TEMPORAL CLASSIFICATION METHOD FOR FORECASTING LOAD PATTERNS FROM AMR DATA *

Heon Gyu Lee[1], Jin-Ho Shin[2], Keun Ho Ryu[1†]
Database/Bioinformatics Lab., School of Electrical & Computer Engineering, Chungbuk National University, Korea[1]
{hglee, khryu}@dblab.chungbuk.ac.kr[1]
Power Information Technology Group, Korea Electric Power Research Institute, Korea[2]
jinho@kepri.re.kr[2]

ABSTRACT. We present in this paper a novel mid and long term power load prediction method using temporal pattern mining from AMR (Automatic Meter Reading) data. Since the power load patterns have time-varying characteristic and very different patterns according to the hour, time, day and week and so on, it gives rise to the uninformative results if only traditional data mining is used. Also, research on data mining for analyzing electric load patterns focused on cluster analysis and classification methods. However despite the usefulness of rules that include temporal dimension and the fact that the AMR data has temporal attribute, the above methods were limited in static pattern extraction and did not consider temporal attributes. Therefore, we propose a new classification method for predicting power load patterns. The main tasks include clustering method and temporal classification method. Cluster analysis is used to create load pattern classes and the representative load profiles for each class. Next, the classification method uses representative load profiles to build a classifier able to assign different load patterns to the existing classes. The proposed classification method is the Calendar-based temporal mining and it discovers electric load patterns in multiple time granularities. Lastly, we show that the proposed method used AMR data and discovered more interest patterns.

KEY WORDS: Load Patterns, Temporal Pattern Mining, Load forecasting, Calendar-based temporal mining.

1. INTRODUCTION

Electricity load patterns prediction (or forecasting) 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 load patterns and recognize new load forecasting methods, employing methods from sciences such as statistics [1] and data mining [2]. In power system, data mining is the most commonly used methods to recognize 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 [3], 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. Also, load demand data is temporal data which has timestamp. Previous researches such as clustering, classification and regression usually did not consider such time factor in temporal data or applied as static factor. Since the power load patterns have time-varying characteristic and very different patterns according to the hour, time, day and week and so on, it gives rise to the uninformative results if only traditional data mining is used. Therefore, if we consider time intervals under multiple time granularities in order to forecast load patterns to load demand data analysis with temporal dimension, we can discover useful load patterns during the given time interval. The purpose of this paper is to investigate the effectiveness and accuracy of classification method within load forecasting. To achieve this purpose, we attempt to apply clustering method and calendar-based temporal classification method and their use in load pattern forecasting. For the forecasting 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 representative load profiles for each class.
2. Temporal associative classification method uses representative load profiles to build a classifier able to assign different load patterns to the existing classes.
   ① Calendar pattern proposed in [4] is applied to AMR (Automatic Meter Reading) load data for the time expression of class association rules. This calendar pattern is

---

† Corresponding Author
* This research was supported by development of AMR system interfacing model on internet GIS environment project of the Korea Electric Power Research Institute (KEPRI)

based on user-specified calendar pattern and represents cyclic pattern.

② CARs (Class Association Rules) are discovered in given time. CARs are special subset of association rules with a consequent limited to class values only.

3. The generated temporal CARs are applied to build classifier for predicting load patterns in AMR load data.

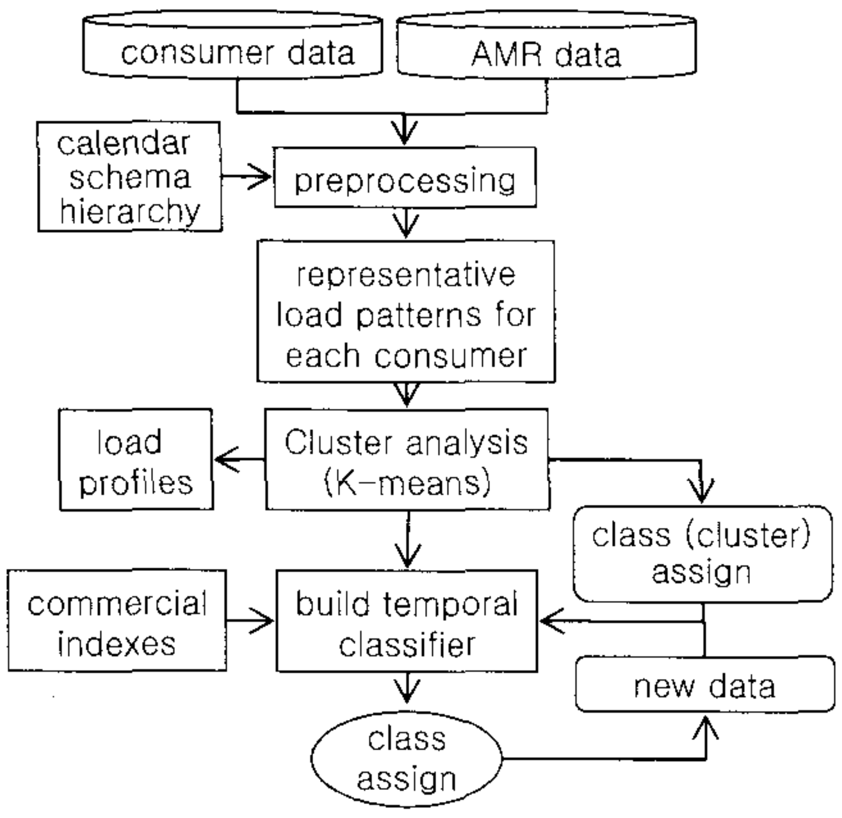

Figure 1. A framework of temporal classification method.

2. CLUSTER ANALYSIS

We describe clustering algorithms for generating the load profiles and class label which will be used classification module. The load pattern associated with any customer contains the information of commercial indexes such as contract assortment, industrial code and electricity use which recoded every 15 minutes. In order to perform clustering, we represent the load pattern for each consumer. The representative daily load pattern of the mth consumer is following:

$$V^{(m)} = \{V_{100}^{(m)}, \ldots, V_h^{(m)}, \ldots, V_H^{(m)}\} \qquad (1)$$

where $h=100, \ldots, H$ with $H=2345$, representing the 15 min. interval between the collected measurements.

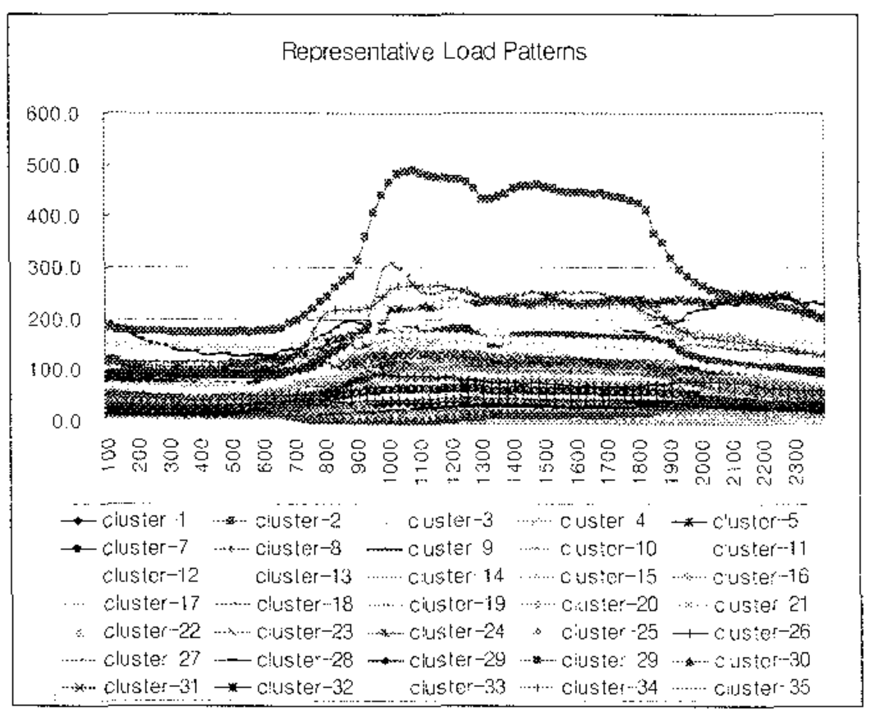

Figure 2. 35 representative load patterns.

In cluster analysis, K-means [6] 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. To define the number of classes, we performed the evaluation of the clusters compactness using the measure sum of the squared error (SSE). The obtained results are showed in Figure 2. It is possible to see that 35 clusters would be good choice, considering the SSN [6].

3. TEMPORAL CLASS ASSOCIATION RULES

In this section, we describe classification method and algorithms of Temporal CARs (Class Association Rules).

3.1 Calendar-based pattern

Let $I=\{a_1,a_2, \ldots,a_m\}$ be a set of items, and a transaction database $DB=<T_1,T_2,\ldots,T_t>$, where $T_i(i \in [1..t])$ is a transaction which contains a set of items in I. The support (or occurrence frequency) of a pattern A, which is a set of items, is the number of transactions containing A in DB. A is a frequent pattern if A's support is no less than a predefined minimum support threshold, ε.

We present a class of calendar related to temporal patterns called calendar patterns. Calendar pattern represents the sets of time intervals in terms of calendar schema specified by user. Calendar schema is a relational schema $R=(f_n:D_n, f_{n-1}:D_{n-1}, \ldots, f_1:D_1)$, where each attribute f_i is a time granularity name such as year, month, day, and so on. Each D_i is a domain value corresponding to f_i. There is given a calendar schema (year:{1,2}, month:{1,2}, week:{1,2,3}, day:{1,2,3,4}). And a calendar pattern on the calendar schema R is a tuple on R of the form $<d_n, d_{n-1},\ldots,d_1>$. Each d_i is a positive integer in D_i or the symbol "*". "*" denotes all the values corresponding to domain and means "every." Exactly, it represents periodic cycles on the calendar pattern such as every week, every month, and so on.

For presentation of calendar pattern, we call a calendar pattern with k symbols a *k-star* calendar pattern (denoted e_k) and a calendar pattern with at least one symbol a star calendar pattern. In addition, we call a calendar pattern with no symbol a basic time interval (denoted e_0) According to the above example 1, the calendar pattern <1,1,*,3> represents time intervals which means the third day of every week of January in the first year.

3.2 Class association rules

CARs (Class Association Rules) are a combination of association rules mining and classification. CARs are a special subset of association rules whose antecedent is an itemsets and consequent are restricted to the classification class label. Let $I=\{a_1,a_2, \ldots,a_m\}$ be a set of all items in DB and Y to be a set of all class labels. CARs r is an implication of the form:

$$X \Rightarrow C \qquad (2)$$

where $X \subset I$, $C \subset Y$. Antecedent of a CARs is also called itemset and a rule itself is called *rule_item*. *Rule_item* is large if the corresponding rule is frequent and accurate if the rule is confident. To find all strong association rules, CBA [7] uses an Apriori-like algorithm to generate large *rule_items*. And *k-rule_item* denotes a rule whose itemset has k items. In each pass, all large *(k-1)-rule_items* are found. These *(k-1)-rule_items* are used to generate candidate *k-rule_item* and selected candidates satisfy minimum support threshold, ε. For each large *rule_item*, the confidence of the corresponding rule is calculated and the rule is added to the set of all rules if the confidence satisfies minimum support threshold, δ.

Support and confidence of *rule_item* in a transaction *DB* is the following:

$$Support = \frac{ruleCount}{|DB|},\ Confidence = \frac{ruleCount}{itemsetCount} \quad (3)$$

where *ruleCount* is the number of items in *DB* that contain the itemset and are labeled with class label and *itemsetCount* is the number of items in *DB* that contain the itemset.

3.3 Temporal CARs algorithm

Given a basic time interval $t\ (e_0)$ under a given calendar schema, we denote the set of transactions whose timestamps are covered by $t\ (e_0)$ as $DB[t]$. Temporal CARs over a calendar schema R is a pair (r, e). Thus, temporal CARs (r, e) hold in DB if and only if the r satisfies ε and δ in $D[t]$ for each basic time interval e_0 covered by e. And CARs are an implication of the form:

$$<X => C, e> \quad (4)$$

X: itemset, C: class label, e: star calendar pattern

We extend apriori [8] to discover large *rule_items*. On the data mining tasks, our algorithm produces the rules $TCAR_k(e)$ that satisfies ε and δ for all possible star calendar pattern on R. The *Temporal CARs* algorithm is given in figure 3. The algorithm generates all the large *rule_items* by making multiple passes over the data.

```
Input transaction (Training data set) D
Output <TCAR_k(e), e> for all star calendar pattern e

forall basic time intervals e_0 do
        L_1(e_0) = {large 1-rule_items in D[e_0]};
        TCAR_1(e_0) = genRules(L_k(e_0))
end
forall (k=2; ∃ calendar pattern e such that L_{k-1}(e)≠Ø;k++) do
        forall basic time intervals e_0 do
                C_k(e_0) = candidateGen(L_{k-1}(e_0), min_sup);
                for each data case d∈D(e_0) do
                      C_d = subset (C_k(e_0), d);
                      for each candidate c∈C_d do
                            c.itemsetCount ++;
                            if d.class=c.class then c.ruleCount++;
                      end
                end
                L_k(e_0) = {c∈C_k | c.ruleCount ≥ min_sup};
                TCAR_k(e_0) = genRules(L_k(e_0), min_conf);
                forall calendar pattern e that cover e_0 do
                      update TCAR_k(e) using TCAR_k(e_0);
        end
end
```

Figure 3. Temporal CARs algorithm.

4. CONSTRUCTING CLASSIFIER

In this section, we describe to build the efficient classifier using Temporal CARs. Generated classifier for each calendar pattern is following format:

$$(r_1, r_2, \ldots, r_n, default\_class, e) \quad (5)$$

where, r_i is the generated Temporal CARs and e is the calendar pattern. However, the number of rules generated by algorithm can be huge. To make the efficient classifier, we need to prune rules generated. To reduce the number of rules generated, we perform two types of rule prune. First rule prune is the pruning from calendar patterns of Temporal CARs. After we discover all large *rule_items*, we remove all the (r, e) if we have other (r, e') and e is covered by e'. Second, we use general and high-confidence rule to prune more specific and lower confidence ones. Before the pruning, all rules are ranked according to the following criteria.

Given two rules r_i and r_j, $r_i > r_j$ (or r_i is ranked higher than r_j) if *1).* $conf(r_i) > conf(r_j)$ *or 2).* $conf(r_i) = conf(r_j)$, $sup(r_i) > sup(r_j)$ *or 3).* $conf(r_i) = conf(r_j)$ *and* $sup(r_i) = sup(r_j)$, *but* r_i *is generated before* r_j.

A rule r_1: $X=>C$ is said a general rule w.r.t. rule r_2: $X'=>C'$, if only if A is a subset of A'. First, we need to sort the set of generated rules for each calendar pattern. This sorting guarantees that only the highest rank rules will be selected into the classifier. And then, given two rules r_1 and r_2, where r_1 is a general rule w.r.t r_2. We prune r_2 if r_1 also has higher rank than r_2.

5. EXPERIMENTAL RESULT

In this section, a case study concerning a database with load patterns from 231 consumers is considered and this information has been collected by KEPRI (Korea Electric Power Research Institute). The collected load patterns were made during a period of three month (January, February and March) in 2007. The instant power consumption for each consumer was collected with a cadence of 15 min. The commercial indexes related with contract assortment power, industrial classification code, and electricity use code are also applied.

5.1 Data preprocessing

To preprocessing the 3 month's training data, we define calendar schema and domain by a hierarchy of calendar concepts.

$$R = (Month: \{1,\ldots,3\}\ week: \{1,\ldots,4\},\ \text{working day}: \{1,\ldots,5\},\ \text{weekend}: \{1,\ldots 2\})$$

The class labels are also assigned representative load patterns of cluster in training data. To compare the load patterns, we use feature of load shape [9], 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.

Table 1. Load curve shape features

Feature	Definition	Period
Load Factor	$s_1 = \dfrac{Pattern_{Avg.\ for\ day}}{Pattern_{Max.\ for\ day}}$	1 day
Night Impact	$s_2 = \dfrac{1}{3}\dfrac{Pattern_{Avg.\ for\ night}}{Pattern_{Avg.\ for\ day}}$	8 hours (11 pm ~ 7 am)
Lunch Impact	$s_3 = \dfrac{1}{8}\dfrac{Pattern_{Avg.\ for\ lunch}}{Pattern_{Avg.\ for\ day}}$	3 hours (12 am ~ 3 pm)

Since the extracted features contain continuous variables, those variables also must be made discrete. Therefore, entropy-based discretization [10] has been used because the intervals are selected according to the information they contribute target variable.

Figure 4 shows the preprocessing results from load pattern data of AMR.

Feature		Type	Description
Date		datetime	YYYYMMDD
Contract power		nominal	Different 29 value
Industrial code		nominal	Different 158 value
Electricity use code		nominal	Different 21 value
AMR Capacity (15min. Interval)	100	continuous	Min.: 0.3 ~ Max.: 490
	115	continuous	Min.: 0.3 ~ Max.: 490
	...	continuous	Min.: 0.3 ~ Max.: 490
	2345	continuous	Min.: 0.3 ~ Max.: 490
class	cluster	nominal	{cluster1, ... cluster 35}

Data preprocessing

Feature		Type	Description
Calendar pattern		nominal	Calendar expression
Contract power		nominal	Different 29 value
Industrial code		nominal	Different 158 value
Electricity use code		nominal	Different 21 value
AMR Capacity (15min. Interval)	S1	nominal.	Discrete value
	S2	nominal.	Discrete value
	S3	nominal.	Discrete value
class	cluster	nominal	{cluster1, ... cluster 35}

Figure 4. Result of preprocessing of AMR data.

5.2 Classifier evaluation

In our experiment, first, we build a classifier based on Temporal CARs from the preprocessed AMR training data. The accuracy was obtained by using the methodology of stratified 10-fold cross-validation (CV-10). One of the criteria for evaluating classifier is the accuracy of the classification results. We would like to be able access how well the classifier can classify. For this purpose, the sensitivity and specificity measures were used and accuracy is defined as:

$$accuracy = sens. \cdot \frac{Positive}{Positive + Negative} + spec. \frac{Negative}{Positive + Negative} \quad (6)$$

$$\left(sens. = \frac{True\_Positive}{Positive} \right), \left(spec. = \frac{Ture\_Negative}{Negative} \right), \quad (7)$$

We have two important thresholds for the classifier performance (min. support and min. confidence). These thresholds control the number of patterns selected for constructing classifier so we used these thresholds.

Figure 5 is the result of testing accuracy according to different minimum confidence and minimum support.

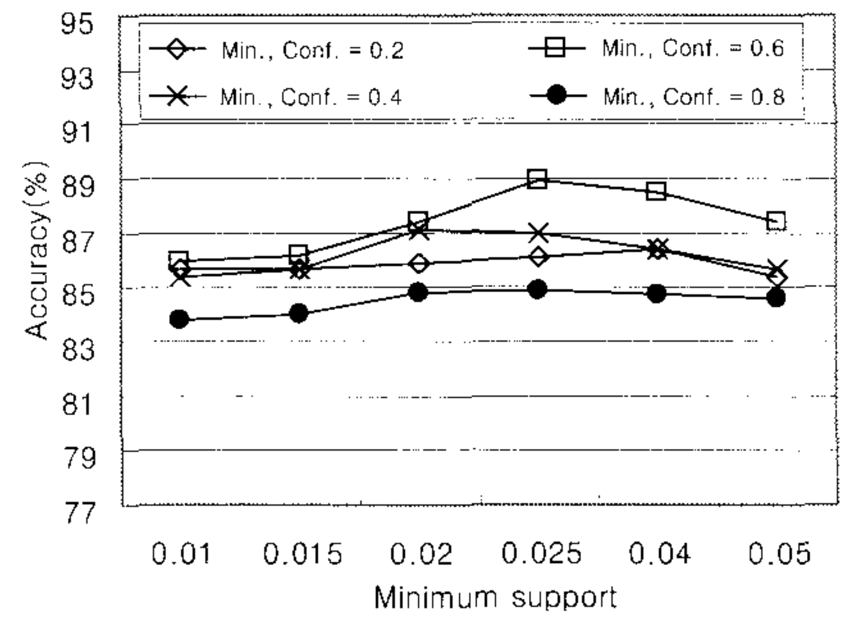

Figure 5. Effect of support and confidence on accuracy.

6. CONCLUSION

In this paper, we proposed a novel temporal pattern mining to predict mid and long term power load patterns.

The proposed main mining tasks include clustering method and temporal classification method. Cluster analysis is used to define load pattern classes and the representative load profiles for each class. Classification method uses representative load profiles to build a classifier able to assign different load patterns to the existing classes. The proposed classification method is the Calendar-based temporal mining and it discovers electric load patterns in multiple time granularities. In experiment, the applied K-means and temporal classifier tested KEPRI AMR data and discovered interest load patterns.

References

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

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

[3] Amjady N., 2001. Short-term hourly load forecasting using time-series modeling with peak load capability. IEEE Trans. Power System. Vol. 16. No. 3, pp. 498-505.

[4] Li Y., Ning P., 2001. Discovering Calendar-based Temporal Association Rules, In Proc. of the 8th Int'l Symposium on Temporal Representation and Reasoning.

[5] Jain A. K., Dubes R. C., 1988. Algorithms for Clustering Data. Englewood Cliffs, NJ: Prentice-Hall.

[6] Park H. K., Kim Y., Park J. H., Ryu K. H., 2007. Cluster Analysis for Region Electric Load Forecasting System. to be appeared in Int'l Symposium Remote Sensing 2007.

[7] Liu B., Ma Y., 1998, Integrating classification and association rule mining. In Proc. of the 4th Int'l Conf. Knowledge Discovery and Data Mining8.

[8] Agrawal R., Srikant R., 1994. Fast algorithms for mining association rules. In Proc. of the 20th VLDB Conference, pp 487-499.

[9] Ernoult M., 1982. Analysis and forecast of electrical energy demand. Revue General d'Electricite, No. 4.

[10] Fayyad U. M., Irani K. B., 1993. Multi-Interval discretization of continuous-valued attributes for classification learning. In Proc. of the 13th Int'l Joint Conf. on Artificial Intelligence, pp. 1022-1027.