Expert Systems with Applications 38 (2011) 14499–14513

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Group *RFM* analysis as a novel framework to discover better customer consumption behavior ☆

Hui-Chu Chang [a,*], Hsiao-Ping Tsai [b]

[a] Department of Information Technology and Communication, TungNan University of Technology, No.152, Sec. 3, Beishen Rd., Shenkeng Dist., New Taipei City 222, Taiwan ROC
[b] Department of Electrical Engineering, National Chung Hsing University, No. 250, Kuo Kuang Road, Taichung 402, Taiwan, ROC

ARTICLE INFO

Keywords:
RFM analysis
Segmentation
Constrained clustering
Cluster distribution

ABSTRACT

The *RFM* model provides an effective measure for customers' consumption behavior analysis, where three variables, namely, consumption interval, frequency, and money amount are used to quantify a customer's loyalty and contribution. Based on the *RFM* value, customers can be clustered into different groups and the group information is very useful in market decision making. However, most previous works completely left out important characteristics of purchased products, such as their prices and lifetimes, and apply the *RFM* measure on all of a customer's purchased products. This renders the calculation of the *RFM* value unreasonable or insignificant for customer analysis. In this paper, we propose a new framework called GRFM (for group *RFM*) analysis to alleviate the problem. The new measure method takes into account the characteristics of the purchased items so that the calculated the *RFM* value for the customers are strongly related to their purchased items and can correctly reflect their actual consumption behavior. Moreover, GRFM employs a constrained clustering method PICC (for Purchased Items-Constrained Clustering) that could base on a cleverly designed purchase pattern table to adjust original purchase records to satisfy various clustering constraints as well as to decrease re-clustering time. The GRFM allows a customer to belong to different clusters, and thus to be associated with different loyalties and contributions with respect to different characteristics of purchased items. Finally, the clustering result of PICC contains extra information about the distribution status inside each cluster that could help the manager to decide when is most proper to launch a specific sales promotion campaign. Our experiments have confirmed the above observations and suggest that GRFM can play an important role in building a personalized purchasing management system and an inventory management system.

1. Introduction

A successful customer-oriented marketing strategy is very important in the sense that it can help to strengthen the relationships between the customers and the business. Understanding customer characteristics and satisfying customer requirements not only can improve the customer loyalty but can make great profit by decreasing the risk of business operation (Cheng & Chen, 2009). It is no wonder that the techniques like customer segmentation or clustering (Management Science, 2003; Wu & Lin, 2005; Yeh, Yang, & Ting, 2008) have been widely used in order to understand the consumption behavior of different groups of customers.

Customer segmentation is a supervised learning process that classifies customers into the predefined classes while customer clustering, on the other hand, groups the customers into non-predefined classes. The discovered group information is very useful in the formulation of proper promotion strategies or pricing policies to improve customer response rate and finally to increase business profit. To identify high-response customers for product promotion, the *RFM* analysis (Miglautsch, 2000) incorporates three variable value, including customers' consumption interval (i.e., *R* value), frequency (i.e., *F* value), and money amount (i.e., *M* value), to model customer's tendency of purchasing. For avoiding ambiguity, the term *RFM* value is represented a single value what using a measuring function to integrate *R* value, *F* value and *M* value. Through the *RFM* analysis, customers' loyalties and contributions can then be properly measured (Wu & Lin, 2005). Because of the success of the *RFM* measure, great efforts have been devoted to customer segmentation or clustering based on the customers' *RFM* values (Cheng & Chen, 2009; Miglautsch, 2000; Yeh et al., 2008).

Although the *RFM* value has been utilized in customer segmentation or clustering, most of the previous works measure the *RFM* value without considering customers' purchasing behavior

☆ This document is a collaborative effort.

* Corresponding author. Tel.: +886 2 86625968; fax: +886 2 86625969.
E-mail addresses: hcchang@mail.tnu.edu.tw (H.-C. Chang), hptsai@iis.sinica.edu.tw (H.-P. Tsai).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.05.034

regarding different products. Hence, the above works fail to provide effective information for promotion of certain products. We summary the reasons for why the characteristics of purchased items should be considered in analyzing customers' purchasing behavior with the *RFM* measure as follows:

- First, there is a dramatic variation in price and lifetime of products. For example, the frequency that a user buys a new notebook is very different from that of buying a new cloth. Moreover, the amount of money spent on the above two items is very different. This implies that the customer loyalty and contribution should be considered with respect to the purchased items. The traditional *RFM* value only provides lump-sum evaluation indices, which are coarse in quantifying customer loyalty and contribution. Previous works (Wu & Lin, 2005; Yeh et al., 2008) that apply a static measuring criterion to all products without addressing their differences thus lack the precision in targeting the most suitable customers.
- Second, the associations between the user and his bought products provide a useful hint about what he would like to buy in the future. Instead of counting all the bought products, which include those rarely bought, the *RFM* value measured only on frequently-bought item sets can do better in predicting the user's purchasing regularity. That is, if the *RFM* value of a customer is measured with respect to different purchased item sets, his requirements can be better satisfied, and a better personalized purchasing management system can be developed to further improve customer relationships.
- Third, sales management and customer management are equally important. A sales manager may want to know "What products are often co-purchased?", while a customer manager may want to know "Who are potential buyers of a certain product item or item set?" And they both may be interested in "What are the consumption interval, frequency, and money amount of a customer over a specific item set?" Therefore, better than the traditional customer oriented *RFM* value, customers' *RFM* values measured over certain purchased items can provide very useful knowledge for building an effective inventory management system.

On the other hand, a customer may be highly interested in the products bought by customers with similar purchasing behavior. Thus, what a customer buys are good targets to promote to the customers with similar purchasing behavior. To discover a good sales policy, we need to figure out the potential buyers, how loyal they are, and how he may contribute to specific products. That is, we need to cluster customers according to their purchased items and calculate their *RFM* value to track their consumption behavior. With this, we then can develop a precise sales policy to better meet the market need.

Therefore, in this paper, we propose a novel Group *RFM* (GRFM for short) framework to identify high loyal and contribution customers; moreover, it discovers potential customers for products promotion. Instead of calculating the customers' *RFM* values on all of the products they have ever purchased, the GRFM calculates customer's GRFM-value what considers customers' purchase patterns as well as the characteristics of products in analyzing customers. Specifically, the GRFM first discovers the frequent patterns, each of which presents a set of products that are purchased frequently in the transactional data set. Then, based on the discovered frequent patterns, customers are clustered into groups, i.e., for each frequent pattern, customers are regarded as a group if they have bought the products in the frequent pattern. By the way, we can tighten the candidates for promoting products in a frequent pattern. Furthermore, we further considers the diverse characteristics of products including their average lifetime and average unit price in evaluate a customer's purchase potential and propose a new measure function that calculates a customer's GRFM-value on the products regarding to each frequent pattern. Therefore, we can obtain the GRFM-values of the customers in a cluster that possesses the characteristics of the purchased items and correctly reflect his loyalty and contributions. Moreover, the GRFM incorporates the PICC (Purchased Items-Constrained Clustering) algorithm, which can reuse the discovered purchase patterns to propose proper sales policies to promptly respond to the market demands. The major contributions of the paper are summarized as follows:

- We propose a new GRFM measure function to evaluate customers' purchase potential with respect to their purchase patterns that involve products with specific characteristics, such as unit price and lifetime. This facilitates the development of a personalized purchasing management system as well as an effective inventory management system. In addition, it can be used in trend analysis and intensity analysis about particular products.
- The GRFM framework incorporates the PICC algorithm to dynamically cluster customers according to a specific demand in terms of constraints, where a constraint is associated with a product category. Therefore the PICC algorithm can base on that information to generate a variety of sales policies according to the clustering results to meet specific demands from users.

The rest of this paper is organized as follows. In Section 2, we review the related works. In Section 2, we give preliminary knowledge to be used in the subsequent sections. In Section 4, we introduce the GRFM framework. Section 5 details our experimental results. Finally, the concluding remarks are provided in Section 6.

2. Related works

The concept of customer segmentation was developed by an American marketing expert, Wendell R. Smith, in the middle of 1950. It is a technology to cluster customers into groups that share similar characteristics and tend to display similar patterns. Later, the *RFM* model is first proposed by Hughes (1994), and it is a model that differentiates important customers from large transaction data. *RFM* method is very effective attributes for customer segmentation (Newell, 1997). Recall that the *RFM* analysis incorporates three important attributes including consumption recency (R), frequency (F), and monetary (M) to model customers' purchasing behavior and measure their loyalty, contribution, and buying potential. In the *RFM* model, recency (R) is, in general, defined as the interval from the time when the latest consumption happens to the present, frequency (F) is the number of consumption within a certain period, and monetary (M) is the amount of money spent within a certain period. An earlier study showed that customers with bigger R, F, and M values are more likely to make a new transaction (Wu & Lin, 2005). Because of the success of the *RFM* model in customer analysis, great efforts have been devoted to customer segmentation or clustering based on the customers' *RFM* values (Miglautsch, 2000; Tsai & Chiu, 2004). For clustering customers based on the *RFM* value, the customers' *RFM* values scoring is key factor. As mentioned in Cheng and Chen (2009), there are two opinions on the importance of the R, F, and M values. While the three parameters are considered equally important in Miglautsch (2000), they are unequally weighted due to the characteristics of industry in Tsai and Chiu (2004). In Miglautsch (2000), each of the R, F, M dimensions is divided into five equal parts and customers are clustered into 125 groups according to their R, F, M values. Consequently, the high potential groups (or customers) can be

easily identified. In Tsai and Chiu (2004), the *RFM* model is utilized in profitability evaluation and a weighted-based evaluation function was proposed. The value of customer C_i is represented by Eq. (1).

$$V(c_i) = W^R * R(c_i) + W^F * F(c_i) + W^M * M(c_i) \quad (1)$$

where $R(c_i)$, $F(c_i)$, and $M(c_i)$ represent customer c_i's *R*, *F*, and *M* values and W^R, W^F, and W^M represent their weights respectively. In general, the *RFM* value measuring is objective (Cheng & Chen, 2009).

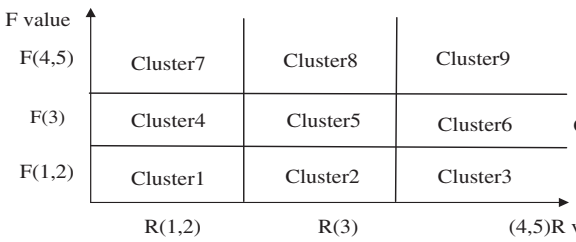
The above *RFM* value measuring methods all adopt a single criterion to measure the *RFM* value of a customer no matter what kinds of products were purchased. However, the characteristics and lifetimes of the purchased products are not always the same, grouping customers in this way can not provide precise quantitative prediction. For example, as shown in Fig. 1, assume that there are 20 transaction records of five customers C01 to C05 in a transaction database T. Each transaction consists of five attributes, including transaction ID, customer ID, date, purchased items, and monetary expense. The clustering method proposed in Wu and Lin (2005) actually creates a customer value matrix according to the calculated *RFM* values for clustering customers. Once the partitions of the axes are decided, each customer is placed in one of the regions of the customer value matrix. The figure shows, by using the values of *R* and *F* for axes, we create nine regions in the matrix, which allows for clustering the customers into nine groups. With this matrix, the customers in the example transaction database can be clustered into three groups, where C02 and C05 in Cluster2 are regarded as the highest in loyalty and contribution. There are several problems with this traditional clustering method, however. First, the method makes C02 and C05 into the same cluster, implying they have the same loyalties and contributions. This is not correct, however. If we look into the details of their purchased items we discover that their preferences over the goods of purchase are quite different. For example, if the business targets a sale promotion about products of clothing to high contribution customers, then the promotion could attract C05 but C02. Since C05 is used to buy clothing but C02 is used to buy office appliances. A second problem arises, where C03 is evaluated to be a customer with lower loyalty than C05 because his *F* value is smaller. Looking into the purchased items, we find C03 is a buyer of 3C products, which

Transaction ID	Customer Name	Date	Purchased Items	expense
T1	C05	3/21	shose,socks	$3,000
T2	C01	3/21	milk,soda	$1,000
T3	C03	3/21	cell phone	$25,000
T4	C04	3/21	beer,cookie	$500
T5	C01	4/12	beer,cookie	$1,200
T6	C02	4/12	chairs	$1,500
T7	C05	4/12	cloth	$2,000
T8	C04	4/12	beaf	$1,000
T9	C05	4/30	shose,socks	$2,500
T10	C01	4/30	milk,soda	$1,000
T11	C05	4/30	jack,dress	$3,000
T12	C02	5/10	pens	$500
T13	C04	5/10	pork	$600
T14	C02	5/10	paper	$300
T15	C02	5/25	lamp	$1,500
T16	C03	5/25	i-pod	$30,000
T17	C01	6/1	milk,soda	$900
T18	C02	6/1	desk	$4,500
T19	C04	6/16	sport shirt	$3,500
T20	C05	6/16	dress	$3,000

Calculating the RFM value of customer. (Today is 6/17)
.R (weeks) score : 0-3 weeks = 5, 3-6 weeks = 4, 6-9 weeks = 3,
9-12 weeks = 2, 12-15 weeks = 1;
.F (times) score : 7 times and up = 5, 5-6 times = 4, 3-4 times = 3,
2 times = 2, 1 time = 1;
.M (expense) score : $15000 and up = 5, $9000-$14444 = 4,
$4000-$8999 = 3 , $1500-$3999 = 2, 1500 down = 1;

Customer Name	RFM array
C01	(5,3,3)
C02	(5,4,3)
C03	(4,2,5)
C04	(5,3,3)
C05	(5,4,4)

Create a customer value matrix



Clustering

Cluster ID	Member
Cluster1	C01,C04
Cluster2	C02,C05
Cluster3	C03

Fig. 1. Example of customers clustering by traditional *RFM*.

usually implies a slow buying frequency and is very different from buying clothes.

Our approach to alleviating the above problems is to consider the characteristics of the purchased items while analyzing the purchasing behavior of the customers. For comparison, a simple clustering result of our approach for the same example transaction database is shown in Fig. 2, which can successfully solve the above problems. First, the clusters are strongly related to purchase patterns and can correctly reflect the purchasing behavior of the customers. This in turn can enable proper promotion plans. For example, the cluster containing customers C04 and C05 have high contribution and loyalty in clothing and should be targeted with clothing promotion plans. Besides, a customer is associated with different (R, F, M) values according to his purchased items. For example, the (R, F, M) value of customer C04 is (1,5,5) for the category of foods, while that is (5,5,3) for the category of clothing. He belongs to two clusters, meaning he is loyal in both foods and clothing. He did buy foods lately. This detailed information about the (R, F, M) values of the customers allows for more correct customer relationship to be managed.

3. Preliminaries

This section first presents the concepts of the constraint-based clustering, then the category hierarchy of products. Last, we introduce notations and definitions used throughout this paper.

3.1. Constraint-based clustering

Constraint-based clustering groups similar objects into clusters while satisfying certain conditions, such as maintaining a fixed number of objects in each cluster. Recently, constrained-based clustering methods have become very popular (Basu, Banerjee, &

Clustering Customers by their purchasing items

Cluster	Members	Kind of buying
Cluster1	C01,C04	Foods
Cluster2	C03	3C
Cluster3	C05,C04	Clothing
Cluster4	C02	office appliances

The calculating criterion of RFM value accordin to purchasing items. (Today is 6/17)

Calculating RFM value criterion of Foods

score / value	1 score	2 score	3 score	4 score	5 score
R (day)	12 up	9~12	6~9	3~6	0~3
F (times)	1	2	3~4	5~6	up 6
M (expense)	$299 down	$300~$499	$500~$699	$700~$999	up $1000

Calculating RFM value criterion of 3C products

score / value	1 score	2 score	3 score	4 score	5 score
R (month)	12 up	9~12	6~9	3~6	0~3
F (times)	0	1	2	3	3 up
M (expense)	$15000 down	$15000~$19999	$20000~$24900	$25000~$29999	$30000 up

Calculating RFM value criterion of Clothing

score / value	1 score	2 score	3 score	4 score	5 score
R (month)	9 up	7~9	4~6	2~3	0~1
F (times)	2	3	4	5	up 5
M (expense)	$1500 down	$1500~$1999	$2000~2499	$2500~$2999	$3000 up

Calculating RFM value criterion of office appliances

score / value	1 score	2 score	3 score	4 score	5 score
R (month)	9 up	7~9	4~6	2~3	0~1
F (times)	1~2	3~4	5~6	7~8	8 up
M (expense)	$4000 down	$4000~$8499	$8500~$11999	$12000~$14999	$15000 up

⇩ RFM Measuring

Cluster	Members	Kind of buying	RFM value
Cluster1	C01,C04	Foods	(1,5,5)
Cluster2	C03	3C	(5,5,5)
Cluster3	C05,C04	Clothing	(5,5,3)
Cluster4	C02	office appliances	(5,2,3)

Fig. 2. Purchased-items-constrained *RFM*-based customer clustering.

Mooney, 2004; Ge, Jin, Wen, Ester, & Davidson, 2007; Wagstaff, Rogers, & Schroedl, 2001; Wong & Li, 2008; Zhang & Hau-San Wong, 2008), because they provide flexibility to attach user specified constraints while clustering. In general, the constraints can be classified into the following two categories.

- Vertical constraint: In this category the clustering methods focus on clustering customers on a portion of attributes of their transaction data sets, e.g., *pattern* clustering (Wong & Li, 2008), where a pattern is composed of some or all attributes which frequently occurs in a transaction data set. As patterns are clustered, the transactions containing these patterns are also clustered. The correlation between a pattern and transaction is straightforward (Wong & Li, 2008). It is noticeable that a pattern can not show the whole aspect of the actual data, so pattern clustering may produce confused results if inappropriate patterns are selected. For example, as shown in Fig. 3, there are 20 (T1 to T20) transactions. Assume two patterns P1 = {*A*,*B*,*C*,*D*} and P2 = {*E*,*F*,*G*,*H*} are merged into a cluster. The corresponding transactions are T5 to T14. We observe that the cluster should be split into two clusters if the similarity threshold is set to 3/4 so that one cluster corresponds to with transactions {*T*5,*T*6,*T*7,*T*8,*T*9,*T*10} and the other to {*T*11,*T*12,*T*13,*T*14} as shown in Fig. 4. The above problem becomes even severer and more time-consuming as the number of patterns increases.
- Horizontal constraint: In this category the clustering process focuses on a set of instance-level constraints. Instance-level constraints are a useful way to express a priori knowledge about which instances should or should not be clustered together (Basu et al., 2004; Wagstaff et al., 2001; Zhang & Hau-San Wong, 2008). There are two types of instance-level constraints:
  (i) Must link (ML): Let M be the set of must-link pairs; then $(x_i,x_j) \in M$ implies the instances x_i and x_j must be assigned to the same cluster.
  (ii) Cannot link (CL): Let C be the set of cannot-link pairs; then $(x_i,x_j) \in M$ implies the instances x_i and x_j should be assigned to different cluster.

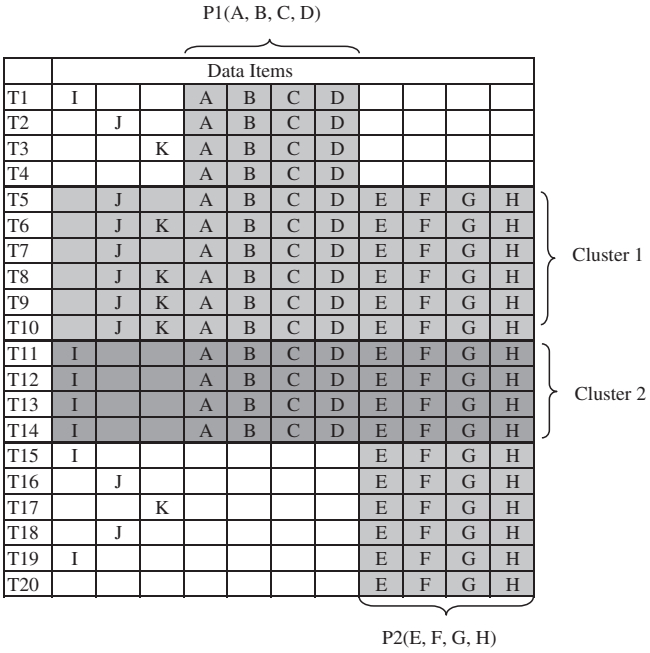


Fig. 4. Result of more adequate clustering.

In horizontal constraint clustering, a penalty weight is given to a clustering which violates a constraint (Basu et al., 2004). In fact, different constraints should have different penalty weights. However, the difference is not easy to be identified. Moreover, pair-wise constraint clustering can not be used when the constraints focus on partial characteristics between the pairs.

3.2. Concept hierarchy for purchased items

A large market-basket database may involve an extreme large volume of products, e.g., Amazon is an on-line shopping mall for

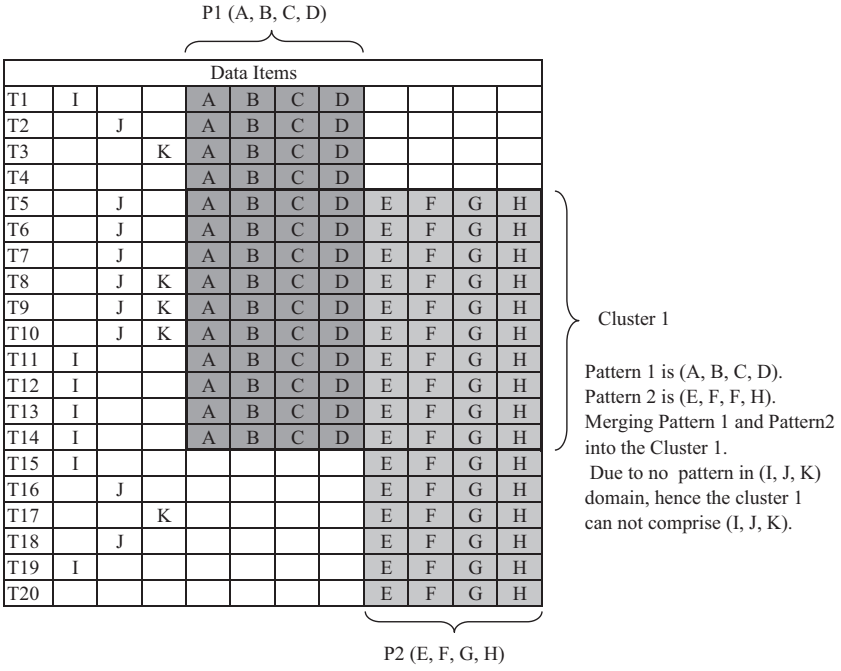


Fig. 3. The result of patterns clustering.

many books, apparel, electronics, etc. Usually, products are categorized such that a collection of subordinate products with similar characteristics are sorted into a super ordinate. A category hierarchy defines a sequence of mappings from a set of low-level product items to higher-level, more general category items. Therefore, data can be generalized by replacing its low-level characteristics, such as a product name, by their higher-level characteristics, such as a category in the category hierarchy (Han & Kamber, 2007). Fig. 5 shows an example five-level category hierarchy for computer products, starting with level 1 at the root (the most general abstraction level). Due to the sparseness of data and voluminousness of products, it is usually difficult to discover interesting purchase patterns at the lowest or primitive level. A trade-off is to analyze data from a higher level. In this work, we refer to the items at level i as items and items at level $i-1$ as categories.

3.3. Data notations

Table 1 lists the symbols and functions that will be used in the subsequent sections. The functions listed in the table are defined by the following equations:

$$VAL(\text{item}_i) = 2^{i-1}, \quad i \text{ is an index}, \tag{2}$$

$$F_T(d_i) = \sum_{j=1} VAL(\text{item}_{ij}) \tag{3}$$

$$F_A(IP_i, \text{ Constrain}_j) = (IP_i - IP_i \bmod 2^{start}) \setminus 2^{end+1} * 2^{end+l} + (IP_i \bmod 2^{start}) \tag{4}$$

A brief explanation about the symbols is in order. First, ***D*** represents a transaction dataset (or database) containing ***M*** categorical data records (or simply data) $\{d_1, d_2, d_3, \ldots, d_M\}$. Each d_i contains some data items. ***I*** represents the data item set that contains all the data items in ***D***. It can be grouped into k subsets SI_i, $i = 1, \ldots, k$, according to their properties. $I = \bigcup_{i=1}^{k}(SI_i)$, and $SI_i \cap SI_j = \emptyset$. Each item in ***I*** is assigned to a unique binary value. The items with the same property are assigned to consecutive binary values; or $SI_i = \{item_l, item_{l+1}, item_{l+2}, \ldots, item_{l+p}\}$ is mapped to $\{2^l, 2^{l+1}, 2^{l+2}, \ldots, 2^{l+p}\}$. With these mapped binary values for the data items, each d_i in ***D*** can then be transformed into an integer IP by transforming function F_T and is counted into $IP\text{-}num_i$. Finally, IP_i and $IP\text{-}num_i$ are stored in a dataset namely ORPA (ORiginal PAtterns).

Taking the dataset in Fig. 6 for example, we have the corresponding data item set $\{A,B,C,D,E,F,G,H,I,J,K\}$, which can be grouped into three subsets, each considered as a data and mapped to an integer. The data mapping table is shown in Table 2. For example, data $\{A,B,C,D\}$ is transformed into $2^0 + 2^1 + 2^2 + 2^3 = 15$. The mapping results of all data in the dataset are shown in Fig. 6. Finally, Table 3 shows the contents of ORPA after all IP and $IP\text{-}num$ are stored.

Sometimes, we want to perform customer clustering subject to a particular constraint like ignoring some type of purchased items. For example, we want to perform customer clustering according to all types of purchased items expect 3C products. Then before clustering, we need to temporarily eliminate the 3C products from each transaction record. For temporary elimination of some attributes, we "mask" them out from the dataset. Equivalently, we adjust each IP_i value by taking out the influence of the masked attributes. The adjusting function F_A is responsible for this. Take the same dataset for example. If the item set {E,F,G,H} is masked, then we need to deduct values of $2^4/2^7$ from each IP_i. Thus, each IP_i needs to be adjusted by the function: $F_A(IP_i, (4,7)) = (IP_i \setminus 2^8) * 2^8 + (IP_i \bmod 2^4)$. The adjusted results of all IP_i are shown in Fig. 7.

4. The GRFM framework

In this section, the GRFM analysis technique is described in detail. The basic framework is shown in Fig. 8, which shows three phases are involved in the GRFM process. The first phase performs data transformation and creates the ORPA table. It first transforms each transaction record in the transaction dataset into an integer. It then creates n ORPA table to store each integer and its occurrence frequency. In other words, ORPA stores the transformed integers corresponding to the original transaction records and their occurrence frequencies. The second phase follows to perform clustering over the ORPA table. To avoid destruction of ORPA, a copy of ORPA is stored as AT. If the user wants to perform constrained clustering, the constraints have to be placed in this phase along with the training instances. According to the constraints, each IP_i (i.e., each record) in AT will then be properly adjusted by F_A. The phase then performs constrained clustering over the new IP_i and produces a clustering result. Finally, the third phase calculates a (R, F, M) value for each customer in each cluster. Since a customer may belong to more than one cluster, a customer may be associated with different (R, F, M) values. The phase also uses the (R, F, M) values to build a cluster RFM cube, which is 3-dimensional as illustrated in Fig. 9. Each block of the cube records the customers who have the same (R, F, M) value. The cube can support a variety of analyses related to the customers' (R, F, M) values. For instance, it can quickly satisfy

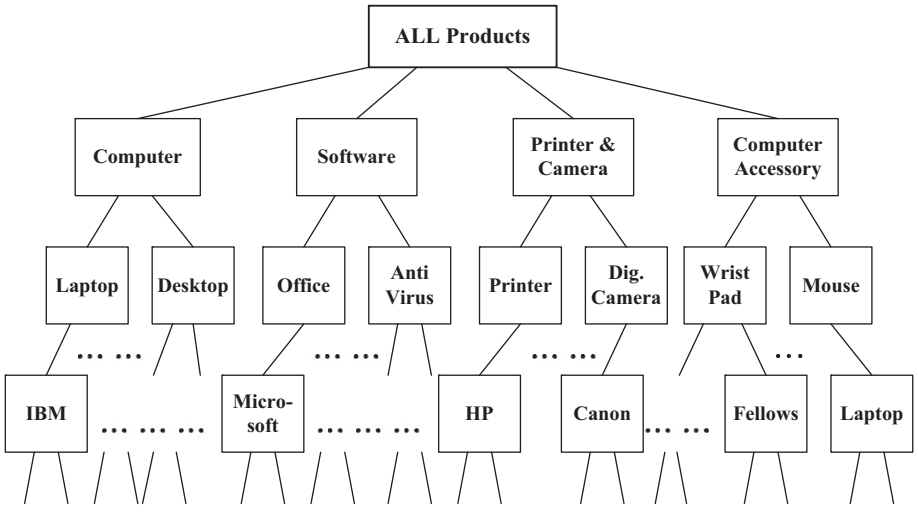


Fig. 5. Example concept hierarchy for computer products.

Table 1
Summary of the notations utilized in this paper.

D	It is original dataset
M	The number of data in
d_i	The ith data in D
I	The data item set of D
N	The number of I
$Item_i$	The ith item in I
SI	The subset of I.
$VAL(item_i)$	The mapping function is utilized to map $item_i$ to an integer
$F_I(d_i)$	The transforming function is utilized to transform d_i into an integer
IP_i	The ith data record in ORPA. (i.e., transformed from d_i)
$IP-num_i$	The number of IP_i
$Const_j$	It is a *pair (start, end)* to mask from $item_{start}$ to $item_{end}$
$F_A(IP_i, Const_j)$	The adjusting function is used to adjust IP_i according to $constrain_j$
$F_{count}(IP_i)$	The function is used to count the number "1" in the binary value of IP_i
$diff(C_i, IP_i)$	The function is used to transform $C_i \oplus IP_i$ into an integer
$Union(C_i, IP_i)$	The function is used to transform the union $C_i \vee IP_i$ into an integer
$Same(C_i, IP_i)$	The function is used to transform $C_i \wedge IP_i$ into an integer
$Dissim(C_i, IP_i)$	The measuring function is used to measure the dissimilarity between IP_i and center C_i

Table 2
Data mapping table.

$\{A,B,C,D\}$	$\{2^0,2^1,2^2,2^3\}$
$\{E,F,G,H\}$	$\{2^4,2^5,2^6,2^7\}$
$\{I,J,K\}$	$\{2^8,2^9,2^{10}\}$

Table 3
ORPA is created after data translation.

Integeral data (Data Pattern)	Number of integer
15	4
767	5
1535	5
240	6

the following user demand: making a sales promotion to low contribution customers, if low contribution is treated as M equal 1. Finally, in this phase, we divide the customers in each cluster into several groups according to an interval-gap set by the user. This allows us to output a distribution status of the member groups and provides further information about when to launch what promotion plans. We describe each phase in detail below.

I. Data transforming and creating ORPA phase

The algorithm in this phase is illustrated in Table 4. First, the purchased items are classified into k categories according to their properties, e.g., computer, cell phone, digital camera ... are belonged to 3C category. Each purchased item is then assigned to a unique binary exponential value, e.g., item i assigned to 2^i, as described before. Note that the items' values in same category are assigned to consecutive binary exponential values, i.e., 2^i, 2^{i+1}, 2^{i+2}... Now, each transaction can be transformed into an integer by summing the binary exponential values of the involved items. The integer is now equivalent to the content of the transaction.

The algorithm then generates an ORPA data table by storing each integer and its occurrence frequency. ORPA is the most important data structure in this framework. It is carefully designed to support the clustering requirement to be done in the next phase. First, it can be used for quick adjustment of its contents to represent new data patterns according to training instance change. This adjustment is equivalent to adjusting the original data, but it does not destroy the original data. Thus, it can be used to rapidly generate a variety of clustering results to meet different clustering requirements. Second, ORPA can be used to roughly estimate a cluster center according to the occurrence frequencies of the integers. This is because a datum with high frequency stands for a concentrated point; hence it could act as a cluster center. Finally, performing the Exclusive-OR operation over any two integers produces a result that can be used to indicate how similar the two corresponding data are. In fact, it also reveals where the two records are different.

II. Constrained clustering phase

The algorithm in this phase is illustrated in Table 5. In this phase, we employ PICC (Purchased-items-Constrained Clustering) as an algorithm for constrained data clustering. The user is prompted to put forward his training constraints. And we expect

Tid	Data Items											Mapped Integer	Total Records	
T1				A	B	C	D					15	Four Records	C1
T2				A	B	C	D					15		
T3				A	B	C	D					15		
T4				A	B	C	D					15		
T5		J		A	B	C	D	E	F	G	H	767	Five Records	C2
T6		J		A	B	C	D	E	F	G	H	767		
T7		J		A	B	C	D	E	F	G	H	767		
T8		J		A	B	C	D	E	F	G	H	767		
T9		J		A	B	C	D	E	F	G	H	767		
T10	I		K	A	B	C	D	E	F	G	H	1535	Five Records	
T11	I		K	A	B	C	D	E	F	G	H	1535		
T12	I		K	A	B	C	D	E	F	G	H	1535		
T13	I		K	A	B	C	D	E	F	G	H	1535		
T14	I		K	A	B	C	D	E	F	G	H	1535		
T15								E	F	G	H	240	Six Records	C3
T16								E	F	G	H	240		
T17								E	F	G	H	240		
T18								E	F	G	H	240		
T19								E	F	G	H	240		
T20								E	F	G	H	240		

Fig. 6. The mapping result of categorical data.

	Data Items											Mapped Integer	Total data
T1				A	B	C	D					15	Four data
T2				A	B	C	D					15	
T3				A	B	C	D					15	
T4				A	B	C	D					15	
T5		J		A	B	C	D	E	F	G	H	527	Five data
T6		J		A	B	C	D	E	F	G	H	527	
T7		J		A	B	C	D	E	F	G	H	527	
T8		J		A	B	C	D	E	F	G	H	527	
T9		J		A	B	C	D	E	F	G	H	527	
T10	I		K	A	B	C	D	E	F	G	H	1295	Five data
T11	I		K	A	B	C	D	E	F	G	H	1295	
T12	I		K	A	B	C	D	E	F	G	H	1295	
T13	I		K	A	B	C	D	E	F	G	H	1295	
T14	I		K	A	B	C	D	E	F	G	H	1295	
T15								E	F	G	H	0	
T16								E	F	G	H	0	
T17								E	F	G	H	0	
T18								E	F	G	H	0	
T19								E	F	G	H	0	
T20								E	F	G	H	0	

C1

C2

Fig. 7. The adjusting result of IP_i when mask {E, F, G, H}.

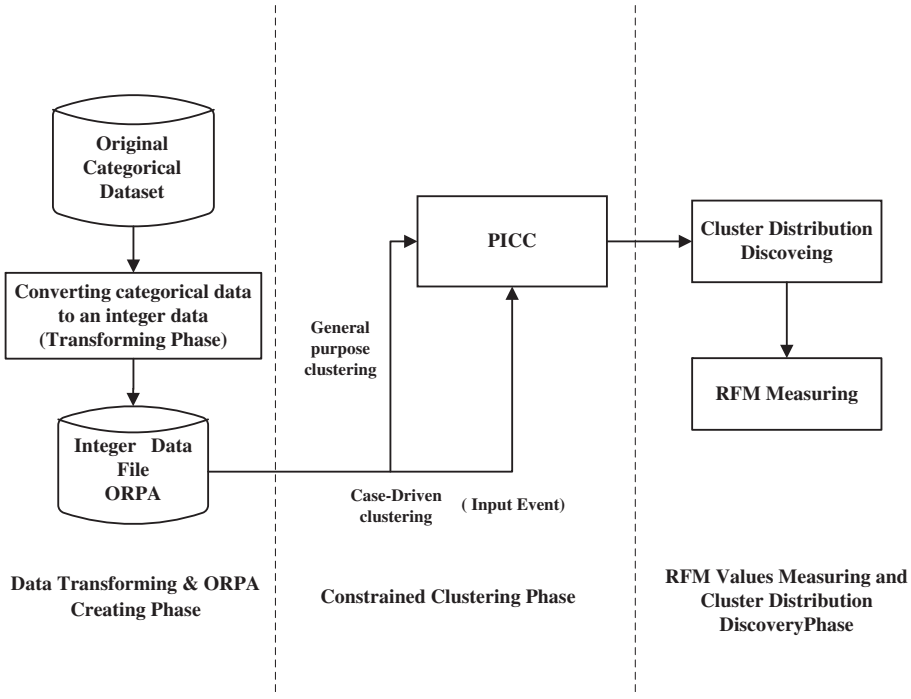


Fig. 8. The framework of GRFM.

the clustering result could satisfy the particular expectation of the user.

When a constraint triggers the related data records, the PICC starts to adjust the corresponding integers in ORPA by using the F_A function. It then uses a dissimilarity function ($Dissim(C_i, IP_i)$) to measure the distance between two transaction records in order to decide whether they should be allocated into the same cluster. Eq. (5) defines the dissimilarity function. If the function value of $Dissim(C_i, IP_i)$ is less than a predefined threshold, then the $cluster_i$ is a candidate cluster, otherwise $transaction_i$ is not clustered to C_i.

$$Dissim(C_i, IP_i) = \frac{F_{count}(diff(C_i, IP_i))}{F_{count}(union(C_i, IP_i))}. \quad (5)$$

Note: F_{count}(bin-data): The function is used to count the number "1" in the binary value.
$diff(C_i, IP_i)$: The function is used to transform $C_i \oplus IP_i$ into an integer.
$union(C_i, IP_i)$: The function is used to transform the union $C_i \vee IP_i$ into an integer.

In the equation, the *Dissim* function first performs the Exclusive-OR operation over the two integers (i.e., cluster center and transaction record) by function $diff(C_i, IP_i)$. The result is then converted to a binary value; the number of 1s contained in the binary value shows how different the two integers are. Moreover, the result of the *diff* operation also represents the difference of the contents between the two records, and therefore, given the same

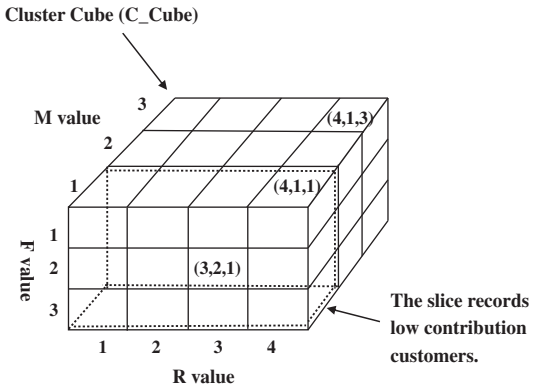


Fig. 9. The cluster cube.

Table 4
Data transforming and creating ORPA phase.

Data transforming and creating ORPA phase
Input: Purchasing data set Output: ORPA Table 1. Classify purchased item according to their properties; 2. Set a 2^n value to every purchased item according to its property; 3. do 4. {transforming purchasing record into IP_i by $F_I(3)$ function and counting appearance times of IP_i; 5. inserting (or updating) IP_i and counter to ORPA; 6. appending IP_i to purchasing record; 7. } 8. Sort up IP_i by its amount in ORPA; 9. end;

Table 5
Constrained clustering phase: PICC (Purchased items based constrained clustering).

Constrained clustering phase: PICC
Input: ORPA Output: Clusters 1. AT table is copied from ORPA 2. **If** (instance trigger) Then 3. {read IP_i and its amount from the AT; 4. adjusting the IP_i value by $F_A(3)$ function; 5. adjusted IP value restores in AT, and adjusted IP frequency is re-counted;} 6. Sort IP values by descending according to their frequencies in AT table ($SN_1 \cdots SN_{cc}$) 7. for i = 1 to cc 8. **if** $Dissim(C_j, SN_i) \leqslant sim - threshold$ **then** 9. {add C_j to candidate cluster set; 10. select appropriate cluster center from candidate cluster set which satisfied certain conditions; (Using function $diff(C_j, SN_i)$ or $Same(C_j, SN_i)$ set a judging criterion) 11. add SN_i into selected cluster; 12. } 13. else 14. create a new cluster C_j and set SN_i to center value of C_j; 15. next

dissimilarity, it can be regarded as a cluster selection mechanism for selecting a cluster from a set of candidate clusters. For example, the binary values assigned to the purchased items are by the decreasing order of their occurrence frequencies. Therefore the items with low purchasing frequencies will be assigned with higher binary values. Now if a transaction record has the same *Dissim* value against two candidate clusters, then we use the $diff(C_i, IP_i)$ function to calculate the differences between the transaction record and the two clusters. A bigger *diff* value here actually means the major difference between the record and the cluster center is over the low purchasing frequency items. And the record should be allocated to the cluster with a bigger *diff* value. Let us illustrate this process by the example illustrated in Fig. 10, which contains the twenty integers created from the sample data records. Suppose we set the threshold of dissimilarity to be less than 1/3, i.e., *dis_threshold* equals 1/3. In addition, we care about the dissimilarity over low purchasing frequency items. In this case, the cluster center with a higher *diff* value will be selected to be the cluster of a transaction record when more than one candidate clusters have the same dissimilarity against the transaction record as shown in Fig. 11. The following Case 1 summarizes this process.

Case 1: Under the same ***dissim*** *value, cluster center C_i with bigger* $\boldsymbol{diff(C_i, SN_j)}$ *is selected.*

1. Sort the integers in the descending order of their concurrence. Let $SN_1 = 287$; $SN_2 = 207$; $SN_3 = 399$; $SN_4 = 286$.
2. Define two clusters C1 and C2 using SN_1 and SN_2, where C1 = {287}, C2 = {207}, as they are ranked higher than the others. Set SN_1 and SN_2 as their respective cluster centers.
3. Calculate the dissimilarity of SN_3 against the two clusters. Since $dissim(C1, SN_3)$ and $diff(C2, SN_3)$ equal 2/7, so that SN_3 has same dissimilarity between C1 and C2. Hence, we calculation the $diff(C1, SN_3)$ value equals 144, and $diff(C2, SN_3)$ value equals 320. We thus can cluster SN_3 into C2, because $diff(C2, SN_3)$ is bigger than $diff(C1, SN_3)$.
4. Repeat the process for SN_4. We have $diff(C1, SN_4)$ equals 1 and $dissim(C1, SN_4)$ equals 1/6. Also $diff(C2, SN_4)$ equals 465 and $dissim(C2, SN_4)$ equals 5/8. Accordingly, SN_4 is clustered to C1. Now we have new C1: {287,286}, and C2: {207,399} and shown as Fig. 11.

As a matter of fact, PICC also use the *same* (C_i, IP_i) function (as shown in Table 1) to calculate the degree of sameness between the transaction record and the cluster. The function performs the AND operation on two integers to produce a binary value, the positions of 1s of which show where they are the same. The AND result can then be treated as a cohesion degree between a transaction record and cluster. Accordingly, we can constrain what features are necessary in a cluster by this mechanism. The AND result can be regarded as a must-link constraint. The following Case 2 illustrates how the constraint is used in clustering.

Tid	Data Items									Mapped Integer	
T1	A	B	C	D	E				I	287	SN_1
T2	A	B	C	D	E				I	287	
T3	A	B	C	D	E				I	287	
T4	A	B	C	D	E				I	287	
T5	A	B	C	D	E				I	287	
T6	A	B	C	D	E				I	287	
T7	A	B	C	D	E				I	287	
T8	A	B	C	D			G	H		207	SN_2
T9	A	B	C	D			G	H		207	
T10	A	B	C	D			G	H		207	
T11	A	B	C	D			G	H		207	
T12	A	B	C	D			G	H		207	
T13	A	B	C	D			G	H		207	
T14	A	B	C	D				H	I	399	SN_3
T15	A	B	C	D				H	I	399	
T16	A	B	C	D				H	I	399	
T17	A	B	C	D				H	I	399	
T18		B	C	D	E				I	286	SN_4
T19		B	C	D	E				I	286	
T20		B	C	D	E				I	286	

A value is 1. B value is 2. C value is 4. D value is 8.
E value is 16. F value is 32. G value is 4. H value is 128.
I value is 256.

Fig. 10. Example dataset for explaining the PICC clustering process.

Tid	Data Items									Mapped Integer
T1	A	B	C	D	E	X	X	X	I	287
T2	A	B	C	D	E	X	X	X	I	287
T3	A	B	C	D	E	X	X	X	I	287
T4	A	B	C	D	E	X	X	X	I	287
T5	A	B	C	D	E	X	X	X	I	287
T6	A	B	C	D	E	X	X	X	I	287
T7	A	B	C	D	E	X	X	X	I	287
T8	A	B	C	D	X	X	G	H	X	207
T9	A	B	C	D	X	X	G	H	X	207
T10	A	B	C	D	X	X	G	H	X	207
T11	A	B	C	D	X	X	G	H	X	207
T12	A	B	C	D	X	X	G	H	X	207
T13	A	B	C	D	X	X	G	H	X	207
T14	A	B	C	D	X	X	X	H	I	399
T15	A	B	C	D	X	X	X	H	I	399
T16	A	B	C	D	X	X	X	H	I	399
T17	A	B	C	D	X	X	X	H	I	399
T18	X	B	C	D	E	X	X	X	I	286
T19	X	B	C	D	E	X	X	X	I	286
T20	X	B	C	D	E	X	X	X	I	286

Fig. 11. Clustering result of Case 1. Under the same *dissim* value, bigger *diff* is selected.

Case 2: Transaction records that contain {A, B, C, D, I} must be grouped in the same cluster, i.e., since {A, B, C, D, I} is transformed to 271, so that the value of $\textbf{same}(C_i, IP_i) \wedge$ *'271')need to be at least 271 for the record to be allocated to the same cluster.*

1. The clusters are the same as the previous case, i.e., C1 = {287}, C2 = {207}, and the respective center values are 287 and 207.
2. Calculate the coherence degrees for SN_3 against the two clusters. Since *same*(C1, SN_3) equals 271, and (271 $\wedge$ 271) equals 271. Therefore, SN_3 is clustered to C1. We skip the calculation against C2 because C2 is smaller than 270, i.e., C2 does not involve the constraint.
3. Repeat the process for SN3. We have *same*(C1, SN_4) equals 286, and (286 $\wedge$ 271) equals 270. Thus, SN_4 can not be clustered because it does not satisfy the constraint. The result is shown in Fig. 12.

III. Cluster distribution Discovering and RFM Measuring Phase

As noted before, this phase contains two tasks inside each cluster, namely, measuring GRFM-values for the customers as well as discovering the cluster distribution status. In order to do this, we propose to employ a new cluster structure to capture relevant information. Each cluster structure, as illustrated in Table 6, contains two parts. The first part contains the features of a cluster, including the cluster center, group amount, *R* (the last period of purchase), *AF* (the average frequency of purchase in periods), *M* (the average expenditure over all the members in the cluster), and Period Amount (the number of periods in the cluster). The second part records all the member groups (to be clear later) in a cluster. Each cluster contains at least one member group, which comprises Start Id, End Id, and the amount of members in the group. Start Id and End Id are used to record the first and the last transaction Ids in a member group. This cluster structure therefore can support the measurement of (*R*, *F*, *M*) values we well as the calculation of distribution status. The first task is shown in Table 7. First, we treat each (*R*, *F*, *M*) value as a point in the 3-dimensional space with *R*, *F*, and *M* as the coordinate axes, respectively. The user is asked to input how the three axes ought to be labelled or partitioned (i.e., into how many partitions) according to his professional knowledge. The system then applies Chebyehev's inequality to the information of *R*, *AF*, and *M* values inside the cluster and calculates the value range for each partition of each axis. The user is allowed

Tid	Data Items									Mapped Integer
T1	A	B	C	D	E				I	287
T2	A	B	C	D	E				I	287
T3	A	B	C	D	E				I	287
T4	A	B	C	D	E				I	287
T5	A	B	C	D	E				I	287
T6	A	B	C	D	E				I	287
T7	A	B	C	D	E				I	287
T8	A	B	C	D			G	H		207
T9	A	B	C	D			G	H		207
T10	A	B	C	D			G	H		207
T11	A	B	C	D			G	H		207
T12	A	B	C	D			G	H		207
T13	A	B	C	D			G	H		207
T14	A	B	C	D				H	I	399
T15	A	B	C	D				H	I	399
T16	A	B	C	D				H	I	399
T17	A	B	C	D				H	I	399
T18		B	C	D	E				I	286
T19		B	C	D	E				I	286
T20		B	C	D	E				I	286

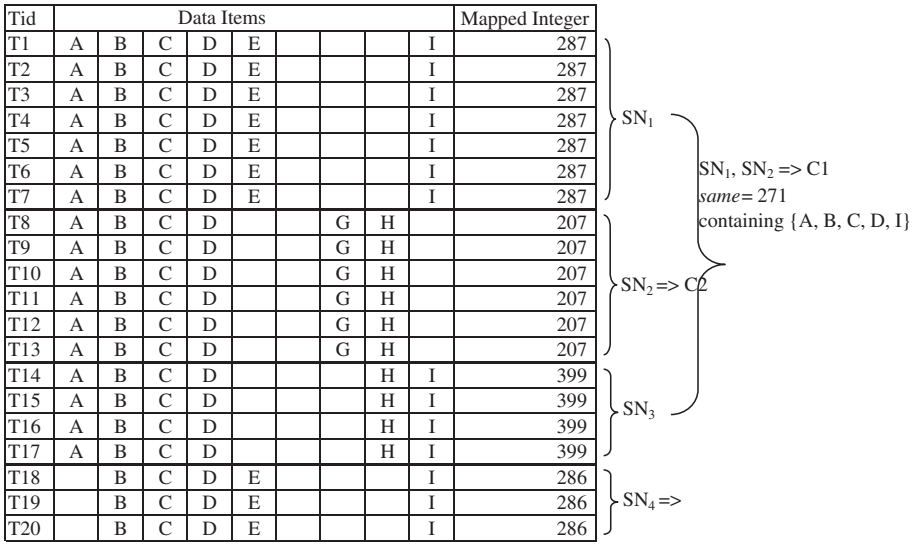


Fig. 12. Clustering result of Case 2. Transaction data has to involve particular items.

Table 6
Cluster structure.

Part	Features
Part I	1.1 Cluster center 1.2 R: It refers the period of last member group appearing 1.3 AF: It refers to average number of purchasing times in a period 1.4 M: It refers to average expenditure of all members in a cluster 1.5 Period Amount: It is indicated how many periods in this cluster
Part II	The part contains at least one member group record Each member group record include: 2.1 Start-ID: The transaction ID of the first member in this group 2.2 End-ID: The transaction ID of the last member at the moment in the group 2.3 Amount: It is numbers of member in the group

Table 7
Measureing the (*R*, *F*, *M*) values for the customers in a cluster.

Cluster distribution discovering and *RFM* measuring phase: *RFM* measuring
1. The user inputs *r*, *f*, and *m*, the number of partitions for *R*, *F* and *M*, respectively, for the given cluster
2. Apply Chebyehev's inequality to compute the range values of each partition for *R*, *F*, and *M*
3. Solicit the user to fine tune the calculated range values
4. Use the range values to measure a 3-dimensional (*R*, *F*, *M*) value for the customers in the cluster
5. According to customers' (*R*, *F*, *M*) value to create a cluster cube

to fine tune the range values. These range values work as the basis for the GRFM to measure the customer's (*R*, *F*, *M*) values inside the cluster. As a matter of fact, these values could be used to create cluster cube for online analysis of customers' behavior. The cube is 3-dimension array to store the *RFM* records. Each record contains two fields; one field records the number of customers with the same GRFM-value, the other makes them into a linked-list. For example, the consumption recency, frequency, and monetary of customers C1, C2, C19 are measured according to user definition values rang for *R*, *F*, and *M* values. Three customers' (*R*, *F*, *M*) values are all (4,1,3), so that, the location (4,1,3) of cluster cube is recodes 3 customer and using linked-list structure link them. The result is shown in Fig. 13.

The second task, discovering cluster distribution status, is illustrated in Table 8. By cluster distribution status, we mean how the transactions behave with respect to the time series in the cluster. To do this, we divide the members of a cluster into member groups according to a pre-defined time interval-gap (or simply interval-gap). The interval-gap is an interval between the consumption

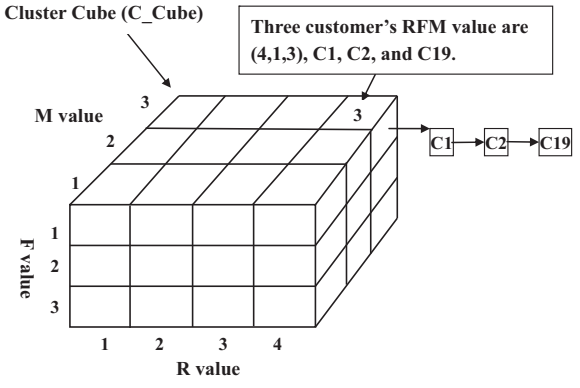


Fig. 13. Cluster cube structure example.

Table 8
Cluster distribution discovery.

Cluster distribution discovering and *RFM* measuring phase: Cluster distribution discovering
Definition: (1) $cgroup_i$ is the number of groups in the i cluster (2) $recency\_of\_cluster_i$ is the period of last member group appearing (3) $frequency\_of\_cluster_i$ is the average member in a particular period (4) $monetary\_of\_cluster_i$ is the average expenditure of all members in a cluster (5) $cgroup\_start_{i,j}$ is the start member in the j member group of the i cluster (6) $cgroup\_end_{i,j}$ is the last member in the j member group of the i cluster (7) $cgroup\_amount_{i,j}$ is the number of the member in the j member group of the i cluster

Input: interval_gap as standard for division cluster
1. each data record is dispatched to belonged cluster according to its IP_i value;
2. do {
3. $cgroup_i$=0, amount_of_money = 0, amount_of_money = 0;
4. while (not end of a cluster) {
5. add a member;
6. If (a member ID − $cgroup\_end_{i,j}$) > interval_gap **then** {
7. create a new group;
8. $cgroup_i$=$cgroup_i$+1;
9. set member ID as $cgroup\_start_{i,j+1}$ and $cgroup\_end_{i,j+1}$;
10. $cgroup\_amount_{i,j+1}$ = 1;}
11. **Else** {
12. $cgroup\_end_{i,j}$ = member ID;
13. $cgroup\_amount_{i,j+1}$ = $cgroup\_amout_{i,j+1}$ + 1;}
14. record $recency\_of\_cluster_i$, $frequency\_of\_cluster_i$ and $monetary\_of\_cluster_i$;
15. } while (has cluster)
16. next
17. output all cluster;

times of two transactions. If the time gap between two transactions is larger than the interval-gap, which implies the consumption is not continuous, then it can be split into two groups. Therefore, we can discover different purchase periods inside a cluster by graphing the appearing times of the member groups. In order to generate the distribution status of the member groups, we use Eq. (6) to compute a *Cluster-Distribution* value. If the *Cluster-Distribution* value is high then the purchased behavior is somewhat fluctuated; on the contrary, it is relatively uniform. Note that the Cluster-Distribution value is not sufficient to outline the marketing status. We have to compute the density of each member group to discover significant purchase periods in the cluster. We use Eq. (7) to compute the *Density-of-Member-Group* value. If the density is high then the period represents the hot time of marketing; for example, the period is hot marketing time during sale. In other words, we can discover the most important marketing period for each product by the member group's density.

$$Cluster-Distribution(i) = \frac{number\ of\ member\ groups}{user-defined-period} \quad (6)$$

Note: *Cluster-Distribution (i)* is the *i*th cluster's distribution.

$$Density\text{-}of\text{-}Member\text{-}Group(i,j) = \frac{Amount}{(End-ID_{(j)}) - (Start-ID_{(j)})} \quad (7)$$

Note: The *Density-of-Member-Group (i,j)* is the *j*th member group in the *i*th cluster.

In summary, a cluster in GRFM provides lots of information, including the cluster center, member groups in the cluster, (*R*, *F*, *M*) values, distribution status and density. The (*R*, *F*, *M*) values can be used by the managers to measure the loyalty and contributions of a customer cluster, and accordingly propose better marketing strategies. The distribution status and density of the clusters

can be used by the managers to propose better product promotion plans and inventory management strategies.

5. Experiments

5.1. Experimental results

In our experiment, the samples of purchase data are randomly generated by the generating program as described in Agrawal and Srikant (1994). There are 20000 transactions that are randomly assigned to 1000 customers. This forms the transaction dataset for training. Each transaction in the dataset contains a customer number (Cid), a transaction number (Tid), purchased items, and monetary. The dataset is then clustered using PICC with respect to the purchased items; i.e., the customers are clustered by their purchased items. We obtain 193 purchase clusters, each containing several customers. As expected, each customer may belong to more than one cluster.

In the first experiment, the interval-gap is set to 500 (unit: transactions), meaning the gap between two transactions in a cluster must be at most 500 transactions. In other words, transactions are treated as the same member set if they are not separated by 500 transactions. Each cluster therefore can have its own distribution status about transactions. After analyzing the distribution status of the clusters, we discovered that a cluster may belong to one of the three cluster purchase behavior types as illustrated in Table 9. For example, a customer belongs to a cyclic purchase cluster, if his purchasing behavior tends to be periodical. Based upon this information, the manager could periodically contact with the customers to improve customer relationships. In fact, the manager could base on the information to build a personalized purchase management system for customers. As for the intermittent purchase behavior cluster, we can use the density of the member group to pinpoint the hot periods for marketing. The manager could then base on this information to build a desirable inventory management system to reduce the risk of over-stock. In Table 10, we show some cluster distribution status. Note that we measure each customer's (R, F, M) value in a cluster according to the center values of R, AF, and M. And every cluster has different numbers of member groups. Since a customer may belong to more than one cluster, he has different (R, F, M) values in different clusters. In order for comparison, we also measure the customers' (R, F, M) values according to Miglautsch's approach (Miglautsch, 2000). Table 11 then lists the measuring results of GRFM and Miglautsch's approach. The table shows GRFM can make better evaluation about customers' (R, F, M) values. In Table 12, we list some interesting factors that affect the customer's (R, F, M) values.

These tables demonstrate that customers tend to purchase items with different features. If an enterprise measures a customer's (R, F, M) value solely according to his consumption time point, consumption frequency, and consumption money, it is clear that it could not make a true appreciation of the his loyalty and contribution. In contrast, GRFM clusters customers according to their purchased items. Thus a customer may be allocated into more than one cluster and therefore assumes different (R, F, M) values in different clusters. Inside a cluster, when a customer is compared with the others with respect to loyalty, the comparison is based on the same purchased items. By this, GRFM can better reveal the actual consumption behaviors of the customers.

Finally, we compare the performance of PICC with the K-means extension clustering method (extended-K-Means)(Huang, 1998)

Table 9
Purchase Cluster Types and Counts (In total 193 clusters with interval_gap of 400 transactions).

Cluster Type	Description of clusters	Amount
Consecutive clusters	The cluster contains only one member group	106 clusters
Intermittent cluster	The member groups are neither permanent nor cyclic cluster	84 clusters
Cyclic cluster	The member groups are cyclic appearance	3 clusters

Table 10
Cluster Purchase Distribution (193 clusters in total with interval-gap set to 500).

Cluster ID	The last period	Average buying times	Periods Amount	Remark
C193	(19072,19461)	84	2 periods	It is an intermittent cluster and the consumption time is centralized
C162	(18614,19257)	51	17 periods	It is an intermittent cluster and the consumption time is not centralized
C1	(19517,19978)	265	3 periods	It is an intermittent cluster and the consumption time is not only centralized but highly dense

Table 11
Comparison of GRFM and Miglautsch's approach in measuring *RFM* values.

Customer Id	*RFM* value by GRFM	*RFM* Value by Miglautsch's approach	Customer characteristics	Promotion policy
Cust_46	5/5/5	5/4/2	The *M* value is different between GRFM and Miglautsch's approach. Although the customer is used to purchase low priced products, inside that cluster, the customer has high contribution and loyalty. Miglautsch's approach misinterprets the customer to be a medium contribution customer	The business should not only keep the customer, but should attract the customer to purchase other products via proper promotion policy
Cust_105	2/3/3 in C_A; 5/3/4 in C_B	5/3/4	In GRFM, the customer belongs to two clusters; i.e., he has different purchase behaviors over different products. In addition, we discover the customer has a change on his purchased products. However, these occurrences can not be discovered by Miglautsch's approach	The business should make more communication with the customer to realize the reason why the customer changes his purchasing behavior
Cust_133	3/4/4	1/1/2	The *R* and *F* values are very different between GRFM and Miglautsch's approach, because the customer is used to purchase products that are purchased infrequently. From the viewpoint of infrequent purchased products,the customer is loyal customer. Miglautsch's approach can not discover this potential loyal customer	Although the customer is used to purchase rare products, to which,he is a loyal customer. Therefore, the business should make more communication with the customer to promote correlative products

Table 12
Factors that Affect customer's *RFM* values and their influences.

Factor	Influence
Price of purchasing items	The *M* value is big if the price of purchasing item is high. *M* value is small, otherwise
Lifetime of purchasing items	The *F* value is big if the lifetime of purchasing item is short or seasonal. *F* value is small, otherwise
New or old customers	The *F* is small and *R* is big if the customer is new customer

and the initial-points refinement *K*-modes clustering method (Ref-*K*-Modes) (Sun, Zhu, and Chen, 2002) in terms of execution time. In addition to execution time, we also use "Scaling" as a criterion for performance comparison. Eq. (8) first defines "Val" to be the degree of how all members in a cluster are close to the cluster center by calculating the average of similar degrees between all transaction records in a cluster and the cluster center. If Val is large, then the cluster's members are close to the cluster center. "Scaling" is then defined in Eq. (9) to be the sum of the "Val" values over all clusters. Therefore, it represents how good of the clustering method in terms of how similar the members are in all clusters.

In our experiments, the samples are randomly generated by the same generator as mentioned before (Agrawal and Srikant, 1994). There are three samples having 10000, 15000, 20000, 25000 and 30000 records, respectively. The results of the comparison are illustrated in Figs. 14 and 15. In Fig. 14, we find the execution time of PICC is less than the other algorithms. Moreover, the execution time of PICC rises slowly with the increasing number of transaction records. In Fig. 15, we find the PICC algorithm has the highest Scaling value in the three algorithms. Finally, the execution times of clustering and re-clustering of PICC are very shot as illustrated in Fig. 16. We would like to point out two more merits of PICC. First, it does not require a cluster number in advance. Second, it allows the setting of constraints for the clustering process when it is asked to ignore some particular products.

$$Val_i = \sum_{j=1}^{m} \frac{\bigcap(CM_i, CR_{i,j})}{CM_{(i)}.count} \tag{8}$$

$$Scaling = \sum_{i=1}^{cn} Val_i \tag{9}$$

Note: $CM_{(i)}$ is the ith cluster center.
$CM_{(i)}.count$ is the number of data items in the ith cluster center.
$CR_{(i,j)}$ is the jth member in the ith cluster.
m is the number of members in a cluster.
cn is the number of clusters.

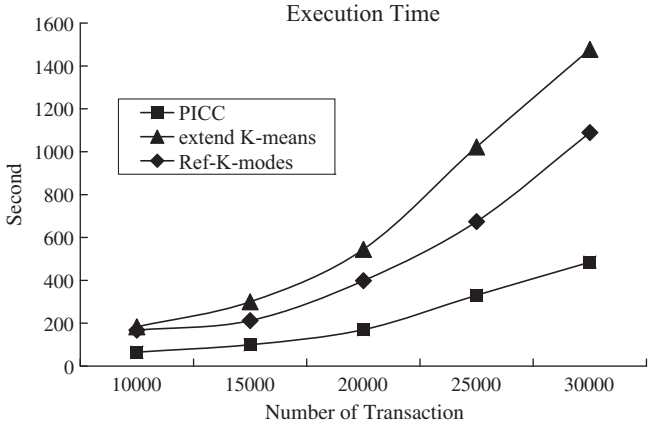


Fig. 14. Comparisons of PICC with extension *K*-means and Ref-*K*-modes with respect to execution time.

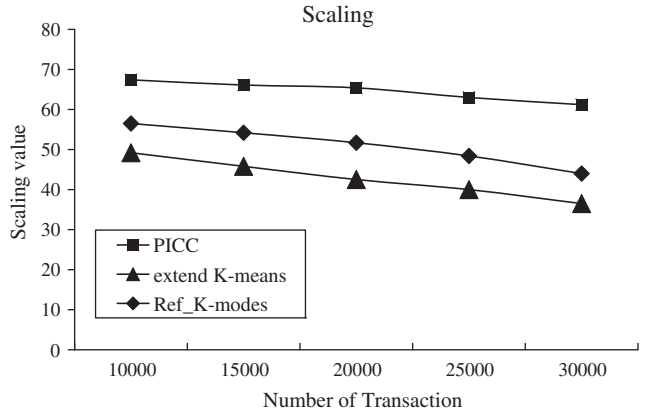


Fig. 15. Comparisons of PICC with extension *K*-means and Ref-*K*-modes with respect to scaling value.

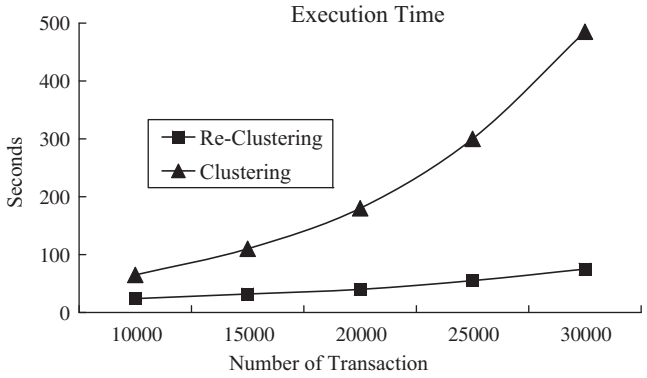


Fig. 16. Comparisons of re-clustering with clustering with respect to execution time.

The experimental results, comparing the PICC with extension *K*-means and refinement initial-points *K*-modes in execution time and Scaling value are illustrated in Figs. 11 and 12. We would like to emphasize that PICC does not require a cluster number in advance.

5.2. Discussion

We summarize four major features of the GRFM framework below:

1. The GRFM framework does not calculate a single (*R*, *F*, *M*) value for a customer. Rather it associates different (*R*, *F*, *M*) values with a customer according to the properties of his purchased items. It thus can better reveal the true purchasing behavior of a customer. In addition, GRFM creates a cluster *RFM* cube for each cluster according the customers' (*R*, *F*, *M*) values in the cluster. These *RFM* cubes not only can support the traditional *RFM* analysis as discussed in Miglautsch's approach (Miglautsch, 2000). but also proposes new analyses. In Table 13, we summary the differences of *RFM*-based analysis between GFRM and (Miglautsch, 2000). Based upon this information, the manager could properly contact with the customers to improve customer relationships and build a personalized purchasing management system for customers.

Table 13
Differences over *RFM*-based analysis between GRFM and Miglautsch's approach.

Analysis requirement	GRFM	Miglautsch's approach
Looking for customers with high contribution and loyalty over some particular products	GRFM can extract the customers with high contribution and loyalty from the cluster cubes of the particular products	Miglautsch's approach can not provide correlative information about business demand
Looking for customers of high loyalty	GRFM can extract and integrate the highly loyal customers from all cluster cubes, among which potentially highly loyal customers can be further discovered	Miglautsch's approach can provide the information but it can not discover potentially highly loyal customers

2. The GRFM framework provides sales information for each purchase cluster, which is clustered with respect to the properties of the purchased items. Base upon the information, the user could obtain integrated sales information, e.g., which purchase cluster is highly loyal and profitable, or which purchase cluster has a potentially high volume of sales. For example, from Table 10, we observed that the members of cluster 1 (C1) are rather centralized in each period; in other words, the purchasing time is very fixed. On the contrary, cluster 162 often appears, while its members are scattered in each period. This information can be re-analyzed by the manager to extract important hidden information. Therefore, the manager could base on this information to build a desirable inventory management system to reduce the risk of over-stock. Note that this information cannot be acquired from traditional *RFM* analysis paradigms.
3. According to Fig. 14, the slope of execution time for the PICC algorithm is less than the other algorithms. The execution time of PICC rises slowly with the increasing amount of data, but the execution time for the other algorithms changes abruptly. Although the *K*-means extension algorithm uses a frequency-based method to update modes (i.e., the means of clusters), it still requires an unknown number of iterations before converging to a good solution. However, PICC has a higher Scaling value than the others algorithms in Fig. 15, which implies PICC could lead the data to converge to a more optimal solution.
4. According to Fig. 16, PICC uses and reuses the comparatively succinct purchase pattern table ORPA to perform clustering to meet different purposes of training. Since PICC does not directly use the original data file for processing, it can perform clustering more rapidly.

5.3. Other application

As a matter of fact, the GRFM framework can be applied in other fields. For example, we can use the framework to cluster students according to their learning styles. As the research of group learning indicates that group learning could be beneficial in students learning (Zheng, Ding, and Tian, 2007). When students with the same learning style are put together for problem solving, they could rapidly generate a variety of possible solutions to solve the problem. However, best learning styles are usually different for different subjects. Thus, an instructor needs to cluster the students by their learning styles according to the requirements of different subjects. Take the Felder–Silverman learning style for example. It defines four aspects of learning, namely, Perception (sensing/intuitive), Input (visual/verbal), Organization (inductive/deductive), and Understanding (sequential/global) (Felder and Silverman, 1988). An instructor thus needs a mechanism to focus on the four aspects while students are being clustered. The PICC algorithm could work as such a mechanism so that the instructor can properly set his constraints and perform constrained clustering. The GRFM framework then can be used to measure the students' learning power with different learning styles. In this case, the *R* value can be defined as the interval from the time when the latest log-in happens to the present; the *F* value can represent the number of log-ins within a certain period; and the *M* value can represent the amount of log-in time within a certain period. Now, the instructor could discover whether some kind of learning styles of the students has more learning power or not. This could effectively help the instructor to develop better teaching strategies.

6. Conclusions

We have described GRFM as a framework to perform purchased items-constrained clustering so as to deeply analyze and utilize the *RFM* value of the customer. It supports cluster analysis from both aspects of customers and their purchased items. Since the analysis takes into account the purchase items, the (*R*, *F*, *M*) values could reveal the true purchasing behavior. GRFM is the same as the traditional *RFM* analysis in the sense that each cluster has the same loyalty and contribution. They are very different in that GRFM allows a customer to belong to different clusters, and thus to be associated with different loyalties and contributions with respect to different characteristics of purchased items. This difference allows GRFM to correctly discover the sales trend for the purchased items. It also facilitates the development of a better personalized purchasing system as well as a desirable inventory management system. Moreover, GRFM provides a clustering method that could reuse original purchase patterns to promptly respond to the market-oriented demands. It converts the original data into corresponding integers and stored them in the ORPA table, which can then be quickly and conveniently adjusted to reflect new types of data patterns. It is equivalent to adjusting original data, but it does not destroy the original data. Therefore, the ORPA table could be reused to satisfy various constraints and reduce clustering time.

References

Agrawal, & Srikant, R. (1994). Fast algorithms for mining association rules. In *Proceedings of the 20th international conference on very large data bases*.

Basu, S., Banerjee, A. & Mooney, R.J. (2004). Active semi-supervision for pairwise constrained clustering. In *Proceedings of the SIAM international conference on data mining*.

Cheng, C.-H., & Chen, Y.-S. (2009). Classifying the segmentation of customer value via rfm model and rs theory. *Expert Systems with Applications*, 4176–4184.

Felder, R., & Silverman, L. (1988). Learning and teaching styles in engineering education. *Engineering Education*.

Ge, R., Jin, Wen, Ester, M. & Davidson, I. (2007). Constraint-driven clustering. In *Proceedings of the 13th ACM KDD international conference on Knowledge discovery in data*.

Han, J., & Kamber, M. (2007). Data Mining Concept and Techniques. *Diane Corra*.

Huang, Z. (1998). Extension to the *k*-means algorithm for clustering large data sets with categorical value. In *Proceedings of the Fourth ACM SIGKDD international conference on knowledge discovery and data mining*.

Hughes, A. (1994). *Strategic database marketing*. Chicago: Probus Publishing Company.

Miglautsch, J. (2000). Thoughts on rfm scoring. *Journal of Database Marketing*, 67–72.

Newell, F. (1997). *The new rules of marketing: How to use one-to-one relationship marketing to be the leader in your industry*. New York: McGraw-Hill.

Management Science, (2003). A comparative research on the methods of customer segmentation based on consumption behavior.

Sun, Y., Zhu, Q., & Chen, Z. (2002). An iterative initial-points refinement algorithm for categorical data clustering. *Pattern Recognition Letters*.

Tsai, C.-Y., & Chiu, C.-C. (2004). A purchase-based market segmentation methodology. *Expert Systems with Applications*.

Wagstaff, K., Rogers, S., & Schroedl, S. (2001). Constrainted *k*-means clustering with background knowledge. In *Proceedings of the 18th international conference on machine learning*.

Wong, K., & Li, C. (2008). Simultaneous pattern and data clustering for pattern cluster analysis. *IEEE Transactions On Knowledge and Data Engineering*, 911–923.

Wu, J., & Lin, Z. (2005). Research on customer segmentation model by clustering. In *Proceedings of the 7th ACM ICEC international conference on electronic commerce*.

Yeh, I.-C., Yang, King-Jung, & Ting, T.-M. (2008). Knowledge discovery on rfm model using bernoulli sequence. *Expert Systems with Application*, 5866–5871.

Zhang, S., & Hau-San, Wong. (2008). Partial closure-based constrained clustering with order ranking. In *Proceeding of 19th international conference on pattern recognition*.

Zheng, Q., Ding, J. & Tian, F. (2007). Assessing method for e-learner clustering. In *Proceedings of the 11th conference on computer supported cooperative work in design*.