An Intelligent Data Processing Method for Residential Load Analysis

Gao-Chao Cui¹, Li Zhu², Dong-Sheng Wang³ and Jian-Ting Cao⁴

¹ Department of Electronic Engineering, Saitama Institute of Technology, Fukaya, Saitama 3690293, Japan
² Department of Cognitive Science, Xiamen University, Xiamen, Fujian 361005, China
³ Department of Electronic Engineering, Saitama Institute of Technology, Fukaya, Saitama 3690293, Japan
⁴ Department of Electronic Engineering, Saitama Institute of Technology, Fukaya, Saitama 3690293, Japan

Abstract
Objective: is to solve the problems existing in residential load data processing including massive amount data storage, processing efficiency and effective analysis. Methods: we proposed an overall processing architecture based on the residential load data characteristics. The improved clustering and associate rule mining algorithms have been fulfilled parallelization calculated through cloud computing. Results: 1) Residential electricity consumption patterns have been found out 2) The efficiency of our method has been varied compared with the current system under different amount of data. Conclusion: big data processing requirements are arising more in different fields, our method would be helpful in grid construction as well as the residential energy consuming modeling.

Keywords: cloud computing, residential load data, cluster model, pattern recognition.

1. Introduction

Smart power is the important component of smart grid. As the majority of power terminal group, residential users’ electricity behavior would have effect on the smart grid construction directly. Energy efficiency analysis and energy-saving diagnosis can improve the residential energy consuming way, promote energy conservation and then increase the proportion of clean power energy. Therefore, to carry the load analysis of family-oriented has important practical significance [1]-[2].

In recent years, the research on theory and practice of power users’ load analysis have been carried out [3]-[5]. In theoretical aspects, the reference [6] applied the fuzzy neural networks algorithm into industrial large users to tackle uncertainties of the load forecasting but the historical data storage and effective utilization have not been solved. The reference [7] did cluster analysis on large users’ actual load curves and the conclusion is that there is more significant difference between the result from load characteristics and the existing industry and pricing. But the power consuming pattern is not got. In the reference 8, a real-time correction on the electricity industry load composition ration in substation was presented, which has a certain value in application. In the practice aspects, the case study of Torino, Italy shows that electrical consumption the monthly variation is compared with thermal energy-use, which realizes creating mathematical models to evaluate monthly behavior of energy consumption as a function of two predictable variables [9]. State Grid tried to build several intelligent districts in China. In the projects, the effective data sets has been collected to do data mining, which could promote the personalized service and provide the data support for development of power demand-side management response policy [10]. Simultaneously, the power industry has inevitably entered the ‘era of big data’. Its demand for big data has been far beyond other basic energy industry. In the business side, electricity consumers could understand the real-time power consuming information through data mining and then adjust the way on electricity using [11]. Therefore, the study of these mass data storage and processing is the technical problem to be solved.

Cloud computing could integrate the computing power and data resources of clusters working together, through the idea of assigning the large computation tasks into multiple distributed computers. Wherein, Hadoop is an open source distributed computing platform that can efficiently store the data on the distributed file system (HDFS)[12]-[14]. Additionally, residential electricity information relates to user privacy while traditional
modeling methods are lack of consideration of privacy protection which is easy to produce a negative impact. And to ensure data security in Hadoop, the redundant data would be stored, lacking of privacy protection mechanism [15]. Through the foregoing analysis, we studied the construction of cloud computing platform within user privacy protection for residential power consuming. Firstly, the architecture and personalized privacy protection model have been given. Moreover, the improved clustering and associate rule algorithms are combined with MapReduce into parallel processing and then transferred into cloud computing platform. Finally, the platform is validated in two type of data mining instances, including the survey and the measured data respectively. In the conclusion section, we discussed the current applications from our research results and the future work.

2. Cloud Computing System Structure for Residential Electricity Load Data Analysis

We add the privacy protection mechanism into construction of the cloud computing platform and then fulfill the data mining algorithms parallelization combined with the cloud computing to get the residential users’ power consumption pattern mining.

2.1 System structure

The technology framework and operation flow of the proposed cloud computing platform including user privacy protection and data mining functions could be shown in Fig.1.

The functions of different layers in the system are summarized as follows: (1) the basic data source: taking two ways to get data acquisition in user power behavior pattern mining, survey-based data and smart meter energy measuring system operational data. This layer is mainly to complete data collection of residential electricity and then put these data from different devices, means and locations into cloud computing data mining platform. (2) distributed computing layer: core bearer of computing and storage, supporting mining operation based on MapReduce framework and cloud storage technology (HDFS, Hadoop district file system). This layer is mainly to solve the efficient scheduling on the massive data distributed storage and computing task. In addition, the user privacy protection implementation module here to make the personalized privacy protection (here, user, the service provider and trusted third party are simplified into a single entity). (3) mining platform layer: achieving encrypted clustering and associate rule et al. mining algorithms, interacting the business layer with distributed computing layer, receiving instruction of business layer, submitting to a computing layer to get the result. Here, it includes job scheduling, data loading, parallel preprocessing, algorithm parallel processing, and parallel output module. Moreover, the encryption processing is added between parallel preprocessing and algorithm parallel processing to achieve the user privacy protection at the algorithm level. (4) Business application layer: the logic layer of each specific application and only business layer related to business logic.

2.2 Personalized privacy protection model

Personalized privacy protection module is located at the distributed computing layer of the system to make out the personalized privacy protection. For different users’ privacy protection requirements, this module could achieve protection for users’ sensitive information (such as user name, address, etc.), avoiding the disclosure of information. This mechanism is divided into two sub-modules including combination of privacy protection and signal sensitive privacy protection. Combination privacy protection means preventing leakage the users’ sensitive
information constituted by plurality attributes. Single-sensitive privacy protection refers to avoid users’ a single sensitive attribute leakage. Furthermore, to ensure that user applications have good scalability related to workloads dynamic changing, the partitioning technology should be applied into data layer. Cloud computing data hierarchical model for residential users’ privacy protection is shown in Fig.2.

3. Mining Algorithms Encryption and Parallelization

Clustering is the task of grouping a set of objects constituted of multiple similar classes or clusters with the effect of objects in the same cluster (group) are more similar to each other than to those in other cluster (group). By clustering residential electricity data and taking the types and using time of appliances as clustering feature, the user’s power consumption patterns could be computed. Clearly, clustering mining would be effectively used for residential load analysis. And then we proposed improved kmeans clustering algorithm with encryption feature in the paper to ensure the protection the sensitive information constituted by plurality attributes. Single-sensitive privacy protection refers to avoid users’ a single sensitive attribute leakage. Furthermore, to ensure that user applications have good scalability related to workloads dynamic changing, the partitioning technology should be applied into data layer. Cloud computing data hierarchical model for residential users’ privacy protection is shown in Fig.2.

3.1 Improved kmeans database encryption method based on user privacy protection [16]

(1) Background of proposed algorithm

In the residential electricity load analysis, it is quite important to protect the information related to the personal privacy. On one hand, a lot of valuable knowledge could be got through these secret data. On the other hand, the effective protection of user privacy could increase user’s engagement enthusiasm. Residential electricity consumption data is related to user behavior and privacy issues so adding privacy protection is very necessary in the data mining.

The objective of adding privacy protection in data mining is [17]-[18] to control on the degree of privacy method in the mining result and keep the sensitive properties at the same time. In this paper, the improved kmeans algorithm for database encryption is proposed based on user privacy protection mechanism.

(2) Algorithm idea

Here, we added the random interference method to realize privacy protection in the improved kmeans algorithm. The algorithm flow is shown in the Fig.3.

In the algorithm, the interference is achieved by 2-dimension rotation transformation, which includes normalization and data interference. The definition of 2-dimension rotation transformation is formula 1.

\[
\text{Angle} = \begin{bmatrix}
0 & \cot \alpha \\
\tan \alpha & 0
\end{bmatrix}
\]  

(1)

Here, \(\alpha\) is pre-setting rotation angle. Suppose, \(T\) is the initial dataset and \(Q_q, Q_s\) is the vector constituted of number \(a\) and \(b\) where \(1 \leq a, b \leq m\), \(m\) is the number of properties, \(n\) is the number of record. Then, the converted dataset can be expressed as \(T' = \text{Angle} \times T\). Expand to \(m\)-dimension rotation transform, according to ‘if \(m\) is even, then \(k = m / 2\); else then \(k = (m+1) / 2\).’

1). Normalization

Normalization could eliminate the differences in the degree of conversion deviation resulting from different properties ranges. Normalized attribute range also makes it
difficult to distinguish the corresponding original property through the value range. Normalized method uses z-score method here.

2). Encryption processing
Though k item matching way, the k pair properties of all the record are computed to 2 dimensional rotation transformation one by one. Here we used the formula 1 to make the encryption processing to realize that even though the appearances of every property have been changed and the distance between two property is remained. The above mechanism is mainly applied to the record processing in the database and is not related to the classical k-means algorithm flow. So it could be regarded as the step between data preprocessing and mining algorithm. Clearly, by this proposed encryption processing, the Aprior algorithm could be calculated after the transformation of original data. So improved Aprior is gave in the parallelization implementation section.

3.2 Algorithm MapReduce realization

In Hadoop, each MapReduce task is initialized to a job that can be divided into map and reduce phases. These two steps are expressed as map and reduce function respectively. Map function could receive the input of the form <key, value> and then there will produce an intermediate output in this form. Hadoop is responsible for the collection of the intermediate key value with the same key value and then transformation to reduce function. The value set will be processed in reduce function and each function will bring about 0 or 1 output in this form.

In summary, the main task is to build the Map and Reduce functions based on the parallel k-means algorithm in Hadoop [19]. This includes the input and output key paired value <key, value> and the Map and Reduce functions specific logic. At the same time, the main computational task is to assign each sample to its nearest cluster The two steps of generatecluster() and calculatecenter() are independent from each other. So it could be performed in parallel. To be suitable for the MapReduce computation, the data to be processed is stored in rows so that it can press row fragmentation and there will be no correlation among the data partitions. In each iteration, the parallel Mask_k_means algorithm just needs to do the same Map and Reduce operation respectively to finish clustering. Furthermore, to optimize MapReduce task, the k number of integers costumed data set is defined to Hadoop.

1. Improved kmeans parallelization procedure

(1) K definition of the integers set
Writable is the core of Hadoop where we can define the basic data types and operations. Although there are many useful data types in Hadoop, a new data type should be made to meet the specific application. Here, we build a new set including k integers.
(2) Map function design
The task of Map function is to complete the computation the distance to the central point and reset a new marker to its category for every record. The input is the record data to be clustering processed and the former iteration clustering center. For $D[0]...D[n]$, the distance between $cid[0]...cid[n-1]$ should be calculated respectively. Here, the nearest distance is $c[i]$ and the number of it is $Ci$. The key of function input <key, value> pair corresponds to offset of the data file from the starting point in the current sample. The value corresponds to the character consisted of each dimensional coordinate value in the sample. The form of intermediate result <key, value> is <cidIndex, record value>. Function implementation is shown as below.

```java
public class k-means_Mapper(cid){
    for(i=0;i<k;i++) {
        if(distance(cid,originalcluster[i])<mindistance) {
            mindistance=distance(cid,originalcluster[i]);
            currentcluster=i;
        }
    }
    EmitIntermediate(currentcluster,cid);
}
```

(3) Combine function design
To reduce the amount of data transmitted and the communication cost in the algorithm, the combine operation should be designed which could merge the output of Map function locally. In the input pair <key, value> of combine function, key is the category of clustering, cidindex and value is assigned to the string list with k index. In MapReduce, the programming should be added job.setCombinerClass(combine.class).

(4) Reduce function design
All the $D[k]$ with the key will be assigned to the same Reduce process because i is key in the MapReduce framework key value pair. And then calculate the new clustering center and store the new center to the objective clustering array DestinationCluster[]. Main code is given as follows.

```java
public class k-means_Reduce(){
    while(currentPoint=cid.Next()){
        num+=currentPoint.get_Num();
        for(i=0;i<N;i++) sum[i]+=currentpoint.point[i];
    }
    for(i=0;i<N;i++)cid[i]=sum[i]/num;
    Emit(key,cid[i]);
}
```

(5) Improved kmeans algorithm processing flow after MapReduce

2. Improved parallelization Apriori
Here, we achieve the parallelization of improved Apriori by changing reading database into the middle value <key, value> and then the potential item sets would be transferred into middle value <key, value>, realizing in the MapReduce (show in Fig.5).

The two important concepts of Apriori are support and confidence. Support represents the probability of event A and B occur simultaneously; Confidence indicates under the event A occurring probability including the event B occurring, shown in formula 2.

$$Support(A \rightarrow B) = P(AB)$$

$$Confidence = P(B | A) = P(AB) / P(A)$$

(2)

Fig. 5 Improved Apriori algorithm flow structure

4. Results

4.1 Data source
We used built cloud platform to do residential load data processing including data source from questionnaire survey and the real measurement load data respectively. (1) Statistical data is from the 1000 residential electricity information obtained through the questionnaire. Purpose of the survey is to obtain basic residential consumption information. Based on this, we designed the questionnaire with 40 questions (multiple choice, text etc.) including family size, regional location, seasonal difference, typical residential appliances, the composition of appliances and power consuming behaviors in different rooms etc. The questionnaires were distributed 1000 and returned 940 copies (713 valid copies).

(2) Real measurement residential load data
This file contains a residential user’s 20705259 residential power consuming measurement data. Time range is from December 16, 2006 to November 26, 2010 and sampling interval is 1 min. After data preprocessing (presence of nearly 1.25% of the default value), take this set as the room sub-metering (unit: kWh) (show in Table 1).

<table>
<thead>
<tr>
<th>Item</th>
<th>Law</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical appliances</td>
<td>Lighting, refrigerator, washing machine, television, computers, air conditioners, hoods</td>
</tr>
<tr>
<td>Using frequency</td>
<td>Using daily: lights, refrigerator, TV, computer, hoods</td>
</tr>
<tr>
<td>Appliance usage period in</td>
<td>Residential appliances using periods differ from each other clearly</td>
</tr>
<tr>
<td>Using time in different rooms</td>
<td>Kitchen: morning, noon, afternoon, evening peaks. Laundry: morning, noon, afternoon, evening average. Living room: noon, evening peak period.</td>
</tr>
</tbody>
</table>

4.2 Experiment results

Six Dell PowerEdge servers have been used to build the platform configured dual-core 2.0 GHz CPU, 4 GB memory, 1000 Mit/s card and 2TB hard disk. The implementation of the platform is based on component model in Eclipse developing environment. The cluster nodes storage management is achieved by Hadoop framework and the standard of module interaction and resources organization and storage is based on WebService and XML technology.

(1) Experiment 1: Multi-user load data analysis
First, conduct the mining on questionnaire and the conclusion is shown as Tab.2. Besides the above univariate statistical data analysis, we also selected some variable to do cross validation. Some rules have been found out to assist in the subsequent modeling study (see Tab.3).

Table 1: Real-time residential data structure

<table>
<thead>
<tr>
<th>Time</th>
<th>Global active power (kW)</th>
<th>Global reactive power (kVAR)</th>
<th>Voltage (V)</th>
<th>Global intensity (A)</th>
<th>Submetering 1 (kWh)</th>
<th>Submetering 2 (kWh)</th>
<th>Submetering 3 (kWh)</th>
</tr>
</thead>
<tbody>
<tr>
<td>17:24</td>
<td>4.216</td>
<td>0.148</td>
<td>234.840</td>
<td>18.400</td>
<td>0.000</td>
<td>1.000</td>
<td>17.000</td>
</tr>
<tr>
<td>17:25</td>
<td>5.360</td>
<td>0.436</td>
<td>233.630</td>
<td>23.000</td>
<td>0.000</td>
<td>1.000</td>
<td>16.000</td>
</tr>
<tr>
<td>17:26</td>
<td>5.374</td>
<td>0.498</td>
<td>233.290</td>
<td>23.000</td>
<td>0.000</td>
<td>2.000</td>
<td>17.000</td>
</tr>
<tr>
<td>17:27</td>
<td>5.388</td>
<td>0.502</td>
<td>233.740</td>
<td>23.000</td>
<td>0.000</td>
<td>1.000</td>
<td>17.000</td>
</tr>
<tr>
<td>17:28</td>
<td>3.666</td>
<td>0.528</td>
<td>235.680</td>
<td>15.800</td>
<td>0.000</td>
<td>1.000</td>
<td>17.000</td>
</tr>
<tr>
<td>17:29</td>
<td>3.520</td>
<td>0.522</td>
<td>235.020</td>
<td>15.000</td>
<td>0.000</td>
<td>2.000</td>
<td>17.000</td>
</tr>
<tr>
<td>17:30</td>
<td>3.702</td>
<td>0.520</td>
<td>235.090</td>
<td>15.800</td>
<td>0.000</td>
<td>1.000</td>
<td>17.000</td>
</tr>
<tr>
<td>17:31</td>
<td>3.700</td>
<td>0.520</td>
<td>235.220</td>
<td>15.800</td>
<td>0.000</td>
<td>1.000</td>
<td>17.000</td>
</tr>
</tbody>
</table>

Through further analysis, we have drawn some important rules. In the Fig.6, we can see that the difference among different users’ air conditioner and light using period. Simultaneously, we do clustering mining on the effective 837 users’ data set and then analyze the number of user’s power behavior patterns. The number of clusters distribution is shown in Fig.7. Clearly, most of users have less than 7 power behavior patterns.

From the above analysis, there are big difference electricity behavior among users. To do clustering analysis on the view of multiple users is neither scientific nor realistic. So our research object is changed into individual residential unit.
(2) Experiment 2: Single user load analysis
Combined the result of experiment 1, parallel kmeans algorithm and real measured big data to finish the single-user power consumption patterns mining. An example description of the user’s power consuming behavior is shown in Fig.8. The power using behavior pattern of the user is very common and the number of it is 4 clusters. From the result of clustering, there are 4 periods in the first category: low peak, morning peak, smooth, evening peak. There four features in the second category: low peak, morning peak, smooth, not clear evening peak. The third category has low peak, smooth, not clear evening peak. The fourth category is smooth on the whole. Such consumption patterns show us the differences between residential user and big power grid users (such as industry, agriculture user) are more random, tighter association with user power using habit and other factors such as holiday.

The number of each cluster is shown in Table 4.

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Number of Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>47104000.000</td>
</tr>
<tr>
<td>2</td>
<td>1298980000.000</td>
</tr>
<tr>
<td>3</td>
<td>55559000.000</td>
</tr>
<tr>
<td>4</td>
<td>647637000.000</td>
</tr>
<tr>
<td>Valid</td>
<td>2049280000.000</td>
</tr>
<tr>
<td>Missing</td>
<td>25980.000</td>
</tr>
</tbody>
</table>

We computed iteration number is 100 and then 10 of them are given in the Table 5.

<table>
<thead>
<tr>
<th>No.</th>
<th>Change in Cluster Centers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25.555 35.869 24.203 34.085</td>
</tr>
<tr>
<td>2</td>
<td>20.227 .4392 32.951 .5774</td>
</tr>
<tr>
<td>3</td>
<td>9.837 .440 32.951 .774</td>
</tr>
<tr>
<td>4</td>
<td>.323 .003 .234 .004</td>
</tr>
<tr>
<td>5</td>
<td>.062 .001 .036 .002</td>
</tr>
<tr>
<td>6</td>
<td>.013 .000 .003 .001</td>
</tr>
<tr>
<td>7</td>
<td>.001 .000 .000 8.782E-5</td>
</tr>
<tr>
<td>8</td>
<td>.000 1.555E-5 .000 .000</td>
</tr>
</tbody>
</table>

The correlation relationship between different rooms could be calculated through the improved Apriori algorithm (shown in the Table 6).

<table>
<thead>
<tr>
<th>Pre-item</th>
<th>Post-item</th>
<th>Support (%)</th>
<th>confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kitchen room</td>
<td>Laundry room</td>
<td>41.234</td>
<td>58.126</td>
</tr>
<tr>
<td>Laundry room</td>
<td>Kitchen room</td>
<td>39.812</td>
<td>57.232</td>
</tr>
<tr>
<td>Kitchen room</td>
<td>Living room</td>
<td>38.016</td>
<td>55.324</td>
</tr>
<tr>
<td>Living room</td>
<td>Kitchen room</td>
<td>42.091</td>
<td>56.879</td>
</tr>
<tr>
<td>Laundry room</td>
<td>Living room</td>
<td>40.287</td>
<td>57.541</td>
</tr>
<tr>
<td>Living room</td>
<td>Laundry room</td>
<td>40.027</td>
<td>57.178</td>
</tr>
</tbody>
</table>

(3) Experiment 3: Parallel algorithm efficiency
This experiment is mainly used to verify the paralleled algorithm efficiency. To get contrast with the existing system, we used the data from one local grid wide area measurement system (WAMS) in 20 days. The size of data
is 1.5TB and the proposed clustering algorithm is applied to process the data and power system breakdown. Dynamic security assessment and early warning system of power system (PDSA) could process data and backup from WAMS. The comparison result processing efficiency between PDSA and the proposed cloud computing platform under different data size conditions is shown in the Table 7. Removing the network random factors interfere, the platform processing efficiency has the optimization effect.

<table>
<thead>
<tr>
<th>Data amount</th>
<th>Response time of PDSA</th>
<th>Response time of proposed platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>1G</td>
<td>12s</td>
<td>28s</td>
</tr>
<tr>
<td>100G</td>
<td>150s</td>
<td>123s</td>
</tr>
<tr>
<td>500G</td>
<td>1007s</td>
<td>236s</td>
</tr>
<tr>
<td>1T</td>
<td>2548s</td>
<td>367s</td>
</tr>
</tbody>
</table>

5. Applications and Conclusions

The cloud computing platform for residential user’s power consuming behavior analysis has been studied based on the massive data set from residential users. First, the platform architecture considering user privacy protection and the implementation of the mechanism have been given. Then, the Mask_k_means algorithm has been designed for preprocessing. Moreover, combined the cloud computing with data mining, the parallelization of the proposed algorithm has been realized. Finally, three validation tests have been complete through the proposed cloud computing platform.

The important results are summarized. Modeling on multi-user is not realistic. But it is reasonable for modeling on single user’s power consuming behavior. The processing advantage becomes clearer with the amount of data increasing. Just as the Fig. 7 shows, we can see this kind of electricity consumption behavior could be classified into four feature-period categories. So the modeling on residential users’ behavior would be built based on feature periods. Moreover, a new grid meter is designed (shown in the Fig. 9). There are several newer features in our meter design compared with the common meter. 1) Data collection; here, the meter could not only getting the real-time data from the electricity devices slots but also could get the data from other data source such as the offline data stored in the client pc. 2) Core CPU module; the chip is used the series of ARM9 and then two main functions can be realized which include the encryption processing in the pre-processing step and the data mining analysis such as the clustering and associate rule processing methods. 3) results storage; Through the serial communication slot, we can upload the results into the client clustering computers and then the historical results could be transferred into the core cpu through this slot.

The future research will focus on the assessment of the energy efficiency for different families and then make out the personalized energy saving policy, combined with the residential user’s power consuming behavior patterns.

References


[16] Zhu Li. The study of daily load curve disaggregation for household electron energy efficiency [D]. Northeast Dianli University, 2014.


Gaochao Cui, PhD student, will receive PhD degree in signal processing, information science, Saitama Institute of Technology, Japan, in 2017. He has published several papers related in signal processing and brain computer interface. Many conferences have invited him to give presentations to the international researchers. In the first year of PhD, he was selected as the member of Junior Research Associate program in RIKEN, Japan and worked out ICA toolbox with other colleagues. Corresponding Author.

Li Zhu, PhD student, will receive PhD degree in computer science, Cognitive Science, Xiamen University. Her research topics are related to data mining, signal processing and cloud computing. She has published several papers, one book and patent related to power system modeling, machine learning and big data processing.

Dongsheng Wang, master student, has interest in computer programming language and large-scale signal processing.

Jianting Cao, PhD, is the professor at Saitama Institute of Technology, Japan. His research interests in algorithms, machine learning and cloud computing.