Graph-based algorithms for comparison and prediction of household-level energy use profiles

Nathaniel Charlton
CountingLab Ltd
billiejoe@countinglab.co.uk

Danica Vukadinović Greetham
Department of Mathematics and Statistics
University of Reading, UK
d.v.greetham@reading.ac.uk

Colin Singleton
CountingLab Ltd
colin@countinglab.co.uk

***Abstract*—We present an efficient graph-based algorithm for quantifying the similarity of household-level energy use profiles, using a notion of similarity that allows for small time–shifts when comparing profiles. Experimental results on a real smart meter data set demonstrate that in cases of practical interest our technique is far faster than the existing method for computing the same similarity measure. Having a fast algorithm for measuring profile similarity improves the efficiency of tasks such as clustering of customers and cross-validation of forecasting methods using historical data. Furthermore, we apply a generalisation of our algorithm to produce substantially better household-level energy use forecasts from historical smart meter data.**

I. Introduction

The demands on future electricity networks are predicted to severely increase [1]. A current global rise of electrical energy demand is fueled by emergent economies and new technologies such as electrical vehicles and heat pumps. At the same time there is a scientific consensus about the need to reduce carbon footprints for environmental reasons.

Accurate forecasts of electrical energy demand at the level of individual households may prove invaluable in the development of the electricity networks of the future. For example, smart storage devices such as batteries can be placed in individual households or neighbourhoods, to optimise usage and mitigate possible demand peaks [2]: a battery can release energy during peak times and recharge when there is a surplus (through users' generation using solar panels, or during quiet periods). While a plethora of forecast methods have been developed for forecasting high-level aggregated energy usage (see [3], [4]) (such as at the regional or national level), much less has been done for household-level forecasting.

In fact, the volatile environment of household-level forecasting requires not only different forecasting techniques, but a different notion of what constitutes a good forecast. Aggregated energy demand is quite smooth and forecast error has been mainly measured using RMSE (root mean square error), MAPE (mean absolute percentage error) or variants thereof. But at the household level one has volatile, non-smooth load functions that are much harder to predict, similar to individual levels of natural gas consumption [5]. Haben et al. [6] demonstrate that at the household level RMSE does a poor job of distinguishing good (useful) forecasts from poor (useless) forecasts. To address this, [6] proposed a new method, the *adjusted error*, of quantifying the similarity between two household-level load curves, typically a forecast and the actual usage. The adjusted error allows for small time permutations when comparing the profiles, while penalising amplitude differences using a chosen L^p norm. [6] gives a way to compute adjusted errors, by reducing the problem to an instance of the assignment problem [7].

In this paper we provide a new algorithm for computing adjusted errors, based on a reduction to the shortest path problem for a particular family of graphs. Experimental results on a real smart meter data set verify that in cases of practical interest our algorithm is far faster than the existing method. We report running times from a profile clustering application, illustrating how our algorithm can improve the efficiency of smart grid applications in practice. Finally, we give a generalisation of our algorithm that we apply to create better household-level energy load forecasts.

The mathematical theory underlying our approach is somewhat involved and space constraints prohibit a full discussion here. In this paper we state without proof the theorems that justify our methods, and refer the reader to a separate technical report [8] for full technical details and proofs.

II. The adjusted error measure

By an *energy use profile* we mean a vector $\vec{x}$ of n non-negative real numbers whose components represent the (electrical) energy use of a household, sampled at n evenly spaced time points. Typically we work with daily profiles read from smart meters installed in houses, reading every hour or every 30 minutes; common values of n are thus 24 and 48.

Figure 1 (top) shows two forecasts A and B plotted against the subsequent actual energy consumption. As argued in [6], Forecast A is a good, informative forecast: it contains the right number of peaks in consumption, at approximately the right times and with approximately the right magnitudes. Yet under RMSE the flat Forecast B scores better, despite giving us no indication of the expected times and magnitudes of the peaks.

The idea of the adjusted error is to make allowances for small discrepancies in time that may be present between the forecast and actual profiles, up to an *adjustment limit* of w time units. A *w-local permutation* is one that rearranges the components of a profile by moving each forwards or backwards by up to w time units. We then consider a forecast good if, among all the possible w-local permutations of the

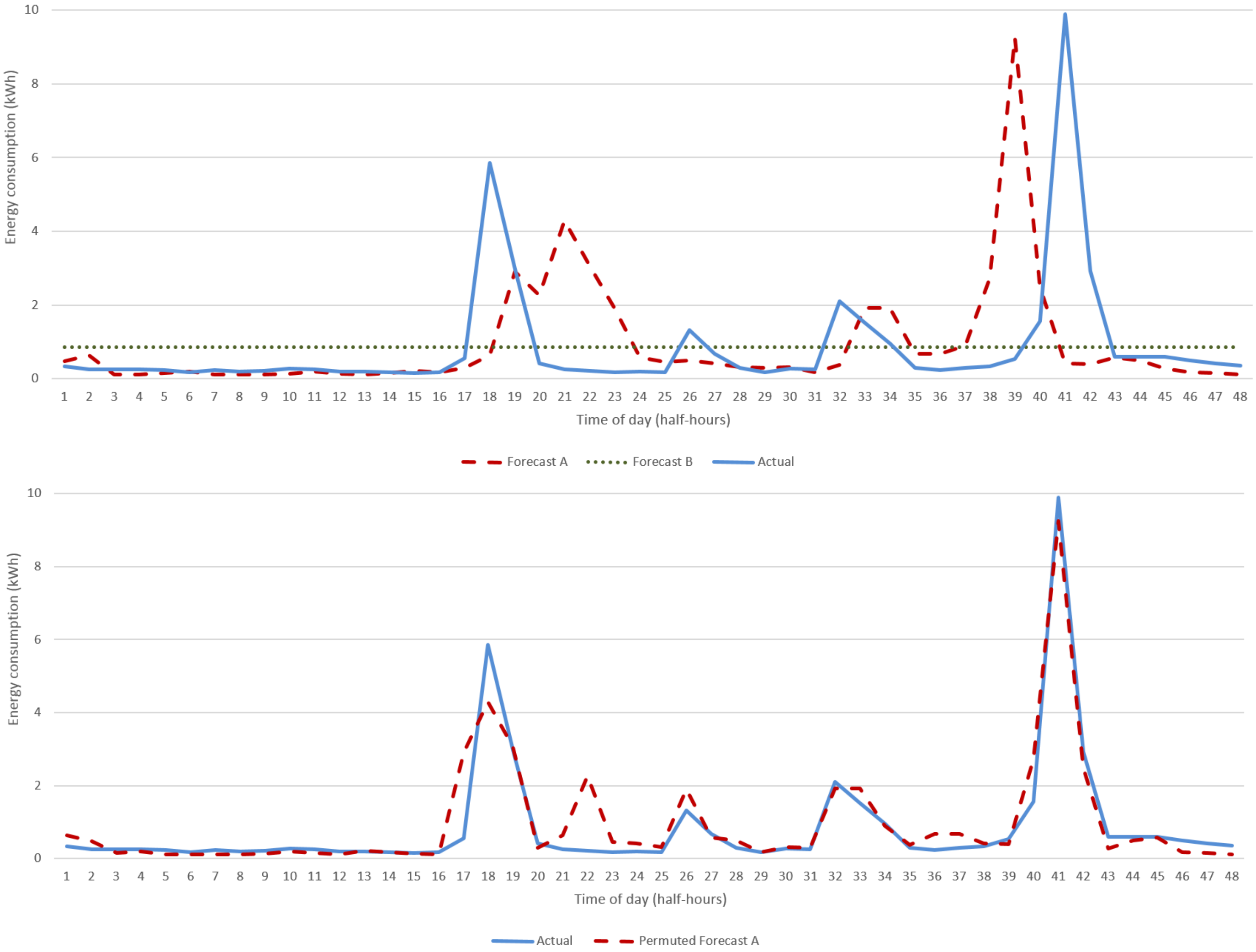


Fig. 1. (top) An informative Forecast A, a flat uninformative Forecast B and the subsequent actual energy consumption (bottom) A locally permuted version of Forecast A matching the actual energy consumption well.

forecast, there is one that is close to the actual consumption at each time point.

Fig.1 (bottom) shows a locally permuted version of Forecast A matched against the subsequent actual consumption. As they are very close, Forecast A will be given a small adjusted error. Because Forecast B is flat, no permuting of its components can bring it closer to the actual profile. We have taken $w = 3$, so that components of the forecast have been shifted forwards or backwards by no more than an hour and a half.

Formally, [6] defines the adjusted error $E_p^w(\vec{x}, \vec{y})$ between an actual profile $\vec{x}$ and a forecast profile $\vec{y}$ (where $p \geq 1$) by

$$E_p^w(\vec{x}, \vec{y}) := \min_{\pi \in \mathcal{P}(w,n)} \left(\sum_{i=1}^{n} |(\pi(\vec{y}))_i - x_i|^p \right)^{1/p} \qquad (1)$$

where $\mathcal{P}(w, n)$ is the set of w-local permutations on n-vectors. [6] gives a way to compute the adjusted error between two profiles in $O(n^3)$ time, by reduction to the assignment problem which is then solved using the Hungarian algorithm [7].

The parameters w and p can be chosen differently for different applications. However if w is set too close to n, adjusted errors are unlikely to yield useful information about profile (dis)similarity: for example if we have daily profiles with $n = 24$, meaning hourly readings, and we set $w = 10$, then a peak in consumption at 8 AM in the first profile could be matched with a peak at 6 PM in the second profile. But even a small value such as $w = 2$ will usually be enough to match events such as cooking dinner, which tend to occur at approximately the same time each day. Thus in practice we are concerned about computing adjusted errors when w is much smaller than n. As for the value of p, although the L^2 norm is very commonly used in statistics generally, [6] recommends using $p = 4$ in order for large errors such as a missed peak to be penalised more severely than smaller errors such as a slightly over- or under-estimated peak amplitude.

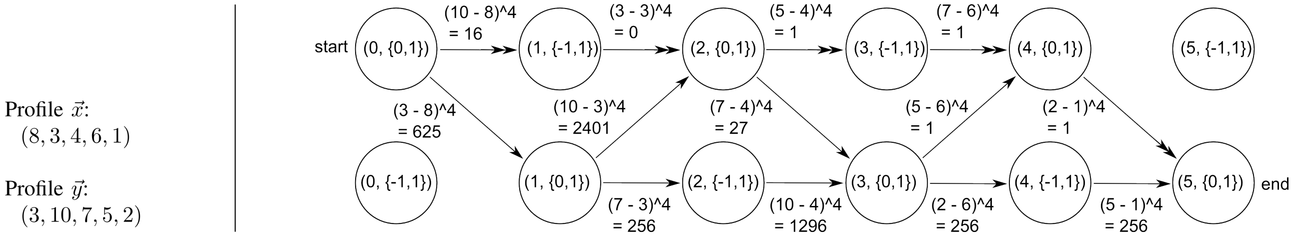


Fig. 2. Two five-point profiles $\vec{x}$ and $\vec{y}$ (left), and the graph we use to compute the adjusted error $E_4^1(\vec{x}, \vec{y})$ between them (right).

III. Our algorithm

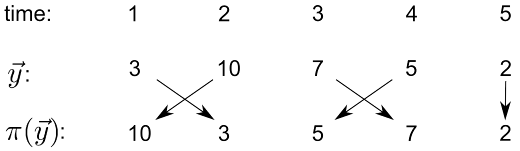
Consider the two profiles $\vec{x}$ and $\vec{y}$ given in Fig. 2 (left); each has just five time points to simplify the presentation. Suppose we wish to compute the adjusted error using the L^4 norm and adjustment limit $w = 1$, i.e. $E_4^1(\vec{x}, \vec{y})$. Our approach is to generate from $\vec{x}$ and $\vec{y}$ a particular graph, shown in Fig. 2 (right), such that the length of the shortest path (or shortest paths) from the "start" node to the "end" node in the graph is equal to the required adjusted error $E_4^1(\vec{x}, \vec{y})$. We now explain informally how the graph is constructed.

Choosing a path from "start" to "end" corresponds to choosing a w-local permutation π of $\vec{y}$: specifically, choosing the ith edge of such a path corresponds to choosing which element of $\vec{y}$ should appear at position i in the permuted profile $\pi(\vec{y})$. Each node is labelled with a pair (i, S) where i gives the number of components of the permuted profile $\pi(\vec{y})$ chosen so far, and the set S records which components of $\vec{y}$ are available when choosing $(\pi(\vec{y}))_{i+1}$. Specifically, if $k \in S$ then we are able to choose to permute the value y_{i+k+1} onto time point $i + 1$. The cost of each edge leaving the node (i, S) is the contribution to the error that comes from comparing the chosen $(\pi(\vec{y}))_{i+1}$ with x_{i+1}.

For example, when we are at the start node $(0, \{0, 1\})$ in Fig. 2 we have chosen 0 components of $\pi(\vec{y})$ so far, and we are about to choose $(\pi(\vec{y}))_1$. The available choices, corresponding to the two outgoing edges, are y_1 and y_2. Suppose we take the horizontal edge, which corresponds to choosing to put $(\pi(\vec{y}))_1 := y_2$. The cost of the edge is $|(\pi(y))_1 - x_1|^4 = (10 - 8)^4 = 16$, which will be the contribution for $i = 1$ to the summation in (1).

The node $(1, \{-1, 1\})$ that we reach has availability set $\{-1, 1\}$ reflecting that the available options for our next choice, the choice of $(\pi(\vec{y}))_2$, are y_1 and y_3; these lie, respectively, at time shifts of -1 and 1 from the time point 2. The availability set does not contain 0, corresponding to a choice of y_2, because we used y_2 already at the previous node.

In fact, however, choosing $(\pi(\vec{y}))_2 := y_3$ is not feasible: components can be shifted by at most $w = 1$, so if we have assigned $(\pi(\vec{y}))_1 := y_2$ and then $(\pi(\vec{y}))_2 := y_3$, we will never be able to assign y_1 to any component of $\pi(\vec{y})$. Thus there is only one edge available from the node $(1, \{-1, 1\})$. Continuing in this way, we gradually choose the permuted profile $\pi(\vec{y})$; when we reach the "end" node we have finished. The shortest path, highlighted with double arrowheads in Fig. 2, corresponds to the following permuted profile:



The following definitions state formally how to construct the appropriate graph for arbitrary profiles $\vec{x}$ and $\vec{y}$.

Definition 1. An *availability set* S is a subset of $\{-w, \ldots, w\}$ such that $w \in S$ and $|S| = w + 1$. Let $\mathcal{S}$ be the set of all such availability sets; there are $\binom{2w}{w}$ of these. Define a distinguished availability set $S^+ := \{0, \ldots, w\}$. □

Definition 2. Given two profiles $\vec{x}, \vec{y} \in {\mathbb{R}_{\geq 0}}^n$, we define a weighted directed acyclic graph $G(\vec{x}, \vec{y})$ as follows. We take as vertices the elements of $\{0, \ldots, n\} \times \mathcal{S}$. We put an edge from vertex $v = (i, S)$ to vertex $v' = (i{+}1, S')$ if there exists $t \in S$ such that $1 \leq i{+}1{+}t \leq n$ and $S' = \mathit{decrease}(S \setminus \{t\}) \cup \{w\}$ (where $\mathit{decrease}(S) := \{m{-}1 \mid m \in S\}$). (There can be at most one such t for each pair of v and v'.) We set the cost of the edge to $|y_{i+1+t} - x_{i+1}|^p$. We define distinguished vertices $v_{\text{start}} := (0, S^+)$ and $v_{\text{end}} := (n, S^+)$. □

Theorem 1. *The length of the shortest path(s) from v_{start} to v_{end} in $G(\vec{x}, \vec{y})$ is equal to the adjusted error $E_p^w(\vec{x}, \vec{y})$.* □

So, we have reduced computation of adjusted errors to a standard graph problem. Thanks to the layered structure of the graphs, visible in Fig. 2, a *topological sort* of the nodes is trivial to obtain: first list the nodes of form $(0, S)$, then those of form $(1, S)$, then those of form $(2, S)$ and so on. Hence we can compute the shortest path from v_{start} to v_{end} in time $O(|E|)$, as detailed in [9, §24.2], where $|E|$ is the number of edges in the graph. Each of the $n{+}1$ layers of G contains at most $\binom{2w}{w}$ nodes, and each node has out-degree at most $w{+}1$. Thus the number of edges $|E|$ is of order $O(nw \cdot \binom{2w}{w})$, which we can relax to $O(nw \cdot 4^w)$. So we can find the required shortest path in $O(nw \cdot 4^w)$. A similar argument shows that we can construct the graph in $O(nw \cdot 4^w)$ time. Thus we have the following result.

Corollary 1. *Our graph-based algorithm computes the adjusted error $E_p^w(\vec{x}, \vec{y})$ in time $O(nw \cdot 4^w)$.* □

For full details and proofs see the technical report [8].

When n is large compared to w, as will be the case in our applications, our running time of $O(nw \cdot 4^w)$, which is linear in n, compares favourably to the $O(n^3)$ running time of the

w	Adj. error computation		Clustering using PAM
	Our new algorithm	Hungarian algorithm	
1	1.0	1920.2	< 0.1
2	3.2	3114.0	< 0.1
3	12.2	4048.9	< 0.1
4	91.3	4796.5	< 0.1
5	885.9	5399.8	< 0.1
6	4906.8	5914.3	< 0.1

TABLE I

RUNNING TIMES (IN SECONDS) OF OUR ALGORITHM AND THE HUNGARIAN ALGORITHM COMPUTING ADJUSTED ERRORS BETWEEN ALL PAIRS OF 2,000 PROFILES, AND OF CLUSTERING THE SAME 2,000 PROFILES INTO 5 CLUSTERS USING THE PAM TECHNIQUE.

existing method. We explore running times in practice in the next section.

IV. Experimental results

We evaluated our algorithm using real data from Ireland's Commission for Energy Regulation [10]. We took 2,000 household-level electricity use profiles, with half-hourly readings (so $n = 48$) and computed the adjusted error between all pairs of these (resulting in 1,999,000 adjusted error computations). Table I compares the CPU time required for this task, using the Hungarian algorithm and using our new algorithm, for values of w from 1 to 6. CPU times reported are measured on a single core of a 2.2Ghz Intel PC, using C++ implementations.

The results show that our new algorithm is very much faster for small values of w. The running time of our algorithm grows more quickly than that of the Hungarian algorithm as w increases, so for sufficiently large w the Hungarian algorithm will be faster; for $n = 48$ this happens at $w \geq 7$. Crucially however, as explained in Section II, in practice w is small and our algorithm delivers orders-of-magnitude savings.

V. Applications of our algorithm

Having a fast algorithm for adjusted error computation is obviously beneficial when using historical data to assess the accuracy of proposed forecasting methods, or to tune their parameters.

Although it has been termed an *error*, $E_p^w(\vec{x}, \vec{y})$ can be used as a measure of (dis)similarity between any two profiles; one need not be a forecast. We can, for instance, use adjusted error as one measure of profile dissimilarity when clustering profiles. Clustering of smart meter profiles has been proposed as a component in various smart grid management activities such as tariff design [11], targeting of behaviour modification initiatives [12] and improving short term load forecasts [13].

Table I also reports the CPU time required for clustering the 2,000 smart meter profiles into 5 clusters using the PAM technique [14]. To use PAM clustering, one defines a distance function between the objects being clustered (here profiles). We took the distance between profiles $\vec{x}$ and $\vec{y}$ to be a weighted sum of the following ingredients:

- the adjusted error $E_4^w(\vec{x}, \vec{y})$,
- the (absolute) difference between the largest component of $\vec{x}$ and the largest component of $\vec{y}$,
- the (absolute) difference between the smallest component of $\vec{x}$ and the smallest component of $\vec{y}$,
- the (absolute) difference between the mean component of $\vec{x}$ and the mean component of $\vec{y}$.

Hence performing the 1,999,000 adjusted error computations was a necessary precursor to performing the clustering.

The CPU times required for the clustering step are tiny compared to those required for computing the adjusted errors. We include them to illustrate that in smart grid applications, the adjusted error computations can give rise to a substantial part (here the vast majority) of the computational burden, so our faster algorithm will be useful in practice.

VI. Generalisation of algorithm for forecasting

As well as computing adjusted errors quickly, we can use a generalisation of our graph-based method to *create* forecasts that perform well under the adjusted error measure. Our idea is as follows. Suppose we have energy use profiles $\vec{x}^1, \dots, \vec{x}^N$ for the previous N Tuesdays for a particular household, and we wish to make a forecast of that household's energy use profile for next Tuesday, using some form of average over the previous N Tuesdays and ignoring for now temperature and seasonal effects. It turns out that when p is even, using a generalisation of the graph construction given in Definition 2, we can compute a profile $\vec{y}$ which minimises the criterion

$$\sum_{j=1}^{N} \left(E_p^W(\vec{x}^j, \vec{y}) \right)^p \tag{2}$$

Intuitively an optimal profile, which we denote $\vec{y}^*$, is one that is not too distant (under the adjusted error measure with an adjustment limit W) from any of the N historical profiles; because people often observe a consistent weekly routine, the profile $\vec{y}^*$ works well as a forecast, as we shall shortly demonstrate. We call this forecast the *permutation merge* (PM) forecast. First we state precisely our formal result.

Theorem 2. *Given profiles $\vec{x}^1, \dots, \vec{x}^N$, we can construct in time $O(nW^N \cdot 4^{NW})$ a directed acyclic graph G with distinguished vertices v_{start} and v_{end}, $O(nW^N \cdot 4^{NW})$ edges and the following properties:*

1) *The length of the shortest path(s) from v_{start} to v_{end} is equal to the minimum value of* (2) *as $\vec{y}$ varies.*
2) *From the shortest path we can read off in $O(n)$ time the optimal profile $\vec{y}^*$.*
3) *G has a layered structured, as in Fig. 2, so that a topological sort is trivial to obtain.* □

Corollary 2. *Our graph-based method computes the optimal profile $\vec{y}^*$, that minimises* (2)*, in time $O(nW^N \cdot 4^{NW})$.* □

This running time is linear in n for fixed N and W. For full details see the technical report [8]. We emphasise that the adjustment limit W used in the PM forecast need not be the same as the adjustment limit w with which one intends to evaluate the forecasts.

We applied the PM forecast to real data from the Commission for Energy Regulation [10]. We took the 543 control households for which there is complete data for the 22 weeks from 3rd May 2010 to 3rd Oct 2010, and produced forecasts of the last 7 weeks of this period. (In Ireland the effect of temperature on demand is small over this period.) Fig. 3 shows the results. The horizontal axis shows the number N of historical profiles (vectors) used to produce the forecast. The vertical axis shows the adjusted error in the forecasts in kWh using $w = 3$ and $p = 4$ summed over the 543 households and 7×7 days. Three forecasts are shown for N from 1 up to 12:

1) a simple mean forecast, where at time i we forecast the mean $(x_i^1 + \cdots + x_i^N)/N$ of the load at time i in the historical profiles,
2) the *AA forecast* [6], a forecast specifically designed to score well under the adjusted error metric, and
3) the PM forecast using $W = 1$.

For $N \geq 3$ the PM forecast outperforms the other forecasts, and, unlike the others, keeps improving as more historical profiles are used. An investigation of which values of W and N give the best PM forecasts for the various values of w is future work, as is the incorporation of weather variables.

Figure 4 shows the PM forecast in action. Figure 4 (top) shows smart meter profiles for a particular household, from successive Tuesdays. Figure 4 (bottom) shows the mean forecast that would be generated from these two historical profiles, and the PM forecast using $W = 1$. Because the evening peaks in the two actual profiles occur one hour apart, the (pointwise) mean forecast does not strongly resemble either of the actuals: it contains two evening peaks, of about half the magnitude. On the other hand, the PM forecast effectively shifts the actual profiles' evening peaks half an hour forward and backward respectively, so that they coincide, *and then* averages them, producing a single evening peak at about the same time, with approximately the same magnitude. By construction, the PM forecast profile is close — in the E_4^1 sense — to both the actual profiles. In this way, the permutation merge method tends to produce better forecasts, as we confirmed in Fig. 3.

The PM forecast works by identifying weekly patterns in a household's energy use that result from weekly patterns in the occupants' behaviour. Factors other than human behaviour, such as solar PV generation on the customer's side of the meter, will not display such weekly patterns and in future work we intend to integrate models for these, using for example weather variables as inputs. Our studies have used smart meter data sampled at 30 minute intervals (i.e. $n = 48$) because the planned smart meter rollout in the United Kingdom will provide data at that resolution [15]. In principle one could apply our methods to higher-resolution data; as resolution increases, however, profiles become increasingly volatile and it remains to be seen how this would affect our forecasts.

VII. Conclusions

We presented an efficient graph-based algorithm for computing *adjusted errors*, which evaluate household-level energy load forecasts by allowing small time-shifts. Using real data we demonstrated that our algorithm is orders of magnitude faster than the existing method based on the Hungarian algorithm. We reported running times from a profile clustering application incorporating adjusted errors, illustrating that the adjusted error computations can be a substantial part of the computational burden for smart grid applications. This shows that our faster algorithm will be beneficial in practice. Finally we generalised our method from dealing with one energy use profile to dealing with N profiles, allowing us to develop a forecasting method which produced substantially improved forecasts on real data.

Acknowledgment

The authors acknowledge support from the Thames Valley Vision project (http://www.thamesvalleyvision.co.uk/).

References

[1] "National Grid: UK Future Energy Scenarios 2012," September 2012, available from http://www.nationalgrid.com/uk/Gas/OperationalInfo/TBE/.

[2] P. Wang, J. Yi, P. Lyons, D. Liang, P. Taylor, D. Miller, and J. Baker, "Customer led network revolution – integrating renewable energy into LV networks using energy storage." in *CIRED Workshop 2012, Lisbon, Portugal*, 2012.

[3] E. A. Feinberg and D. Genethliou, "Load forecasting," in *Applied Mathematics for Restructured Electric Power Systems*, ser. Power Electronics and Power Systems, J. H. Chow, F. F. Wu, and J. Momoh, Eds. Springer US, 2005, pp. 269–285.

[4] J. W. Taylor and A. Espasa, "Energy forecasting," *International Journal of Forecasting*, vol. 24, no. 4, pp. 561 – 565, 2008.

[5] M. Brabec, O. Konár, E. Pelikán, and M. Malý, "A nonlinear mixed effects model for the prediction of natural gas consumption by individual customers," *International Journal of Forecasting*, vol. 24, no. 4, pp. 659 – 678, 2008.

[6] S. Haben, J. Ward, D. Vukadinovic-Greetham, P. Grindrod, and C. Singleton, "A new error measure for forecasts of household-level, high resolution electrical energy consumption," University of Reading, Tech. Rep. MPS-2012-14, 2012.

[7] J. Munkres, "Algorithms for the assignment and transportation problems." *J. SIAM*, vol. 5, pp. 32–38, March 1957.

[8] N. Charlton, D. Vukadinovic-Greetham, and C. Singleton, "On minimum cost local permutation problems and their application to smart meter data," University of Reading, Tech. Rep. Mathematics Report Series 2/2013, 2013, http://www.reading.ac.uk/maths-and-stats/research/maths-report-series.aspx.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition*. The MIT Press and McGraw-Hill Book Company, 2001.

[10] Commission for Energy Regulation (CER), "Smart metering electricity customer behaviour trials," 2007, data available from Irish Social Science Data Archive, http://www.ucd.ie/issda/data/commissionforenergyregulation/.

[11] G. Chicco, R. Napoli, P. Postolache, M. Scutariu, and C. Toader, "Customer characterization options for improving the tariff offer," *Transactions On Power Systems*, vol. 18, no. 1, 2003.

[12] I. Dent, T. Craig, U. Aickelin, and T. Rodden, "Finding the creatures of habit; clustering households based on their flexibility in using electricity," in *Digital Futures 2012, Aberdeen, UK*, 2012.

[13] M. Espinoza, C. Joye, R. Belmans, and B. De Moor, "Short-term load forecasting, profile identification, and customer segmentation: A methodology based on periodic time series," *IEEE Transactions on Power Systems*, vol. 20, no. 3, pp. 1622–1630, 2005.

[14] L. Kaufman and P. Rousseeuw, "Clustering by means of medoids," in *Statistical Data Analysis Based on the L1-Norm and Related Methods*, Y. Dodge, Ed. North-Holland, 1987, pp. 405–416.

[15] Department of Energy and Climate Change (DECC), "Smart metering equipment technical specifications (version 2.0)," January 2013, available from https://www.gov.uk/government/consultations/smart-metering-equipment-technical-specifications-second-version.

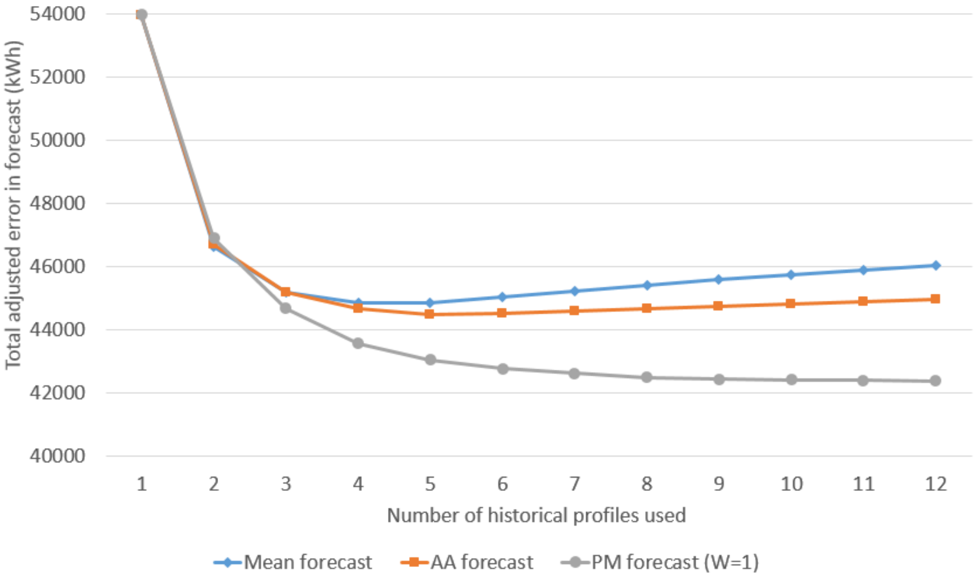


Fig. 3. Comparison of three household-level forecasting methods under the adjusted error measure (with $w = 3$ and $p = 4$). The PM forecast we have introduced performs better than the existing AA forecast [6] and a simple mean forecast.

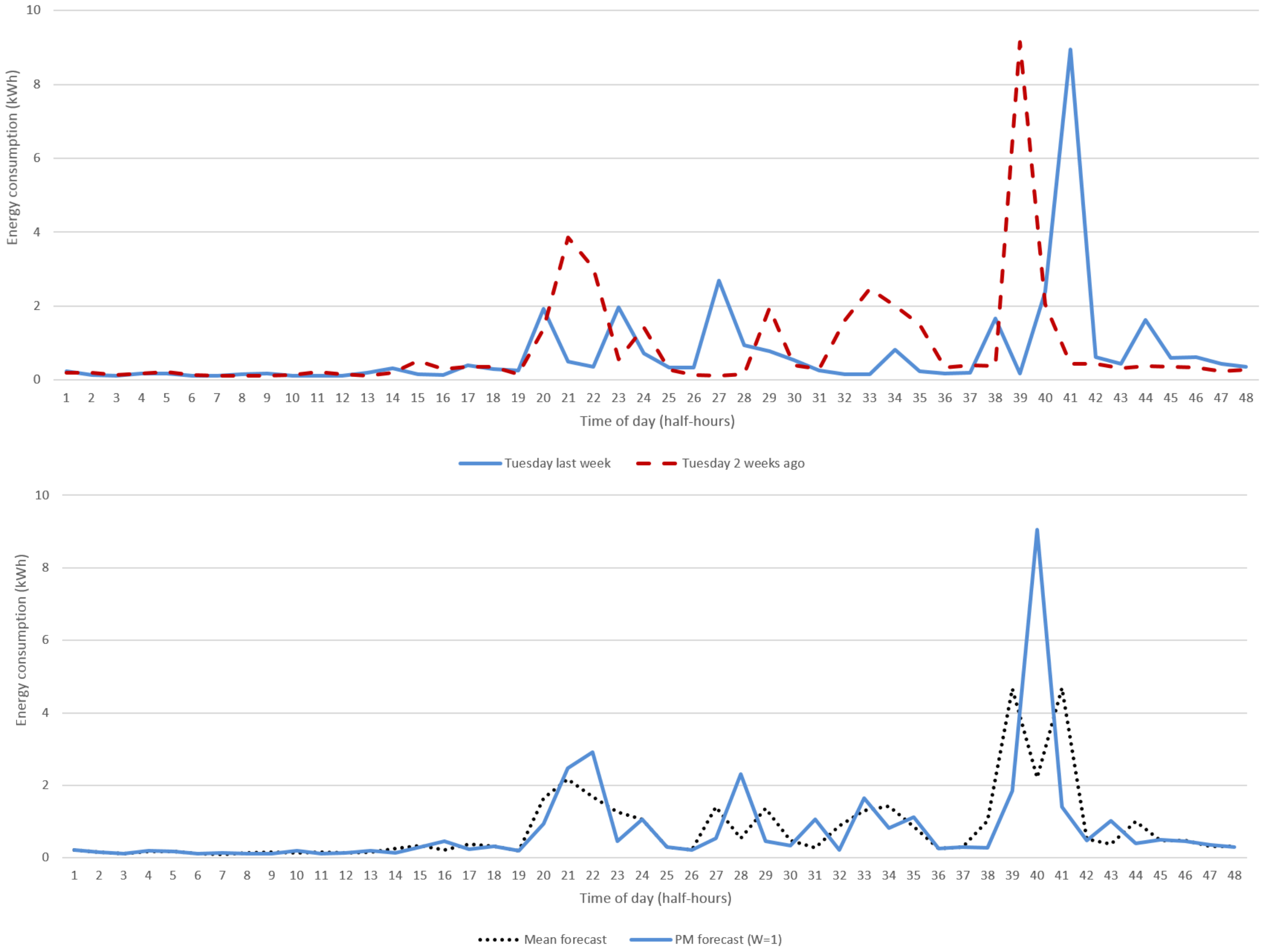


Fig. 4. (top) Actual energy use profiles from successive Tuesdays (bottom) The mean forecast and PM (permutation merge) forecast for the following Tuesday. The PM forecast is much more similar in shape to the recent actual profiles.