



An Efficient Sparse Coding-Based Data-Mining Scheme in Smart Grid

Dongshu Wang, Jialing He, Mussadiq Abdul Rahim,
Zijian Zhang^(✉), and Liehuang Zhu

Beijing Institute of Technology, Beijing, China
zhangzijian@bit.edu.cn

Abstract. With the availability of Smart Grid, disaggregation, i.e. decomposing a whole electricity signal into its component appliances has gotten more and more attentions. Now the solutions based on the sparse coding, i.e. the supervised learning algorithm that belongs to Non-Intrusive Load Monitoring (NILM) have developed a lot. But the accuracy and efficiency of these solutions are not very high, we propose a new efficient sparse coding-based data-mining (ESCD) scheme in this paper to achieve higher accuracy and efficiency. First, we propose a new clustering algorithm – Probability Based Double Clustering (PDBC) based on Fast Search and Find of Density Peaks Clustering (FSFDP) algorithm, which can cluster the device consumption features fast and efficiently. Second, we propose a feature matching optimization algorithm – Max-Min Pruning Matching (MMPM) algorithm which can make the feature matching process to be real-time. Third, real experiments on a publicly available energy data set REDD [1] demonstrate that our proposed scheme achieves a for energy disaggregation. The average disaggregation accuracy reaches 77% and the disaggregation time for every 20 data is about 10s.

Keywords: Smart Grid · Energy disaggregation · Sparse coding
Data mining

1 Introduction

Data mining is an efficient technique that shows the interesting patterns or knowledge from huge amount of data. These patterns play an important role in marketing, business, medical analysis, intrusion detection, and other applications where these patterns are of paramount importance for strategic decision making [21].

In Smart Grid, Energy disaggregation, the task of separating the whole energy signal of a residential, commercial, or industrial building into the energy signals of individual appliances, is a kind of important data mining method for energy-saving and by Energy disaggregation, we can disaggregate electricity consumptions to infer what device is used, how often this device is used. If this data

is used by the supplier, amount of advertising fees may be saved and they can make a precise advertising for each family. They can even know which device needs to be fixed [24, 25].

Studies on energy disaggregation date back to about thirty years ago. The early approaches look for sharp edges (corresponding to device on/off events) in both the real and reactive power signals, and cluster devices according to these changes in consumption [2–6]. A number of different directions has been explored, [22, 23] computing harmonics of steady-state power to determine more complex device signatures, [24] analyzed the transient noise of a device circuit when the device changes state. But now for costs and convenience more and more researches focus on disaggregate electricity using low-frequency-resolution [7–9]. In this paper, we use a novel clustering algorithm based on sparse coding to disaggregate electricity in low frequency data set.

In this paper, we propose a new scheme—ESCD to make Energy disaggregation. The main contributions are as follows:

1. In the scheme we first proposes a new clustering algorithm—Probability Based Double Clustering (PBDC) based on Fast Search and Find of Density Peaks Clustering (FSFDP) algorithm in energy disaggregation, which can cluster the device consumption features fast and efficiently.
2. Our scheme proposes a feature matching optimization algorithm – Max-Min Pruning Matching (MMPM) algorithm, without any constraint condition, our algorithm can make the feature matching process to be real-time.
3. Experiments show that our scheme (ESCD) has a better Energy disaggregation accuracy and real-time performance than the sparse-coding based schemes before.

2 Related Work

The basic problem in this paper is energy disaggregation (the task of determining the component appliance contributions from an aggregated electricity signal) [1]. The user’s behavior of electricity consumption can be obtained by energy disaggregation, the solutions of which are varied. According to different types of basic data sets, the solutions can be divided into two categories. The first is the scheme based on the high frequency data [10–12], and the second is based on the low frequency data [7–9, 13–18]. The schemes based on the high frequency data can obtain more accurate analysis results. But the hardware for data sampling is very expensive. It requires one or more than one sensor per appliance to perform Appliance Load Monitoring (ALM). Namely Intrusive Load Monitoring (ILM) that is contrary to the issue studied in this paper. The scheme based on the low frequency data just requires only a single meter for per house or a building, the scheme is called Non-Intrusive Load Monitoring (NILM). The NILM schemes are divided into supervised [15–17] and unsupervised [7–9, 18] disaggregation algorithms. Most unsupervised disaggregation algorithms are developed based on the HMM. They can have good results in estimating the total electricity consumption per device, while the analysis for the time-sharing of each

device is not good compared to the supervised ones [26]. Most supervised disaggregation algorithms are based on the sparse coding. For example, [16] uses a data set provided by Plugwise which contains ten broad categories of electrical devices from several houses. It models the entire signal of each device over a long period of time, such as a week, as a sparse linear combination of the atoms of an unknown dictionary [27]. However, the drawbacks of the algorithm are that it requires a large training dataset to capture all possible times that the same device may operate and the classification times cost too long. [15] uses the data set REDD, it proposes a concept: powerlets – a small period time, the electricity consumption of device measured by ‘powerlets’ and decoded by Dissimilarity-based Sparse subset Selection (DS3) [19] algorithm. The scheme can achieve the energy disaggregation accuracy of 72%, and the energy disaggregation speed is about 15 s. It is an efficient and accurate scheme, but it requires a lot of constraints, which causes the bad effect on the actual effect of this scheme. We propose a scheme in this paper, which improves the algorithm in [20]. We first use FSFDP algorithm and propose a feature matching optimization algorithm – max-min Pruning algorithm.

3 Preliminaries

We make a brief introduction to fast search and find of density peaks clustering and sparse coding.

3.1 Fast Search and Find of Density Peaks Clustering (FSFDP)

Clustering algorithm is an effective tool for data mining. Specifically, clustering analysis automatically groups things in the absence of category tag information. Each packet is self-identifying and differentiated from other groups [28].

Fast search and find of density peaks Clustering is a clustering algorithm, it is based on the idea that cluster centers are characterized by a higher density than their neighbors, and by a relatively large distance from points with higher densities. In FSFDP, the number of clusters arises intuitively, outliers are automatically spotted and excluded from the analysis, and clusters are recognized regardless of their shape and the dimensionality of the space in which they are embedded [21].

Assume that the data set $S = \{x_i\}_{i=1}^N$, $I_S = \{1, 2, \dots, N\}$ to be clustered is the corresponding set of indices, and d_{ij} represents the defined distance between the data points x_i and x_j . For any data point x_i in S , we can define two quantities of ρ_i and δ_i , which are the two characteristics of the clustering center mentioned above: local density and distance, respectively.

Local density ρ_i is calculated by two ways: Cut-off kernel or Gaussian kernel.

Cut-off kernel and Gaussian kernel are calculated by Eqs. 1 and 2:

$$\rho_i = \sum_{j \in I_S \setminus \{i\}} \chi(d_{ij} - d_c) \chi(x) = \begin{cases} 1, & x < 0; \\ 0, & x \geq 0. \end{cases} \quad (1)$$

$$\rho_i = \sum_{j \in I_s \setminus \{i\}} e^{-\left(\frac{d_{ij}}{d_c}\right)^2} \quad (2)$$

The parameter $d_c > 0$ is the cutoff distance and must be specified in advance. ρ_i represents the number of data points in the S where the distance is less than d_c (regardless of the itself). In Gaussian kernel, the distance between the data point j and i ($d_{ij} < d_c$) is greater, the value of ρ_i is greater. The only difference is that the cut-off kernel is a discrete value and the Gaussian kernel is a continuous value, the probability of the latter is conflicting (i.e., the local density values of the same data points with the same distance) are smaller.

We calculate the distance δ_i as follows:

Let $\{q_i\}_{i=1}^N$ be a descending order of $\{\rho_i\}_{i=1}^N$, $\rho_{q_1} \geq \rho_{q_2} \geq \dots \geq \rho_{q_N}$.

$$\delta_{q_i} = \begin{cases} \min_{q_j, j} \{d_{q_i, q_j}\}, i \geq 2; \\ \max_{j \geq 2} \{d_{q_i, q_j}\}, i = 1. \end{cases} \quad (3)$$

$$\delta_i = \begin{cases} \min_{j \in I_s^i} \{d_{ij}\}, I_s^i \neq \Phi; \\ \max_{j \in I_s} \{d_{ij}\}, I_s^i = \Phi. \end{cases} \quad (4)$$

The set $I_s^i = \{k \in I_s : \rho_k > \rho_i\}$, when x_i has the maximum local density, δ_i represents the distance between the data point and x_i with the largest distance from x_i in S, otherwise δ_i represents the shortest distance that in all data points with local density greater than x_i . We can choose some appropriate value of ρ_i and δ_i as critical value to cluster.

3.2 Sparse Coding

Sparse coding is a class of unsupervised methods for learning sets of over-complete bases to represent data efficiently. The aim of sparse coding is to find a set of basis vectors ϕ_i such that we can represent an input vector X as a linear combination of these basis vectors [21]:

$$X = \sum_{i=1}^k a_i \phi_i \quad (5)$$

We define sparsity as having few non-zero components or having few components not close to zero. The requirement that our coefficients a_i be sparse means that given a input vector, we would like as few of our coefficients to be far from zero as possible. The choice of sparsity as a desired characteristic of our representation of the input data can be motivated by the observation that most sensory data such as natural images may be described as the superposition of a small number of atomic elements such as surfaces or edges. Other justifications such as comparisons to the properties of the primary visual cortex have also been advanced [22].

We define the sparse coding cost function on a set of m input vectors as [23]:

$$\min_{a_i^{(j)}} \phi_i \sum_{j=1}^m \left\| X^{(j)} - \sum_{i=1}^k a_i^{(j)} \phi_i \right\|^2 + \lambda \sum_{i=1}^k S(a_i^{(j)}) \quad (6)$$

where $S(\cdot)$ is a sparsity cost function which penalizes a_i for being far from zero. We can interpret the first term of the sparse coding objective as a reconstruction term which tries to force the algorithm to provide a good representation of X and the second term as a sparsity penalty which forces our representation of X to be sparse. The constant λ is a scaling constant to determine the relative importance of these two contributions [29].

Although the most direct measure of sparsity is the ‘L0’ norm ($S(a_i) = 1(|a_i| > 0)$), it is non-differentiable and difficult to optimize in general. In practice, common choices for the sparsity cost $S(\cdot)$ are the L_1 penalty $S(a_i) = |a_i|$ and the log penalty $S(a_i) = \log(1 + a_i^2)$.

In addition, it is also possible to make the sparsity penalty arbitrarily small by scaling down a_i and scaling ϕ_i up by some large constant. To prevent this from happening, we will constrain $\|\phi^2\|$ to be less than some constant C . The full sparse coding cost function including our constraint on ϕ is

$$\begin{aligned} \min_{a_i^{(j)}} \phi_i \sum_{j=1}^m \left\| X^{(j)} - \sum_{i=1}^k a_i^{(j)} \phi_i \right\|^2 + \lambda \sum_{i=1}^k S(a_i^{(j)}) \\ \text{subject to } \|\phi\|^2 \leq C, \forall i = 1, \dots, k \end{aligned} \quad (7)$$

4 Efficient Sparse Coding-Based Data-Mining (ESCD) Scheme

In this section we propose a novel energy disaggregation scheme, ESCD Scheme. Based on sparse coding, this is an efficient and fast scheme for energy disaggregation. First, we carry out a detailed description of the problem. We assume that there are n electrical devices in a building, where $x_i(t)$ denotes the energy signal of device i at time t . $y(t)$ denotes the aggregate energy signal, recorded by a smart meter, at time t . We can write

$$y(t) = \sum_{i=1}^n x_i(t) \quad (8)$$

Given only the whole power consumption $y(t)$, the goal of energy disaggregation is to recover the power signal of each of the appliances, i.e., to estimate $x_i(t)$ for $i \in \{1, 2, 3, \dots, n\}$. It is very difficult to solve the formula (8) because $x_i(t)$ represents the i -th electricity consumption of the device at the time t , and the electrical device is operated by the person. We can not judge the human behaviour at time t that we can’t get $x_i(t)$.

In this paper, we model each electrical device with sparse coding. For every $i = 1, 2, 3, \dots, n$, we learn a feature matrix from training set whose frequency

is q Hz, $X_i \in R^{m_i \times T_w}$, T_w we call sliding window time represents a continuous period of time from training time T_{tr} and $T_w \ll T_{tr}$. m_i represents the number of features in the i -th electrical device. If we have got enough features from the i -th electrical device, we can use the approximation

$$\begin{aligned} T_w(x_i(t)) &\approx X_i a_i(t), T_w(x_i(t)) \\ &= \left(x_i(t), x_i\left(t + \frac{1}{q}\right), x_i\left(t + 2 \times \frac{1}{q}\right), \dots, x_i(t + T_w)\right) \end{aligned} \quad (9)$$

We call the vector $T_w(x_i(t))$ sliding window, $a_i(t)$ is the activations of feature matrix X_i , $a_i(t)$ is sparse that contain mostly zero elements and only a 1 elements. We propose a matching algorithm to get the $a_i(t)$ so that we can calculate $X_i a_i(t)$ to get an approximate solution for each device. Our scheme based on sparse coding is consist of two stages, learning feature matrix and energy matching.

4.1 Learning Feature Matrix

We propose a Probability Based Double Clustering (PBDC) algorithm to learning each of the electrical devices feature matrix. As described above, the time length of training data is T_{tr} , frequency is q Hz, there are $(T_{tr} - T_w + 1) \times q$ sliding windows, for each sliding window, we have $T_w \times q$ elements. Assume that every sliding window is a data point $P_i \in (1, (T_{tr} - T_w + 1) \times q)$, after removing the repeated points we get k unique points set P_{uniq} and a vector representing the repeat times of data points $R = (r_1, r_2, r_3, \dots, r_k)$, we define the distance between two point $d_{ij} = \|P_i - P_j\|_2$, the distance matrix D is a $k \times k$ scale symmetric matrix with a diagonal of 0, $D = \{d_{ij}, i, j \in \{1, 2, 3, \dots, k\}\}$, PBDC algorithm is based on FSFDP, the FSFDP algorithm needs to manually specify the minimum value of ρ and δ to determine the boundary of the cluster that to determine the number of clustering centres, this operation will lead to the instability of the number of clustering centres and can not get the scheme to be automated.

In this paper, the algorithm is improved by setting a upper limit of the number m , when clustering we will compare the difference of m and the number of clusters, and make a automatically correct until the number of clusters is similar to m . In this improved algorithm, it is only necessary to specify the upper limit of the number m of clustering centres before clustering instead of manually specify the minimum value of ρ and δ during clustering. Therefore, we can have an efficient clustering and the number of clustering centres can be stabilized at a certain controllable value. Every clustering centre is an electrical device feature, a controllable number of electrical device feature is necessary for reduce the algorithm complexity of energy matching.

After first clustering, we find some problem, the clustering result is not what we really want. According to the power consumption of electrical equipment and the relationship between time, as shown in Fig. 1.

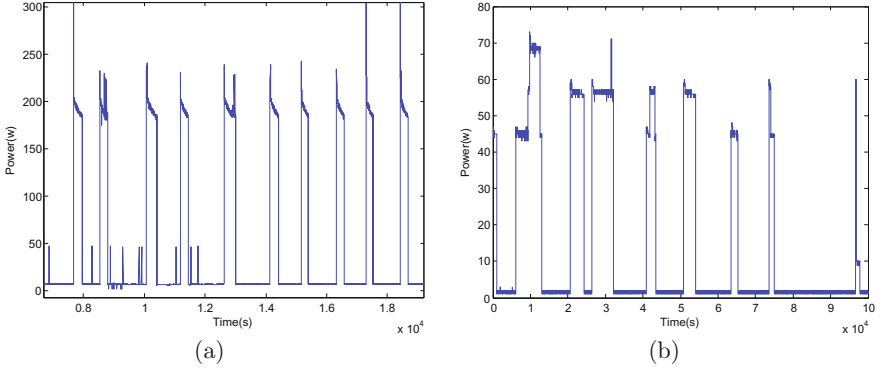


Fig. 1. (a) and (b) are the electricity consumption of refrigerator and lighting.

Algorithm 1. Learning Feature Matrix

Input: Device training data set D_{tr} ,
 Sliding window size w ,
 The number of feature vector m .

Output: Device feature matrix X .

1. Compute P_{uniq} and R from D_{tr}
 2. Compute distance matrix D :
 3. **for** $i = 1, \dots, size(R)$ **do**
 4. **for** $j = i, \dots, size(R)$ **do**
 5. $d_{ij} = \|P_i - P_j\|_2$
 6. **end for**
 7. **end for**
 8. First Clustering: improved FSFDP($D, m/10$)
 9. Second Clustering:
 10. **for** $k = 1, \dots, size(clusters)$ **do**
 11. $C_i = num(cluster_i) / \|R\|_1$
 12. improved FSFDP($D_i, m * C_i$)
 13. clustering result is collected to X
 14. **end for**
 15. **return** X
-

It is clear that after the first clustering, the centre point represents the larger distance from the data points, and the result is just judged by the data value difference, this will lead a result that some data points with small value is buried in large. In fact, we need a more comprehensive clustering to make a second clustering for improving this result.

In the second clustering, we need to do cluster for every clusters calculated in the first clustering. And we set the number of cluster by the probability set C that includes the probability C_i of each cluster. Considering the repeated points, the probability C_i of each cluster can be calculated by formula 10 and 11.

$$C_i = \frac{\text{num}(\text{Cluster}_i)}{\sum_{i=1}^k r_i} \quad (10)$$

$$\text{num}(\text{Cluster}_i) = \sum r_i, r_i \in \{\text{Cluster}_i\} \quad (11)$$

Assume that we need about m features in each electrical device, we can set the upper limit $m \times C_i$ for each cluster and clustering again. The algorithm details describe in Algorithm 1.

4.2 Energy Matching

According to the first stage, we obtain the feature matrix of each device X_i , as seen in formula (9), in this stage, we need to calculate $a_i(t)$. Considering the constraint of $a_i(t)$ that $a_i(t)$ has only one element is 1 and other is 0.

Algorithm 2. MMPM algorithm

Input: Device feature matrix $X_1 X_2 \dots X_n$

Test data Y .

Pruning threshold u

Output: Disaggregation result $\tilde{Y}_1 \tilde{Y}_2 \dots \tilde{Y}_n$

1. get the windows size w and each device feature numbers m_i
 2. **while** $X_1 \neq NULL$
 3. get one feature vector form X_1
 4. compute the remainder energy $T_w(y(t))$ and arg_{max}
 5. **if** ($arg_{max} > \mu || \min(T_w(y(t))) + \mu < 0$)
 6. break;
 7. **end if**
 8. overlay record the feature vector into \tilde{Y}_1
 9. ...
 10. **while** $X_n \neq NULL$
 11. get one feature vector form X_n
 12. compute the remainder energy $T_w(y(t))$ and arg_{max}
 13. **if** ($arg_{max} > \mu || \min(T_w(y(t))) + \mu < 0$)
 14. break;
 15. **end if**
 16. overlay record the feature vector into \tilde{Y}_n
 17. **end while**
 18. ...
 19. **end while**
-

In general, in order to get a globally optimal solution with n electrical devices, if the number of feature matrix in each electrical device X_i is m_i , the algorithm complexity is $O\left(\prod_{i=1}^n m_i\right)$. We can find that the complexity of the algorithm increases exponentially with each additional device, and the complexity of this

algorithm is unacceptable. We need to do some algorithm optimization, consider that n is a constant determined by the number of equipment and m_i is the number of electrical device feature, it will lead to a poor accuracy if we reduce the m_i . We propose a Max-Min Pruning Matching (MMPM) algorithm.

The purpose of the algorithm is to perform the pruning optimization of the matching algorithm under the condition of guaranteeing the global optimum. This optimization can be divided into two operations: minimum pruning and maximum pruning, set the maximum pruning parameters for arg_{max} , the pruning threshold for μ . We first discuss the minimum pruning.

The goal of maximum pruning can make sure that we can get the global optimal solution and end the loop as soon as possible when we get the global optimal solution. To do this option, we should get the order j of the maximum element in $T_w(y(t))$. And then sort in each feature matrix $X_i, i \in \{1, 2, 3, \dots, n\}$ by the j -th element in descending order, we set the maximum element of feature matrix X_i in j -th column max_i . We calculate the maximum pruning parameters as below

$$arg_{max} = y(t + (j - 1)q) - \sum_{i=n-i}^n max_i \quad (12)$$

When $arg_{max} > \mu$, it means that the arg_{max} in remaining loop will large than the pruning threshold, the remaining loop should be cutoff.

The goal of minimum pruning aids to cut the invalid loop when the sliding windows vector is too small. In each loop, we get a remainder energy $T_w(r(t))$ which is the difference between total energy $T_w(y(t))$ and the upper loop, we make a minimum pruning judgement condition

$$\min(T_w(r(t))) + \mu < 0 \quad (13)$$

If this judgement condition is set up, that proves it the remaining loop will make the $\min(T_w(r(t)))$ more and more small, we should cut off the remaining invalid loop. The algorithm details describe in Algorithm 2.

5 Experiment Analysis

In this section, we evaluate our propose scheme on the real-world REDD data set [1], a publicly available data set for electricity disaggregation. The data set consists of power consumption signals from six different houses, where for each house, the whole electricity consumption as well as electricity consumptions of about twenty different devices are recorded. The signals from each house are collected over a period of two weeks with a low frequency sampling rate of 1/3 Hz. The House 5 data set is excluded because of its data contains very few fluctuation that we could not extract enough features to do energy disaggregation.

In the experiment, in every house we use a month of recorded electricity signals that include 5 important household appliances, one week for learning feature matrix and the rest for energy matching. We set the size of feature matrix as $20 \times m$ that means the size of a feature vector is set to 20 and the numbers

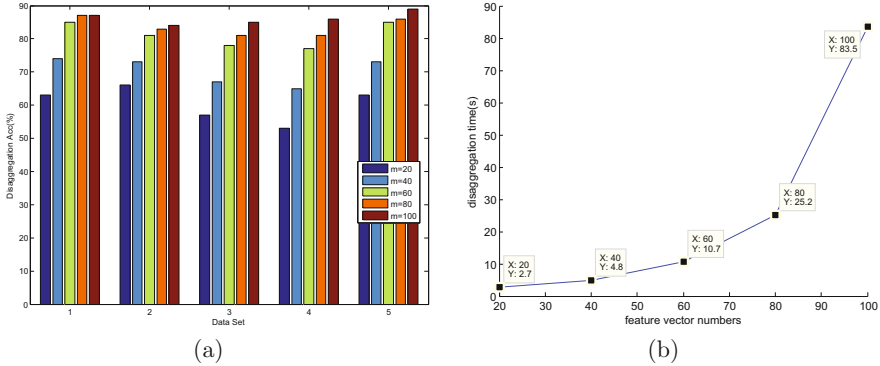


Fig. 2. (a) and (b) are the relationship of disaggregation accuracy and time with feature vector numbers.

of feature vector is set to m . The size of feature vector too small or too large may lead to a low performance because that a small size could cause excessive training and a large size could confuse the features. In order to make a real-time energy matching the numbers of feature vector m must be less than 100. With these settings, we can get five feature matrix for five household appliances, it takes about 10s to perform energy matching on a temporal window. We compute the disaggregation accuracy, similar to [1]:

$$acc_{\text{energy matching}} = 1 - \frac{\sum_{t \in \psi} \sum_{i=1}^M \left\| T_w(x_i(t)) - \tilde{T}_w(x_i(t)) \right\|_1}{2 \sum_{t \in \psi} \left\| T_w(x_i(t)) \right\|_1} \quad (14)$$

where $\psi = \{1, T_w + 1, 2T_w + 1, \dots\}$ and the 2 factor in the denominator comes from the that the absolute value results in double counting errors. We compare our method with the PED algorithm [15], FHMM algorithm (in its supervised setting) [1] and a Simple Mean prediction algorithm, which estimates the total consumption percentage of each device and predicts that the whole electricity signal breaks down according to this percentage at all time.

Figure 2 shows the disaggregation accuracy and time of our algorithm for different houses as a function of feature vector numbers m when size of a feature vector is set to 20. Table 1 shows the disaggregation results for all the six houses (exclude the House 5) in the REDD data set. Our algorithm performs better than PED, FHMM and the naive Simple Mean on the data set, achieving about 5.4% higher accuracy overall. Figure 3 shows the actual and estimated energy consumption obtained by our method for refrigerator and lighting in the House 1. Our scheme captures transients and different steady states in each device.

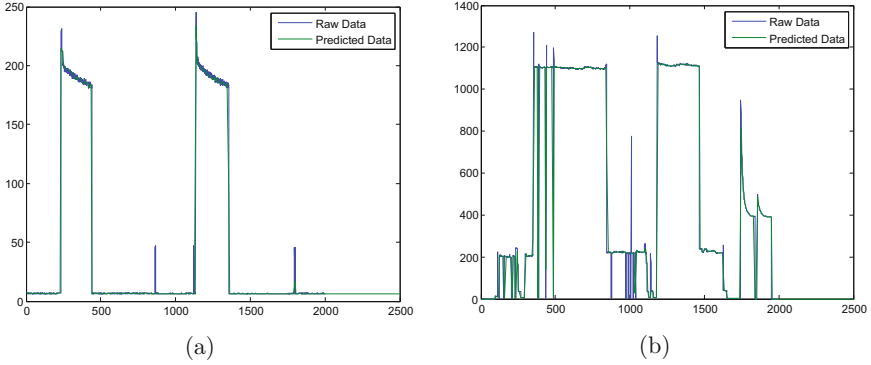


Fig. 3. (a) and (b) the actual and estimated energy consumption obtained by ESCD scheme for refrigerator and lighting in the House 1

Table 1. Energy disaggregation accuracies (%)

	House 1	House 2	House 3	House 4	House 6	Average
Simple	41.4%	39.0%	46.7%	52.7%	33.7%	42.7%
FHMM	71.5%	59.6%	59.6%	69.0%	62.9%	64.5%
PED	81.6%	79.0%	61.8%	58.5%	79.1%	72.0%
ESCD	84.3%	82.7%	70.2%	71.0%	78.9%	77.4%

6 Conclusion

In this paper, we propose a new algorithm for energy disaggregation which consists of the two steps of learning feature matrix of power consumption signatures and a energy matching for disaggregation. To learn feature matrix, based on FSFDP algorithm we propose the Probability Based Double Clustering (PBDC) algorithm to learning each of the electrical devices feature matrix, the PBDC algorithm can make a second clustering according to the first clustering points distribution probability and ensure that the clustering can extract sufficient feature vector and avoid excessive training. After we calculate feature matrix of each devices, we propose a Max-Min Pruning Matching (MMPM) energy matching algorithm for disaggregation. The MMPM algorithm can minimize computational complexity make the disaggregation be a real-time calculation. Our experiments are based on a real energy data set, we show that our scheme provides promising results for energy disaggregation.

Acknowledgment. This work is partially supported by China National Key Research and Development Program No. 2016YFB0800301.

References

1. Kolter, J.Z., Johnson, M.J.: REDD: a public data set for energy disaggregation research. In: Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA, vol. 25, pp. 59–62 (2011)
2. Pinkas, B.: Cryptographic techniques for privacy-preserving data mining. *SIGKDD Explor. Newsl.* **4**(2), 12–19 (2002)
3. Hart, G.W.: Nonintrusive appliance load monitoring. *Proc. IEEE* **80**(12), 1870–1891 (1992)
4. Berges, M., Goldman, E., Matthews, H.S., Soibelman, L.: Learning systems for electric consumption of buildings. In: *ASCI International Workshop on Computing in Civil Engineering* (2009)
5. Shaw, S.R., Abler, C.B., Lepard, R.F., et al.: Instrumentation for high performance nonintrusive electrical load monitoring. *J. Sol. Energy Eng.* **120**(3), 224–230 (1998)
6. Patel, S.N., Robertson, T., Kientz, J.A., Reynolds, M.S., Abowd, G.D.: At the flick of a switch: detecting and classifying unique electrical events on the residential power line. In: *9th International Conference on Ubiquitous Computing (UbiComp 2007)* (2007)
7. Shao, H., Marwah, M., Ramakrishnan, N.: A temporal motif mining approach to unsupervised energy disaggregation. In: *Proceedings of the 1st International Workshop on Non-Intrusive Load Monitoring*, Pittsburgh, PA, USA, 7 May 2012
8. Zhong, M., Goddard, N., Sutton, C.: Interleaved factorial non-homogeneous hidden Markov models for energy disaggregation (2014). arXiv preprint: [arXiv:1406.7665](https://arxiv.org/abs/1406.7665)
9. Lange, H., Berges, M.: Efficient inference in dual-emission FHMM for energy disaggregation. In: *AAAI Workshop: AI for Smart Grids and Smart Buildings* (2016)
10. Norford, L.K., Leeb, S.B.: Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms. *Energ. Build.* **24**, 51–64 (1996)
11. Shaw, S.R., Leeb, S.B., Norford, L.K., Cox, R.W.: Nonintrusive load monitoring and diagnostics in power systems. *IEEE Trans. Instrum. Meas.* **57**, 1445–1454 (2008)
12. Gupta, S., Reynolds, M.S., Patel, S.N.: ElectriSense: single-point sensing using EMI for electrical event detection and classification in the home. In: *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, Copenhagen, Denmark, pp. 139–148, 26–29 September 2010
13. Srinivasan, D., Ng, W., Liew, A.: Neural-network-based signature recognition for harmonic source identification. *IEEE Trans. Power Del.* **21**, 398–405 (2006)
14. Kim, H., Marwah, M., Arlitt, M., Lyon, G., Han, J.: Unsupervised disaggregation of low frequency power measurements. In: *Proceedings of the 11th SIAM International Conference on Data Mining*, Mesa, AZ, USA, 28–30 April 2011
15. Elhamifar, E., Sastry, S.: Energy disaggregation via learning powerlets and sparse coding. In: *AAAI*, pp. 629–635 (2015)
16. Kolter, J.Z., Batra, S., Ng, A.Y.: Energy disaggregation via discriminative sparse coding. In: *Advances in Neural Information Processing Systems*, pp. 1153–1161 (2010)
17. Gupta, M., Majumdar, A.: Nuclear norm regularized robust dictionary learning for energy disaggregation. In: *2016 24th European Signal Processing Conference (EUSIPCO)*, pp. 677–681. *IEEE* (2016)
18. Kolter, J.Z., Jaakkola, T.: Approximate inference in additive factorial HMMs with application to energy disaggregation. *J. Mach. Learn. Res.* **22**, 1472–1482 (2012)

19. Elhamifar, E., Sapiro, G., Sastry, S.S.: Dissimilarity-based sparse subset selection. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(11), 2182–2197 (2016)
20. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. *Science* **344**(6191), 1492–1496 (2014)
21. Lee, H., Battle, A., Raina, R., et al.: Efficient sparse coding algorithms. In: *Advances in Neural Information Processing Systems*, pp. 801–808 (2007)
22. Hoyer, P.O.: Non-negative sparse coding. In: *Proceedings of the 2002 12th IEEE Workshop on Neural Networks for Signal Processing*, pp. 557–565. IEEE (2002)
23. Bao, C., Ji, H., Quan, Y., et al.: Dictionary learning for sparse coding: algorithms and convergence analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(7), 1356–1369 (2016)
24. Du, X., Guizani, M., Xiao, Y., Chen, H.H.: Secure and efficient time synchronization in heterogeneous sensor networks. *IEEE Trans. Veh. Technol.* **57**(4), 2387–2394 (2008)
25. Hei, X., Du, X., Wu, J., Hu, F.: Defending resource depletion attacks on implantable medical devices. In: *Proceedings of IEEE GLOBECOM 2010, Miami, Florida, USA, December 2010*
26. Yao, X., Han, X., Du, X., Zhou, X.: A lightweight multicast authentication mechanism for small scale IoT applications. *IEEE Sens. J.* **13**(10), 3693–3701 (2013)
27. Xiao, Y., Rayi, V., Sun, B., Du, X., Hu, F., Galloway, M.: A survey of key management schemes in wireless sensor networks. *J. Comput. Commun.* **30**(11–12), 2314–2341 (2007)
28. Du, X., Xiao, Y., Chen, H.H., Wu, Q.: Secure cell relay routing protocol for sensor networks. *Wirel. Commun. Mob. Comput.* **6**(3), 375–391 (2006)
29. Du, X., Guizani, M., Xiao, Y., Chen, H.H.: A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks. *IEEE Trans. Wirel. Commun.* **8**(3), 1223–1229 (2009)