Int. J. Advance Soft Compu. Appl, Vol. 6, No. 1, March 2014
ISSN 2074-8523; Copyright © SCRG Publication, 2014

Subspace Clustering and Temporal Mining for Wind Power Forecasting

Heon Gyu Lee, and Yong Ho Shin

Convergence Technology Research Lab., ETRI, South Korea
e-mail: hg_lee@etri.re.kr
School of Business, Yeungnam Univeristy, South Korea
e-mail: yhshin@ynu.ac.kr (corresponding author)

Abstract

Wind power energy has received the biggest attention among the new renewable energies. For achieving a stable power generation from wind energy, the accurate analysis and forecasting of wind power pattern is required. In this paper, we propose subspace clustering method for generating clusters of similar wind power patterns from data to be analyzed and the calendar–based temporal associative classification rule mining for reflecting temporal information of wind power on the classification/prediction model. The experiments show that the optimal cluster is constructed by applying PROCLUS algorithm and it has 88.6% accuracy of prediction under application of temporal associative classification rules.

Keywords: *Wind power patterns, subspace clustering, temporal associative classification rules, temporal pattern mining*

1 Introduction

Currently, the wind power energy is getting the spotlight in the new renewable energies. Thanks to the use of natural wind, the wind power is unlikely to be depleted and has the advantage of less environmental pollution. On the other hand, electricity production by wind power is quite erratic due to the irregularity of wind energy sources. Therefore, electricity by wind power may not be able to be supplied to meet the demand of power and output changes drastically in a short period of time. Such disadvantage of fluctuation and irregularity make it very difficult to predict the exact generation of power. Among the existing studies on prediction of wind power generation, Kusiak and Verma predicted the state of generator and generation in the short-term stages [1]. Saurabh and Hamidreza [2] divided the time zone into short-term and long-term and predicted the generation

using artificial neural net and time-series model. RBF (Radial Basis Function) [3] can be used for forecasting the short-term generation; however there is a shortcoming that the performance of the RBF is sensitive to the form of RBF basis functions and the underlying parameters. The irregular changing characteristics of power output by surge or sharp drop of generation is referred to as ramping [4] and it is measured with the power ramp rate PRR representing the generation output in a given time interval. PRR can be analyzed by multivariate time series based statistical methods [5]. However, PRR can only check swings of power generation but wind power forecasting is impossible.

In this paper, cluster analysis and temporal classification techniques are applied in order to predict an accurate and efficient wind power patterns in the short-term and long-term stage in respect.

Especially this study uses subspace clustering method for generating clusters of similar wind power patterns from data to be analyzed and applies the calendar–based temporal mining techniques for reflecting temporal information of wind power on the classification/prediction model. The framework proposed in this study for the prediction of wind power pattern is as follows.

- Data preprocessing: To perform cleansing for preprocessing the missing value and outliers included in the raw data acquired from wind turbine at first, then feature extraction, the adjustment of time granularity and discretization steps are followed.
- Subspace cluster analysis: to generate a similar group of wind power patterns from the preprocessed data
- Temporal associative classification: to build the prediction model considering temporal characteristics and wind power pattern features by extending the FP-growth method and the temporal pattern mining technique.

2 Data Preprocessing

2.1 Data Collection and Cleansing

For this study we collected one-year data measured at a wind turbine from April of 2010 to April of 2011. Actual measurement was performed at every 10 minute. The data attributes consist of the exact time, the wind speed and the generation at every collection timestamp. The data collected from the wind turbine were cleansed so as to remove the missing value and outliers negatively affecting the clustering and classification model.

All the outliers contained raw data are removed and missing values are assigned as substitute values provided by using PASW statistics 18 method [6]. Also, since

the 10-minute interval data are so detailed, the collected data is generalized to one hour-interval.

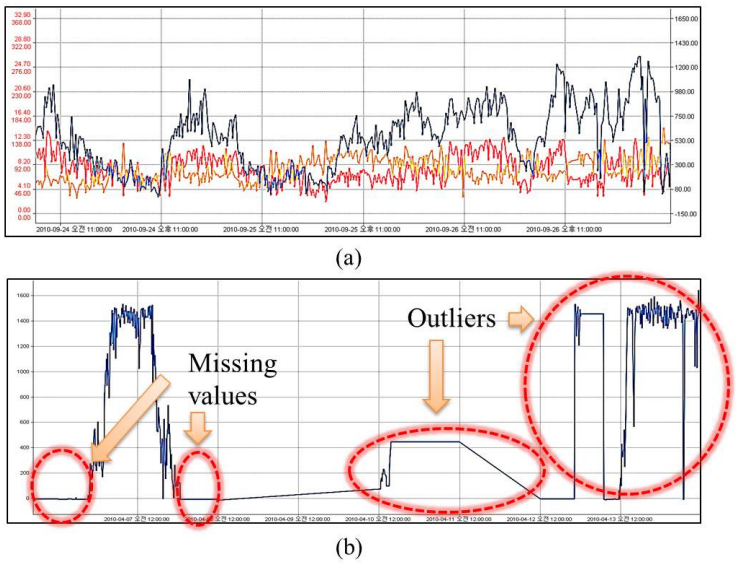

Fig. 1: Graphical representation of collected data;
(a) Normal data (b) Abnormal data.

2.1 Feature Extraction and Discretization

The collected attribute data from the wind turbine are not enough for the accurate forecasting of wind power pattern. So we need an additional feature extraction for predicting exact wind power pattern. At first, we extract PRR feature representing the rate of generation by using the irregular property of wind causing the surge or sharp drop of generation. Then, the value of minimum, maximum, average and standard deviation are computed with respect to wind speed and the quantity of generation. The entire features used for cluster analysis and classification are shown in Table 1.

Table 1: Extracted features from raw data

Feature	Description
ws_MEAN	Average value of wind speed
ws_MIN	Minimum value of wind speed
ws_MAX	Maximum value of wind speed
ws_SD	Standard deviation of wind speed
wp_MEAN	Average value of power generation
wp_MIN	Minimum value of power generation
wp_MAX	Maximum value of power generation
wp_SD	Standard deviation of power generation

PRR	$PRR = \frac{P(t+\Delta t) - P(t)}{\Delta t}$, ($P(t)$ is the current quantity of generation Δt is time difference)
timestamp	Timestamp of measurement <year/month/day/hour/minute/second>

All the extracted features are the continuous values. They do not matter for cluster analysis but should be transformed into nominal valued in order to be applied for the frequent pattern-based classifiers. In this step discretizing filter, a preprocessor of Weka's implementation [7], is applied.

3 Generating Wind Power Clusters Using Subspace Clustering Methods

Subspace clustering methods might report several clusters for the same object in different subspace projections, while subspace clustering methods are restricted to disjoint sets of objects in different subspace. These subspace projections also can be identified into three major approaches characterized by the underlying cluster definition and parameterization of the resulting clustering. First, cell-based subspace clustering discretizes the data space for efficient detection of dense grid cells in a bottom-up fashion. It was introduced in the CLIQUE [8] approach which exploits monotonicity on the density of grid cells for pruning. SCHISM [9] extends CLIQUE using a variable threshold adapted to the dimensionality of the subspace as well as efficient heuristics for pruning. In contrast, DOC [10] uses variable cells represented by hypercube. Second, density-based clustering defines clusters as dense areas separated by sparsely populated areas. In SUBCLU [11], a density monotonicity property is used to prune subspaces in a bottom-up fashion. A FIRE [12] extends this paradigm by using variable neighborhoods and an approximative heuristics for efficient computation. Lastly, cluster-oriented method optimizes the overall clustering result. This method defines properties of the entire set of clusters, like the number of clusters, their average dimensionality or more statistically oriented properties. PROCLUS [13] extends the k-medoids algorithm by iteratively refining a full-space k-medoid clustering in a top-down manner. DOC algorithm is used as the cell-based method, FIRES algorithm is used as density-based method and PROCLUS of cluster-oriented method is also used for cluster analysis of the given wind power data. In a cluster analysis, the determination of optimal number of groups heightening the similarity of inner cluster is very important. The conventional method is to evaluate reproducibility of clustered results or to use MIA(Mean Index Adequacy), an adequacy measure [14]. In the clustering methods, since that the attributes involved in the underlying cluster creation can only be applied, it is impossible to use these two methods

using all the data attributes. Therefore, clustering, a kind of unsupervised method, and classification, a supervised method are combined to determine the optimal cluster. The process steps of such evaluation method are shown in Fig. 2.

- First, we partition the training data into two parts. The ratio is 8:2. The larger data set is used as the training set and the smaller one is used as a test set.
- Second, run the three subspace clustering algorithms on training set to produce cluster label(A cluster label is as class label at the supervised learning step.).
- The labels created by three cluster methods are added into a training set.
- A classification model is created by applying the supervised learning method to the added training set. In this step, the decision tree(C4.5) induction is used considering its performance. The decision tree is made up of a set of nodes that classify the past realizations of the objective variable. Each classification is achieved by separation rules according to the numerical or categorical values of the explanatory variables. The classification rules of each node are derived from a mathematical process that minimizes the impurity of the resulting nodes, using the available learning set. The main advantage of the decision tree is the easy interpretability of the results and the supply of probability values without assuming normal distributions.
- Finally, the classifier having the highest accuracy is chosen among three created classifiers using the test set. At last, the subspace clustering algorithm generating the class label of this classifier is optimal clustering of wind power data.

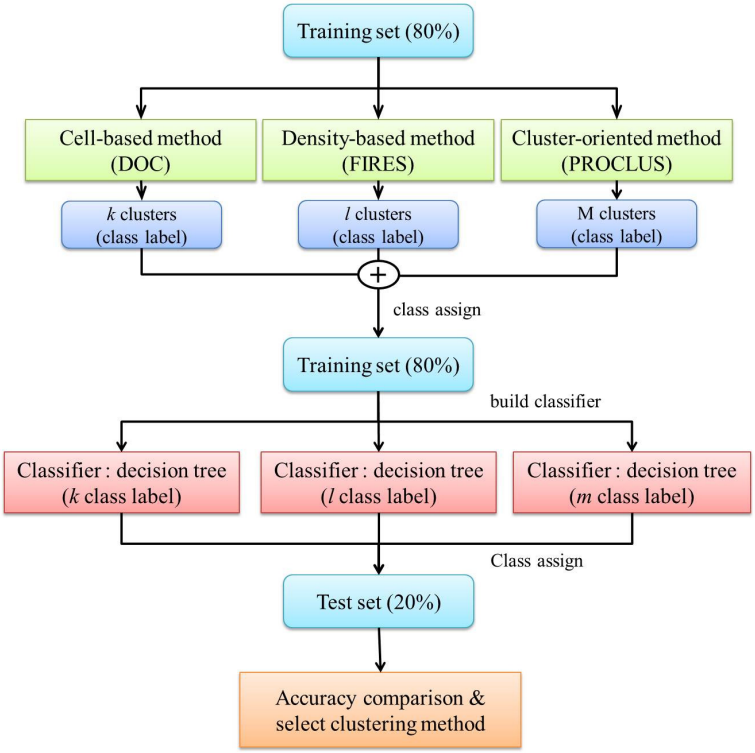

Fig. 2: Determination process for the optimal number of cluster.

Cell-based DOC algorithm proceeds in two steps: first step is to define what an optimal cluster is. Since clusters are discovered one at a time, the extent to which this definition models the "natural" clusters in the data determines the quality of the result. DOC developed a Monte Carlo algorithm for iteratively computing projective clusters. Second step is to design a fast and accurate method for computing one such optimal cluster, or a good approximation for it. DOC proposed a few simple heuristics to reduce the number of data scans and speed up the algorithm. Density-based FIRES is based on an efficient filter-refinement architecture that scales at most quadratic with regard to the data dimensionality and the dimensionality of the subspace clusters. It consists of three steps. First step is the pre-clustering; all 1-D clusters called base clusters are computed. Second is the generation of subspace cluster approximations. The base clusters are merged to find maximal dimensional subspace cluster approximations. The third step is post-processing of subspace clusters, refines the cluster approximations retrieved after second step. Cluster-oriented PROCLUS algorithm proceeds in three phases: an initialization phase, an iterative phase, and a cluster refinement phase. The general approach is to find the best set of medoids by a hill climbing process similar to the one used in CLARANS [15], Initialization phase reduces the set of points when doing the hill climbing, while at the same time trying to select representative points from each cluster in this set. The second phase is to

find a good set of medoids during hill climbing process. It also compute a set of dimensions corresponding to each medoid by using Manhattan segmental distances so that the points assigned to the medoid best form a cluster in the subspace determined by those dimensions. Finally, refinement phase uses one pass over the data in order to improve the quality of the clustering. The important parameter settings and performance evaluation results of the three kinds of subspace clustering algorithm are described in detail in section 6.

4 Generating Wind Power Patterns Using Temporal Mining

This section discusses the temporal mining technique to discover the representative patterns of wind power considering the temporal characteristics within each cluster group having similar wind power properties. Fig. 3 shows the classification/prediction procedure considering time periodicity starting from creation.

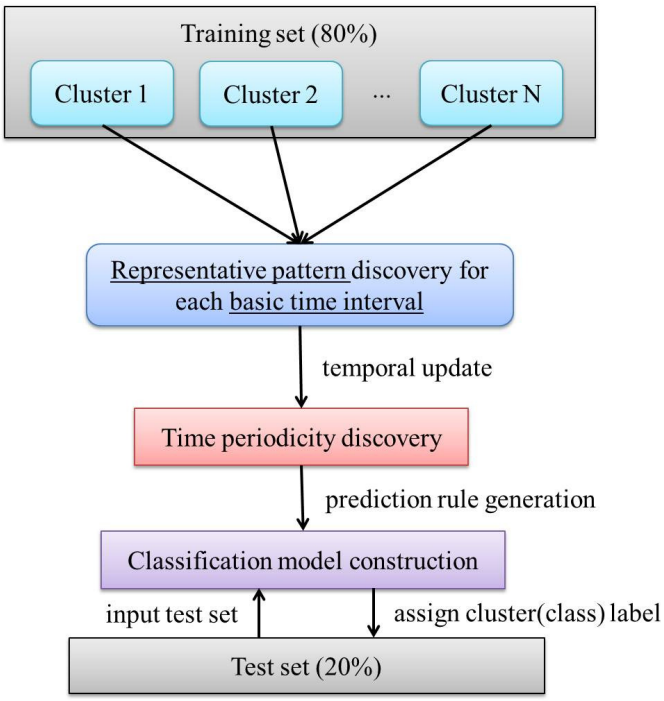

Fig. 3: Wind power pattern forecasting procedure

4.1 Calendar-based Temporal Mining

The calendar schema is a relational schema determined by calendar's concept

hierarchy [16, 17].

Definition 4.1 *A calendar schema (CS) is defined as a set of the calendar based time granularity and the possible domain of in such unit. The form is as follows.*

$$CS = (G_n : D_n, \cdots, G_1 : D_1) \qquad (1)$$

G_i is a time granularity in a calendar concept such as year, month, day for $1 \le i \le n$. Each D_i is domain value of G_i as a set of positive integer numbers. When a calendar schema is $(G_n, G_{n-1}, \cdots, G_1)$, $1 \le i \le n$, time granularity G_i is included uniquely into G_{i+1}. For example, a schema of (month, day) is valid because “day” is included in a specific “month”. In case of (year, month, week), "week" does not belong to a specific "month". In case $CS = (month : \{1 \sim 12\}, day : \{1 \sim 30\})$, {1, 20} is valid for expressing 20th day of January but {2, 31} is not a valid set.

Definition 4.2. *A calendar pattern (CP) is an instance of a given schema* $CS = (G_n : D_n, \cdots, G_1 : D_1)$ *and can be expressed as* $CP = \{d_n, \cdots d_1\}$. *Here each* d_i *is a domain value of* D_i *or symbol "※". If* d_i *is "※", it means all the value of domain* D_i *and can be interpreted as "every". Also it means time periodicity for current domain.*

For example, when a calendar schema is given as (month:{1~3}, day:{1~7}), calendar pattern {※, 3} expresses time interval {1, 3}, {2, 3}, {3, 3} and “Wednesday of every month”. Depending on the number "※" included in a calendar pattern, the expression is classified. A calendar pattern containing i "※"s is called as $i - star\ pattern(CP_i)$ and other patterns including no "※" is called as "basic time interval".

Table 2: Expression of calendar pattern

# of symbol	Calendar pattern	Expression
0	Basic time interval	CP_0
1	1-star pattern	CP_1
...	...	...
i	i-star pattern	CP_i

4.2 Generating Associative Classification Rules Using Wind Power Patterns

A representative wind power pattern for each cluster can be generated by the associative classification rule mining [18].
D is a transaction DB, $I = \{i_1, i_2, \cdots, i_n\}$ denotes all *itemset* and $C = \{c_1, c_2, \cdots, c_m\}$ is called as cluster (or class) label.

Definition 4.3. *CARs(Class Association Rules) is expressed as a form like* $X \to c_i$, *the antecedent of CARs is itemset, rule itself is named as ruleitem.*

Definition 4.4. σ, the support of *CARs* and confidence, δ;

$$\sigma = \frac{ruleitem\ count}{|D|} \tag{2}$$

$$\delta = \frac{ruleitem\ count}{itemset\ count} \tag{3}$$

CARs are a set of ruleitem satisfying minimum support and minimum confidence. Temporal mining is performed by adding calendar pattern by definition 3 and 4. If calendar pattern CP for a calendar schema CS is given, transaction of timestamp is expressed as $D(CP)$ included in a CP. Grammatically, the associative classification rules are expressed as <CAR, CP> form. For instance, in case CS={year:{1999~2001}, month:{1~12} and day:{1~31}, rule $<(A \wedge B) \to cluster_1, \{*,2,3\}>$ is showing a calendar and cyclic expression. This rule is valid in a set of basic time interval such as {1999, 2, 3}, {2000,2,3}, {2001,2,3}, means that the rule $<(A \wedge B) \to cluster_1>$ is established during from 1999 to 2001. The wind power data is a very huge data automatically obtained from sensors in a given time interval. So FP-tree structure should be included in temporal concept for generating an efficient associative classification rules. For design and construction of temporal FP-tree, the following are modifications to the temporal FP-tree that make it suitable for frequent pattern mining.

① Count value of every node is replaced in the FP-tree with the class distribution, where each element of the distribution stores a number of transactions of the class containing a pattern from this node to the root.
② All item count value in the frequent -item table of the temporal FP-tree is now replaced with item class distribution.
③ Each node in the item prefix sub-tree contains basic time interval pattern, CP_0.

For a given $CS = (G_n, \cdots G_1)$, all the ruleitems are created in order to satisfying the condition σ and δ in each basic time interval. Temporal FP-tree is a tree structure defined as the below:

① The tree has a root labeled as null, a set of item prefix sub-trees as the children of the root, and a frequent item table.
② Each node in the item prefix sub-tree contains three fields: item-name, class count for each class c_i, basic time interval and node-link.
③ The number of entries in frequent item table is equal to the number of distinct elements in the FP-tree and each entry contains three fields: item-name, class count for each class c_i and head of node-link.

For example, in case that transaction DB is given as Fig. 4 and minimum support threshold (σ) is 2, a descending sort is performed considering the class distribution for 1-item at the first DB scan (e(5), b(3), e(3)).
The first transaction contains item e, b and class c_3 that are timestamp, <1,1,1>. A class frequent pattern {c_3: e, b} insert into FP-tree by generating new nodes and branch for basic time interval <1,1,1>. The second transaction {c_3: a,b,e} has item a that is not class frequent with class c_3. Since the pattern {c_3: 5,2} has a common prefix with the prefix-path and same basic time interval in the tree no new nodes are inserted and only counts are updated. The third transaction {c_2: c,e,g} is sorted into {c_3: e,c,g}. Item g is not frequent for class c_2, and class frequent pattern {c_2: e,c} has a different basic time interval, so the pattern {c_2: e,c} is inserted by generating new nodes and branch. For the fourth transaction {c_2: c,e,f}, we insert a pattern {c_2: e,c} by just updating the counts. No item in the last transaction is class frequent.

ID	Items	Class	CP_0
1	{b,d,e}	c_3	<1,1,1>
2	{a,b,e}	c_3	<1,1,1>
3	{c,e,g}	c_2	<1,2,1>
4	{c,e,f}	c_2	<1,2,1>
5	{b,c,e}	c_1	<2,1,1>

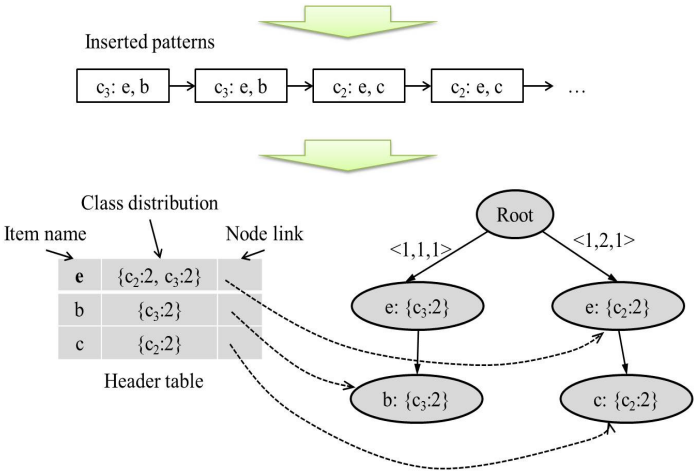

Fig. 4: An example of temporal FP-tree construction

The algorithm for the temporal FP-tree construction is shown as Fig. 5.

Input: (1) transaction *DB*, (2) minimum support, σ.
Output: TFP-tree corresponding to *DB* and satisfying σ.
1. Scan *DB* once and collect the set of *FP(frequent patterns)* and their class supports.
2. Sort *FP* in support descending order as *L*, the list of class frequent items.
3. Create the root *R* of *FP-tree* and label it as "null."
4. for each transaction $d \in DB$ do {
   c_d = class of transaction *DB*;
   Select only class frequent *x* into a record *P*;
   Sort *P* in the order of *L*.
   Call *insert_tree(P, c_d, R);*
}

Procedure *insert_tree(P, c, R)* {
1. *P=[p|P-p]*, where *p* is the first pattern of *P* and *P-p* is the remaining list.
2. if *R* has a child *N* such that *N.calendar-pattern* CP_0 = *p.calendar-pattern* CP_0 then
3. if *N.item-name=p* then *N.count(c)=N.count(c)+1*;
4. else {
5. create a new node *N* on the same branch
6. for all class c_i do
7. if $c_i=c$ then *N.count*(c_i)*=1* else *count*(c_i)*=0*;
8. *N.item-name=p; N.parent=R;*
9. *N.node-link=H(p).head; H(p):head=N;*
10. *}*
11. *H(p).count=H(p).count + 1;*
12. else create new Branch link from the root.
13. if *P-p*$\neq\Phi$ then
14. call *insert_tree(P-p, c, N)* recursively
}

Fig. 5: Temporal *FP*-tree construction algorithm

```
Input: (1) Temporal FP-tree, T for DB, (2) minimum class support σ and minimum confidence δ.
Output: A set of CARs for each basic time interval.
Method Call FP-growth(T, null), //null is the initial suffix.
Procedure FP-growth(T, μ) {
1. CAR = Φ;
2. if T contains single path P in one calendar pattern then
3.       for each combination λ of the nodes in P do {
4.          generate pattern p = λ ∪ μ;
5.             for each class c_i do {
6.               σ (X→c_i) among nodes in λ;
7.                  if σ (X→c_i) ≥ σ and δ (p→c_i) ≥ δ then
8.                       CAR = CAR ∪ (p→c_i);
9.               }
10.    }
11. else for each μ_i in the header of Tree do {
12.          generate pattern λ = μ_i ∪ μ ;
13.          for each class c_i do {
14.             if σ (λ→c_i) ≥ σ and δ (λ→c_i) ≥ δ then
15.                  CAR = CAR ∪ (λ→c_i);
16.          }
17.           construct λ's conditional pattern base;
18.           construct λ's conditional FP-tree T_λ;
19.           if T_λ≠Φ then call FP-growth(T_λ, λ);
20. }
 }
```

Fig. 6: Temporal FP-growth algorithm

After completing a temporal FP-tree construction, all the associative classification rules should be extracted through the FP-growth method. The algorithm for mining all associative classification rules using temporal FP-growth is described in Fig. 6. After all classification rules were discovered in basic time interval, rules is used to update i-star patterns CP_i. In this phase, we used updating algorithm was introduced in [16]. Fig. 8 shows a process to renew all the rules having basic time interval into i-star patterns.

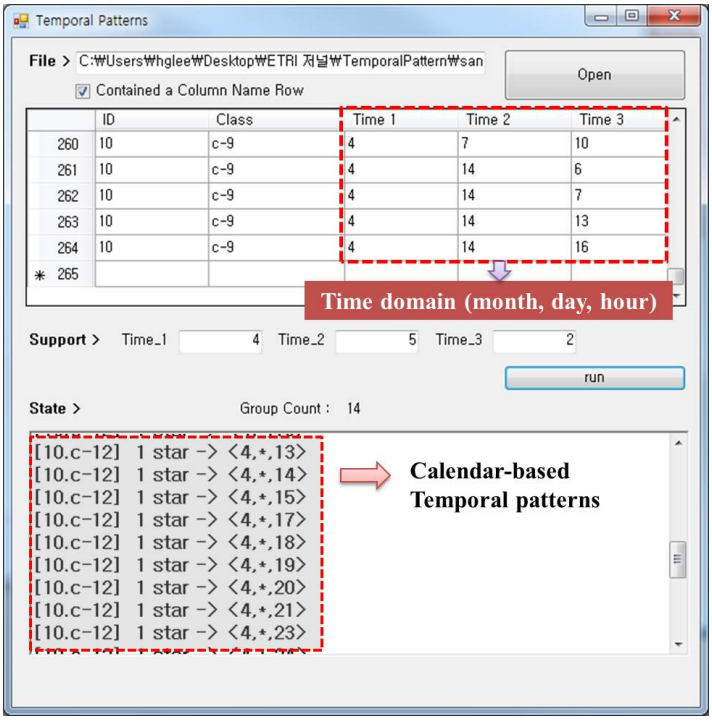

Fig. 7: Timer pattern extraction program for wind power data

For example, the temporal patterns shown in the lower part of Fig. 7, "[10, c-12] --> 1 star --><4,*,13>, <4,*,14>, <4,*,15>, <4,*,17>, <4,*,18>, <4,*,19>, <4,*,20>, <4,*,21>, <4,*,23> " is interpreted as "The wind pattern of ID=10 corresponds cluster=12 at 13, 14, 15, 17, 18, 19, 20, 21, 23 o'clock every day of April". Here if the wind power pattern of ID=10 shows that maximum wind speed is smaller than 17.8 m/s and average generation is between 20.336kW and 79.2083kW, then it means "A window power pattern which has the largest level of wind speed as 17.8 m/s and the average generation is larger than 20.336kW and smaller than 79.2083kW at 13, 14, 15, 17, 18, 19, 20, 21, 23 o'clock every day of April carries a representative pattern of cluster 12". After extracting all the associative classification rules and temporal patterns, the rules for classification/prediction of new data are generated. Generated classification rules for each calendar pattern is following format:

$$(rule_1, rule_2, \ldots, rule_n, CP_i) \qquad (4)$$

where, $rule_i$ is the generated associative rules and CP_i is the calendar-based temporal pattern. However, the number of rules generated by temporal FP-growth method can be huge and associative classification rules include redundant rules. To remove the redundant rules and make the efficient classifier, we use the rule

ranking method. This method is used for rule ranking and redundant pattern pruning. Rule ranking is needed to select the best pattern in case of overlapping rules. Redundant rules are defined following definition.

Definition 4.5. *Redundant rule*: A rule *r* is redundant if and only if any of its subrule has higher rank than *r*. Redundant rule will should be deleted.

Definition 4.6. *Rule ranking, given two rules r_i and r_j, $r_i > r_j$ (or r_i is ranked higher than r_j) if*
(1) confidence, $\delta(r_i) >$ confidence, $\delta(r_j)$ or
(2) $\delta(r_i) = \delta(r_j)$ but support, $\sigma(r_i) >$ support, $\sigma(r_j)$.

If itemset X –s a subset of X', rule r_1: {X→c} is called as general rule for r_2: {X'→c}. For two different rule r_1 and r_2, in case that both rules are genera rule and r_1 has higher rank than r_2, rule r_2 is so redundant rule as to be remove from the associative classification rules. After a set of associative classification rules is selected for classification model, we classify new data set. First, we select the rules whose temporal patterns can cover basic time intervals of cases. Then, we find the associative classification rules matching case in the selected one and classify class labels of the found rules. If all the rules matching the new data have same class label, the new data is assigned to that label. Otherwise, we classify the new data as class label of the rule with higher ranking.

5 Experiment Results

5.1 Subspace Cluster Analysis

For unsupervised techniques like clustering, it is difficult to provide appropriate parameter setting without prior knowledge about the data is available. To speed up parameter setting process for users and give them more information to base their parameter choice on, the system supports parameter bracketing for direct feedback. This means that users can pick the most appropriate one(s) from subsequent runs of the subspace clustering algorithms according to given parameter settings' combination. By directly comparing the results of different parameter settings, parameterization is no longer a guess, but becomes an informed decision based on the visual analysis of the all mined subspace clusters by overview browsing and object ranking. Parameter bracketing is used to pick the most appropriate one for subsequent runs of the subspace clustering algorithms by directly comparing the results of different parameter settings. Table 3, 4, 5 shows the parameter bracketing for algorithms respectively.

Table 3: Parameter bracketing for DOC

Parameter	From	Offset	Op	Step	To
Alpha	0.001	10	*	3	0.1
Beta	0.1	0.1	+	4	0.4
Maxtter	1024	0	+	1	1024
k	9	1	+	5	13
w	5	1	+	4	8
Total number of experiments : 288					

The description and setting for the parameters for *DOC* algorithm in Table 1 is as follows.

- Alpha: it can be viewed as a minimum required cluster density, if Alpha is too small, we may execute too many outer iterations. On the other hand, if Alpha is too large, we can miss an input cluster entirely. The given range is 0.001~0.1.
- Beta: represents the user's "opinion" on the relative importance of points versus number of dimensions in a cluster. When Beta is small, the algorithm chooses more dimensions for a cluster at the cost of throwing away cluster points. At the other end, when Beta is large, the algorithm merges input clusters. The given range is 0.1~0.4.
- Maxtter: upper-bound the number of inner iterations. The given value is 1024.
- k: # of clusters. The given range is 9~15.
- w: interval of length, define a hyper-cube of given side length. The given range is 5~8.

Table 4: Parameter bracketing for *FIRES*

Parameter	From	Offset	Op	Step	To
Graph_K	1	3	+	4	10
Graph_MU	1	3	+	4	10
Graph_MINCLU	1	1	+	4	4
Base_DBSCAN_ EPSILON	0.4	0	+	1	0.4
Base_DBSCAN_ MINPTS	6	0	+	5	6
Graph_SPLIT	0.66	0	+	1	0.66
Post_DBSCAN_ EPSILON	2	0	+	1	2
Post_DBSCAN_ MINPTS	6	0	+	1	6
Post_	25	0	+	1	25

Minimumpercent					
Total number of experiments : 320					

FIRES algorithm has main parameters (*Graph_K, Graph_MU and Graph_MINCLU*) and other parameters of minor interest.

- *Graph_K*: it is used to find k-most-similar clusters from 1D base clusters. The given range is 1~10.
- *Graph_MU*: it is used to merge those base clusters which are all mostly similar to each other. The given range is 1~10.
- *Graph_MINCLU*: it is used to generate best merge clusters. The given range is 1~4.

Parameter settings of minor interest are listed as bellow (it is following the optimized setting of parameters in OpenSubspace toolkit for *FIRES*):

- *Base_DBSCAN_EPSILON*: the given value is 0.4.
- *Base_DBSCAN_MINPTS*: the given value is 6.
- *Graph_SPLIT*: the given value is 0.66.
- *Post_DBSCAN_EPSILON*: the given value is 2.
- *Post_DBSCAN_MINPTS*: the given value is 6.
- *Pre_Minimumpercent*: the given value is 25.

Table 5: Parameter bracketing for *PROCLUS*

Parameter	From	Offset	Op	Step	To
Average dimensions	2	1	+	8	9
Number of clusters	8	1	+	9	15
Total number of experiments : 72					

Parameter description and setting of *PROCLUS:*

- Number of clusters: the given range is 8~15.
- Average dimensions: The given range is 2~9.

Clustering is performed after parameter setting of three subspace clustering algorithms, the generated cluster label at that time is added into the training set to evaluate the algorithms including *DOC, FIRES, PROCLUS.* The generated cluster labels are regarded as class label for the purpose of testing the supervised learning. *C.45*, a kind of decision tree induction is used for testing the accuracy. After first decision tree model is constructed as training set, and then we can evaluate how accurate the algorithm can classify by applying the test set on the underlying model. The performance indices of the classified results are the rate of accurate classification, the rate of the inaccurate classification and *RMSE*(Root Mean Squared Error) shown in Table 6. Moreover, the last column of Table 6 is the number of clusters generated by performing a specific clustering algorithm. Based

the replicated experiment and the test result of subspace clustering algorithms, *PROCLUS* is adequate for accurate cluster constructing and the optimal number of cluster is 15.

Table 6: Result s of applying decision tree for each clustering algorithm

Cluster	Correctly classified objects	Incorrectly classified objects	Root mean squared error	# of cluster
DOC	39.33%	60.67%	0.38	5
FIRES	72,67%	27.33%	0.22	10
PROCLUS	**88.64%**	**11.36%**	**0.11**	15

5.2 Forecasting Wind Power Pattern Using Temporal Mining

We perform experiments to compare temporal classifier with state-of-the-art classifiers: CMAR(Classification Based on Multiple Class Association Rules), Bayesian classifier(TAN), and the widely known SVM(Support Vector Machine). We use WEKA's implementation of TAN, and SVM and LUCS-KDD software [19] of CMAR. The parameters of the four classifiers were set as follows.

- For the temporal associative classifier (temporal FP-growth), the minimum support threshold=1.2%; the minimum confidence threshold=80%.
- For the CMAR, the minimum support was set to 1.2%, the minimum confidence=80% and the database coverage was set to 3.75 (critical threshold for a 5% significance level, assuming degree of freedom equivalent to 1).
- For the SVM, the soft margin allowed errors during training. We used RBF kernel function [20].
- The parameters of Bayesian were default values.

In our experiment, we build the four classifiers from the preprocessed wind power training data. To evaluate classification performance w. r. t. the number of instances and class labels, we used an accuracy, RMSE and weighted average of TPR(True Positive Rate)/FPR (False Positive Rate). The results of classifiers comparison are shown in Table 7.

Table 7: A description of summary results

Classifier	Accuracy	RMSE	TPR	FPR
Temporal associative classifier	**88.6%**	**0.108**	**0.886**	**0.047**

CMAR	**88.3%**	**0.109**	**0.883**	**0.040**
SVM	75.76%	0.153	0.758	0.076
Bayesian classifier	72.34%	0.165	0.723	0.051

Fig. 9 shows the predicted result of each classifier with respect to the given representative patterns of cluster.

With respect to performance comparison, the algorithm considering temporal characteristics is expected to show more accurate result. According to the results shown in Table 7 and Fig. 9, our temporal associative classifier and CMAR perform very well. They archive higher accuracy than Bayesian classifier(TAN) and SVM

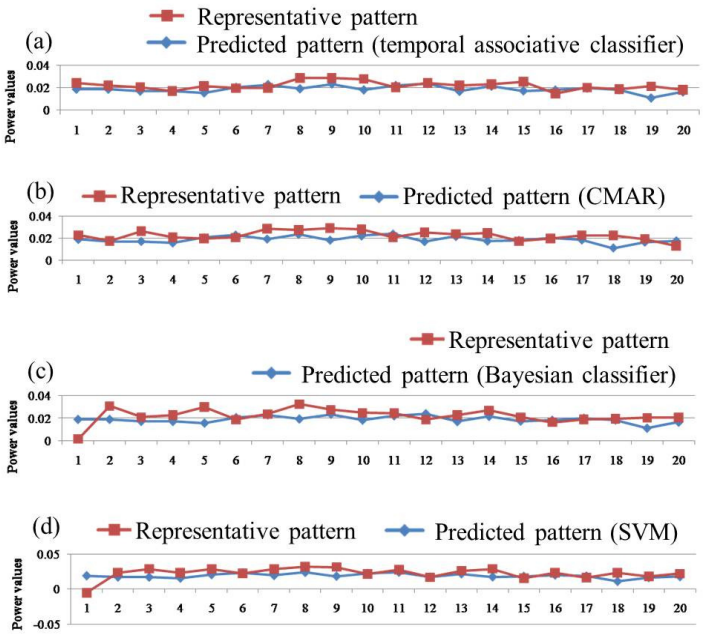

Fig. 8. Comparisons for forecasting results of wind power patterns.

6 Conclusion

This paper suggests an associative classification approach extended from the existent FP-growth algorithm, subspace clustering method and temporal mining for accurate prediction of wind power pattern based on data measured at a wind turbine. The experimental results show that such approach is similar to CMAR method and the prediction is more accurate wind power pattern than a Bayesian classifier and SVM. This paper has a limit not to extract sufficient temporal patterns from turbine data having time information. That is why the data used for experiments is just one year date, therefore no periodicity is found in terms of

yearly and monthly scale. If the wind turbine data is sufficient later, temporal patterns are likely to have many impacts on the prediction result of wind power pattern.

ACKNOWLEDGEMENTS.
This work was supported by the Postal Technology R&D program of MKE/IITA. [2006-X-001-02, Development of Implementation Technology for SMART Post]

References

[1] Kusiak, A. and Verma, A. 2011. Prediction of Status Patterns of Wind Turbines: A Data-Mining Approach, *Journal of Solar Energy Engineering-Transactions of the ASME*, Vol.133, No.2, 1-10.
[2] Saurabh, S. and Hamidreza, Z. 2010. A Review of Wind Power and Wind Speed Forecasting Methods with Different Time Horizons, *North American Power Sym.*, 1-8.
[3] Rutger, W. and Leon, J. M. 2006. Short-term Wind Power Prediction with Radial Basis Network, *Neural Network World,* Vol.16, No.2, 151-161.
[4] Ferreira, C. Gama, J. Matias, L. and Botterud, A. 2010. A Survey on Wind Power Ramp Forecasting, *Argonne National Laboratory*, 1-6.
[5] Zheng, H. and Kusiak, A. 2009. Prediction of Wind Farm Power Ramp Rates: A Data-Mining Approach, *Journal of Solar Energy Eng.,* 1-8.
[6] *SPSS Korea Data_solution Inc*, 2011.
[7] Witten, IH, and Frank, E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques 2nd Edition*, San Mateo, CA: Morgan Kaufmann.
[8] Agrawal, R. Gehrke, J. Gunopulos, D and Raghavan, P. 1998. Automatic subspace clustering of high dimensional data for data mining applications, *SIGMOD,* Vol.27, No.2, 94-105.
[9] Sequeira K. and Zaki, M. 2004. SCHISM: A new approach for interesting subspace mining, *In Proc. IEEE ICDM*, 186-193.
[10] Procopiuc, M. Michael, J. Agarwal, K. and Murali, T.M. 2002. A Monte Carlo algorithm for fast projective clustering, *In Proc. ACM ICMD*, 418-427.
[11] Kailing, K. Kriegel, H.P. and Kroger, P. 2004. Density-connected subspace clustering for high-dimensional data, *In Proc. SIAM*, 246-257.
[12] Kriegel, H.P. Kroger, P. Renz M. and Wurst, S. 2005. A generic framework for efficient subspace clustering of high-dimensional data, *In Proc. ICDM*, 250-257.
[13] Aggarwal, C. Wolf, J. Yu, P. Procopiuc, C. and Park, J. 1999. Fast algorithms for projected clustering, *In Proc. ACM SIGMOD*, 61-72.
[14] Chicco, G. Napoli, R. Postulache, P. Scutariu, M. and Toader, C. 2004. Load Pattern-Based Classification of Electricity Customers, *IEEE Trans. Power System*, Vol.19, 1232-1239.
[15] Ng, R. and Han, J. 1994. Efficient and Effective Clustering Methods for

Spatial Data Mining, *In Proc. VLDB*, 144-155.

[16] Lee, H.G. Shin, J.H. and Ryu, K.H. Application of Calendar-Based Temporal Classification to Forecast Customer Load Patterns from Load Demand Data, 2003, *In Proc. IEEE CIT*. 149-154.

[17] Li, Y. and Ning, P. 2011. Discovering Calendar-based Temporal Association Rules, *In Proc. Int'l Symposium on Temporal Representation and Reasoning*, 111-118.

[18] Hoque, A.H.M.S, and Mondal, S.K. 2011. Implication of association rules employing FP-growth algorithm for knowledge discovery, *In Proc. ICCIT*, 514-519.

[19] Coenen, F. Leng, P. and Goulbourne, G. 2004. Tree Structures for Mining Association Rules, *Data Mining and Knowledge Discovery*, Vol.15, No.7, 391-398.

[20] C. Lim, and J-H. Chang. 2011. Adaptive Kernel Function of SVM for Improving Speech/Music Classification of 3GPP2 SMV, *ETRI Journal*, Vol 33, No.6, 871-879.