전력 부하 패턴 자동 예측을 위한 분류 기법

Piao Minghao○ 박진형 이헌규 류근호
충북대학교 데이터베이스/바이오인포매틱스 연구실
{bluemhp, neozean, hglee, khryu}@dblab.chungbuk.ac.kr

Classification Methods for Automated Prediction of Power Load Patterns

Piao Minghao○ Jin-Hyung Park Heon-Gyu Lee Keun-Ho Ryu
Database/Bioinformatics Laboratory, Chungbuk National University

Abstract

Currently an automated methodology based on data mining techniques is presented for the prediction of customer load patterns in long duration load profiles. The proposed our approach consists of three stages: (i) data pre-processing: noise or outlier is removed and the continuous attribute-valued features are transformed to discrete values, (ii) cluster analysis: k-means clustering is used to create load pattern classes and the representative load profiles for each class and (iii) classification: we evaluated several supervised learning methods in order to select a suitable prediction method. According to the proposed methodology, power load measured from AMR (automatic meter reading) system, as well as customer indexes, were used as inputs for clustering. The output of clustering was the classification of representative load profiles (or classes). In order to evaluate the result of forecasting load patterns, the several classification methods were applied on a set of high voltage customers of the Korea power system and derived class labels from clustering and other features are used as input to produce classifiers. Lastly, the result of our experiments was presented.

1. Introduction

Electrical customer load patterns prediction has been an important issue in the power industry. Load patterns prediction deals with the discovery of power load patterns from load demand data. It attempts to identify existing customer load patterns and recognize new load forecasting methods, employing methods from sciences such as statistical analysis [1], [2] and data mining techniques [3], [4], [5]. In power system, data mining 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 using data mining is usually made by building models on relative information, weather, temperature and previous load demand data. Such prediction is aimed at short-term prediction [6], since mid- and long-term prediction may not be reliant because the results of prediction contain high forecasting errors. However, mid- and long-term (load patterns for longer period) forecasting on load demand is very useful and interest.

The main objective of our work is to forecast monthly load patterns 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 their use in load pattern forecasting. For the forecasting customer load patterns, the main tasks are the following: and a framework of our approaches is showed in Figure. 1.

① Cluster analysis is performed to detect load pattern classes and the load profiles for each class.
② Classification module is performed using customer load profiles to build a classifier able to assign different customer load patterns to the existing classes.

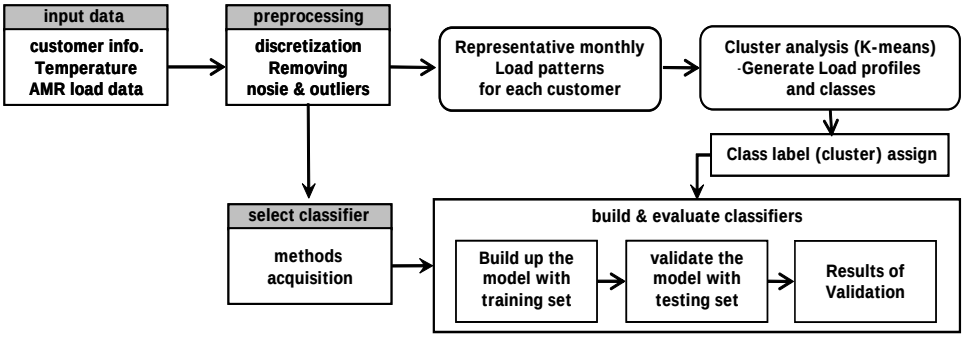

Figure1. Load pattern prediction framework

③ The classifiers are evaluated to select a suitable classification method.

2. Data Collection and Preprocessing

A case study concerning a database with load patterns and power usage from 1049 high voltage consumers is considered and this information has been collected by KEPRI (Korea Electric Power Research Institute). The collected load patterns from AMR were made during a period of ten months (from Jan. to Oct.) in 2007. The instant power consumption for each consumer was collected with a cadence of 15 min. The commercial index related with customer electricity use code, and max. load demand and temperatures are also applied. To compare the load patterns, we use features of load shapes [7], able to capture relevant information about the consumption behavior, must be create the classifier. These features must contain information about the daily load curve shape of each consumer and presented in Table 1.

Lastly, since the extracted features contain continuous variables, those variables also must be made discrete. Therefore, entropy-based discretization has been used because the intervals are selected according to the information they contribute target variable. Due to the decision tree's discretization [8], all continuous contributed variables are cut up into a number of intervals. Let T partition the set D of examples into the subsets D1 and D2. Let there be k classes C1,...,Ck. Let P(Ci, Dj) be the proportion of examples in Dj that have class Ci. The class entropy of a subset Dj, j=1, 2 is defined as,

$$Ent(D_j) = -\sum_{i=1}^{k} P(C_i, D_j)\log(P(C_i, D_j)) \qquad (1)$$

Suppose the subsets D1 and D2 are induced by partitioning a feature A at point T. Then, the class information entropy of the partition, denoted E(A, T ; D), is given by:

$$E(A,T;D) = \frac{|D_1|}{D} Ent(D_1) + \frac{|D_2|}{D} Ent(D_2) \qquad (2)$$

A binary discretization for A is determined by selecting the cut point TA for which E(A, T ; D) is minimal amongst all the candidate cut point. The same process can be applied recursively to D1 and D2 until some stopping criteria is reached.

The Minimal Description Length Principle is used to stop partitioning. Recursive partitioning within a set of values D stop if

$$Gain(A,T;D) < \frac{\log_2(N-1)}{N} + \frac{\delta(A,T;D)}{N}, \qquad (3)$$

where N is the number of values in the set D,

$$Gain(A, T\ ;\ D) = Ent(D)$$

$$\delta(A,T;D) = \log_2(3^k - 2) - [k \cdot Ent(D) - k_1 \cdot Ent(D_1) - k_2 \cdot Ent(S_2)]$$

, and ki is the number of class labels represented in the set Di. Fig. 2 shows the data preprocessing for load demand data.

Table1. Load curve shape features

Shape Feature	Definition
L1: Load Factor (24h)	
L2: Night Impact (8h:23pm~07am)	
L3: Lunch Impact (3h:12am~03pm)	

Feature		Type	Description
Customer Electricity Use code		nominal	Different 21 values
Max load demand		continuous	Min.: 0.32 ~ Max.: 5544
Temperature		continuous	Min.: -15.34 ~ Max.: 35.23
AMR daily Power usage (15 min. interval)	0	continuous	Min.: 0.32 ~ Max.: 5544
	15	continuous	Min.: 0.32 ~ Max.: 5544
	…	continuous	Min.: 0.32 ~ Max.: 5544
	2345	continuous	Min.: 0.32 ~ Max.: 5544
Class	Cluster	nominal	{cluster1, …, cluster 12}

Data preprocessing

Feature		Type	Description
Customer Electricity Use code		nominal	Different 21 values
Max load demand		nominal	Discrete values
Temperature		nominal	Discrete values
Daily Load factors	0	nominal	Discrete values
	15	nominal	Discrete values
	…	nominal	Discrete values
	2345	nominal	Discrete values
Class	Cluster	nominal	{cluster1, …, cluster 12}

Figure 2. Data preprocessing for AMR data.

3. Generating Representative Load profiles using K−means

We describe clustering algorithms for generating the load profiles and class label which will be used classification process. The load pattern associated with any customer contains the information of commercial indexes such electricity use and load factors which recoded every 15 minutes. The representative monthly load pattern (i.e. April, June, Sep., Oct. 2007) of the mth consumer is following:

$$V(m)=\sum_{i=1}^{k}V(m)_i, V(m)_i=\{V_0(m)_i,\ldots,V_t(m)_i,\ldots, \quad (4)$$

where t=0, …, T with T=2345, representing the 15 min. interval between the collected measurements. In cluster analysis, K-means is used to group the load patterns and the optimal clusters are obtained. The use of clustering in this step detects the number of classes as an input of the classification model.

In order to evaluate the performance of the clustering algorithm, adequacy measure (MIA: Mean Index Adequacy [9]) is applied. The purpose of adequacy measure is to obtain separated and compact clusters that make the load patterns (curves) well identified. Let's suppose a set of M load patterns separated in k clusters with k=1,… , K and K is the total number of clusters. Each cluster center is formed by a subset C(k) of load patterns, where r(k)is a pattern assigned to cluster k. MIA is defined as the average of the distances between each input vector assigned to the cluster and its center. The K-means algorithm was used to generate class labels based on the MIA measure. It is possible to see that 12 clusters would be good choice, considering the MIA.

4. Classification Methods for Forecasting Load Patterns

In this section, we describe several classification methods to forecasting customer load patterns.

4.1 CMAR (Classification based on Multiple Association Rules)

CMAR is the associative classification. Associative classification uses association mining techniques that search for frequently occurring patterns in large data sets. The patterns may generate rules, which can be analyzed for use in classification. CMAR [10] generates rules using the FP-growth algorithm. In the pruning phase, CMAR selects only positively correlated rules. This means that for a rule $r: p_1 \wedge p_2, \ldots, p_k \rightarrow c$, (A literal pis a attribute-value pair and a rule r consist of a conjunction of literals $p_1, p_2, \ldots, p_k$, associated with a class label c.) the algorithm checks whether $p_1 \wedge p_2, \ldots, p_k$ is positively correlated with by chi-square testing (the chi-square test method measures the significance of associations). Only the rules that are positively correlated, i.e., those having χ^2 value larger than a significance level threshold, are used for later classification. All of the other rules are pruned. Also CMAR prunes rules based on database coverage. That is, CMAR removes one data object from the training dataset after it is covered by at least υ rules (υ expresses the database coverage parameter). That allows more selected rules. In the testing phase, for a new sample, CMAR collects the subset of rules matching the sample from the total set of rules. If all the rules have the same class, CMAR assigns this class to the new sample. If the rules are not consistent in the class label, CMAR divides the rules into groups according to the class label and yields the label of the "strongest"group. The "strength" of a group of rules is computed using weighted chi-square.

4.2 CPAR (Classification based on Predictive Association Rules)

CMAR adopts method of frequent itemset mining to generate candidate rules, which include all conjunctions of attribute-value pairs. However, CMAR

generates a large number of rules. CPAR [11] takes a different approach to rule generation, based on a rule generation algorithm for classification known as FOIL [12]. FOIL builds rules to distinguish positive examples from negative ones. FOIL repeatedly searches for the current best rule and removes all the positive examples covered by the rule until all the positive examples in the dataset are covered. For multi-class problems, FOIL is applied to each class: the examples for each class are used as positive examples and those of other classes as negative ones. The rules for all classes are merged together to form the resulted rule set. During classification, CPAR employs a somewhat different multiple rule strategy than CMAR. If more than one rule satisfies a new one, P, the rules are divided into groups according to class, similar to CMAR. However, CPAR uses the best k rules of each group to predict the class label of P, based on expected accuracy. By considering the best k rules rather than all of the rules of a group, it avoids the influence of lower ranked rules. The accuracy of CPAR on numerous data sets was shown to be close to that of CMAR. However, since CPAR generates far fewer rules than CMAR, it shows much better efficiency with large sets of training data.

4.3 Support Vector Machine

A SVM is an algorithm for the classification of both linear and nonlinear data. It transforms the original data in a higher dimension, from where it can find a hyper-plane for separation of the data using essential training examples called support vectors. The SVM is a basically two class classifier and can be extended for multi-class classification. In our model each object is mapped to a point in a high dimensional space, each dimension of which corresponds to features. The coordinates of the point are the frequencies of the features in the corresponding dimensions. SVM learns, in the training step, the maximum-margin hyper-planes separating each class. In testing step, it classifies a new object by mapping it to a point in the same high-dimensional space divided by the hyper-plane learned in the training step. For our experiments, we used the sequential minimal optimization (SMO) algorithm [13].

4.4 C4.5 (Decision Tree)

C4.5 is a decision tree generating algorithm, based on the ID3 algorithm [14]. It contains several improvements, especially needed for software implementation. These improvements contain:

① Handling both continuous and discrete attributes - In order to handle continuous attributes, C4.5 creates a threshold and then splits the list into those whose attribute value is above the threshold and those that are less than or equal to it.

② Handling training data with missing attribute values - Missing attribute values are simply not used in gain and entropy calculations.

③ Handling attributes with differing costs.

④ Pruning trees after creation - C4.5 goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with leaf nodes.

5. Experiments and Results

In this section, we evaluate our experiments in building a customer load pattern prediction model.
In our experiment, we evaluate the classifiers performance. The accuracy was obtained by using the methodology of stratified 10-fold cross-validation. One of the criteria for evaluating classifier is the accuracy of the classification results. We want to be able access how well the classifier can classify. For this purpose, the mean absolute error, root mean squared error, and accuracy were used.

The parameters of the CMAR were set as follows: the min. support was set to 0.4%, the min. confidence to 70%, and the database coverage was set to 3.75 (critical threshold for a 5% "significance"level, assuming degree of freedom equivalent to 1). More specifically for the CPAR algorithm, the minimum gain set to 0.7, gain similarity ratio to 0.95 and the weight decay factor to 0.67. The best ten rules were used for prediction. For the SVM, the soft margin allowed errors during training. We set 0.1 for the soft margin value. C4.5 parameters were default values. We tested both the C4.5 tree method and the rule method.

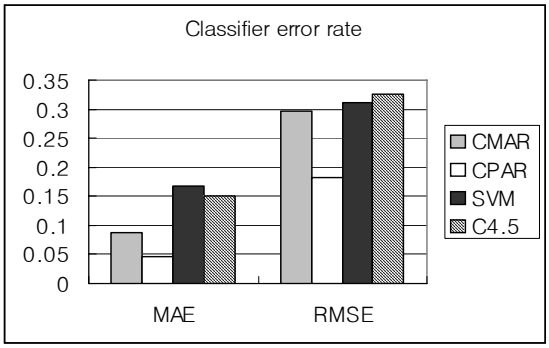

Figure 3. Comparison of classifier error rate.

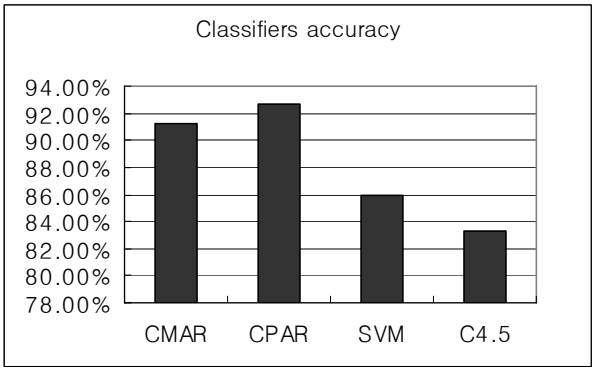

Figure 4. Comparison of classifier accuracy.

As shown in Fig. 3, the CPAR algorithm shows the lowest error rate than others. The error rate is almost about half of decision tree both on mean absolute error and root mean squared error. At Figure 4, the CPAR and CMAR show the highest accuracy.

6. Conclusion

The purpose of this paper is to find useful features and the automated methodology to predict the power load patterns. In this study, we applied k-means clustering to create load pattern classes and the representative load profiles for each class. To compare the load patterns, we used features of load curve shapes such as load factor, night impact and lunch impact, and temperature and max load demand. These features contain information about the daily load curve shape of each consumer. For forecasting the load patterns, we applied several classification methods such as CMAR, CPAR, SVM and C4.5 on the data set of high voltage customers of the Korea power system. In order to evaluate the performance of classifiers, the mean absolute error, root mean squared error, and accuracy were used. In our experiments, CPAR algorithm outperformed the other classifiers.

Acknowledgements

This work is supported by a Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MOST) (R01-2007-000-10926-0).

References

[1] Chris Perry, "Short-Term Load Forecasting using Multiple Regression Analysis," Rural Electric Power Conference, 1999.

[2] Alexander Bruhns, Gilles Deurveilher, Jean-Sebastien Roy, "A non-linear regression model for mid-term load forecasting and improvements in seasonality," 15th PSCC, 2005.

[3] 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.

[4] 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.

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

[6] Henrique Steinherz Hippert, Carlos Eduardo Pedreire, Reinaldo Castro Souza, "Neural Networks for Short-Term Load Forecasting: A review and Evaluation," IEEE Transactions on Power Systems, Vol.16, No.1, February 2001.

[7] Vera Figueiredo, Fatima Rodrigures, Zita Vale, Joaquim Borges Gouveia, "An Electric Energy Consumer Characterization Framework Based on Data Mining Techniques," IEEE Transactions on Power Systems, Vol.20, No.2, May 2005.

[8] James Dougherty, Ron Kohavi, Mehran Sahami, Supervised and Unsupervised Discretization of Continuous Features, Machine Learning: Proceeding of the 12th International Conference, Morgan Kaufmann Publishers, 1995.

[9] George J. Tsekouras, Nikos D. Hatziargyriou, Evangelos N. Dialynas, "Two-Stage Pattern Recognition of Load Curves for Classification of Electricity Customers," IEEE Transactions on Power Systems, Vol.22, No.3, August 2007.

[10] Li W., Han, J. and Pei, "CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rule," Proc ICDM 2001, pp369-376, January 2001.

[11] Yin, X. and Han, CPAR: "Classification based on Predictive Association Rules," Proc. SIAM Int. Conf. on Data Mining (SDM'03), San Fransisco, CA, pp. 331-335 January 2003.

[12] Coenen, LUCS-KDD implementations of FOIL, http://www.cxc.liv.ac.uk/~frans/KDD/Software/FOIL_PRM_CPAR/foil.html, Department of Computer Science, The University of Liverpool, UK, February 2004.

[13] John C. Platt, "Sequential Mining Optimization: A Fast Algorithm for Training Support Vector Machines," Microsoft Research Technical Report MSR-TR-98-14, 1998.

[14] Quinlan, J.R. , "C4.5: Programs for Machine Learning, Morgan Kauffman," 1993.

[15] K. Liu, S. Subbarayan, R. R. Shoults, M. T. Manry, C. Kwan, F. L. Lewis, J. Naccarino, "Comparison of Very Short-Term Load Forecasting Techniques," IEEE Transactions on Power Systems, Vol.11, No.2, May 1996.