this, as usually little to no water is being used in the house relative to the highest recorded water consumption. The running time is also kept low by only keeping track of subsequences with non-zero probabilities. This is especially significant in higher-order models where $\mathbf{T}$ is quite sparse.

To ensure there is at least one possible path with a non-zero probability at each point in time, we ensure that every entry in the 0 column of $\mathbf{T}$ is non-zero (i.e. every row can transition to the OFF hidden state). This maintains the sparsity of $\mathbf{T}$ while allowing any path to *zero-out*. $\mathbf{E}$ is smoothed by averaging out the 0 and 1 counts in each row before normalizing. Effectively, the singular counts are spread out to emissions with no count. The intuition behind this is that the number of fluke single observations give an indication of the probability of a previously unseen emission. In cases where there are no 1 counts, all of the 0 counts are set to 1.

4 Experimental Results

Here we provide the results of a formal evaluation of the performance of our model on AMPds. By hand-labelling the dishwasher water data, we are able to conduct a quantitative analysis. 185 dishwasher runs were extracted from the second year of AMPds. These were divided into 10 sets with 18 or 19 samples each. 10-fold cross-validation was used to evaluate the performance of first-order,

Table 2. Results of different n-order HMMs. Disaggregation in not performed in the first row where the aggregate water reading is just assumed to be the dishwasher.

Order	Explained variance	Mean squared error (L^2/min^2)	Training time (μs per point)	Testing time (μs per point)
—	−33.648582	9.238157	—	—
1	0.000000	0.249550	3.70	55.1
2	0.932241	0.015590	7.75	94.1
3	**0.938049**	**0.014253**	**32.5**	**113**
4	0.902036	0.022544	195	125

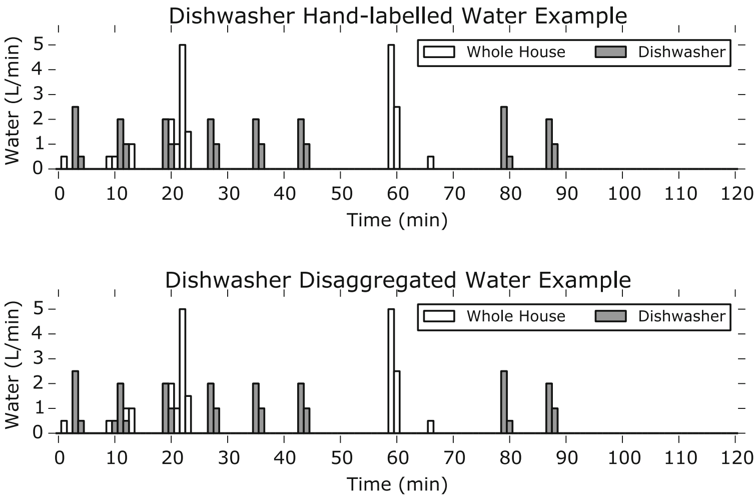

Fig. 7. Results example: (top) hand-labelled ground truth for one dishwasher example, and (bottom) output of third-order HMM on this example. The labeller was able to use knowledge from nearby actions to more accurately label difficult cases; e.g. there are multiple ways to determine 3 L in the second burst of water.

second-order, third-order, and forth-order models. The results[1] are presented in Table 2 along with average running times per data point.

The first-order model is not able to capture any of the relationships in the dataset, as shown by the Explained Variance of 0. In fact, it just labels the dishwasher's water consumption as 0 L at every point in time. This acts as a good baseline because it shows that even only looking at times when the dishwasher is running, most of the time it is not consuming water. A mean squared error of $0.25\,\mathrm{L}^2/\mathrm{min}^2$ is trivial to achieve.

As expected, the second-order model performs considerably better. The third-order model shows minor improvements over this. Once we reach the fourth-order model, we begin to see diminishing returns. This is due to the extreme sparsity of such a high order model. Note that the testing time (Capped Viterbi Algorithm) does not grow exponentially as with the training time (including smoothing).

Figure 7 (top) shows the hand-labelled ground truth for the example from Figs. 4 and 5. The output of the third-order model when run on these examples is shown in Fig. 7 (bottom). We are able to almost capture the exact water usage.

5 Conclusion

The field of household water disaggregation has tended towards studies that focus on teasing out low-level differences between similar fixtures. In doing so,

[1] Classification measures [8] were not used. As NILM has predetermined classification for us, we only need to measure the amount of error in our results.

non-standard sensors must be introduced to collect additional data. By situating the problem in terms of a hierarchy, we were able to pinpoint the level to which disaggregated water information is helpful to homeowners.

This work is the first to present non-intrusive water disaggregation using the results from a non-intrusive load monitoring algorithm. There is no need to install water sub-meters to build a model of water consumption. Further, our work allows a data model to be built that does not require tuning by an expert. Our water disaggregator achieves very high accuracy results having mean squared errors of under $0.02\text{L}^2/\text{min}^2$. Future work may include combining our method with other aforementioned methods to disaggregate human water usage.

Acknowledgments. Research was supported by NSERC, including an Alexander Graham Bell Canada Graduate Scholarship, and a number of awards given by SFU's Dean of Graduate Studies: a Provost Prize of Distinction, a C.D. Nelson Memorial Graduate Scholarship, and a Graduate Fellowship.

References

1. DeOreo, W.B., Heaney, J.P., Mayer, P.W.: Flow trace analysis to assess water use. J. Am. Water Works Assoc. **88**(1), 79–90 (1996)
2. Dong, H., Wang, B., Lu, C.T.: Deep sparse coding based recursive disaggregation model for water conservation. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, pp. 2804–2810 (2013)
3. Forney Jr., G.D.: The viterbi algorithm. Proc. IEEE **61**(3), 268–278 (1973)
4. Froehlich, J., Larson, E., Campbell, T., Haggerty, C., Fogarty, J., Patel, S.N.: HydroSense: infrastructure-mediated single-point sensing of whole-home water activity. In: Proceedings of the 11th International Conference on Ubiquitous Computing, pp. 235–244 (2009)
5. Hart, G.W.: Prototype nonintrusive appliance load monitor. MIT Energy Laboratory and Electric Power Research Institute Technical report (1985)
6. Kim, Y., Schmid, T., Charbiwala, Z.M., Friedman, J., Srivastava, M.B.: NAWMS: nonintrusive autonomous water monitoring system. In: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems, SenSys 2008, pp. 309–322. ACM, New York (2008)
7. Makonin, S., Bajic, I.V., Popowich, F.: Efficient sparse matrix processing for nonintrusive load monitoring (NILM). In: 2nd International Workshop on Non-Intrusive Load Monitoring (2014)
8. Makonin, S., Popowich, F.: Nonintrusive load monitoring (NILM) performance evaluation. Energy Effi. **8**, 1–6 (2014)
9. Makonin, S., Popowich, F., Bartram, L., Gill, B., Bajic, I.V.: AMPds: a public dataset for load disaggregation and eco-feedback research. In: Electrical Power & Energy Conference, pp. 1–6 (2013)
10. Mayer, P.W., DeOreo, W.B., Opitz, E.M., Kiefer, J.C., Davis, W.Y., et al.: Residential End Uses of Water. American Water Works Association, Denver (1999)
11. Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
12. Schantz, C., Donnal, J., Sennett, B., Gillman, M., Muller, S., Leeb, S.: Water non-intrusive load monitoring. IEEE Sens. J. **PP**(99), 1 (2014)

13. Srinivasan, V., Stankovic, J., Whitehouse, K.: WaterSense: water flow disaggregation using motion sensors. In: Proceedings of the Third ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings, BuildSys 2011, pp. 19–24. ACM, New York (2011)
14. Viterbi, A.J.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans. Inf. Theory **13**(2), 260–269 (1967)