AMR 데이터에서의 전력 부하 패턴 분류

Minghao Piao*, 박진형*, 이헌규*, 신진호**, 류근호*
*충북대학교 데이터베이스/바이오인포매틱스 연구실
**한국전력연구원 전력 정보 기술 그룹
e-mail : *{bluemhp, neozean, hglee, khryu}@dblab.chungbuk.ac.kr
**jinho@kepri.re.kr

Power Load Pattern Classification from AMR Data

Minghao Piao*, Jin-Hyung Park*, Heon-Gyu Lee*, Jin-Ho Shin**, Keun-Ho Ryu*
*Database/Bioinformatics Laboratory, Chungbuk National University
** Power Information Technology Group, Korea Electric Power Research Institute

Abstract

Currently an automated methodology based on data mining techniques is presented for the prediction of customer load patterns in load demand data. The main aim of our work is to forecast customers' contract information from capacity of daily power consumption patterns. According to the result, we try to evaluate the contract information's suitability. The proposed our approach consists of three stages: (*i*) data preprocessing: noise or outlier is detected and removed (*ii*) cluster analysis: SOMs clustering is used to create load patterns and the representative load profiles and *(iii)* classification: we applied the K-NNs classifier in order to predict the customers' contract information base on power consumption patterns. According to the our proposed methodology, power load measured from AMR(automatic meter reading) system, as well as customer indexes, were used as inputs. The output was the classification of representative load profiles (or classes). Lastly, in order to evaluate K-NN classification technique, the proposed methodology was applied on a set of high voltage customers of the Korea power system and the results of our experiments was presented.

1. Introduction

Electrical customer load patterns classification has been an important issue in the power industry. Load patterns prediction deals with the discovery of power load patterns from load demand data. Therefore, accurate classification models are essential to the operation and planning of an electricity utility.

Customer classification helps an electric utility to make important decisions including decisions on purchasing, load switching, and also helps to develop the infrastructure. It is extremely important for electric energy generation and transmission, distribution and electrical markets. In power system, data mining [1, 2, 3] is the most commonly used methods to determinate load profiles and extract regularities in load data and thus has been the target of some investigations for its used in load pattern forecasting. In particular, it promises to help in the detection of previously unseen load patterns by establishing sets of observed regularities in load demand data. These sets can be compared to current load pattern for deviation analysis. Load patterns prediction [4, 5] using data mining is usually made by building models on relative information, temperature and previous load demand data.

The main aim of our work is to forecast customers' contract information from capacity of daily power usage dataset and customer information in terms of accuracy for the classification processes. To achieve this objective, we attempt to apply clustering and classification techniques and the main tasks are the following:

1. Data preprocessing is performed to detect and remove the noise and outlier patterns using k-means clustering technique.
2. Cluster analysis is performed to detect load pattern classes and the load profiles for each class.
3. Classification is performed using customer load patterns to build a classifier able to assign different customer load patterns to the existing classes.

In this paper, we applied SOMs (self-organizing maps) method to determine the number of clusters and performed customer classification base on K-NNs (k- nearest neighbors). Details about preprocessing are represented in Section 2 and clustering is detailed in Section 3. Section 4 presents our performance study. Section 5 summarizes this paper.

2. Data preprocessing

The AMR data is the power consumption of the customers during each 15 minutes interval. It is incomplete, noisy and inconsistent. For data mining practice, we applied data preprocessing to the AMR data.

2.1 Data cleaning.

This process includes: ignore missing values, smooth noisy data and identify or remove outliers. Tuples that have "null" values or 0 are removed. For handling the outlier, we redefined 2 concepts of outliers [6]:

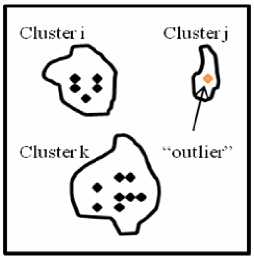


Figure 1

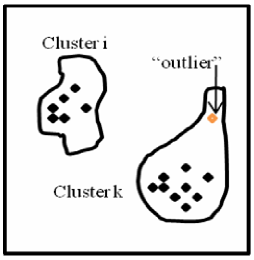


Figure 2

1. The outlier is an element of the group which is located far from the center of cluster. It illustrates that the tuple has low similarity between other tuples because it contains outliers in the tuple and has long distance in the cluster (Figure 1).
2. The outlier is an element of group which is located far from the other groups. It has low similarity between other clusters because it contains only one element (Figure 2).

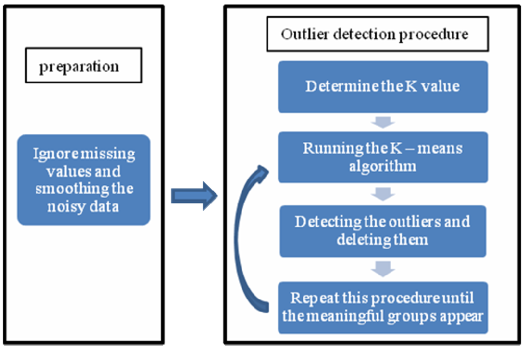
We used k-means to detect outliers. The k-means is a clustering analysis algorithm that groups objects based on their feature values into k disjoint clusters. Objects are classified into the same cluster where represents similar patterns. The similarity between objects is calculated by Euclidean distance as below:

$$dist(x,y) = \sqrt{\sum_{k=1}^{H}(x_k - y_k)^2} \qquad (1)$$

where:

Symbol	Description
x_k	k^{th} feature of object x.
y_k	k^{th} feature of object y.
dist	The distance between two objects in the Euclidean space.

The data may contain outliers which do not belong to a bigger cluster. This does not disturb the K-means clustering process even the number of outliers is small. The clustering algorithm may occur that clusters are very close to each other. This can have several reasons: Either the number of clusters k has been badly chosen or the training data is very homogeneous, e.g. because it does not contain any anomalous traffic or because the anomalous traffic looks very similar to normal traffic.


preparation
Ignore missing values and smoothing the noisy data
Outlier detection procedure
Determine the K value
Running the K – means algorithm
Detecting the outliers and deleting them
Repeat this procedure until the meaningful groups appear

(Figure 3) Procedure of outlier detection.

2.2 Data transformation.

When applying the data mining methods to the raw data, it does not produce a useful result since the numerous features has different weights. Hence, we convert the raw data into relevant values.

$$nor(x) = \frac{x_i}{max(x_i)} \qquad (2)$$

where:

Symbol	Description
x_i	i^{th} feature of object x.
nor	The normalized value of i^{th} feature of object x.

3. Clustering analysis

When we try to analyze the customers who represent similar behaviors, the best way is grouping them into group base on their behavior. To accomplish that, we can use clustering method. The first task is to determine the number of clusters. We use k-means to evaluate the reproducibility of the clustering and according to the result we select the optimal number of the clusters.

3.1 Evaluation for reproducibility of the clustering.

Data partitioning was applied in supervised learning. However, we can use it to determine the number of clusters for clustering. If the number of clusters is optimal and the clustering algorithm will produce the similar result when it runs on two data set which from same mechanism. In this paper, we use data partitioning and K-means to select the optimal number.

First, we partitioned the training data set into 3 parts. The ratio is 4:4:2. The larger two data set used as training set and the smaller one used as test set.

Second, run the k-means on two training data set to produce *Rule1* and *Rule2*.

Third, apply the *Rule1* and *Rule2* on the test set and produce a confusion matrix to evaluate the result. If the selected number of clusters is optimal, the matrix will show strong homologous characteristic. As shown in tables the number of clusters is selected as 4.

<Table 1> Confusion Matrix for k=3

K=3		Rule1		
		Cluster1	Cluster2	Cluster3
Rule2	Cluster1	225	65	0
	Cluster2	0	22	0
	Cluster3	0	27	73

<Table 2> Confusion Matrix for k=4

K=4		Rule1			
		Cluster1	Cluster2	Cluster3	Cluster4
Rule2	Cluster1	205	0	0	16
	Cluster2	1	4	0	0
	Cluster3	5	0	0	97
	Cluster4	0	2	73	9

<Table 3> Confusion Matrix for k=5

K=5		Rule1				
		Clus1	Clus2	Clus3	Clus4	Clus5
Rule2	Clus1	120	0	0	0	27
	Clus2	0	4	0	0	0
	Clus3	6	0	0	70	1
	Clus4	0	2	69	9	0
	Clus5	21	0	0	21	62

<Table 4> Confusion Matrix for k=6

K=6		Rule1					
		Clu1	Clu2	Clu3	Clu4	Clu5	Clu6
Rule2	Clu1	98	0	0	0	51	3
	Clu2	0	4	0	0	0	0
	Clu3	29	0	0	36	1	3
	Clu4	0	1	27	22	0	7
	Clu5	0	0	0	9	14	65
	Clu6	0	0	41	0	0	1

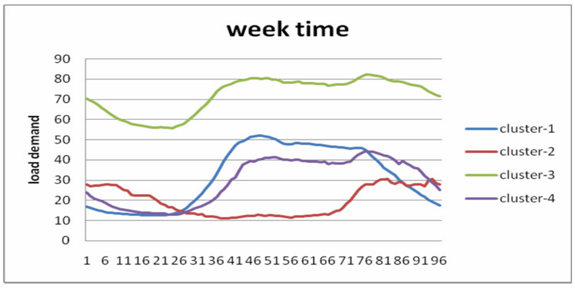
3.2 The application of the SOMs.

We used the K value which selected in Section 3.1 to determine the size of maps of SOM. SOM training algorithm resembles vector quantization algorithms, such as K-means [7]. It means the K value selected will give the result of SOM, respectively.

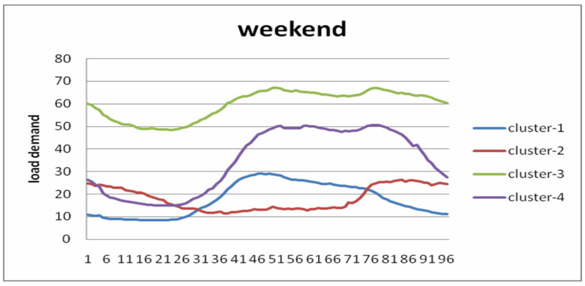
SOMs operate in two modes: training and mapping. Training builds the map using input examples. Mapping automatically classifies a new input vector. A SOM consists of components called nodes or neurons. Associated with each node is a weight vector of the same dimension as the input data vectors and a position in the map space. The self-organizing map describes a mapping from a higher dimensional input space to a lower dimensional map space.

In each training step, one sample vector *x* from the input data set is chosen randomly, and the distances between it and all the weigh vectors of the SOMs are calculated using Euclidean distance. It is important to note that different training (randomly) sessions usually produce a different map even for the same data set. Notice that these maps conserve the relative position between the elemental cells but not their absolute position for different training sessions.

We partitioned training data set into week time and weekend. For each groups of data, we clustered data set into 4 clusters using 4 x 1 maps and add contract information to each groups. Figure 4 and 5 represent representational patterns for week time and weekend.



(Figure 4) week time



(Figure 5) weekend

The SOM provides a very simple visual explanation of the clustering procedure. However, its result was influenced by the given K value, outlier detection and the size of maps. It is important to decide the size of maps and the value of N and M for N by M maps. For example, 4 by 1 and 2 by 2 maps produce the different results.

4. Customer classification base on the K-NN.

In our study, we try to give contract information to customers who only have pattern information, and also give information to evaluate the customers' consumption behavior. This work will base on the power consumption patterns. Therefore, we need an algorithm that can be used in pattern recognition.

It is reasonable to assume that observations which are close together will have the same classification, or at least will have almost the same probability distributions on their respective classification. Thus, to classify the unknown sample *x* we may wish to weighting the evidence of the nearby's most heavily. The simplest nonparametric decision procedure of this form is the nearest neighbor rule, which classifies *x* in the category of its nearest neighbor [8].

In pattern recognition, the K-nearest neighbor (K-NN) is a method for classifying objects based on closest training examples in the feature space. K-NN is a type of instance - based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. Object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors.

$$dist(x,y) = \sqrt{\frac{1}{H}\sum_{k=1}^{H}(x_k - y_k)^2} \qquad (1)$$

where:

Symbol	Description
x_k	k^{th} feature of object x.
y_k	k^{th} feature of object y.
H	Number of dimension.
dist	The distance between two objects in the Euclidean space.

We applied K-NN on the data that has been labeled by SOMs and produced a classifier. When we apply the K-NN in pattern recognition, we try to make the K value be small. It is in order that let the points to be close enough to *x* to given an accurate estimate of the probabilities of the true class *x*. Single-NN always choose the closest neighbor for input sample. Hence, we apply the Single-NN to forecasting the customer's contract information base on their load patterns. If k = 1, then the object is simply assigned to the class of its nearest neighbor. For any number *n* of samples, the Single-NN rule has strictly lower probability of error than any other K-NN rule against certain classes of distributions [8]. Suppose the training data set is clustered very well, after predicted, the matrix about the original class labels and predicted class labels will show strong correspondence. It means the given value of the clusters is optimal and Single-NN works directly on the input data.

The performance of the weekend and week time is shown in below.

<Table 5> Confusion Matrix for weekend

Cluster-1	Cluster-2	Cluster-3	Cluster-4	
315	39	0	0	Cluster-1
40	226	29	0	Cluster-2
0	31	171	22	Cluster-3
0	0	13	293	Cluster-4
Total	1179	Correctly classified		85.2%

<Table 6> Detailed Accuracy by Class for weekend

TP	FP	Precision	Recall	Class
0.89	0.048	0.887	0.89	Cluster-1
0.766	0.079	0.764	0.766	Cluster-2
0.763	0.044	0.803	0.763	Cluster-3
0.958	0.025	0.93	0.958	Cluster-4

<Table 7> Confusion Matrix for week time

Cluster-1	Cluster-2	Cluster-3	Cluster-4	
372	33	0	0	Cluster-1
27	254	25	0	Cluster-2
0	29	205	5	Cluster-3
0	0	10	219	Cluster-4
Total	1179	Correctly classified		89.1 %

<Table 8> Detailed Accuracy by Class for Week time

TP	FP	Precision	Recall	Class
0.919	0.035	0.932	0.919	Cluster-1
0.83	0.071	0.804	0.83	Cluster-2
0.858	0.037	0.854	0.858	Cluster-3
0.956	0.005	0.978	0.956	Cluster-4

Between the input data, some customers were predicted into different groups. The first case is that the customer was labeled differently with his own contract information. It is because that his consumption behavior is different with others and more like another group. The second case is that the customer was labeled differently in week time and weekend. It is because their power consumption is totally different during the week time and weekend. Such kinds of customers are the target to electricity utilities, and they have to make examination when it is serious.

5. Conclusions

In this paper, the proposed main mining tasks include cluster analysis and customer classification method. Cluster analysis is used to define load pattern classes and the representative load profiles for each class. For each class, they are mapped into private contract information from load profiles. Classification method uses representative load profiles to build a classifier able to assign different load patterns to the existing classes. Finally, existing contract information and the predicted information are compared to evaluate the suitability.

In experiment, the applied SOMs and K-NNs classifier tested KEPRI AMR data. The performance has been shown that methods have been used is admissible for this case.

Acknowledgements

This work is supported by development of AMR system interfacing model on internet GIS environment project of the Korea Electric Power Research Institute (KEPRI) and a Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MOST) (R01-2008-000-10926-0).

References

[1] S. J. Huang, K. Shih, "Short-term load forecasting via ARMA model identification including non-Gaussian process considerations," IEEE Trans. Power System, Vol. 18, No. 2, pp. 673-679, 2003.

[2] G. Chicco, R. Napoli, P. Postulache, M. Scutariu, C. Toader, "Customer characterization options for improving the tariff offer," IEEE Trans. Power System, Vol. 18, pp.381-387, 2003.

[3] B. Pitt, D. Kirchen, "Applications of data mining techniques to load profiling," In Proc. IEEE PICA, pp. 131-136, 1999.

[4] Heon Gyu Lee, Jin-ho Shin, Keun Ho Ryu, "Application of Calendar-Based Temporal Classification to Forecast Customer Load Patterns from Load Demand Data," to be appeared in IEEE CIT 2008.

[5] Heon Gyu Lee, Jin-Ho Shin, Hong Kyu Park, Young-il Kim, Bong-Jae Lee, Keun Ho Ryu, "Temporal Classification Method for Forecasting Power Load Patterns From AMR Data," Korean Journal of Remote Sensing, Vol. 23, No. 5, pp.393-400, 2007.

[6] Kyung – A Yoon, Oh – Sung Kwon, Doo – Hwan Bae, "An Approach to Outlier Detection of the Software Measurement Data using the K – means Clustering Method," IEEE ESEM 2007, pp.443-445, 2007.

[7] J. Hartigan, M. Wong, "A K-means clustering algorithm," Appl.Stat., Vol.28, No. 1, pp.100-108, 1979.

[8] T. M. Cover, P. E. Hart. "Nearest Neighbor Pattern Classification," IEEE Transactions on Information Theory, Vol.13, No.1, pp.21-27, Jan.1967.