Advanced Engineering Informatics xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

An unsupervised hierarchical clustering based heuristic algorithm for facilitated training of electricity consumption disaggregation systems

Farrokh Jazizadeh [a,*], Burcin Becerik-Gerber [b], Mario Berges [c], Lucio Soibelman [d]

[a] Sonny Astani Dept. of Civil and Environmental Engineering, Univ. of Southern California, KAP 217, 3620 South Vermont Ave., Los Angeles, CA 90089-2531, United States
[b] Sonny Astani Dept. of Civil and Environmental Engineering, Univ. of Southern California, KAP 224C, 3620 South Vermont Ave., Los Angeles, CA 90089-2531, United States
[c] Civil and Environmental Engineering Department, Carnegie Mellon University, Porter Hall 113, Pittsburgh, PA 15213-3890, United States
[d] Sonny Astani Dept. of Civil and Environmental Engineering, Univ. of Southern California, KAP 210A, 3620 South Vermont Ave., Los Angeles, CA 90089-2531, United States

ARTICLE INFO

Article history:
Received 14 February 2014
Received in revised form 6 August 2014
Accepted 11 September 2014
Available online xxxx

Keywords:
Clustering
Unsupervised learning
Non-intrusive load monitoring
Training

ABSTRACT

Provision of training data sets is one of the core requirements for event-based supervised NILM (Non-Intrusive Load Monitoring) algorithms. Due to diversity in appliances' technologies, in-situ training by users is often required. This process might require continuous user-interaction to ensure that a high quality training data set is provided. Pre-populating a training data set could potentially reduce the need for user-system interaction. In this study, a heuristic unsupervised clustering algorithm is presented and evaluated to enable autonomous partitioning of appliances signature space (i.e. feature space) for applications in electricity consumption disaggregation. The algorithm is based on hierarchical clustering and uses the characteristics of a cluster binary tree to determine the distance threshold for pruning the tree without *a priori* information. The algorithm determines the partition of a feature space recursively to account for multi-scale nature of the binary cluster tree. Evaluation of the algorithm was carried out using metrics for accuracy and cluster quality (proposed in this study) on a fully labeled data set that was collected and processed in a real residential setting. The algorithm performance in accurate partitioning of the feature space and the effect of different feature extraction techniques were presented and discussed.

1. Introduction

Non-Intrusive Load Monitoring (NILM) as a low-cost alternative approach to individual load level monitoring has been the subject of numerous studies in the recent decades. NILM methodologies apply few sensing points (commonly at the circuit panel level) and use signal processing, pattern recognition, and inference algorithms for recognizing the underlying patterns of appliances' operational states and consequently the energy consumption associated with an operational schedule. In general, the NILM approaches could be categorized into two main classes: event-based and non-event based. In the existing event based NILM solutions, the load disaggregation problem is tackled by tracking the events on aggregated signal time series. Events are defined as variations in the time series caused by appliances' operational state changes such as turning on or off a light bulb. In recent years, non-event based NILM approaches have been also the subject of a few research studies, where mainly Hidden Markov Models (HMM) are used for decomposing aggregated power time series into constituent components (i.e., the power trace of individual contributors in the aggregated power draw) [1–4].

In event-based NILM approaches, detection of underlying state changes (associated with each event) is commonly defined as a classification (i.e., a supervised learning) problem, which requires provision of a training data set. Although, the application of generalized training data (using cross-building training process) has been the subject of a number of research studies, in most of the cases, upon installation of a NILM system, in-situ training has to be carried out due to the diversity in the design and manufacturing technologies used in different appliances, as well as various sources of noise in a data acquisition process. Considering these sources of complexity and the ad hoc nature of the problem, the training process needs to be continuous and requires supervision of a classification algorithm for a period of time to ensure that all the variations in the training data is introduced to the NILM system. This supervision calls for numerous user interactions with

* Corresponding author. Tel.: +1 213 400 2413.
E-mail addresses: jazizade@usc.edu (F. Jazizadeh), becerik@usc.edu (B. Becerik-Gerber), marioberges@cmu.edu (M. Berges), soibelman@usc.edu (L. Soibelman).

http://dx.doi.org/10.1016/j.aei.2014.09.004
1474-0346/© 2014 Elsevier Ltd. All rights reserved.

the NILM system, which is considered as one of the challenges in wide adoption of NILM systems.

To address these challenges and to facilitate the training process, one possible solution is to provide a pre-populated training data set, for which users can provide labels. Using the pre-populated training data set could potentially reduce the number of interactions, which in turn could improve user experience with the training process. Such training data sets provide the information about possible appliances operational states in a specific setting and thus provide the ground for smart communication with users (instead of communicating for all detected events). Pre-populating a training data set requires grouping the events' signatures (i.e., the signal characteristics in vicinity of events) into similar classes, which then could be labeled in the training process. Such a data set includes a number of examples for each class (e.g., the class of turn-on event of a television) and therefore, labeling one of those examples results in labeling the rest. This could be achieved by using clustering algorithms, which are used to group signatures into similar clusters, where the members of each cluster are similar to each other compared to the signatures in other clusters based on a predefined similarity metric. Although clustering is an unsupervised approach, for the majority of the clustering algorithms determining the representative number of clusters calls for *a priori* information. However, as noted, our objective is to reduce the challenges of the training process. The number of appliances' state changes, associated with the number of events, depends on the number of appliances and their operational states. Differences in number of appliances in each building, their internal components (e.g., a refrigerator could include compressor, defrosting module, and light fixture), and the fact that not all of the operational states (e.g., the refrigerator compressor operation and defrost operation) could be observed and detected by users compound the problem. Consequently, determination of the number of appliances state transitions (i.e., the number of clusters) is not a trivial task and it requires close monitoring of appliances by trained users. Furthermore, due to the ad-hoc nature of the signature space, determining any other generalized parameters (such as threshold values in stopping rules) for autonomous clustering could also be a challenging task. Accordingly, in this study, we propose a heuristic algorithm based on hierarchical clustering to achieve the objective of autonomous clustering of similar events' signatures, associated with appliances operational state transitions without the need for prior information.

The paper is structured as follows. First a research background covering NILM research, as well as clustering techniques, is presented in Section 2. Section 3 describes different components of our methodology in generating pre-populated training data sets, including our proposed heuristic algorithm. The experimental set up including the test bed and data acquisition for validation of the algorithm is presented in Section 4. The evaluation of algorithm performance, including the performance metric and the sensitivity analysis for exploring the effect of different features on the algorithm, is presented in Section 5. Section 6 concludes the paper by providing a summary of the study, followed by the future directions of the authors' research.

2. Research background

2.1. Event-based NILM research

Event-based NILM approaches comprise the main body of research since their introduction [5] three decades ago. A majority of the studies on NILM has focused on improvements to the methodologies, used for electricity disaggregation, through the introduction of different features [6] that represent appliances' operational characteristics to enhance the performance of pattern recognition algorithms. Features are the constituents of the events' signatures, which are the signal characteristics associated to events. Fig. 1 illustrates a short segment of real power time series and a number of events. Application of different features depends on signal resolution and consequently the data acquisition and processing system. Steady state features, that are commonly associated with low-resolution (e.g. 1 Hz) power metrics, are the features related to the segments of a power time series that are in the steady condition between the events (Fig. 1a).

Steady state real and reactive power metrics were used in Hart's seminal research [5] to improve the performance of the NILM system for appliances with a similar power draw. Since then, many studies used different combinations of steady state features. Real power, reactive power, power factor, RMS (Root Mean Square – used to identify the effective value of the alternating raw signal) current, and RMS voltage are the features that were used in some of the recent studies [7–9]. However, as illustrated in Fig. 1, the increase in signal resolution could potentially increase the information related to events and therefore various studies have focused on the application of the features associated with the high-resolution signals. These feature extraction methods have been coupled with different pattern recognition algorithms, such as nearest neighbor [10,11], neural networks [12,13], and Bayes classifiers [11]. Application of various feature extraction methods along with their pertinent methods for pattern recognition has been reviewed in [6].

As another approach, different sets of features were introduced in previous research, using generated electric noise as features for event detection and classification [14,10]. In [14], Patel et al. used a unique voltage transient noise, which is generated as a result of the operation of electromechanical components of a circuit (i.e., switching appliances on/off) and could be sensed at one outlet. The frequency component of the noise was used as a feature for pattern recognition. However, in [10], Gupta et al. pointed to the challenges such as the computational complexity and dependency of transient features on wiring and introduced the application of steady-state voltage noise features. Features are extracted from continuous electromagnetic interference (EMI) generated by appliances' switch mode power supply (SMPS). Modern appliances, with electronic components, generate high frequency EMI, which could be captured using high sampling frequencies about 500 kHz and higher. These features have been found to be transferable for a number of electronic appliances across different residential settings [10]. However, this approach could be used only for appliances that are equipped with SMPS components. Furthermore, the challenges related to the variation in appliances' manufacturing technologies and the assessment of temporal stability are yet to be explored.

Despite the above-mentioned improvements in the field of NILM, the methods for training data set provision remained undiscussed until recent years. Except for the sensor assisted approaches (for example [15–17]), Berges et al. [11,18] proposed a framework as a user-centered event-based NILM system to facilitate user interaction for training. In this framework, communication between a user and a NILM system has been facilitated through event detection and classification algorithms to continuously improve pattern recognition performance. However, continuous interaction between a user and a NILM system could potentially decrease the success of a NILM system set up. Accordingly, in this paper, we proposed a methodology for pre-populating training data sets through a heuristic clustering algorithm, explored the performance of the algorithm, and evaluated the effect of different feature extraction methods. The proposed algorithm enables unsupervised clustering to be utilized for facilitated training of an event based NILM approach.

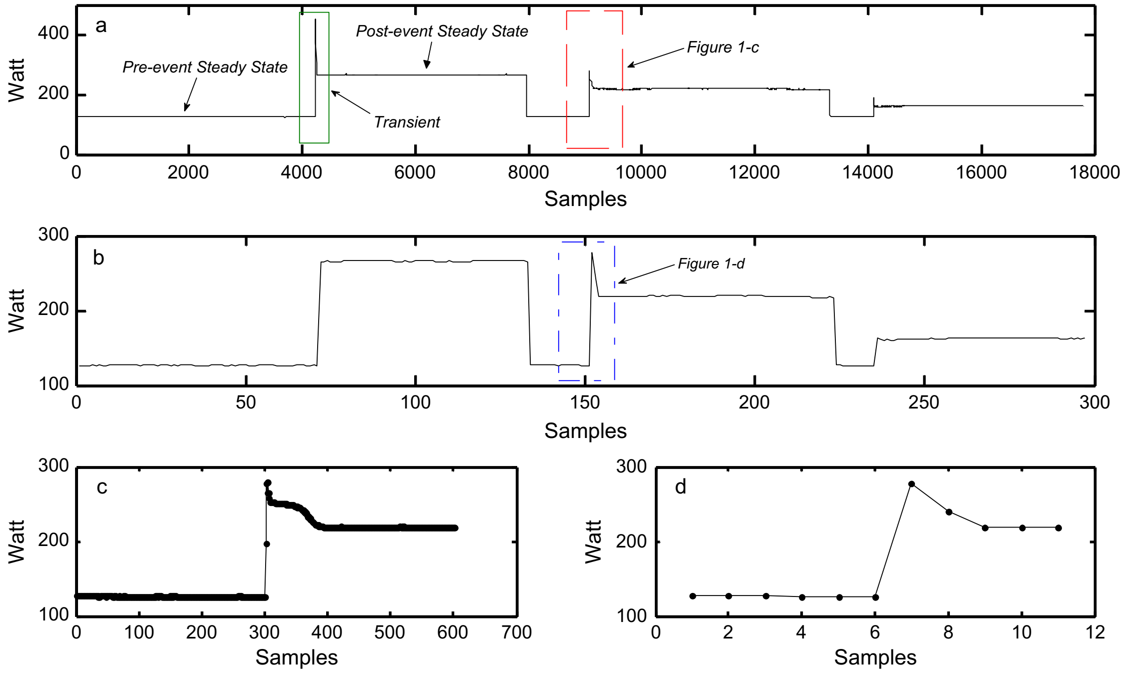

Fig. 1. Samples of real power time series at different resolutions: (a) a segment of the real power time series in 60 Hz; (b) the same segment of power time series down-sampled to 1 Hz; and (c) a sub-segment of the power time series in (a) highlighted by dash line five seconds before and five seconds after the event; (d) a sub-segment of the power time series in (b) highlighted by dash-dot line 5 s before and 5 s after the event.

2.2. Data clustering techniques

Although categorization of clustering methods could be carried out at different levels of granularity, as comprehensively presented in [19], one possible general categorization of the clustering techniques includes centroid-based clustering, connectivity-based (i.e. hierarchical), density-based, distribution based, and graph based. Centroid-based clustering algorithms, commonly represented with well-known *K*-means algorithm, require the number of clusters as *a priori* information and they could result in spherical representation of clusters, which might not be a desirable condition for all applications [20]. Distribution based clustering techniques consider a mixture of distributions with unknown parameters, representing the underlying clusters in the data, and solve the clustering as a model (i.e., distribution) identification problem. Acceptable solution is obtained through maximum likelihood principles by applying expectation–maximization (EM) algorithm [21]. Accordingly, although this class of techniques enables unsupervised determination of number of clusters and distribution parameters, they could be susceptible to overfitting; in other words, a more complex model could represent the data better. This drawback could be handled with constraints on the model complexity, which in turn may call for external information about the data and the underlying patterns.

Density-based clustering is carried out by determining dense regions of objects in the data space, separated by regions of low density. Therefore, in general form, this class of algorithms searches for the dense regions of the data space. DBSCAN [22] is a well-known algorithm in this class, which requires input information of region size and minimum number of elements in that region [20]. Mean shift [23] is another well-known and representative algorithm in this class, which is based on kernel density estimation and relies on (and sensitive to) the kernel size as the input parameter. In the graph-based category, spectral clustering is a common technique, which uses a weighted graph on the data and global eigenvectors of the similarity matrix, associated to that graph [24], to identify the clusters. Effective spectral clustering algorithms (e.g., [25]) with successful results on clustering non-convex clusters of data could be found in literature. These algorithms rely on the number of clusters as an input parameter; however, specific to spectral clustering, eigengap heuristic has been introduced, which uses the relatively large variations between the eigenvalues for selecting the number of clusters [26] and relies on determination of thresholds. In the hierarchical clustering category, algorithms yield nested grouping patterns at different levels of granularity and similarity levels [19]; this is achieved through a number of cluster tree generation algorithms including agglomerative and divisive algorithms [20] and different object and cluster dissimilarity metrics [19]. Number of clusters or pruning distance threshold is common metric for separation of the clusters in hierarchical algorithms [20]; however, we argue that the nested structure of the cluster tree could provide information about individual objects' connectivity at different levels of granularity, which could be used for autonomous clustering. Accordingly, in our proposed heuristic algorithm, we adopted the hierarchical clustering as the base approach to use the information contained in the structure of the cluster tree for autonomous clustering of electricity measurement data without provision of *a priori* input information.

3. Methodology for pre-populated training data set generation

In our approach, we use power metrics time series (also illustrated in Fig. 1) as the source of information. Transient features and higher harmonic contents of the current waveform provide more information that could potentially improve the algorithm performance. Accordingly, in our approach, the data acquisition and processing system is set to enable the application of these features. In the following sub-sections, data acquisition and processing system and the event detection algorithm are described, followed by the heuristic clustering algorithm.

3.1. Data acquisition and processing

The data acquisition system in this study is comprised of voltage and current sensors at the main feed (i.e. the main circuit breaker panel in a building unit – e.g. an apartment unit). The sampled current $i(t)$ and voltage $v(t)$ waveforms are then processed to generate the power time series. Due to the presence of non-linear loads and their associated current harmonics, the definition of reactive power is a challenging problem and there is no standard solution to it [27,28]. However, approximate approaches have been developed for this purpose [29–31]. In these approaches, based on the definitions of fundamental powers, a properly shifted harmonic voltage waveform could be used as a reference in order to compute power at a harmonic frequency. A periodic current waveform $\tilde{\imath}(t)$ could be represented in terms of continuous time Fourier series [29]:

$$\tilde{\imath}(t) = a_0 + \sum_{k=1}^{\infty} a_k \cos\left(k\frac{2\pi}{T}t\right) + \sum_{k=1}^{\infty} b_k \sin\left(k\frac{2\pi}{T}t\right) \tag{1}$$

in which T is the period, k is the harmonic index and a_k and b_k are the Fourier series coefficients at time t:

$$\begin{aligned} a_k &= \frac{1}{T}\int_t^{t+T} \cos\left(k\frac{2\pi}{T}\tau\right)d\tau, \quad b_k \\ &= \frac{1}{T}\int_t^{t+T} \sin\left(k\frac{2\pi}{T}\tau\right)d\tau \quad k \geqslant 1 \end{aligned} \tag{2}$$

Considering the fundamental component of the voltage waveform V, the current harmonic powers could be defined as:

$$P_k(t) = \frac{V}{2}a_k, Q_k(t) = \frac{V}{2}b_k \tag{3}$$

The Fourier series coefficients are scaled versions of the Fourier transform, evaluated at harmonic frequencies. In this study, the approximation approach for calculating power metrics presented by [30] as spectral envelope coefficient computation is adopted. Considering the digitally sampled current waveforms, the computation of spectral envelope coefficients are carried out using a Short-time Fourier Transform (STFT) on current and voltage waveforms, and results in $I(t)$ and $V(t)$. To account for the phase of the voltage relative to the window of the STFT, the phase shift ($\theta(t)$) between the first harmonic component of current, and voltage signal are obtained after the application of the STFT, and real and reactive power components [18] are:

$$P_k(t) = |I_k(t)| \cdot \sin(\theta(t)) \cdot |V_1(t)| \tag{4}$$

$$Q_k(t) = |I_k(t)| \cdot \cos(\theta(t)) \cdot |V_1(t)| \tag{5}$$

in which P_k and Q_k are kth real and reactive power quantities, I_k is the kth harmonic component of the transformed current waveform and V_1 is the first harmonic component of the transformed voltage.

3.2. Event detection algorithm

One of the fundamental steps in an event-based NILM approach is to detect the occurrence of events. In our approach, events are defined as sharp variations in the fundamental frequency component of real power time series that are associated with the appliances' state changes in a building. Since we intended to use the information in the transients between steady states in power time series, higher sampling frequency for data acquisition (the details are presented in Section 4) was used. Consequently, a probabilistic event detection algorithm was adopted in this study to avoid false positives due to the presence of noise. The event detection approach is based on the Generalized Likelihood Ratio (GLR) test that was proposed [32] and improved [33] in previous research studies.

3.3. Agglomerative hierarchical clustering

Given the signatures' (i.e., feature vectors) set for events, *FV*, has been extracted, the objective is to cluster similar signatures associated with events. Feature vectors are the vectors that represent the extracted samples for events. The collection of samples is called the feature space hereafter. As noted, in our approach, we propose a heuristic for hierarchical clustering to enable automated clustering of the feature vectors without provision of any *a priori* information as an input. Hierarchical clustering is a connectivity-based clustering approach that could provide clusters at different levels of granularity in a feature space. In this study, the agglomerative clustering [34], which is a bottom-up approach is used. Hierarchical clustering uses distance measures between feature vectors and connectivity measures between clusters to find all possible partitioning in a feature space in the form of a binary tree. The agglomerative hierarchical clustering algorithm starts with all feature vectors as singleton clusters. The pairwise distance matrix, D_{pw}, of all singleton clusters is generated. Clusters are merged to form binary clusters. Distance matrix is updated and iterative merging continues until only one binary cluster remains, which contains all of the sub-clusters and singleton clusters. Fig. 2 presents the algorithm for building the binary cluster tree.

For building the cluster binary tree, distance and linkage metrics are used to find the distance between the feature vectors, $dist_x(x_i, x_j)$, and clusters, $dist_c(c_i, c_j)$, respectively. The L_p-norm of the $x_i - x_j$ vectors could be used to evaluate different distance functions, including the common distance metrics, namely Euclidean (L_2-norm) and the Manhattan (city block) (L_1-norm) distances:

$$dist_x(x_i, x_j) = \|x_i - x_j\|_p = \left(\sum_{i=1}^{D} |(x_{mi} - x_{ni})|^p\right)^{\frac{1}{p}} \tag{8}$$

where D is the number of features in each feature vector and p could be any integer number to represent different distance metrics. Since clusters include more than one feature vector, the distance between clusters is defined in the form of linkage metrics. These linkage metrics could consider the distance between different objects in clusters, which results in different representation of linkage metrics including a single linkage, the distance between two closest feature vectors in two clusters, a complete linkage, the distance between two most distant feature vectors in each cluster, and an average linkage, the average of pairwise distances between feature vectors in each cluster. Depending on the nature of the data that is the subject of clustering, one of these linkage metrics might result in better performance. These linkage metrics are defined as follows:

```
Input: $FV$
$C_b \leftarrow \forall \boldsymbol{x} \in FV$
$N_C \leftarrow size(C)$
For each $\boldsymbol{x}_i$
    $D_{pw} \leftarrow dist_x(\boldsymbol{x}_i, \boldsymbol{x}_j)$
End
While ($\exists c \in C$ $covering\ all\ N_C\ clusters$)
    $C_b \leftarrow merge\ clusters(D_{pw})$
    $N_C \leftarrow size(C)$
    $D_{pw} \leftarrow dist_c(c_i, c_j)$
End
Output: Binary Cluster Tree
```

FV: the set of events' signatures — N_C : the number of clusters
C_b: the binary cluster set — D_{pw}: the pairwise distance matrix

Fig. 2. The agglomerative hierarchical binary cluster tree generation algorithm.

$$dist_{single}(c_i, c_j) = \min_{x_m \in c_i, x_n \in c_j} d(x_m, x_n) \tag{9}$$

$$dist_{complete}(c_i, c_j) = \max_{x_m \in c_i, x_n \in c_j} d(x_m, x_n) \tag{10}$$

$$dist_{average}(c_i, c_j) = \frac{1}{N_{c_i} N_{c_j}} \sum_{x_m \in c_i} \sum_{x_n \in c_j} d(x_m, x_n) \tag{11}$$

where C_k is the kth cluster and N_{C_k} is the number of feature vectors in cluster k.

3.4. Heuristic algorithm

The algorithm in Fig. 2 generates a binary cluster tree. Fig. 3 illustrates the generated binary tree (i.e., the dendrogram of the tree) using the power signatures from a residential setting. The details of the experimental set up and data processing are provided in Section 4. The visualization is based on the signatures' data obtained for all of the turn-on events on one phase of the residential setting.

As Fig. 3 shows, the dendrogram represents the structure of the feature vector space. This dendrogram presents the cluster tree with one hundred leaf nodes. Each leaf node may include a number of sub-clusters. The inverse U shape connections (highlighted for two clusters in Fig. 3 with a dashed dot line) represent a binary cluster that is the result of merging two sub-clusters. The root node covers all feature vectors and sub-clusters. The height of the connecting lines shows the distance between two clusters. As noted, once the tree is generated, it contains clusters at different levels of granularity. Grouping the feature vectors into clusters could be carried out by pruning the tree based on criteria such as number of clusters, and a distance threshold. As discussed earlier, providing the number of clusters is not a viable approach since it requires the knowledge of unique appliance state transitions. A common approach in estimating number of clusters for any clustering algorithm is to evaluate clustering through elbow phenomenon [35]. In elbow phenomenon, the number of clusters is estimated based on the relative variation of the within-cluster dispersion or between-cluster distances for different number of clusters (k). The optimum k is selected when changes in these measures are flattened [36]. In our heuristic, we use the elbow phenomenon by considering the between-cluster distances that are calculated through the agglomerative tree generation. Therefore, an inconsistency threshold could be used for the purpose of pruning. When the heights of two successive links are close together, the links are considered consistent and there is no clear distinction between the clusters. In order to find the clusters, an inconsistency threshold has to be determined. In this study, the global inconsistency at the cluster tree level is used to avoid pruning based on a local inconsistency threshold. The local inconsistency could happen due to the multistate nature of the signatures.

As shown in Fig. 3, a distance threshold gives a horizontal pruning level in the tree for separating clusters. By moving the distance threshold (represented by the dashed grid lines in Fig. 3) clusters at different levels of granularity could be obtained. We use the cluster tree structure characteristics to determine the distance threshold. As the structure of the tree indicates, the difference between distances are small close to the leaf nodes and by moving towards the root node the rate of between-cluster distance escalates. The distance between clusters represents the similarity/dissimilarity. Therefore, we use the distance growth rate as a metric for pruning of the tree. Fig. 4a illustrates the variations in the distance measure between the binary clusters along the tree – moving from leaf nodes to the root.

The distance measure, *dist* (Fig. 4a), is obtained using the distance and linkage metrics, as described in eqs. (8)–(11). The gap metric, δ, is defined as follows:

$$\delta = dist_{i+1} - dist_i \tag{12}$$

As illustrated in Fig. 4a, at a point on the cluster tree, the distances start growing very fast. This is the point of pruning the tree into constituent sub-clusters. In order to find the associated location on the tree, the gap measure curve, δ, does not provide enough information. As it could be observed in Fig. 4b, finding the pruning level based on the gap metric calls for introducing a threshold. Therefore, we use the gap metric growth rate, defined as follows:

$$\Delta = (\delta_{i+1} - \delta_i)/\delta_i \tag{13}$$

The distance threshold for pruning the tree, τ_d^*, is obtained by:

$$\tau_d^* = \underset{l_t}{\operatorname{argmax}} \Delta \tag{14}$$

where l_t represents different levels of the tree. However, as noted for inconsistency pruning threshold, the local inconsistency in the

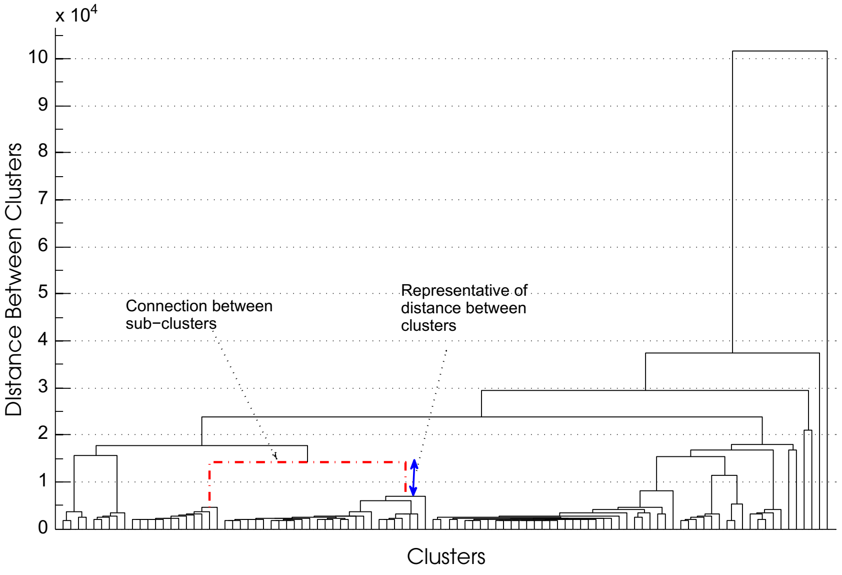

Fig. 3. Dendrogram of binary cluster tree for feature vectors of all turn-on events in a data set from a residential setting; the tree shows up to one hundred leaf nodes.

Please cite this article in press as: F. Jazizadeh et al., An unsupervised hierarchical clustering based heuristic algorithm for facilitated training of electricity consumption disaggregation systems, Adv. Eng. Informat. (2014), http://dx.doi.org/10.1016/j.aei.2014.09.004

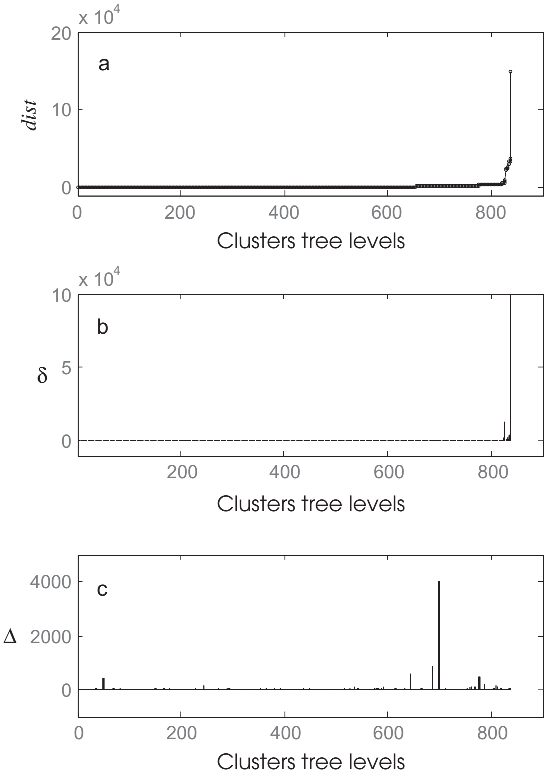

Fig. 4. Variation of different metrics - used for detecting the structure of cluster tree – along the cluster tree from leaf nodes to the root; (a) between cluster distances along the tree; (b) slope measure representing distance growth rate (Eq. (12)); and (c) slope growth measure representing slope growth rate (Eq. (13)).

tree structure might result in a τ^*_d value in the lower parts of the cluster tree. As it could be seen in Fig. 4c, the maximum of Δ could occur in a lower level compared to what a human user observes. As noted, this local inconsistency could be related the multistate nature of the feature space. Consequently, this could result in a number of clusters that does not represent the natural (i.e., compatible with the appliances' state transitions) separation of the data in the feature space. As illustrated in Figs. 3 and 4, to address this issue, we argue that the pruning distance threshold needs to be sought for in the upper parts of the cluster tree (if it exists), close to the root node. The question is how to find the border. In order to find the upper part of the tree, again the information contained in the structure of the tree could be used. Fig. 5a shows the histogram of the δ values for the cluster tree. In order to find the upper segment of the tree, where the distance growth rate is higher, we proposed a histogram segmentation method. As the histogram illustrates, histogram bins with higher δ values are less populated and bins with lower δ values are highly populated. Therefore, the segmentation point of the tree is selected as the point, where there is balance in weighted sum of the δ values in the histogram. The δ segmentation threshold is obtained by finding the minimum value of absolute difference of the weighted sum of δ values in the histogram:

$$\tau^*_\delta = \underset{c\in C}{\operatorname{argmin}} \left| \underbrace{\sum_{i=1}^{p\in w^l} c_i n_i - \sum_{j=1}^{q\in w^r} c_j n_j}_{\Delta_{hw}} \right| \tag{15}$$

where c_k is the center of the kth histogram bins, n_k is the number of elements in each histogram bin, C is the set of all histogram center values, p is the number of histogram bins in the w^l, and q is the number of histogram bins in the w^r. w^l and w^r are the two contiguous summation windows with a common side that moves along the histogram of δ. Once the τ^*_δ is obtained, the upper part of the cluster binary tree is determined by moving upwards from the leaf nodes and finding the level at which, the median of the δ values from that point to the root is greater than τ^*_δ. This remaining set is called δ^u and is used in Eqs. (13) and (14). In this process, the number of histogram bins is considered to be equal to the size of the δ set. Fig. 5 illustrates the histogram segmentation process for the cluster tree. If the point of segmentation does not exist (for example in case of clustering very similar feature vectors) the histogram segmentation approach results in δ^u equal to δ.

Fig. 6a shows the dendrogram of the cluster tree for a sample feature space. This is the same dendrogram presented in Fig. 3, but in this dendrogram, the number of leaf nodes was increased so that the multi scale nature of the tree could be observed. As noted, data processing [37] showed that due to the nature of the signatures in the feature space, the cluster tree has a multi-scale structure. By evaluating the dissimilarity measure over the entire feature space, the information contained in sub-components of the tree (sub-trees) does not contribute to the information retrieval for pruning threshold determination. Although the leaf nodes in Fig. 6a do not show all of the singleton clusters, the scale effect in the structure of the tree is observed. Accordingly, by evaluating the pruning threshold and clustering the feature vectors over the entire feature space, part of the feature space remains as one cluster of feature vectors that might be associated to multiple appliance state transition classes. In the context of this paper, this cluster is defined as the *residual cluster*. Accordingly, we use recursion in the clustering process to account for the scale effect in the cluster tree. In other words, once the clusters in each scale are determined, the residual cluster introduces a different structure for the cluster tree (as it has been illustrated in Fig. 6b–d), in which the natural separation of the feature vectors of smaller scales is amplified, thus resulting in the application of our proposed pruning threshold determination technique at different scales. Fig. 6b illustrates the residual cluster after pruning the cluster tree, shown in Fig. 6a. This pattern is recursively repeated for Fig. 6c and d. Each recursion provides the opportunity for further separation of the data. However, as we approach the lower scale the dissimilarity between clusters is faded (as it could be observed in Fig. 6d). Since the pruning at each recursion is performed on a horizontal level, in each recursion, a number of clusters with one or two feature vectors are generated. Accordingly, in order to control the number of clusters, these clusters are eliminated in our approach.

A dispersion measure for all of the clusters in the feature space is used to determine the residual cluster. The cluster with maximum dispersion value is determined as the residual cluster. The dispersion is measured by using the dispersion index:

$$I_d = \frac{\sigma^2}{\mu} \tag{16}$$

in which, σ^2 is the vector containing the element-wise variance of the feature vectors in each cluster and μ is the vector of element-wise mean of the feature vectors in each cluster. The algorithm recursively searches the tree for natural separations in the feature space until it reaches to a point that no more separation in the residual tree could be achieved (i.e. the number of clusters in the recursion is equal to 1). The pseudo code and the corresponding flowchart of the proposed heuristic algorithm are presented in Fig. 7. In this figure, the agglomerative hierarchical binary cluster tree generation algorithm, presented in Fig. 2, is used as the aggTree function.

Please cite this article in press as: F. Jazizadeh et al., An unsupervised hierarchical clustering based heuristic algorithm for facilitated training of electricity consumption disaggregation systems, Adv. Eng. Informat. (2014), http://dx.doi.org/10.1016/j.aei.2014.09.004

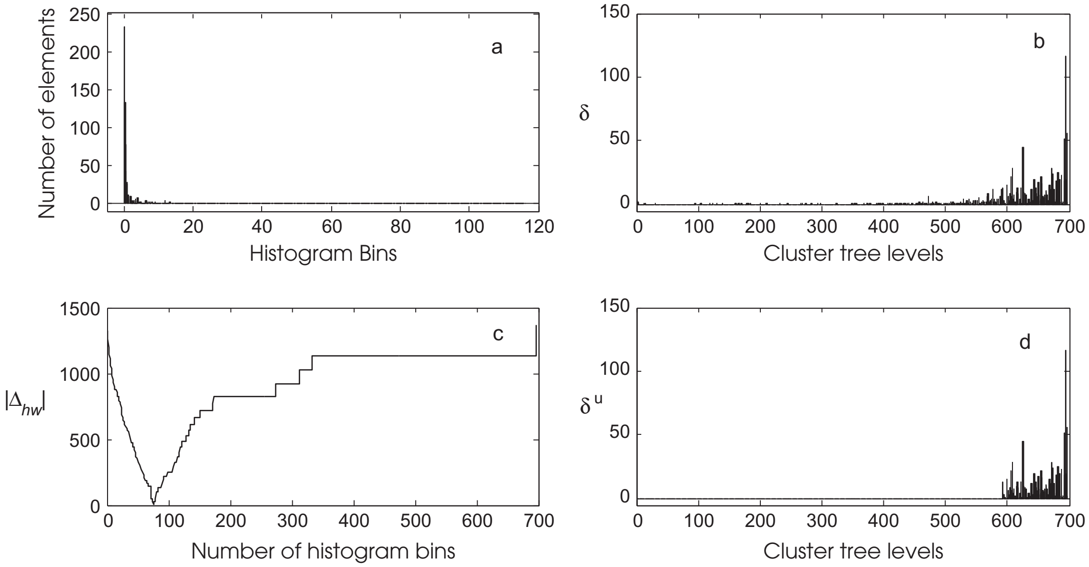

Fig. 5. Distance growth rate histogram segmentation process; (a) the histogram of the δ values of the cluster tree; (b) variation of δ values along the cluster tree; (c) variation of the $|\Delta_{hw}|$ (the absolute difference of the weighted sum of δ) along the tree; and (d) variation of δ^u values along the cluster tree.

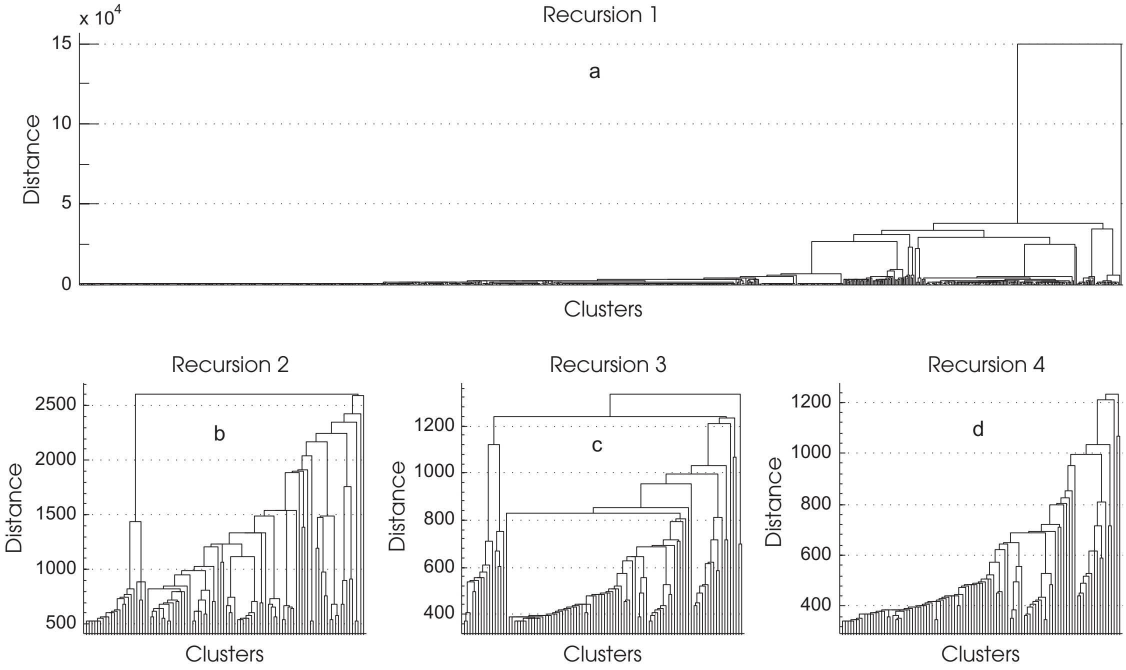

Fig. 6. Dendrogram of the cluster tree at different scales through recursive clustering; sub-graphs (a)–(d) represent the dendrogram of the cluster tree for the residual feature space.

3.5. Feature extraction

As noted, signal characteristics in the vicinity of event locations are extracted as feature vectors. Although signatures could be extracted from current or voltage waveforms, in this study, only the power metric signatures (i.e., the signatures that are extracted from processed real or reactive power time series) are used as the features. In addition, since transient features provide more information that may improve the separation of the signatures in the feature space, these features were used for evaluating the algorithm performance. As described in Section 3.1, due to the presence of non-linear loads, higher harmonic components of the current waveform and consequently the power metrics could potentially provide more information to enhance the cluster separation. Accordingly, the feature vectors are considered as power time series segments close to the event locations. These segments are extracted using two windows, one before, w^b and one after, w^a, the event index. Feature vectors are extracted as vectors with elements of real power segment followed by the reactive power segment. The *basic feature vector* is the feature vector of real and

```
Input: $FV$
$N_{cr}$ ← 0
$C$ ← $\emptyset$
$FV^r$ ← $FV$

Function RH ($FV^r$)
    While $N_{cr} \neq 1$
        $T_c$ ← aggTree($FV^r$)
        $dist$ ← $extract\ dist\ from\ T_c$
        $\delta$ ← $claculate\ \delta\ using\ dist$
        $(N_h, C_h)$ ← $histogram\ (\delta)$
        $\tau_\delta^* = \text{argmin}_{c \in C_h} \left|\sum_{i=1}^{p \in w^l} c_i n_i - \sum_{j=1}^{q \in w^r} c_j n_j\right|$;
                        $c \in C_h, n \in N_h$
        $\delta^u$ ← $update\ \delta\ using\ \tau_\delta^*$
        $\Delta$ ← $calculate\ \Delta\ using\ \delta^u$
        $\tau_d^* = \text{argmax}_{l_t} \Delta$
        $C^r$ ← $prun\ the\ tree\ to\ clusters\ using\ \tau_d^*$
        $C$ ← $C \cup C^r$
        $N_{cr}$ ← $size(C)$
        $FV^r$ ← $\text{argmax}_{c \in C} I_d(c)$
        RH ($FV^r$)
    End

End Function
```

FV: The set of events' signatures
FV^r: The set of residual feature vectors
N_{cr}: The number of clusters in each recursion
C: The set of clusters

T_c: The cluster tree
N_h: Number of elements in histogram bins
C_h: The values for histogram centers
C^r: The set of clusters in each recursion

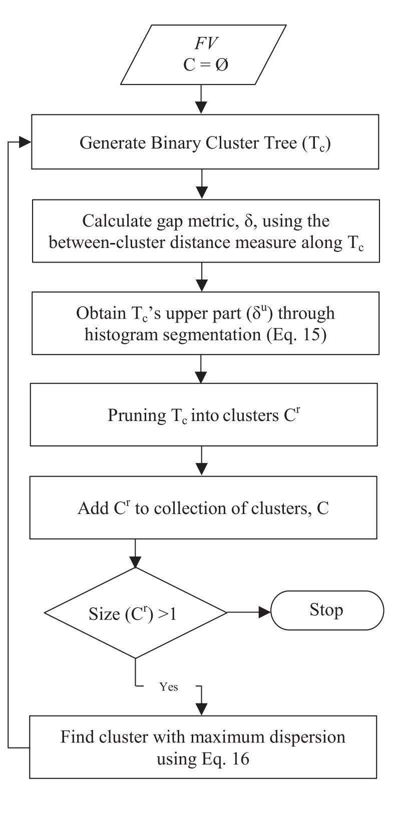
FV
C = Ø
Generate Binary Cluster Tree (T$_c$)
Calculate gap metric, δ, using the between-cluster distance measure along T$_c$
Obtain T$_c$'s upper part (δ^u) through histogram segmentation (Eq. 15)
Pruning T$_c$ into clusters C^r
Add C^r to collection of clusters, C
Size (C^r) >1
Stop
Yes
Find cluster with maximum dispersion using Eq. 16

Fig. 7. Recursive hierarchical clustering (RH) algorithm – pseudo code and the corresponding flowchart.

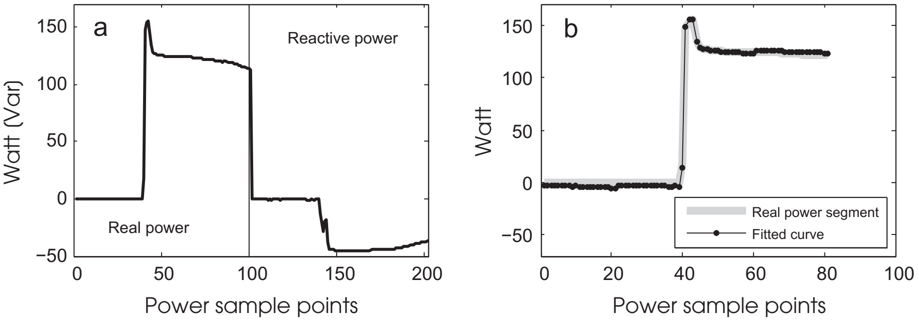

Fig. 8. Appliance state change (event) feature vectors examples; (a) the basic feature vector, comprised of real and reactive power time series segments and (b) the modeled feature vector through linear regression analysis.

reactive power for the fundamental frequency. Fig. 8a illustrates a basic feature vector. The feature vectors in their general form are as follows:

$$x_n = \{p_1[n], q_1[n], \ldots, p_k[n], q_k[n]\}, k \in \{1, 2, \ldots, K\} \tag{17}$$

where p and q are the real and reactive power components and k is the number of harmonic components, used for feature extraction. K is the total number of harmonics, which is up to the first nine harmonics of the fundamental frequency in this study. The feature vectors could be comprised of all, even, or odd harmonics of the fundamental frequency.

In order to explore the effect of reducing the noise in the signatures, in this study (similar to the approach used in [11] for classification) we also use regression analysis to model the transients and use lower-dimensional feature vectors. Due to the complexity of the transient shapes, higher order regression analysis using basis functions is used. As illustrated in Fig. 8, the combination of the polynomial and Fourier basis functions could be used to model the shape of transients with high accuracy while the accuracy declines in case of using each individual basis function. In this approach, the signatures are modeled as follows:

$$f(x) = \sum_{i=1}^{r} \alpha_i x^i + \sum_{j=1}^{s} \left[\beta_j \sin\left(\frac{2\pi jx}{T}\right) + \gamma_j \cos\left(\frac{2\pi jx}{T}\right)\right] \tag{18}$$

where T is the period, r is the highest degree of polynomial and s is the number of Fourier basis functions. Coefficient vector, namely $\{\alpha, \beta, \gamma\}$ is used as the feature vector that represents the shape of the feature vector.

4. Performance evaluation of the algorithm

In this study, the algorithm performance for autonomous clustering of appliances signatures was evaluated through an experimental study in a residential setting to account for real world situations, where aggregated data with the presence of noise is captured. The following sections present the experimental set up and data set preparation, the metrics for evaluating the algorithm performance, and the findings and discussion.

4.1. Experimental set up

To capture the transients and the higher harmonic contents of the current waveform, a high frequency sampling data acquisition system was used. The data acquisition system included a National Instrument A/D card, NI 9215 DAQ, with the capability of sampling at 100 kHz. Considering the harmonic contents of the load and the Nyquist sampling theorem, a sampling frequency of more than 1 kHz was needed to capture the first 9 harmonics. The selected A/D card has the capability to gather a wide range of harmonic contents. At the sensing node, one voltage and two current transformers were used. Since the voltage waveforms in the split phase system (the typical system in the U.S. – the description of the split phase system could be found in [5]) have 180° phase difference, the voltage measurement is performed on one phase only. For capturing the voltage waveform, Pico TA041–25 MHz ± 700v differential probe was used. For current waveforms, the measurement was performed on each phase using Fluke i200 AC current clamps. The analogue signals captured by these sensors were digitized through NI DAQ and stored/processed on a local PC computer.

An important part of the data acquisition system is the ground truth data collection. One way to acquire the ground truth is to follow the activity of the appliances and prepare a log of all events. This process is very complicated, specifically in cases that the number of appliances' state transitions is relatively large. Therefore, the ground truth data was obtained by plug level metering for appliances that were plugged into a receptacle and by ambient light sensors for lighting fixtures. Since the aggregated power time series has a relatively high resolution, in order to avoid errors in using ground truth labels on the aggregated signal, high-resolution power data at the plug level have to be provided. Therefore, in this study, Enmetric Powerports [38], off-the-shelf plug level meters were used. These sensors provided 1 Hz power signal sampling rate through their integrated application programming interface (API). Collecting the ground truth for lighting was performed using custom made sensing nodes comprised of Linksprite DiamondBack microcontrollers, equipped with a WiFi module and AMBI™ light intensity sensors, which transferred their data through a local WiFi network, established in the test bed.

The test bed is an occupied residential unit (a one bedroom apartment), in which all of the appliances and lights were monitored with the above mentioned appliance level monitoring system. The main circuit breaker was equipped with current and voltage sensors. The data acquisition was carried out for two weeks, while the occupants were using the appliances. The collected voltage and current waveforms were processed into power metrics, (i.e., real and reactive power time series) for the first nine harmonics of the current waveform using Eqs. (4) and (5). At the appliance level, the collected data included the power time series for plug level meters and light intensity signal for the lighting fixtures. The signals collected at appliance level were matched with the real power times series obtained at the main feed and the events were labeled for the validation of the algorithm. Accordingly, each appliance's state transition was labeled with a unique label so that the evaluation of the algorithm performance could be accurate. Table 1 shows the list of appliances, as well as, the labels associated to their different internal operational states (e.g., Refrigerator has two turn-on states (11101, 11103) and two turn-off states (11102, 11104). These states are related to the operation of compressor and defrost components).

As Table 1 depicts, the data set is a fully labeled data set with all appliances' state transitions uniquely labeled. The appliances with two sub-labels (e.g., toaster turn-on (16301) events and its turn-off (16302) events) are the ones with binary operational states. However, these binary operational states for an appliance such as hair dryer could still have different signatures because of the difference in power draw in various modes of operation (i.e., low, medium, high). Since in a NILM system, the sign of the signature (whether the signature is related to a turn-on or turn-off event) could be evaluated and the analysis could be independently carried out on each phase, the data were separated for turn-on and turn-off events. Fig. 9 illustrates all of the turn-on events on one of the phases (which is called phase A, hereafter) in the test bed. As this figure shows, the feature vectors were extracted using the event detection algorithm, explained in Section 3.2. For the feature extraction process, the before and after event windows, w^b and w^a, were set to be 40 and 60 samples with power resolution of 60 Hz (two third of a second before and one second after the event), respectively. Since the event detection algorithm was used for the labeling process, a number of false positives were also detected. The label 0 in the data set is associated to those false positive or unimportant (change in power draw less than 20 W) events that do not represent actual events on power time series. The label 200 in Fig. 9 is for the lighting fixtures' feature vectors that could not be labeled by a unique label/sub-label. The remaining labels are provided in Table 1.

4.2. Performance evaluation metrics

Different performance evaluation metrics have been introduced for clustering algorithms. In one category, the performance metrics are distance based such as the Dunn index [39], which uses the ratio between the minimum distance between clusters (inter-cluster) and maximum distance between feature vectors in each cluster (intra-cluster). In our study, since the labels of the feature vectors are known, the evaluation is carried out externally following a procedure similar to the one used for supervised learning problems by matching the ground truth labels and the labels of the clusters. However, since the number of clusters is not necessarily equivalent to the number of sub-labels (see Table 2 as an example), in order to use a confusion matrix and consequently *F*-measure metrics we use a mapping approach in addition to a complementary metric for cluster quality evaluation. The outcome of the clustering algorithm could be represented in a matrix similar to a confusion matrix. The evaluation metrics are described through the presentation of the confusion matrix resulted from running the heuristic algorithm on the data, shown in Fig. 9. Table 2 presents the association matrix, which shows the association between the clusters assigned by the algorithm and the sub-labels of the feature vectors.

In Table 2, the first row represents the numerical labels autonomously assigned to the clusters by the algorithm. The first column shows the labels of the ground truth data classes and the last column shows N_o values, which are the numbers of feature vectors associated to each ground truth label. The association matrix is mapped to a conventional confusion matrix as follows:

- The cluster labels are defined as the class label associated with the majority of the feature vectors in a cluster. Once all the clusters are labeled, the clusters with the same labels are merged and labeled with associated class label. In merging the clusters, the entire column related to each cluster is moved to be merged.

Table 1
The list of the appliances in the test bed and the labels for their associated operational states.

Appliance	Phase	Label code	Sub-label codes
Refrigerator	A	111	11101,11102,11103,11104
AC	A, B	180	18001, 18002, 18003, 18004, 18005, 18006
Toaster	A	163	16301, 16302
Kettle	A	162	16201, 16202
TV	A	129	12900, 12901, 12902, 12903, 12904, 12905, 12906
Iron	B	182	18201, 18202
Hair Dryer	B	181	18101, 18102
LCD monitor	B	122	12201, 12202
Laptop	B	120	12001, 12002
Desk Lamp	B	151	15101, 15102
Washing Machine	B	183	18301, 18302, 18303, 18304, 18305, 18306, 18307, 18308, 18309, 18310
Bathroom Light	A	145	14501, 14502
Bedroom Light	A	144	14401, 14402
Kitchen Light 1	A	141	14101, 14102
Kitchen Light 2	A	143	14301, 14302
Kitchen Fan Light	A	142	14201, 14202
Unknown	A, B	300	–

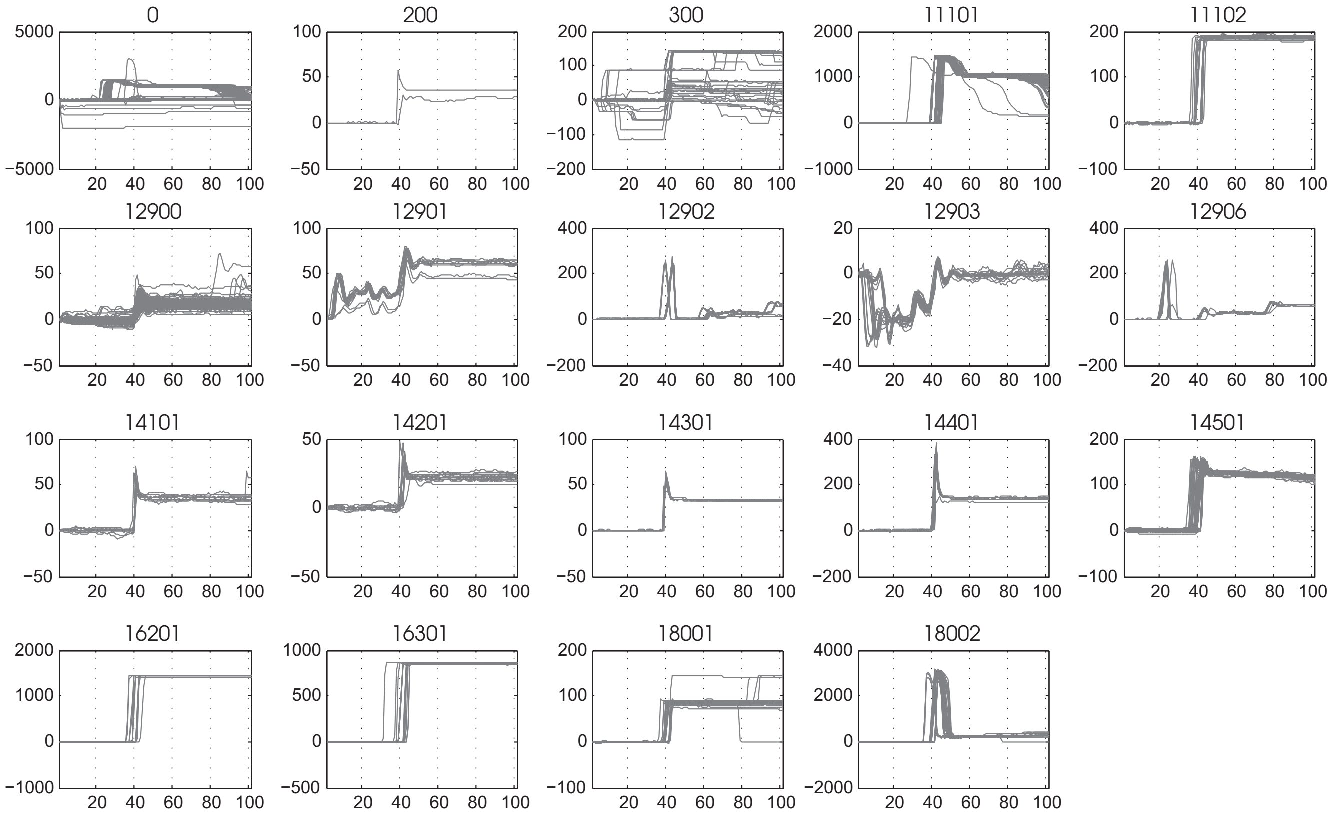

Fig. 9. Feature vector classes for phase A turn-on events, manually labeled by a user referring the ground truth sensors' data.

- Upon the completion of the mapping, precision, recall and the *F*-measures were calculated for the confusion matrix.

Another important factor in evaluation of the algorithm performance is the number of clusters associated with each ground truth label, as well as the density of the clusters. In this context, this factor is called the cluster quality. As noted in Section 3.4, in each recursion, the clusters with one or two feature vectors are removed in order to control the number of clusters. Accordingly, in evaluating the effect of feature extraction methods and the distance and linkage metrics, the number of clusters and eliminated feature vectors play an important role. The ideal condition is to have the same number of clusters as the number of ground truth labels, with each cluster containing all original ground truth feature vectors. This condition is interpreted as the ideal cluster quality. Two rows of the association matrix (presented in Table 2) are described in detail to shed more light on the concept of cluster quality. The row for label 11101 represents the class of signatures associated with the refrigerator compressor turn-on events. This class of signatures covers 126 signatures. The ideal condition for us is to have all these signatures in one cluster. However, in the clustering process, the signatures were clustered into two clusters (with labels 5 and 16) and five signatures were ended into small clusters and thus were ignored. On the other hand, the class represented by label 14401 covers 31 signatures of a lighting fixture. In the clustering process, 30 signatures were clustered under cluster label 6 and 1

Table 2
The matrix representing the association between cluster labels and ground truth labels (association matrix).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	N_o
0												9		3																		16	41
200																												1				1	2
300			4								4								1				3		3	5						1	36
11101					4											117																	126
11102							37																										37
12900																												1		3		171	180
12901																						12											14
12902																								6			19						25
12903																													18				20
12906																				10													12
14101																												11					12
14201																															12	8	22
14301																												8					8
14401						30												1															31
14501																		139															139
16201				7											5																		15
16301																	19																20
18001																			3		36					1							41
18002	5	3						3	8	3			12																				58

signature was clustered with the signatures from another class (14501) as a false negative and no signature was clustered into small clusters. Therefore, the clustering result for class label 14401 is almost ideal while the result for class label 11101 could be better. Although the ideal situation might not happen (due to the variation of signatures – refer to Fig. 9 for illustration of these variations), it is used in this study as the benchmark for the cluster quality analyses. The evaluation of clusters' quality could be carried out qualitatively by looking at the association matrix. However, to facilitate the comparison, we introduce a cluster quality (CQ) graph as a visual metric. Fig. 10 shows the CQ graph for the data presented in Fig. 9 using the basic feature vectors in addition to city block distance and single linkage metrics. The CQ graph is presented on the left side. The *x* axis shows the number of clusters and the *y* axis shows the density of clusters. On the right side, a portion of the CQ graph has been zoomed in for clarification. The interpretation of the CQ graph could be carried out by looking at the distance between the markers and the ideal point of (1, 1). The farther the markers lie from the ideal point, the lower the quality of the clustering is. Fig. 10 shows a relatively high quality of clustering since most of the classes have high densities with one or two clusters.

In addition to the CQ graph metric for comparison, to provide a quantitative metric to be used for the comparison between different features and distance metrics, we introduce the cluster quality index (*CQI*) to represent the information contained in CQ graph.

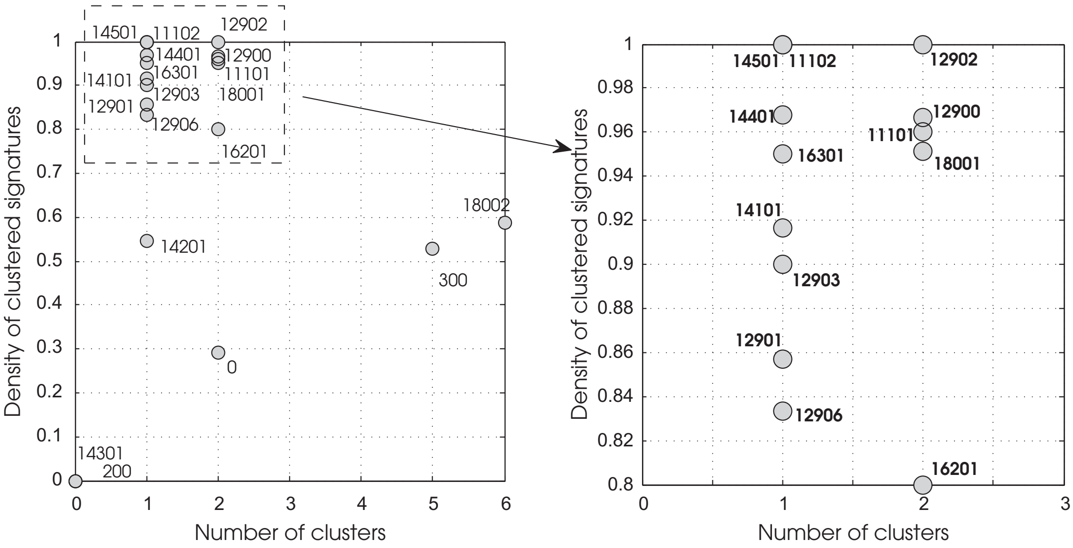

Fig. 10. Cluster quality (CQ) metric representation; the left side shows the results for all clusters and the right side shows the congested area in a larger scale.

Please cite this article in press as: F. Jazizadeh et al., An unsupervised hierarchical clustering based heuristic algorithm for facilitated training of electricity consumption disaggregation systems, Adv. Eng. Informat. (2014), http://dx.doi.org/10.1016/j.aei.2014.09.004

$$CQI = \frac{1}{N_l}\sum_{1}^{N_l}\frac{\rho_l}{N_{cl}^l} \tag{22}$$

$$\rho_l = \left(\frac{N_r^{FV}}{N_a^{FV}}\right)_l \tag{23}$$

in which, ρ_l is the density associated with feature vectors related to each class label and is defined as the ratio between the number of remaining feature vectors, N_r^{FV}, and the actual number of feature vectors for corresponding label, N_a^{FV}; N_{cl}^l is the number of clusters for each label l, and N_l is the number of labels. Although absolute value of the *CQI* could be used for evaluation, lower values of the *CQI* are not necessarily an indication of poor performance. For example, if in a clustering procedure, ρ_l is equal to one for all labels and the algorithm returns one cluster for half of the labeled feature vectors and two clusters for the other half, a *CQI* value of 0.75 is obtained. Although this is a desirable condition, the absolute *CQI* is relatively low. Accordingly, it is emphasized that in this study, the relative *CQI* values are used for the comparison between different feature extraction methods and distance metrics.

4.3. Algorithm evaluation

The evaluation of the algorithm was carried out for different feature extraction methods and for different distance and linkage functions. As noted in Section 4.1, the feature extraction was carried out using the event detection algorithm to account for realistic variations in feature vectors, which happen due to the differences in detected event indices, detected by the algorithm. The feature vectors with label 0 could be eliminated from the feature space before clustering. In order to remove these feature vectors the ΔP, change in first harmonic real power draw at the point of event, was calculated. If |ΔP| is less than a certain threshold, the associated feature vector could be eliminated. However, the difference in appliances power draw and the fact that the location of the detected events on real power time series do not necessarily coincide with the actual point of events, finding the ΔP threshold is a challenging task for some of the events. Accordingly, although a large portion of the feature vectors with label 0 could be excluded from the cluster analysis, complete elimination of zero labeled feature vectors could result in the loss of some of the important events. Consequently, a portion of these feature vectors remained in the feature space.

Since the dissimilarity measures between clusters and therefore the structure of the feature space play an important role in the performance of the algorithm, 5-fold cross validation was used in the evaluations. In addition, to avoid biased conclusions the results were reported as average values across five rounds of 5-fold cross validations. The reported performance metrics (presented in the following tables) are then mean values of the performance metrics. Combination of all distance and linkage functions (Eqs. (8)–(11)) and different feature extraction methods could result in a large number of combinations for algorithm evaluation. Therefore, the effect of distance and linkage functions was evaluated using a basic feature vector (i.e., $[P_1, Q_1]$) on phase A turn-on events. Euclidean and city block distance metrics along with three linkage metrics were taken into account. Accordingly, six combinations were evaluated and the results are presented in Table 3. CQI_m is modified *CQI* for which the data related to feature vectors with labels 0 and 300 were removed. The feature vectors in these two classes could be associated to multiple clusters and thus their data could result in inaccurate reduction of *CQI*. In order to determine the optimum results, P_I or the index for proximity to ideal case was defined. P_I is defined as the distance between the pair of *F*-measure and CQI_m to point (1, 1). The case with minimum P_I is the optimum condition. As the results in Table 3 shows, the *single linkage* along with the *city block distance* metrics resulted in the better combination. Since the objective is to facilitate training by reducing the number of true clusters, which represent the appliances state transition, higher *CQI* value is desirable.

The effects of the feature extraction methods were evaluated using the city block distance and single linkage metrics. Five cases were compared: (1) the real and reactive power of the fundamental frequency component (basic feature vectors), (2) the basic feature vectors using a kernelized distance function, (3) the real and reactive power for the first nine harmonics, (4) the real and reactive power for the first five odd harmonics, and (5) the features representing the reduced noise version of the transients using higher order linear regression (Eq. (18)). The case no. 4 was considered due to the fact that odd harmonic components of the current waveform contain more information and eliminating even harmonics could potentially improve the algorithm performance. Since case no. 5 represents noise-reduced features for transients, the dimension of feature vectors is reduced, and different *r* (i.e., the degree of polynomial basis functions) and *s* (i.e., the number of Fourier basis functions) values were used to extract features. The regression coefficient vectors for real power segment and reactive power segment were combined and used as the feature vector. Fig. 11 shows the sensitivity analysis of the algorithm performance for different model parameters on the data from phase A turn-on events. As this figure shows, changes in the values of *r* and *s* does not change the results dramatically. However, slightly better results were obtained for *r* = 1 and *s* = 5 – 20. For case no. 2 the application of kernelized distance (Eq. (24)), using polynomial kernel function with degree 1, was investigated to explore whether it is effective in magnifying the dissimilarities. The kernelized distance was observed to improve the results in some of the individual runs and therefore, we explored their effect through cross validation.

$$\begin{aligned} d_k(x_m, x_n) &= x_m^T x_m + x_n^T x_n - 2x_m^T x_n \\ &= k(x_m, x_m) + k(x_n, x_n) - 2k(x_m, x_n) \end{aligned} \tag{24}$$

Table 4 shows the results of the analyses for the turn-on events on Phase A. The best results were obtained in cases, where basic feature vectors and higher harmonic contents were used (i.e. cases no. 1, 3, and 4). For this part of the data, the addition of the higher harmonic contents did not improve the performance. However, using the odd numbered harmonics showed better performance.

Table 3
Performance of the heuristic algorithm for turn-on events on phase A for different linkage and distance metrics (average values across five 5-fold cross validation results).

Case no.	Distance	Linkage	Precision	Recall	*F*-Measure	*CQI*	CQI_m[a]	P_I
1	City block	Single	0.91	0.93	0.90	0.55	0.60	0.41
2	Euclidean	Single	0.91	0.95	0.91	0.45	0.49	0.51
3	City block	Average	0.88	0.91	0.86	0.48	0.53	0.49
4	Euclidean	Average	0.92	0.92	0.89	0.41	0.45	0.56
5	City block	Complete	0.90	0.89	0.85	0.39	0.42	0.60
6	Euclidean	Complete	0.91	0.92	0.89	0.37	0.40	0.61

[a] The cluster quality index by excluding the feature vectors related to 0 and 300 labels.

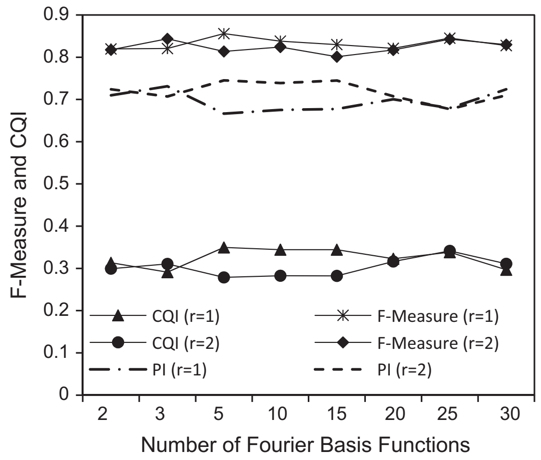

Fig. 11. Variation of *F*-Measure and CQI indices for different number of Fourier basis functions and degree of polynomial basis functions.

Using the kernelized distance resulted in equally high *F*-measure value; however, the highest CQI_m value was obtained in cases no. 1 and 3. The application of the linear regression for modeling the transients resulted in relatively poor performance compared to other cases. The association matrix and *CQ* graph of the entire turn-on events data on phase A (for a single run on the entire data set using basic feature vectors – case no. 1) are presented in Table 2 and Fig. 10, respectively. As these illustrations show the algorithm showed promising performance for case no. 1.

The same cases were taken into account for turn-off events on phase A. The problem could be more challenging in case of turn-off events since the information related to the dynamics of load variation is missing in turn-off events. Fig. 12 illustrates the feature vectors for turn-off events on phase A. As this figure shows, there are apparent similarities between some of the feature vectors (e.g., labels 11103 and 14502).Table 5 shows the evaluation results. Similarly, the feature extraction method in case no. 1, 3 and 4 resulted in better performance considering lower P_I values. The performance of the algorithm for these three cases and for both

Table 4
Performance of the heuristic algorithm for turn-on events on phase A for different feature extraction methods (average values across five 5-fold cross validation results).

Case no.	Feature vector	Precision	Recall	*F*-Measure	*CQI*	CQI_m[b]	P_I
1	$[P_1, Q_1]$	0.91	0.93	0.90	0.55	0.60	0.41
2	$[P_1, Q_1]$ (kernelized distance)	0.90	0.95	0.90	0.42	0.46	0.55
3	$[P_1, Q_1, \ldots, P_9, Q_9]$	0.92	0.93	0.90	0.53	0.58	0.43
4	$[P_1, Q_1, P_3, Q_3, \ldots, P_9, Q_9]$	0.91	0.93	0.89	0.55	0.60	0.42
5	$[\alpha, \beta, \gamma]_{P,Q}$[a]	0.87	0.79	0.76	0.29	0.31	0.73

[a] The set of coefficient in Eq. (18) for real and reactive power.
[b] The cluster quality index by excluding the feature vectors related to 0 and 300 labels.

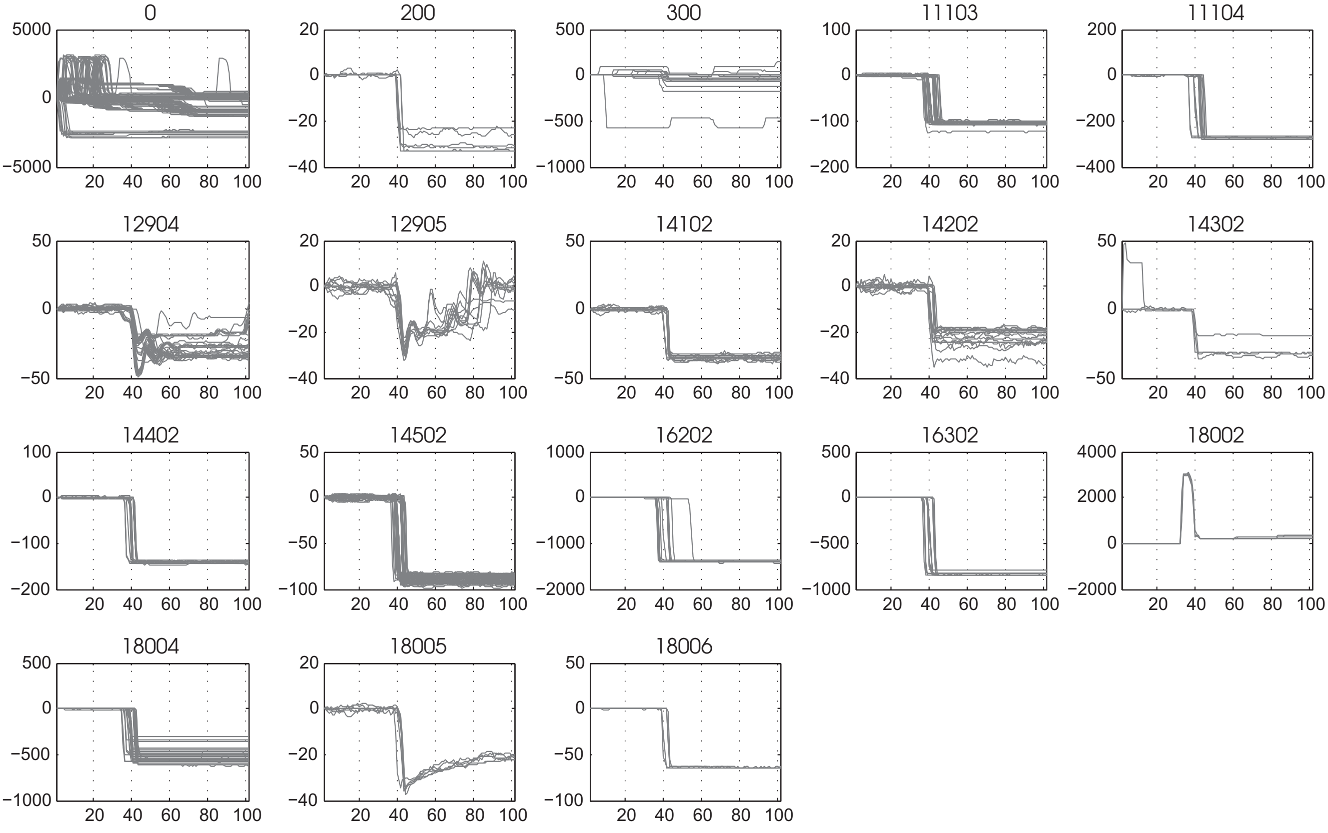

Fig. 12. Feature vectors clusters for phase A turn-off events, manually labeled by user, using the ground truth sensors' data.

Table 5
Performance of the proposed heuristic algorithm for turn-off events on phase A (average values across five 5-fold cross validation results).

Case no.	Feature Vector	Precision	Recall	*F*-Measure	*CQI*	CQI_m[b]	P_I
1	$[P_1, Q_1]$	0.96	0.89	0.91	0.46	0.51	0.50
2	$[P_1, Q_1]$ (kernelized distance)	0.97	0.85	0.89	0.37	0.41	0.60
3	$[P_1, Q_1, \ldots, P_9, Q_9]$	0.97	0.87	0.90	0.45	0.49	0.52
4	$[P_1, Q_1, P_3, Q_3, \ldots, P_9, Q_9]$	0.94	0.89	0.90	0.46	0.51	0.50
5	$[\alpha, \beta, \gamma]_{P,Q}$[a]	0.90	0.84	0.83	0.31	0.35	0.68

[a] The set of coefficient in Eq. (18).
[b] The cluster quality index by excluding the feature vectors related to 0 and 300 labels.

Table 6
Performance of the heuristic algorithm for turn-on and turn-off events on phase B for different feature extraction methods (average values across five 5-fold cross validation results).

Type	Case no.	Feature vector	Precision	Recall	*F*-Measure	*CQI*	CQI_m[b]	P_I
On	1	$[P_1, Q_1]$	0.89	0.92	0.87	0.35	0.41	0.60
On	2	$[P_1, Q_1]$ (kernelized distance)	0.95	0.92	0.91	0.29	0.34	0.67
On	3	$[P_1, Q_1, \ldots, P_9, Q_9]$	0.91	0.91	0.88	0.36	0.43	0.59
On	4	$[P_1, Q_1, P_3, Q_3, \ldots, P_9, Q_9]$	0.92	0.93	0.89	0.30	0.35	0.65
On	5	$[\alpha, \beta, \gamma]_{P,Q}$[a]	0.80	0.85	0.78	0.20	0.24	0.79
Off	6	$[P_1, Q_1]$	0.90	0.85	0.85	0.40	0.44	0.58
Off	7	$[P_1, Q_1]$ (kernelized distance)	0.92	0.82	0.85	0.38	0.39	0.62
Off	8	$[P_1, Q_1, \ldots, P_9, Q_9]$	0.89	0.85	0.85	0.45	0.49	0.53
Off	9	$[P_1, Q_1, P_3, Q_3, \ldots, P_9, Q_9]$	0.88	0.86	0.84	0.40	0.43	0.59
Off	10	$[\alpha, \beta, \gamma]_{P,Q}$[a]	0.80	0.79	0.74	0.14	0.15	0.88

[a] The set of coefficient in Eq. (18) for real and reactive power.
[b] The cluster quality index by excluding the feature vectors related to 0 and 300 labels.

turn-on and turn-off events data sets was similar with subtle changes in accuracy and *CQI* values which indicates that the addition of the higher harmonic contents did not affect the algorithm performance for the feature space on this phase. Furthermore, as the result for both of the analyses show, the application of the kernelized distance using a polynomial kernel function with degree 1 did not result in improved performance. Although the *F*-measure values are almost similar for both turn-on and turn-off events' cases, the turn-on events on Phase A have higher *CQI* values.

The evaluation of the performance for phase B turn-on and turn-off events were also carried out. In general, in our experimental test bed, the noise level on phase B was higher. The results are presented in Table 6. As the results for both turn-on and turn-off events show, the better performance was again observed in cases that basic feature vectors and higher harmonic contents were used. However, for the feature space (both turn-on and turn-off events) on this phase the algorithm was more sensitive to the application of different harmonic contents and the use of the first 9 harmonics showed relatively better performance. The relatively lower *F*-measure values for the turn-off events could be an indicator of the reduced information related to the missing dynamics of load variation in turn-off events.

The aforementioned observations for the data analyses on both phases show that the application of different feature extraction methods for autonomous clustering (using the proposed approach in this study) could depend on the structure of the feature space. Therefore, depending on the nature of the feature space, different features could result in different performance metrics' values. However, in general, the results showed the capability of the heuristic algorithm in accurate partitioning of the feature space. As a general observation, based on the results of our analyses, the application of basic feature vectors and higher harmonic contents along with city block distance and single linkage metrics could result in an acceptable partitioning of the feature space. As shown, depending on the structure of the feature space, the application of different harmonic contents might result in better partitioning of the feature space. To find the optimum feature extraction method that fits well with the feature space structure, an unsupervised method of cluster quality evaluation is required. Metrics such as inter-cluster and intra-cluster distance optimization could be a solution and the authors plan to investigate it as part of their future research. However, more analysis on different data sets could provide the ground for drawing statistically significant conclusions on the effect of different feature extraction methods.

5. Conclusion and future directions

Integration of human behavioral factors into energy management of the buildings has been shown to be effective in efficiency improvement [40–42]. NILM as a cost-effective stepping-stone solution for behavioral driven energy management has been the subject of several research studies in recent decades. The training process for commonly used supervised learning algorithms in event-based NILM applications is one of the major obstacles for wide adoption of the technology as a commercial solution. Part of the problem is related to the need for in-situ training, which is a consequence of diversity in different appliances' design and manufacturing technologies. Pre-populating the training data set, with the objective of reduced continuous user-interaction for training, could potentially facilitate a successful NILM application. Accordingly, in this study, we proposed and evaluated a heuristic for autonomous clustering of the feature space using hierarchical clustering algorithm to enable partitioning of the feature space to similar samples related to each appliances' state transition. In this heuristic algorithm, the characteristics of the binary cluster tree were used to determine the distance threshold for pruning the tree. To account for the multi-scale nature of the cluster tree, the algorithm finds the natural partitions of the feature space at different scales in a recursive fashion.

The algorithm was evaluated using the data collected from a real residential setting for two weeks. The power time series in the data set was fully labeled using the ground truth sensor network for different appliances operational states. For the performance evaluation, accuracy metrics and a customized metric for cluster quality were used. The evaluation of the algorithm was carried out for different distance and linkage metrics by using different feature extraction methods. The evaluations demonstrated the potential of the proposed algorithm in accurate partitioning of the feature space with high *F*-measure values (above 0.85 for the majority of the cases) for various evaluation scenarios. The assessment of different feature extraction methods showed that the application of basic feature vectors (real and reactive power for the fundamental frequency in proximity of the events) and higher harmonic contents of the power time series results in acceptable (with high accuracy and relatively high cluster quality index) partitioning of the feature space. However, depending on the feature space structure, each one of these feature extraction methods could relatively improve the quality of clustering. Unsupervised determination of the better feature extraction method requires an autonomous internal cluster quality evaluation approach, which will be part of the future directions of this research.

As it was demonstrated, the application of the proposed algorithm leads to grouping of thousands of events into a limited number of clusters, representing the signatures of appliances' operational schedules. Identification of the possible appliances' operational states provides the ground for more efficient communication with users. This not only could reduce the number of interactions, but also has the potential to improve the quality of labeling process. For example, event detection algorithms can sometimes detect more than one event for a given appliance state transition. From a user's perspective, this state transition corresponds to a single event, however, the power signal does not need to agree with this. In these situations, the last detected event (among all the consecutive sequential events) is always presented for user interaction, leaving the precedent events unlabeled. However, by using pre-populated training data, the NILM system (which is aware of the signature space and the corresponding labeling history) could present different instances in the events sequence at different interaction occasions. Moreover, semantic information, extracted from clustered data such as frequency of the events in the clusters, could facilitate rule-based intelligent communication. Accordingly, investigating the algorithm's performance on more data sets, developing an unsupervised performance evaluation for feature selection automation, and exploring the proposed algorithm efficacy in conjunction with event based NILM solutions for improved user experience and labeling process are among the authors' future research directions.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant #0930868. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] H. Kim, M. Marwah, M. Arlitt, G. Lyon, J. Han, Unsupervised disaggregation of low frequency power measurements, in: 11th SIAM International Conference on Data Mining, SDM 2011, April 28, 2011–April 30, 2011, pp. 747–758.

[2] J.Z. Kolter, M.J. Johnson, REDD: a public data set for energy disaggregation research, in: Proceedings of the SustKDD Workshop on Data Mining Applications in Sustainability, 2011, pp. 1–6.

[3] O. Parson, S. Ghosh, M. Weal, A. Rogers, Non-Intrusive Load Monitoring Using Prior Models of General Appliance Types, AAAI Press, 2012.

[4] J. Zico Kolter, Tommi Jaakkola, Approximate inference in additive factorial HMMs with application to energy disaggregation, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2012.

[5] G.W. Hart, Nonintrusive appliance load monitoring, Proc. IEEE 80 (1992) 1870–1891.

[6] M. Zeifman, K. Roth, Nonintrusive appliance load monitoring: review and outlook, in: 2011 IEEE International Conference on Consumer Electronics, ICCE 2011, January 9, 2011–January 12, 2011, pp. 239–240.

[7] A.G. Ruzzelli, C. Nicolas, A. Schoofs, G.M.P. O'Hare, Real-time recognition and profiling of appliances through a single electricity sensor, in: 2010 7th Annual IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010, pp. 1–9.

[8] S. Drenker, A. Kader, Nonintrusive monitoring of electric loads, IEEE Comput. Appl. Power. 12 (1999) 47–51.

[9] A. Marchiori, D. Hakkarinen, Qi. Han, L. Earle, Circuit-level load monitoring for household energy management, IEEE Pervas. Comput. 10 (2011) 40–48.

[10] S. Gupta, M.S. Reynolds, S.N. Patel, ElectriSense: Single-point sensing using EMI for electrical event detection and classification in the home, in: 12th International Conference on Ubiquitous Computing, UbiComp 2010, September 26, 2010–September 29, 2010, pp. 139–148.

[11] M. Berges, E. Goldman, H.S. Matthews, L. Soibelman, K. Anderson, User-centered nonintrusive electricity load monitoring for residential buildings, J. Comput. Civ. Eng. 25 (2011) 471–480.

[12] J. Wang, S. Wang, Wireless sensor networks for home appliance energy management based on ZigBee technology, in: 2010 International Conference on Machine Learning and Cybernetics, ICMLC 2010, July 11, 2010–July 14, 2010, pp. 1041–1044.

[13] D. Srinivasan, W.S. Ng, A.C. Liew, Neural-network-based signature recognition for harmonic source identification, IEEE Trans. Power Del. 21 (2006) 398–405.

[14] S.N. Patel, T. Robertson, J.A. Kientz, M.S. Reynolds, G.D. Abowd, At the flick of a switch: detecting and classifying unique electrical events on the residential power line, in: 9th International Conference, UbiComp 2007, 2007, pp. 271–288.

[15] S. Giri, M. Bergés, A. Rowe, Towards automated appliance recognition using an EMF sensor in NILM platforms, Adv. Eng. Inform. 27 (2013) 477–485.

[16] A. Schoofs, A. Guerrieri, D.T. Delaney, G. O'Hare, A.G. Ruzzelli, ANNOT: automated electricity data annotation using wireless sensor networks, in: 2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2010, pp. 9.

[17] F. Jazizadeh, S. Ahmadi-Karvigh, B. Becerik-Gerber, L. Soibelman, Spatiotemporal lighting load disaggregation using light intensity signal, Energy Build. 69 (2014) 572–583.

[18] M. Berges, A framework for enabling energy-aware facilities through minimally-intrusive approaches, Ph.D. Thesis, Carnegie Mellon University, United States, 2010.

[19] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. (CSUR) 31 (1999) 264–323.

[20] M. Halkidi, Y. Batistakis, M. Vazirgiannis, On clustering validation techniques, J. Intell. Inform. Syst. 17 (2001) 107–145.

[21] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. Roy. Stat. Soc. Ser. B (Methodol.) (1977) 1–38.

[22] M. Ester, H. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: International Conference on Knowledge Discovery in Data Bases and Data Mining, Montreal, Canada, 1996, pp. 226–231.

[23] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 603–619.

[24] B. Nadler, M. Galun, Fundamental limitations of spectral clustering, Adv. Neural Inform. Process. Syst. (2006) 1017–1024.

[25] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, Adv. Neural Inform. Process. Syst. 2 (2002) 849–856.

[26] U. Von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (2007) 395–416.

[27] M.E. Balci, M.H. Hocaoglu, Comparison of power definitions for reactive power compensation in nonsinusoidal conditions, in: 11th International Conference on Harmonics and Quality of Power, 2004. 2004, pp. 519–524.

[28] J.L. Wyatt Jr., M. Ilic, Time-domain reactive power concepts for nonlinear, nonsinusoidal or nonperiodic networks, in: IEEE International Symposium on Circuits and Systems, 1990, 1990, pp. 387–390.

[29] K.D. Lee, Electric Load Information System Based on Non-Intrusive Power Monitoring, Ph.D. Thesis, Massachusetts Institute of Technology, Dept. of Mechanical Engineering, 2003.

[30] S.R. Shaw, S.B. Leeb, L.K. Norford, R.W. Cox, Nonintrusive load monitoring and diagnostics in power systems, IEEE Trans. Instrum. Measur. 57 (2008) 1445–1454.

[31] S.B. Leeb, S.R. Shaw, J.L. Kirtley Jr., Transient event detection in spectral envelope estimates for nonintrusive load monitoring, IEEE Trans. Power Del. 10 (1995) 1200–1210.

[32] D. Luo, L.K. Norford, S. Leeb, S. Shaw, Monitoring HVAC equipment electrical loads from a centralized location-methods and field test results, ASHRAE Trans. 108 (2002) 841–857.

[33] M. Berges, E. Goldman, H.S. Matthews, L. Soibelman, K. Anderson, User-centered non-intrusive electricity load monitoring for residential buildings, J. Comput. Civ. Eng. 25 (6) (2011) 471–480.

[34] W.H. Day, H. Edelsbrunner, Efficient algorithms for agglomerative hierarchical clustering methods, J. Classif. 1 (1984) 7–24.

[35] R. Tibshirani, G. Walther, T. Hastie, Estimating the number of clusters in a data set via the gap statistic, J. Roy. Stat. Soc.: Ser. B (Stat. Methodol.) 63 (2001) 411–423.

[36] A.D. Gordon, Classification, second ed., Chapman and Hall – CRC, London, 1999.

[37] F. Jazizadeh, B. Becerik-Gerber, M. Berges, L. Soibelman, Unsupervised clustering of residential electricity consumption measurements for facilitated user-centric non-intrusive load monitoring, Comput. Civ. Build. Eng. (2014) 1869–1876.

[38] Enmetric-Systems. <http://www.enmetric.com/platform#Hardware>, 2012 (accessed 10.12.2012).

[39] J.C. Dunn, Well-separated clusters and optimal fuzzy partitions, J. Cybern. 4 (1974) 95–104.

[40] F. Jazizadeh, A. Ghahramani, B. Becerik-Gerber, T. Kichkaylo, M. Orosz, User-led decentralized thermal comfort driven HVAC operations for improved efficiency in office buildings, Energy Build. 70 (2014) 398–410.

[41] R.K. Jain, J.E. Taylor, G. Peschiera, Assessing eco-feedback interface usage and design to drive energy efficiency in buildings, Energy Build. 48 (2012) 8–17.

[42] D. Parker, D. Hoak, A. Meier, R. Brown, How much energy are we using? Potential of residential energy demand feedback devices, in: Proceedings of the 2006 Summer Study on Energy Efficiency in Buildings, American Council for an Energy Efficient Economy, Asilomar, CA, 2006.

Please cite this article in press as: F. Jazizadeh et al., An unsupervised hierarchical clustering based heuristic algorithm for facilitated training of electricity consumption disaggregation systems, Adv. Eng. Informat. (2014), http://dx.doi.org/10.1016/j.aei.2014.09.004