Subspace Projection–Based Clustering and Temporal ACRs Mining on MapReduce for Direct Marketing Service

Heon Gyu Lee, Yong Hoon Choi, Hoon Jung, and Yong Ho Shin

A reliable analysis of consumer preference from a large amount of purchase data acquired in real time and an accurate customer characterization technique are essential for successful direct marketing campaigns. In this study, an optimal segmentation of post office customers in Korea is performed using a subspace projection–based clustering method to generate an accurate customer characterization from a high-dimensional census dataset. Moreover, a traditional temporal mining method is extended to an algorithm using the MapReduce framework for a consumer preference analysis. The experimental results show that it is possible to use parallel mining through a MapReduce-based algorithm and that the execution time of the algorithm is faster than that of a traditional method.

Keywords: Direct marketing, customer characterization, subspace projection, temporal associative classification, temporal mining, MapReduce framework.

I. Introduction

Direct marketing is the process of identifying potential customers of products and promoting the products accordingly [1]. Marketing campaigns seek to elicit actions, such as an order, a visit to a store, or a request for further information, from selected groups of consumers in response to communication from the marketer. Retailer ads, catalog distribution, postcards, flyers, promotional letters, and real samples or publications are all included in advertising techniques. Direct marketing is a new service providing the target delivery area and address information requested by a business looking to advertise using the postal delivery address customer relationship management (CRM) and data mining techniques. The most important factor in the success of direct marketing is securing customer lists [2]. Therefore, efficient data analysis techniques are needed in direct marketing for the selection of an appropriate customer list that includes temporal and spatial information. According to research by the US Postal Service (USPS), 80% of US residents receive direct mail, which they spend twenty-five minutes per day reading [3]. The USPS has been studying a way to simultaneously increase the response rate and decrease their expenses with direct mail services [2].

An accurate customer segmentation and reliable analysis regarding the bulk data on customer preference accumulated in real time are required for an effective direct marketing campaign. From the perspective of customer characterization, a detailed cluster analysis with various customer characteristics is required [4]. In addition, a market basket analysis [4]–[6] with a large amount of real-time product purchasing information

Manuscript received Aug. 4, 2014; revised Sept. 15, 2014; accepted Oct. 24, 2014.

This work was supported by the Postal Technology R&D program of MSIP (10039146, Development of Implementation Technology for SMART Post).

Heon Gyu Lee (hg_lee@etri.re.kr), Yong Hoon Choi (ychoi@etri.re.kr), and Hoon Jung (hoonjung@etri.re.kr) are with the IT Convergence Technology Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Yong Ho Shin (corresponding author, yhshin@ynu.ac.kr) is with the School of Business, Yeungnam Univeristy, Kyeongsan, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.15.2314.0068

and a temporal pattern analysis [7]–[13] for detecting preference changes over time are also important from the perspective of customer preference. A direct marketing campaign can be improved when customer groups based on similar lifestyles are formed and characterized. Thus, customer segmentation is performed using demographical, geographical, and economical information. Clusters detected using a clustering method are used to generate a profile based on the type of lifestyle definition. Business enterprises can identify customer groups using such profiling information and target their marketing services based on the different customer lifestyles available. However, census data (such as demographical, geographical, and economic information) used for detailed customer segmentation are high-dimensional data with around 100 different attributes. Therefore, traditional clustering methods, such as *k*-means and self-organizing map (SOM), are inefficient because they consider complete sets of attributes to detect clusters. Traditional methods such as *k*-means and SOM are based on the Euclidean distance measure and focus only on the global property by considering all dimensions in a dataset. As the number of dimensions in a dataset increases, distance measures become increasingly meaningless. Thus, a clustering algorithm using a subspace projection method is necessary for high-dimensional data. Subspace clustering methods seek to find clusters existing in subspaces or projections of a given high-dimensional data space, where a subspace is defined using a subset of attributes of a full space [14]. The mining of association rules [7] or sequential patterns [8] is mainly applied to analyze customer purchase patterns. However, for rating the customer preference of product changes over time, temporal characteristics are to be considered when a market basket analysis is applied. Data mining techniques using temporal characteristics are divided into periodic (or cyclic) pattern [9]–[11] and temporal relation [12] analyses. A periodic analysis for time-interval data discovers the cyclic pattern for repeated events during a time granularity. A temporal relation analysis uses temporal operators to show events based on a temporal expression. Previous works on temporal pattern mining for discovering useful patterns from temporal data have been conducted, but they do not consider large-scale temporal data. Among the above methods, a periodic pattern analysis is appropriate for a customer's product preference rating. Moreover, with the fast growth of online shopping, a large amount of historical purchase data is accumulated in real time. Thus, a distributed processing system to handle bulk data is needed since there is a limitation in executing a market basket analysis algorithm on a single machine. Recently, Google has used a distributed file system to manage bulk online data efficiently and provides a MapReduce framework [15]–[17] to support an efficient distributed programming environment.

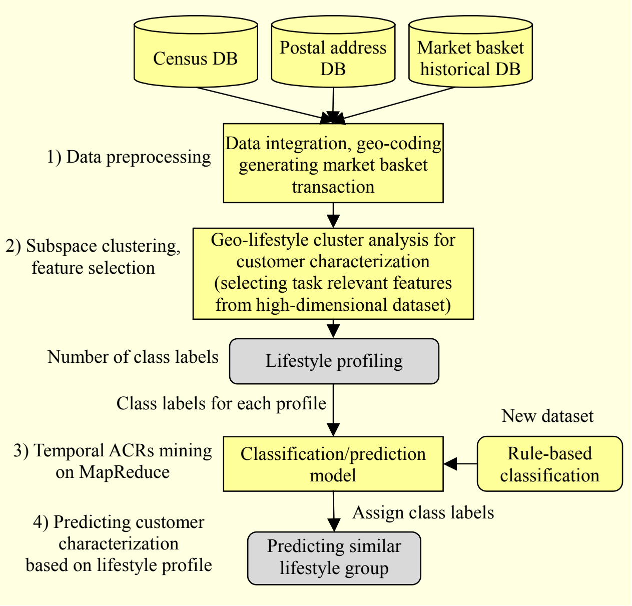

Census DB
Postal address DB
Market basket historical DB
1) Data preprocessing
Data integration, geo-coding generating market basket transaction
2) Subspace clustering, feature selection
Geo-lifestyle cluster analysis for customer characterization (selecting task relevant features from high-dimensional dataset)
Number of class labels
Lifestyle profiling
Class labels for each profile
New dataset
3) Temporal ACRs mining on MapReduce
Classification/prediction model
Rule-based classification
4) Predicting customer characterization based on lifestyle profile
Assign class labels
Predicting similar lifestyle group

Fig. 1. Cluster analysis and Temporal Associative Classification Rules (TACR) mining framework for direct marketing service.

A MapReduce program is composed of a map procedure that conducts filtering and sorting, along with a reduce procedure that conducts a summary operation. MapReduce can conduct such operations simultaneously with multiple machines and leads to faster operations for tasks with high cost. An algorithm using the MapReduce framework is proposed herein to conduct a market basket analysis using temporal characteristics. Temporal mining using the MapRedue framework is a challenging topic because the problem of mining temporal patterns from large-scale time interval data is very complicated.

There are two main purposes in this paper. First, optimal cluster groups for the customers of Korea Post, the Korean postal service, are formed with the necessary subspace from a high-dimensional census dataset, and accurate customer characterization information of each group is given to the postal operators and business marketers through profiling. Second, temporal pattern and purchasing pattern mining techniques are suggested using the MapReduce framework to extract the product preference information quickly and accurately. Figure 1 illustrates the data mining framework for a direct marketing service, and the methodology proposed in this study for providing marketing information is as follows:

- Data integration and preprocessing — the census information, postal address, and historical purchase information are combined with the spatial information. These integrated datasets are preprocessed at the block level (MicroArea) for privacy protection.
- Subspace cluster analysis — groups having a similar

Table 1. Description of census dataset.

Statistics items		Description of detailed attributes
Level 1	Level 2	
General statistical info.	Population	Total population, average age, population density, ageing index, old-age dependency ratio, childhood dependency, total dependency ratio
	Household	Total household, homeownership rate, homerentalship rate, home (and so on), average household size
	House	Total houses
	Facility	Apartment, single house, residence (and so on), total businesses
Population	Age	Under 4, between 5–9, between 10–14, between 15–19, between 20–24, between 25–29, … , over 65
	Education level	Elementary, middle, high, college, and graduate school
	Sex/marital status	Single_man, married_man, bereavement_man, divorced_man, single_woman, married_woman, bereavement_woman, divorced_woman
	Religion	Religion-has, Buddhist, Christian, Catholic, Confucianism, Won Buddhism, Jeungsangyo, Chondoist, Daejonggyo, and so on
Household	Family composition	1 generation, 2 generations (parents-child), 3 generations (grandparents-parents-child), 4 generations, 1 person household, unrelated household
	Houseownership	Homeownership, home-rental-ship (and so on), free
	Heating equipment	Central heating, district heating, city gas boiler, oil-fired boiler, propane gas boiler, electric boiler, and so on
	House size	1 bedroom, 2 bedrooms, 3 bedrooms, 4 bedrooms, 5 bedrooms, studio, 1 living room, 2 living rooms, no kitchen, 1 kitchen, 2 kitchens
House	Construction year	# of houses by construction year
	Floor space	Under 7, between 7–9, between 10–14, between 15–19, between 20–29, between 30–39, between 40–49, between 50–69, over 70
	House type	Multiplex-house, single-house, apartment, town-house, residential & commercial complex, and so on
Economic indices	Income	Under $20,000, between $20,000–$40,000, over $40,000

lifestyle are formed from spatial and census information, and then, based on the group, profiling is performed for characterization.

- TACRs on MapReduce — profiled clusters are used as a class label for prediction, and purchased products are transformed into a transaction database to discover valid TACRs in the given time domain. In this step, the exploration of associative classification rules (ACRs) and temporal patterns uses the MapReduce framework to enable distributed programming.

II. Customer Characterization Based on Subspace Projection Method

1. Data Integration and Description

Census data and postal addresses are integrated for the characterization of customer grouping through a geo-lifestyle cluster analysis and profiling. Census data are provided in the form of an open application programming interface with geographic information system (GIS) information by the National Statistics Korea, and postal address information is given by Korea Post. Census information provides demographic, geographical, and economic information, along with their subconcept information, hierarchically. Moreover, the average income information of the target customers is given from credit card companies and is added to the census information. All the attribute values from the census information are continuous statistical data. The attributes of the census information for data integration is shown in Table 1. Census information is provided at the MicroArea[1)] level and contains about 60 households (for privacy); in addition, maps are given in the form of a polygon — that is, a spatial object of the MicroArea. A postal address in text form is transformed into a spatial form by setting the standard GIS coordinates. A spatial join, one of the GIS operations, is used to integrate the census information from different units (points and polygons). By joining MicroArea (expressed as area) and addresses (defined as point objects), a spatial join operator calculates their inclusion relation. Through

[1)] MicroArea: statistics-collecting unit, a spatial concept while protecting resident's personal information and 1/24 of the small town such as eup, myeon, and dong (local district level) in size.

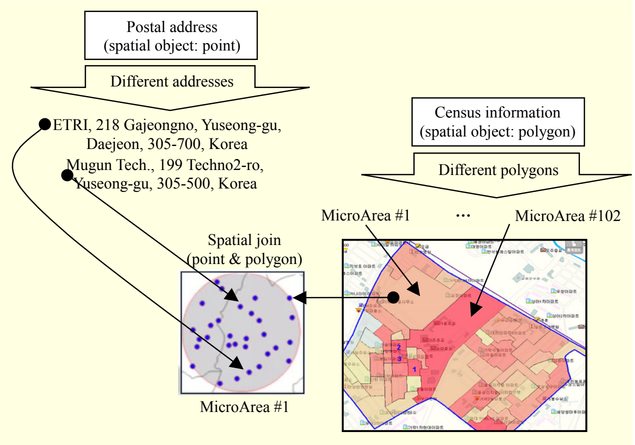


Fig. 2. Example of data integration using a spatial join operation (polygons present a MicroArea, and points are postal addresses).

this operation, customers' marketing information corresponding to their addresses can be extracted by block unit while protecting personal information. The process of data integration is illustrated in Fig. 2.

2. Projected Clustering (PROCLUS) Method

PROCLUS [18] is a *k*-medoid-type method that first generates *k*-cluster centers for a high-dimensional dataset using a dataset sample. It then refines the subspace clusters iteratively. In each iteration, for each of the current *k*-medoids, PROCLUS considers the local neighborhood of the medoid in the full dataset and discovers a subspace for the cluster by minimizing the standard deviation of the distances of the points in the neighborhood to the medoid on each dimension. Once all of the subspaces for the medoids are determined, each point in the dataset is assigned to the closest medoid according to the corresponding subspace. Clusters and outliers are discovered. In the next iteration, new medoids replace existing ones if doing so improves the clustering quality [19]. The PROCLUS algorithm is shown in Fig. 3. PROCLUS, a subspace projection method, uses the OpenSubspace of the Java WEKA [20] data mining tool. It supports up-to-date performance measures to facilitate study on subspace clustering. For an unsupervised method such as a cluster analysis, it is difficult to determine the appropriate parameter settings without prior knowledge on the data. For instance, there is a difficulty in finding the number of clusters for users in the case of the *k*-means algorithm. The PROCLUS algorithm supports parameter bracketing for the most-suitable parameter setting process. However, the best *range* must be determined through more repeated iterations because the algorithm using a range to set parameters, as opposed to using a fixed value. The PROCLUS algorithm has two parameters, *k* and *l*, which indicate the number of clusters and number of dimensions, respectively. A total of 50 individual results needed to be analyzed to find (*k*, *l*); that is, the best parameter pair, if the user has set $k = \{1 \text{ to } 5\}$ and $l = \{11 \text{ to } 20\}$.

```
PROCLUS (number of clusters; $k$, average dimensions; $l$ )
{$C_i$ is the $i$th cluster}
{$D_i$ is the set of dimensions associated with cluster $C_i$}
{$M_{\text{current}}$ is the set of medoids in current iterations}
{$M_{\text{best}}$ is the best set of medoids found so far}
{$N$ is the final set of medoids with associated dimensions}
{$A$, $B$ are constant integers}
begin // {1. Initialization Phase}
$S$ = random sample of size $A_k$
$M$ = GREEDY ($S$, $B_k$) // {2. Iterative Phase}
BestObjective = $\infty$
$M_{\text{current}}$ = Random set of medoids $\{m_1, m_2, \ldots, m_k\} \subset M$
repeat
        {Approximate the optimal set of dimensions}
        for each medoid $m_i$ $M_{\text{current}}$ do
          begin
              Let $i$ be distance to nearest medoid from $m_i$
                  $L_i$ = Points in sphere centered at $m_i$ with radius end
        $L = \{L_1, \ldots, L_k\}$
        $(D_1, D_2, \ldots, D_k)$ = FindDimensions ($k$, $l$, $L$)
        {Form the clusters}
        $(C_1, \ldots, C_k)$ = AssignPoints $(D_1, \ldots, D_k)$
        ObjectiveFunction = EvaluateClusters $(C_1, \ldots, C_k, D_1, \ldots, D_k)$
        if ObjectiveFunction < BestObjective then
          begin
                BestObjective = ObjectiveFunction
                $M_{\text{best}} = M_{\text{current}}$
                Compute the bad medoids in $M_{\text{best}}$ end
        Compute $M_{\text{current}}$ by replacing the bad medoids in $M_{\text{best}}$ with random points
        from $M$
        until (termination_criterion)
 // {3. Refinement Phase}
$L = \{C_1, \ldots, C_k\}$
$(D_1, D_2, \ldots, D_k)$ = AssignPoints $(D_1, \ldots, D_k)$
$N = (M_{\text{best}}, D_1, D_2, \ldots, D_k)$
return($N$) end
```

Fig. 3. PROCLUS clustering algorithm.

III. TACR Mining Based on MapReduce

In this section, the methodology using the MapReduce framework is proposed for a market basket analysis using the TACRs algorithm. The TACRs algorithm was applied for intrusion detection in our previous work [21]. Therefore, the concept of TACRs is briefly recalled first, and a description of an extended algorithm using a distributed programming method based on MapReduce will follow.

1. Calendar-Based Temporal Patterns

A calendar schema (CS) is a relational schema that is based on the calendar concept hierarchy [9], [21]. A CS is defined to comprise a set of calendar-based time granularities and corresponding possible domain values. It is of the following form:

$$CS = (G_n{:}D_n, \ldots, G_1{:}D_1), \tag{1}$$

where G_i is a time granularity in a calendar concept, such as a year, a month, or a day, for $1 \le i \le n$. Each D_i, a positive integer, is a domain value of G_i. When the CS is $(G_n, G_{n-1}, \ldots, G_1)$, $1 \le i \le n$, time granularity G_i is included uniquely into G_{i+1}. For example, the schema of (month, day) is valid because "day" is included in a specific "month." In the case of (year, month, week), "week" does not belong to a specific "month." CS = (month:{1–12}, day:{1–30} {1, 20}) is valid for expressing the twentieth day of January, but {2, 31} is not a valid set. A calendar pattern (CP) is an instance of a given schema CS = $(G_n{:}D_n, \ldots, G_1{:}D_1)$ and can be expressed as CP = $\{d_n, \ldots, d_1\}$. Here, each d_i is a domain value of D_i or a symbol "*." If a particular d_i is a symbol, then it implies that all values of domain D_i can be included; hence, this symbol can be interpreted as denoting "every." In addition, "*" means the time periodicity for the current domain. For example, when a CS is given as (month:{1–3}, day:{1–7}), calendar pattern {*, 3} expresses the time intervals {1, 3}, {2, 3}, and {3, 3} — that is, the Wednesdays of the first three months of the year. Depending on the number of "*"s included in a calendar pattern, the expression is classified. A calendar pattern containing i "*"s is called an i-star pattern (CP_i), and other patterns including no "*"s are called a basic time interval (CP_0) [9].

2. TACR Mining Using MapReduce Framework

Customer purchase patterns for each cluster can be generated using associative classification rule mining [22]–[24]. Let D denote a transaction database, $I = \{i_1, i_2, \ldots, i_n\}$ an *itemset*, and $C = \{c_1, c_2, \ldots, c_m\}$ a cluster (or class) label. ACRs are expressed in the form $X \to c_i$, while the antecedent of an ACR is an *itemset*, and a rule itself is called a *ruleitem*.

The support and confidence of ACRs, denoted by σ and δ, respectively, are defined as follows:

$$\sigma = \frac{\text{ruleitem count}}{|D|}, \quad \delta = \frac{\text{ruleitem count}}{\text{itemset count}}. \qquad (2)$$

ACRs are a ruleitem set satisfying the minimum support and minimum confidence. Temporal mining is performed by adding a CP. If a CP for a CS is given, then the transaction of a timestamp is expressed as D(CP). Grammatically, the ACRs are expressed in the form <ACR, CP>. For instance, in the case of CS = (year:{1999–2001}, month:{1–12}, day:{1–30}), rule <$(A \wedge B) \to \text{cluster}_1$>, {*, 2, 3}> shows a calendar and cyclic expression. This rule is valid in a set of basic time intervals, such as {1999, 2, 3}, {2000, 2, 3}, and {2001, 2, 3}, and means that the rule <$(A \wedge B) \to \text{cluster}_1$> is established. In addition to the support and confidence of the ACRs, temporal ACRs use a new threshold, f, called the frequency. Threshold f (%) is defined as the minimum frequency. TACRs satisfy the support and confidence with respect to frequency f in transaction DB if <ACR, CP> holds during no less than $100 \times f$ (%) basic time intervals covered by CP [9], [21]. Therefore, the mining of TACRs results in the discovery of a set of rules that satisfy both the minimum support and minimum confidence from the transaction database. Such mining can discover changing rules as time goes on by considering the frequency f.

Market basket data are huge amounts of data automatically obtained in a given time interval. Thus, the MapReduce structure should be included in the temporal concept for generating efficient associative classification rules. MapReduce's basic data structure is composed of pairs of <key, value>, where either element in <key, value> can be defined as an integer, real number, string, byte string, or arbitrary complex data structures [15]–[16]. For example, the *key* and *value* elements become the URL and HTML contents, respectively, when a group of web pages is needed to be expressed by such a <key, value> pair. Before the input data of MapReduce are created, they need to be segregated at the block level for a MapReduce operation. These are distributed into the data blocks in a specified size. After that, the mapper and reducer tasks start to execute the map function and reduce function independently. The mapper reads the given data and generates the pairs of <key, value> as the basic data structure for the applied transaction. The reducer generates the output in pairs of <key, value> from all the values having the same intermediate key.

During this process, the transaction is performed for the group of intermediate values. The mapper task and reducer task process the data in parallel. The support of frequent *ruleitems* in each basic time interval for forming all temporal ACRs is calculated first. Therefore, the suggested algorithm is processed by the following three steps [21]:

1. First, all large *ruleitems* are generated regarding the given CS.
2. Next, real rules for building the classification model are generated from them based on the confidence level.
3. Finally, the generated rules in the basic time units are renewed when considering the containment of the CP.

It will not be useful to use conventional algorithms, since the parcel acceptance data come at a very large scale. Thus, the *a priori* map and reduce functions for the distribution programming of [25] are applied for the candidate generation and support counting. The *ruleitems* discovered from the basic time interval are also used in CP renewal steps to find the time cycle. The single-pass counting algorithm proposed in [17], along with the MapReduce framework, is extended to generate frequent a priori–based itemsets. In our proposed algorithms,

```
Map (key, value = ruleitems in transaction $t_i(CP_0)$):
Input: database a database partition $D_i(CP_0)$
for each $t_i \in D_i(CP_0)$ do
    for each ruleitem $i \in t_i$ do
        output <$i$, 1>;
    end
end

Reduce (key = ruleitem, value = count):
for each key $y$ do // initial $y.count$ = 0
    for each value $v$ in $y$'s value list do
        $y.count$ + = $v$;
    end
    if $y.count \geq$ minimum support
        output <$y$, $y.count$>; // collected in $R_1$
    end
end
```

Fig. 4. MapReduce algorithm for R_1.

```
Map (key, value = ruleitems in transaction $t_i(CP_0)$):
Input: database a DB partition $D_i(CP_0)$ and $R_{k-1}$ ($k \geq 2$)
read $R_{k-1}$ from DistributedCache;
candidateGen($R_{k-1}$);
for each $t_i \in D_i(CP_0)$ do
    $C_t$ = subset ($C_k$, $t_i$);
    for each candidate ruleitem $c \in C_t$ do
        output <$c$, 1>;
    end
end

Reduce (key = ruleitem, value = count):
for each key $y$ do // initial $y.count$ = 0
    for each value $v$ in $y$'s value list do
        $y.count$ + = $v$;
    end
    if $y.count \geq$ minimum support
        output <$y$, $y.count$>; // collected in $R_k$
    end
end
```

Fig. 5. MapReduce algorithm for R_k ($k \geq 2$).

each map function computes the count value of each candidate from its own partition, and each candidate and its count value are then presented as the output. The processing result of the map function, candidates, and their count values are collected and summed for the reduce function. The communication cost is decreased through the count distribution process for the map and reduce functions. The algorithms are performed as follows:

- The transaction database is divided based on the basic time intervals and database partitioning.
- The algorithm in Fig. 4 discovers the *ruleitems* containing one item through a single transaction scan.
- First, the <*ruleitem*, 1> is output regarding each *ruleitem* contained in the transaction based on the map function.
- The reduce function collects the counts of each *ruleitem* and returns the result of <*ruleitem*$_1$, count>, R_1.

Figure 5 is an algorithm for *ruleitems* containing more than two items. It operates as follows:

1. First it reads R_1 to generate candidate C_2 in the DistributedCache of Haddop [26]. It is possible to generate a candidate by calling the candidateGen function.
2. Next, the map function scans transaction t_i with the subset function to identify the candidates in all t_i.
3. Finally, <candidate, 1> is generated regarding all candidates.

```
Map (key, value = discovered rules $ACRs(CP_0)$):
Input: $ACRs$ for each $CP_0$
for each $cp \in CP_0$ do
    for each $ACRs$ $i \in cp_i$ do
        output <$i$, 1>;
    end
end

Reduce (key = i-star patterns for each ACRs, value = count):
for each key i-star pattern do // initial i-star pattern.count = 0
    for each value $p$ in i-star pattern's value list do
        i-star pattern.count + = $p$;
    end
    if i-star pattern.count $\geq$ minimum frequency
            // collected in TCARs
            output <i-star pattern, i-star pattern.count>;
    end
end
```

Fig. 6. MapReduce algorithm for temporal pattern update.

The reduce function collects all count values of the candidates and generates <*ruleitem*$_2$, count> as large 2-*ruleitems*. The exploration process from large 3-*ruleitems* to *k*-*ruleitems* is identical to that of large 2-*ruleitems*.

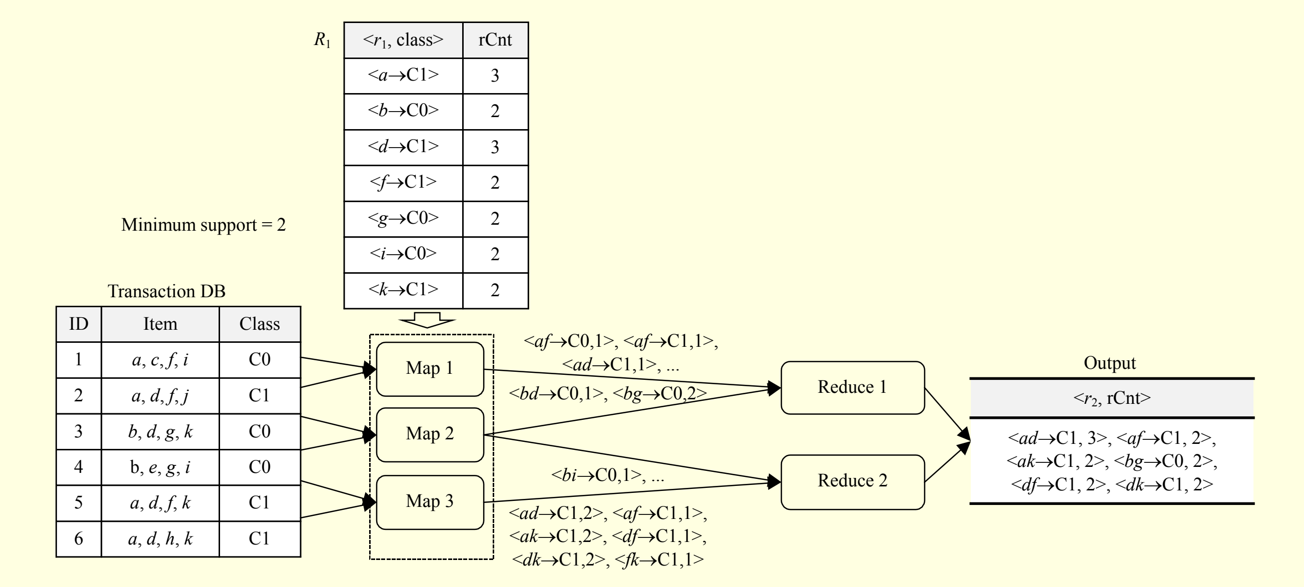
Figure 6 shows an updating algorithm for *i*-star patterns containing more than one "*" for each rule in the CP$_0$. This updating algorithm operates as follows:

1. First, each ACR is divided by each CP$_0$.
2. The map function generates mappers containing one "*" and outputs <ACRs, 1> by each map function.
3. Reducers are generated at the reduce function step for each ACR, and count values of 1-star patterns belonging to the key, ACRs, are collected.
4. Finally, the pair of <ACRs, 1-star pattern> is generated satisfying the minimum frequency.

The renewal process from the 2-star pattern to *i*-star pattern is the same as the 1-star pattern update process.

Examples for the ACRs discovery process and 1-star pattern update in the basic time intervals are shown in Figs. 7 and 8. After finding 1-*ruleitem* by the algorithm described in Fig. 4, Mapper reads R_1 and generates candidate 2-*ruleitems*. In the case where the minimum support is two, the Reducer sums up the counts and outputs frequent 2-*ruleitems* into R_2, as shown in Fig. 7. At the CP updating phase, Mapper composes 1-star patterns including just one "*" and outputs ACRs relevant to each 1-star pattern.

Reducer sums up the count of 1-star patterns satisfying each of the ACRs and outputs an <ACRs, ACRs(CP$_1$)> that meets the minimum frequency. The same procedure is applied to *i*-star patterns update.

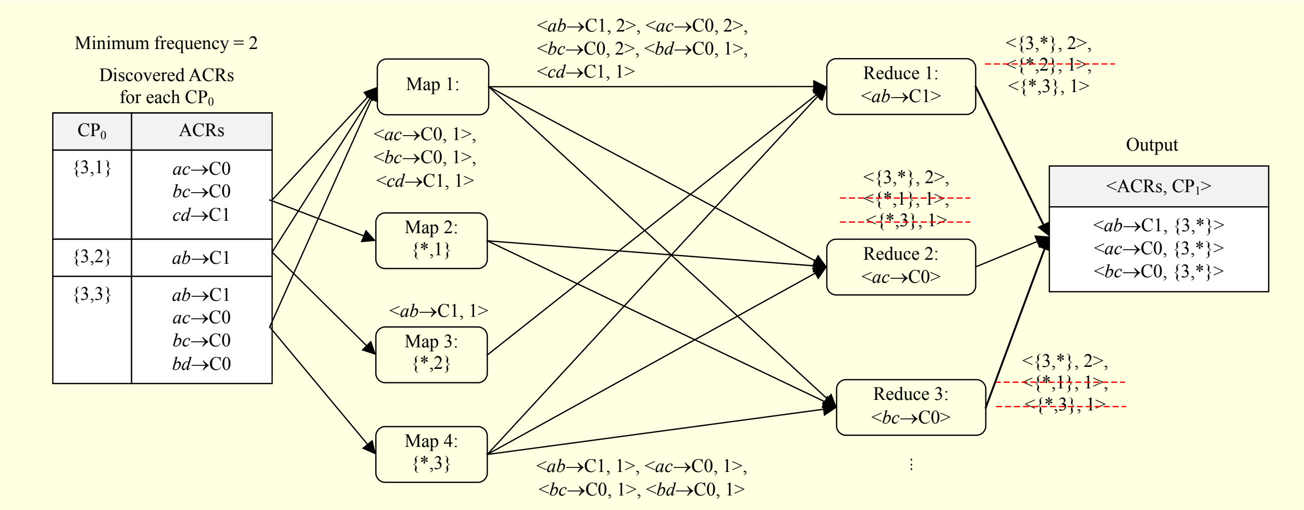


$<r_1$, class>	rCnt
<*a*→C1>	3
<*b*→C0>	2
<*d*→C1>	3
<*f*→C1>	2
<*g*→C0>	2
<*i*→C0>	2
<*k*→C1>	2

ID	Item	Class
1	*a, c, f, i*	C0
2	*a, d, f, j*	C1
3	*b, d, g, k*	C0
4	b, *e, g, i*	C0
5	*a, d, f, k*	C1
6	*a, d, h, k*	C1

$<r_2$, rCnt>
<*ad*→C1, 3>, <*af*→C1, 2>, <*ak*→C1, 2>, <*bg*→C0, 2>, <*df*→C1, 2>, <*dk*→C1, 2>

Fig. 7. Example of R_2 generation phase.



CP_0	ACRs
{3,1}	*ac*→C0 *bc*→C0 *cd*→C1
{3,2}	*ab*→C1
{3,3}	*ab*→C1 *ac*→C0 *bc*→C0 *bd*→C0

<ACRs, CP_1>
<*ab*→C1, {3,*}> <*ac*→C0, {3,*}> <*bc*→C0, {3,*}>

Fig. 8. Example of CP updating phase.

IV. Experimental Results and Discussion

1. Datasets and Preprocessing

The census information for customer characterization is mainly for a metropolitan area reflecting a high population density and consumer sentiment. The results of a spatial informatization performance show that the dataset to be used includes 5,269,364 mail addresses and 15,987 features from the census information. The PROCLUS clustering method forms similar groups with the subspace of all dimensions for the class of prior knowledge included in the training datasets. Next, PROCLUS classifies the objects of test datasets to identify the group to which they belong. Therefore, the class information (prior knowledge) is needed. In this experiment, four types of customer groups segmented by credit rating are given, and the prior class labels are defined with this information. The prior

Table 2. Data description for TACR mining.

Attribute	Description
Timestamp	Product purchase date <year/month/week/day/hour(time-intervals)>
PostAddr	Postal address for customer identity
ItemCd	Product code
ItemName	Product name
Class	Cluster ID

class labels are considered as the original clusters, and the classes grouped by the PROCLUS method become discovered clusters, which are compared and analyzed. The data for a market basket analysis is derived from the e-Commerce purchase history of Korea Post. The data attributes include the purchase date, customer address, product code, and product name. The dataset uses a total of 18,680,419 cases from July 2008 to December 2009. In particular, the week is added to the time-stamped time granularity, and the purchase time is sectioned into morning, afternoon, evening, or night. The preprocessed data for temporal ACR mining are shown in Table 2.

2. PROCLUS Cluster Analysis for Profiling Customer Characterization

The preprocessed census data containing 99 dimensions and 15,987 MicroArea block objects are clustered in this subsection. The PROCLUS clustering method supports parameter bracketing for the most-appropriate-parameter setting process. The parameter settings of the PROCLUS algorithm for census data are given in Table 3. Because the census data have 99 attributes, the range of the average dimensions is set from 1 to 99, and the number of clusters is set from 3, which is lower than the original number of clusters (4 to 10). Each time the experiment is repeated, two parameter values are incremented by 1.

As PROCLUS clusters similar objects in the training datasets and classifies which group out of the defined clusters the test data objects belong to, two methods of clustering and classification are included. Therefore, an evaluation measure such as the sum of the squared error (from a traditional clustering method) is inappropriate. The PROCLUS algorithm of OpenSubspace considers each cluster label after clustering as the prediction target class of the classification problem. The current study uses evaluation measures such as F_1-value, accuracy, entropy, and coverage to evaluate the clustering algorithm. Formal definitions of these measures are given below [14].

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}), \tag{3}$$

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}), \tag{4}$$

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \tag{5}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{6}$$

$$\text{Entropy} = -\sum_{i=1}^{c} p(i/t)\log_2 p(i/t), \tag{7}$$

and

$$\text{Coverage} = \frac{\sum_{i=1}^{c} N_i}{|D|}. \tag{8}$$

Table 3. Parameter bracketing for PROCLUS algorithm.

Parameter	From	Offset	Op	Step	To
Avg. dimensions	1	1	+	99	99
# of clusters	3	1	+	7	10
Total number of experiments: 693					

Table 4. Evaluation measures for PROCLUS algorithm.

Evaluation measures	Values
F_1-value	0.76
Accuracy	0.875
Coverage	0.86
1.0 entropy	0.72
Outliers	384
Parameters	Clusters = 5, dimensions = 47

Table 5. Confusion matrix of original vs. discovered groups.

Original groups	Discovered groups (cluster: 5, dimension: 47)				
	C1	C2	C3	C4	C5
Cluster 1 (4,572/4,578)	0	**3,810**	254	508	0
Cluster 2 (3,591/3,609)	57	228	114	**2,691**	**501**
Cluster 3 (3,802/4,133)	79	159	**2,448**	159	**957**
Cluster 4 (3,426/3,455)	**3,426**	0	0	0	0

Table 4 shows the results of the PROCLUS algorithm with five clusters and 47 dimensions when applying the parameter bracketing in Table 3. The census dataset was applied with 47 dimensions depending on the parameter set with the range shown in Table 3, and the optimal number of clusters determined was five. Based on the results in Table 4, PROCLUS was used along with the clustering results shown in Table 5.

For cluster 1, cluster 2, cluster 3, and cluster 4 (the original groups of Table 5), specific clusters detected from the PROCLUS analysis results are profiled into the customer characterization. Different subgroups can be formed from the original groups using the confusion matrix in Table 5. For example, cluster 2 comprises five clusters, C1 through C5. However, C1, C2, and C3 are removed. Since the number of members included in the cluster is small, these are considered to be minor groups. Therefore, one C2 cluster is created for the

*True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN)

http://dx.doi.org/10.4218/etrij.15.2314.0068

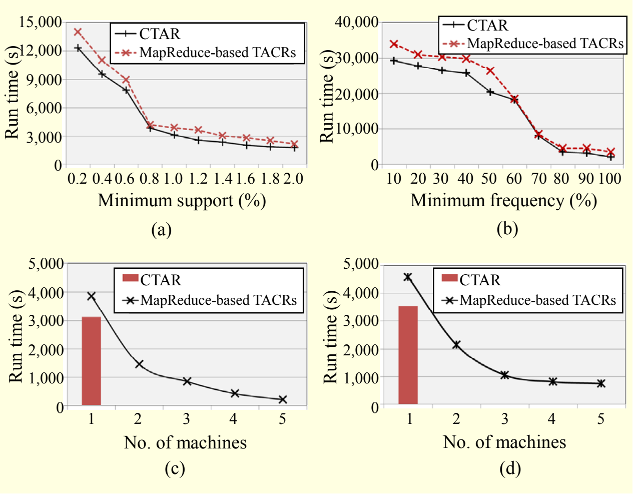


Fig. 9. Runtime of MapReduce-based TACRs vs. CTAR: (a) scalability with support, (b) scalability with frequency, (c) scalability with no. of machines: support set to 1.0%, and (d) scalability with no. of machines: frequency set to 80%.

cluster 1 group, and two subgroups, C4 and C5, are created for the cluster 2 group. The groups selected as the cluster in Table 5 are displayed in bold font.

3. TACRs Mining Performance Evaluation

In this subsection, we evaluate the performance of the proposed algorithms, including their scalability with respect to the support and frequency thresholds. While the minimum support is a parameter affecting the generation of *ruleitems*, the minimum frequency is a parameter affecting the generation of *i*-star CPs. We carried out experiments on our market basket data from Korea Post. For the performance evaluation of parallel mining, five machines equipped with a 2.67 GHz Intel(R) Core(TM) CPU and 8 GB of RAM were used in the experiments. Java ver. 1.6 and Hadoop 0.20 were used to develop the algorithm. The transaction DB used in the experiments is composed of 17 distinct items, and the average length of the transactions is 12. The execution times between the algorithm proposed in this paper and the traditional classification based on temporal class-association rules (CTAR) [21] algorithm are compared when executed on one machine (see Fig. 9). The experimental results shows that the CTAR algorithm is faster in both the large *ruleitems* discovery phase and the *i*-star pattern updating phase in terms of the default time, as shown in Figs. 9(a) and 9(b). The use of a Hadoop distributed file system results in an additional cost when communicating between each mapper and reducer. However, when multiple machines are used, the proposed MapReduce-based methods show a superior performance since the algorithm with MapReduce shows an increase in the execution speed depending on the number of employed machines. The proposed algorithms show better results even when only two machines are used, as illustrated in Figs. 9(c) and 9(d). Figure 9(c) shows a runtime comparison in terms of the machine quantity when the support is set to 1%, and Fig. 9(d) shows a runtime comparison of the machine quantity change when the frequency is set to 80%.

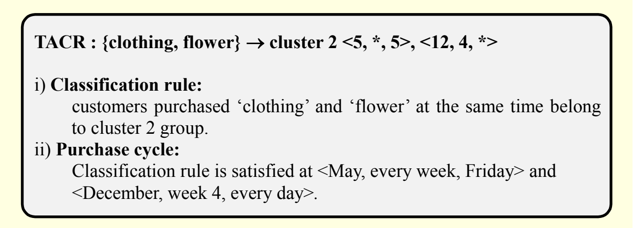
Table 6. Detailed accuracy results based on class. Optimal parameters of the algorithm were set as follows: support (0.8%), confidence (68%), and frequency (62%).

Class label (customer group)	Precision	Recall	F_1-value
Cluster 1	0.671	0.873	0.759
Cluster 2	0.745	0.845	0.792
Cluster 3	0.993	0.726	0.839
Cluster 4	0.738	0.878	0.802
Cluster 5	0.693	0.432	0.532
Cluster 6	0.995	1.000	0.998
MAE	0.0692		
RMSE	0.263		

Table 7. Confusion matrix for predicted class.

Actual class	Predicted class					
	C1	C2	C3	C4	C5	C6
Cluster 1	87.29%	0%	0.42%	0%	12.29%	0%
Cluster 2	0%	84.54%	0%	0%	14.98%	0.48%
Cluster 3	3.85%	12.98%	72.6%	0.48%	4.81%	5.29%
Cluster 4	0.98%	9.76%	0%	87.80%	0%	1.46%
Cluster 5	40.36%	5.45%	0.45%	0%	44.18%	9.55%
Cluster 6	0%	0%	0%	0%	0%	100%

The accuracy of the proposed classification model in predicting the customer characterization group was evaluated based on the precision, recall, and F_1-value of (3) through (5). Thresholds such as the support, confidence, and frequency involved in the generation of classification rules were experimented on in the same way as our previous works [21]; thus, the detailed process of the test was omitted, and we used the optimal parameter setting values from the test results. All evaluation measures were obtained using the methodology of a stratified 10-fold cross validation for all six classes. Table 6 shows the performance evaluation results of the TACRs model for the different classes. In addition, the error rate measures of the classification use the mean absolute error (MAE) and root-mean-square error (RMSE). All classes (six clusters) were



TACR : {clothing, flower} → cluster 2 <5, *, 5>, <12, 4, *>

i) **Classification rule:**
customers purchased ‘clothing’ and ‘flower’ at the same time belong to cluster 2 group.

ii) **Purchase cycle:**
Classification rule is satisfied at <May, every week, Friday> and <December, week 4, every day>.

Fig. 10. Example of discovered TACR.

discovered by the PROCLUS clustering algorithm.

To evaluate the performance with respect to the number of instances and classes, a confusion matrix is used. Table 7 records the accuracy of the classifiers used in a confusion matrix. According to the results shown in Tables 6 and 7, our TACRs method performs very well. Temporal ACR mining discovers the frequent *ruleitems* for customer-purchased items and represents the purchasing cycle that occurs repeatedly under the given time schema. The class included in *ruleitems* is the profiled cluster information found through a geo-lifestyle cluster analysis. An example of the data mining execution result for direct marketing is as follows.

For example, when {clothing, flower}→cluster 2 <5,*,5>, <12,4,*> is applied for direct marketing, it is possible to conduct a targeted marketing campaign with the rules found from TACRs mining, such as is shown in Fig. 10.

V. Conclusion

The most important factor for a successful direct marketing is to obtain a list of customers. In this paper, a data mining method including temporal and spatial information is proposed for an appropriate selection of a customer list. Initially, Korea Post customers were segmented into groups according to style. The census data including demographic, economic, and geographical information are analyzed for customer grouping at the MicroArea block level with a consideration of personal information. The PROCLUS algorithm provided by OpenSubspace was used because the census information is a high-dimensional dataset. The cluster analysis showed that registered customers of Korea Post are segmented into six groups of different characteristics. A temporal associative classification method applied with the MapReduce framework was developed as a new data mining method for direct marketing. A distributed processing method using multiple machines, rather than a single machine, was developed to find product purchase cycles and purchase patterns from a large amount of customer purchase history data. The distributed processing method is compared with a traditional method for various parameter changes of the TACRs algorithm, and it was proven that the TACRs algorithm handles the market basket data analysis faster. As the contribution of the data mining techniques proposed in this paper, the GIS-based profiled customer characterization information and parallel mining method enabling a faster analysis of huge amounts of product purchase information are made available to Korea post for the first time. Future research will use the suggested data mining techniques on real big data related to marketing collected over years, analyze the results, and apply MapReduce-based TACRs to various fields.

References

[1] X. Ling and C. Li, “Data Mining for Direct Marketing: Problems and Solutions,” *Int. Conf. Knowl. Discovery Data Mining*, 1998, pp. 73–79.

[2] A. Hall, *Direct Mail that Makes Cents*, NPF2014 (Nat. Postal Forum), Washington D.C., 2014.

[3] B. Greshan and C. Kiani, *Mail Meets Mobile Technol.*, NPF2014 (Nat. Postal Forum), Washington D.C., 2014.

[4] L. Sing’oei and J. Wang, “Data Mining Framework for Direct Marketing: A Case Study of Bank Marketing,” *J. Comput. Sci. Issues*, vol. 10, no. 2, Mar. 2013, pp. 198–203.

[5] D. Zakrzewska and J. Murlewski, “Clustering Algorithms for Bank Customer Segmentation,” *Int. Conf. Intell. Syst. Des. Appl.*, Sept. 8–10, 2005, pp. 197–202.

[6] H.-J. Oh, C. Lee, and C.-H. Lee, “Analysis of the Empirical Effects of Contextual Matching Advertising for Online News,” *ETRI J.*, vol. 34, no. 2, Apr. 2012, pp. 292–295.

[7] R. Agrawal, T. Imieliński, and A. Swami, “Mining Association Rules between Sets of Items in Large Databases,” *Int. Conf. Manag. Data*, 1993, pp. 207–216.

[8] R. Agrawal and R. Srikant, “Mining Sequential Patterns,” *Int. Conf. Data Eng.*, 1995, pp. 3–14.

[9] Y. Li et al., “Discovering Calendar-Based Temporal Association Rules,” *Data Knowl. Eng.*, vol. 44, no. 2, Feb. 2003, pp. 193–218.

[10] B. Őzden, S. Ramaswamy, and A. Silberschatz, “Cyclic Association Rules,” *Int. Conf. Data Eng.*, Orlando, FL, USA, Feb. 23–27, 1998, pp. 412–421.

[11] K. Verma and O.P. Vyas, “Efficient Calendar Based Temporal Association Rule,” *ACM SIGMOD Record*, vol. 34, no. 3, Sept. 2005, pp. 63–70.

[12] Y.J. Lee et al., “Mining Temporal Interval Relational Rules from Temporal Data,” *J. Syst. Softw.*, vol. 82, no. 1, 2009, pp. 155–167.

[13] S. Laxman and P.S. Sastry, “A Survey of Temporal Data Mining,” *Sadhana*, vol. 31, no. 2, Apr. 2006, pp. 173–198.

[14] E. Müller et al., *OpenSubspace: Weka Subspace-Clustering Integration*, 2013. Accessed May 11, 2014. http://dme.rwth-aachen.de/OpenSubspace/

[15] J. Dean and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters,” *Operating Syst. Des.*

Implementation, vol. 6, 2004, pp. 137–150.

[16] S. Parsa and M. Hamzei, "Locality-Conscious Nested-Loops Parallelization," *ETRI J.*, vol. 36, no. 1, Feb. 2014, pp. 124–133.

[17] M.-Y. Lin, P.-Y. Lee, and S.-C. Hsueh, "A Priori-Based Frequent Itemset Mining Algorithms on MapReduce," *Int. Conf. Ubiquitous Inf. Manag. Commun.*, 2012, no. 76.

[18] C. Aggarwal et al., "Fast Algorithms for Projected Clustering," *Int. Conf. Manag. Data*, Philadelphia, PA, USA, 1999, pp. 61–72.

[19] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed., San Francisco, CA, USA: Morgan Kaufmann, 2006, pp. 508–511.

[20] B. Durrant et al., *Weka 3: Data Mining Software in Java*, Machine Learning Group at the University of Waikato, 2014. Accessed July 12, 2014. http://www.cs.waikato.ac.nz/ml/weka/

[21] J.S. Kim et al. "CTAR: Classification Based on Temporal Class-Association Rules for Intrusion Detection," *Int. Workshop WISA*, Jeju, Rep. of Korea, Aug. 25–27, 2003, pp. 84–96.

[22] W. Li, J. Han, and J. Pei, "CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules," *IEEE Int. Conf. Data Mining*, San Jose, CA, USA, 2001, pp. 369–376.

[23] F. Thabtah, "A Review of Associative Classification Mining," *Knowl. Eng. Rev.*, vol. 22, no. 1, Mar. 2007, pp. 37–65.

[24] T.H. Nhan, J.W. Lee, and K.H. Ryu, "Spatiotemporal Pattern Mining Technique for Location-Based Service System," *ETRI J.*, vol. 30, no. 3, June 2008, pp. 421–431.

[25] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," *Int. Conf. Very Large Databases*, 1994, pp. 487–499.

[26] K. Shvachko et al., "The Hadoop Distributed File System," *IEEE Symp. Mass Storage Syst. Technol.*, Incline Village, NV, USA, May 3–7, 2010, pp. 1–10.

Heon Gyu Lee received his BS degree in computer science from Kyonggi University, Suwon, Rep. of Korea, in 2002 and his MS and PhD degrees in computer science from Chungbuk National University, Cheongju, Rep. of Korea, in 2005 and 2009, respectively. Since 2009, he has been working for the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea, where he is now a senior member of the researching staff. His main research interests are data mining, databases, bioinformatics, pattern recognition, and knowledge-based information retrieval.

Yong Hoon Choi received his BS degree in mechanical engineering from the University of Ulsan, Rep. of Korea, in 1997 and his MS and PhD degrees in industrial and manufacturing systems engineering from Iowa State University, Ames, USA, in 1999 and 2003, respectively. Since 2004, he has been working for the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea, where he is now a director of the Logistics Process Research Team. His main research interests are signal processing; process design and monitoring; SCM; and logistics automation.

Hoon Jung received his BS degree in industrial engineering from Kyunghee University, Seoul, Rep. of Korea, in 1989 and his MS and PhD degrees in industrial engineering from Iowa State University, USA and the University of Missouri, MO, USA, in 1997 and 2001, respectively. Since 2002, he has been working for the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea, where he is now a senior researcher. His main research interests are supply chain management, logistics information systems, and logistics strategy.

Yong Ho Shin received his BS degree in industrial engineering from Seoul National University, Rep. of Korea, in 1993 and his MS and PhD degrees in industrial and system engineering from KAIST, Daejeon, Rep. of Korea, in 1995 and 2003, respectively. From 2007 to 2009, he worked for the Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea. Since 2009, he has been with the School of Business, Yeungnam University, Kyeongsan, Rep. of Korea, where he is now an associate professor. His main research interests are operations management, logistics management, data mining, and knowledge management systems.