Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Behavioral rules of bank's point-of-sale for segments description and scoring prediction

Mehdi Bizhani[a] and Mohammad Jafar Tarokh[a*]

[a]*Department of Industrial Engineering, K.N. Toosi University of Technology,Tehran, Iran*

A R T I C L E I N F O

Article history:
Received 1 March 2010
Received in revised form
1 July 2010
Accepted 10 July 2010
Available online 10 July 2010

Keywords:
Banking industry
RFM scoring
Merchant segmentation
Behavioral rule

A B S T R A C T

One of the important factors for the success of a bank industry is to monitor their customers' behavior and their point-of-sale (POS). The bank needs to know its merchants' behavior to find interesting ones to attract more transactions which results in the growth of its income and assets. The recency, frequency and monetary (RFM) analysis is a famous approach for extracting behavior of customers and is a basis for marketing and customer relationship management (CRM), but it is not aligned enough for banking context. Introducing RF^*M^* in this article results in a better understanding of groups of merchants. Another artifact of RF^*M^* is RF^*M^* scoring which is applied in two ways, preprocessing the POSs and assigning behavioral meaningful labels to the merchants' segments. The class labels and the RF^*M^* parameters are entered into a rule-based classification algorithm to achieve descriptive rules of the clusters. These descriptive rules outlined the boundaries of RF^*M^* parameters for each cluster. Since the rules are generated by a classification algorithm, they can also be applied for predicting the behavioral label and scoring of the upcoming POSs. These rules are called behavioral rules.

1. Introduction

Electronic payment systems are now widely spread among customers and merchants (shop owners) using debit, credit or gift cards and it can substantially reduce the social cost of a country's payment system (Humphrey, et al., 2001). For every transaction, the bank charges the merchant with fee (Kotler, et al., 2005), which is one way of generating revenue in retail banking (Garland, 2002). After a successful transaction, the amount is subtracted from the customer's account, and in the settlement process the fee is added to a special bank's account and the remained money is deposited to the merchant's account. Therefore, the bank profits in three ways from POS transactions: first it takes fees from merchants and raises its income. Second, the merchant's account is deposited which results to the bank's assets growth and third, the settlement process may take some time, based on the bank policy and it usually takes one day. During this time, the money is subtracted from customers account, but it is not deposited in merchants' account, so the bank can use it without paying any interest.

To gain more profit and reduce cost, bank needs to monitor its current POSs and new upcoming POSs to retain profitable POSs, prevent churn and persuade inactive to active ones. To achieve these goals, it is necessary to know the current POSs status and group them, properly. One of the applications of clustering is data reduction (Halkidi, et al., 2001), which is applied when the number of data items are

* Corresponding author. +98 912 3844762, Fax: +9821 22500163
E-mail addresses: mjtarokh@kntu.ac.ir (M. J. Tarokh),

doi: 10.5267/j.ijiec.2010.04.002

very large and their processing becomes very difficult, so instead of processing the entire data set, only the clusters' representatives of the defined clusters are considered. Some related works are summarized in Table 1.

Table 1

Related works of applying clustering as data reduction technique

Title	Description	Reference
An efficient approach for building customer profiles from business data	• Clustering data items in order to extract "natural" groups of customers • Selecting the most important attributes for each group • Building a set of customer profiles based on the group of customers	(Romdhane, et al., 2010)
Classifying the segmentation of customer value via RFM model and RS theory	• Using RFM model to produce quantitative value as input attributes • Clustering the customer value • Mining classification rules that help enterprises driving an excellent CRM	(Cheng & Chen, 2009)
Applying knowledge engineering techniques to customer analysis in the service industry	• Clustering customers' RFM values over five time periods • Assigning low/high to each cluster based on its centroid in comparison to the overall mean of data items of each period • Tracking customer shifts among segments • Extracting the dominant transition paths that the majority of customers follow • Predicting the next shift possible path for each customer by examining the dominant paths	(Ha, 2007)

For POS segmentation, the clustering techniques of data mining are applied to obtain common behavioral characteristics of each segment. For calculating the behavior, the RFM technique is used with new definition of F and M as F^* and M^* parameters to be aligned to banking context. After the segmentation, the clusters are labeled based on RF^*M^* scoring. Using the labeling and the R, F^* and M^* parameters, the behavioral rules are extracted. For each important and valuable segment, there are specific rules that define the behavioral characteristic of that segment. Since these rules are generated from a classification method, they can also be applied for predicting the behavioral status of the new upcoming POSs.

The rest of the paper is organized as follow. In section 2 behavioral analysis is described and RFM model is explained, the new definition of F^* and M^* and their calculation formulas are introduced, and the meaning and application of behavioral rules are outlined. In section 3, clustering is defined, learning vector quantization (LVQ) as one of the clustering technique is explored, and Dunn and Silhouette indices are illustrated as clustering validation methods to evaluate the clustering result. LVQ method has a general form, so some of its variants like ULVQ, SOM and ALVQ are summarized, and these variants are used to find a more exact result. In section 4, the rule-based classification is described as a group of algorithms to extract classification rules as descriptive and predicative models and RIPPER algorithm as one example is explained. Section 5 is about the result of implementing the theories on a real case to show the application.

2. Behavioral Analysis

2.1. RFM analysis

The RFM analytic model is proposed by Hughes (Hughes, 1994). It is a model to distinguish customers based on three behavioral variables (attributes), i.e. last purchase interval, customer

purchase frequency and monetary value. Recency (R) of the last purchase is the interval between the last purchase and a present time reference, so the lower recency is more valuable. Frequency (F) of the purchases is the number of transactions in a particular period, so the higher frequency is more valuable. Monetary (M) value of the purchases is the total amount of money paid by the customer over a particular period, so the higher monetary is more valuable. RFM also generates new information about customer's payments preferences, and can be used as an appropriate predictor of future behavior based on the past behavior and it can even be preferred over demographics information (Hughes, 1994). The R, F, and M are used together to calculate the RFM score, which is a simple numerical score and it is used as a comparison criteria for customers (i.e. customers with higher RFM score is more valuable). The RFM score is calculated as follows,

$$RFM\ Score = \alpha \times R_{score} + \beta \times F_{score} + \gamma \times M_{score}\ , \quad (1)$$

where α, β, and γ are the weights of R, F, and M respectively, and they mention the relative importance of the three variables. In (Cheng & Chen, 2009), all the weights are assigned equally to one. It is important that R, F, and M are normalized before calculating the RFM score, so the real R, F, and M are mapped to a scoring discrete grades, R_{score}, F_{score}, and M_{score}, as mentioned in (Hughes, 1994) (Cheng & Chen, 2009). For each scoring parameter, all the values in all records are sorted in descending order of their values (note that for recency, lower number shows more value), and then they are divided equally to five partitions. The five partitions are assigned 5, 4, 3, 2 and 1 score in descending order of their values and the RFM score can vary between 3 and 15 for α=β=γ=1.

2.2. F^* and M^* in banking context

There is a subtle difference between using RFM in CRM context and in this article. In the literature, the RFM is used as an analytical method to estimate the end customers' loyalty and purchase behavior based on the RFM score (Yeh, et al., 2009), where customers with high scores are usually the most profitable, the most likely to repeat a behavior and the company can concentrate its promotional programs on those customers. In banking context, the merchants are the bank's customer; each merchant has its own end customers who purchase products or services from them. The merchant can use the RFM analysis model for its CRM. However, in this article we want to evaluate and segment the POSs, not the customers of the merchants. Therefore, the F and M parameters are modified. To be more specific, suppose a merchant has a POS for 100 days and during this time there are 200 transactions on his/her POS, and another merchant has a POS for 60 days and 120 transactions. Although the first one totally has more transactions, both have two transactions per day, and they have the same activity level. Therefore, the F^* parameter is defined as the average number of transactions per day for every POS. If there is a high frequency for a POS, consequently there is a high monetary, but to detect the monetary behavior of a merchant the M^* parameter is defined as the average amount of spent-money per transaction. This new parameter can help us differentiate transactions' monetary behavior as expensive or inexpensive transactions, which may affect the settlement and taking-fee policy in the bank.

2.2.1. Calculating R, F^* and M^*

For every POS the following parameters are calculated:
F: total number of transactions for a POS terminal
M: total monetary amount of transactions for a POS terminal
$GlobRcvDt_{max}$: the last received date of transactions of all POS terminals
$RcvDt_{max}$: the last received date of transaction of a POS terminal
$RcvDt_{min}$: the first received date of transaction of a POS terminal
D = $GlobRcvDt_{max}$ - $RcvDt_{min}$: duration (in days) for a POS terminal

Finally the R, F^* and M^* parameters can be obtained:
R = $GlobRcvDt_{max}$ - $RcvDt_{max}$: recency (in days)
F^* = F / D: number of transactions per day for a POS terminal

M^*=M / F: monetary unit value per transaction for a POS terminal

2.2.2. RF^*M^* Scoring

Like the RFM scoring, we can define RF^*M^* scoring, and all the process is as the same as RFM, but instead of F and M, F^* and M^* are used. For simplicity, all the weights are assigned equally to one. In this article RF^*M^* score is used for both preprocess and segmentation, which are explained in more detail in section 5. In preprocess, the POSs with RF^*M^* scoring value below the average are omitted as not interesting ones.

2.3. Behavioral rule

Suppose we have some rules like "if $R>r_1$ and $F^*<f_1$ and $M^*<m_1$ then the POS is *label*$_1$" and "if $R<r_2$ and $F^*<f_2$ and $F^*>f_3$ and $M^*>m_2$ then the POS is *label*$_2$" where r_i, f_j, m_k are some constants and *label*$_1$ and *lable*$_2$ are meaningful business-related labels. The boundaries over R, F^* and M^* associated with a meaningful label helps us view the system execution from an abstract level and an assessment of values to compare the segments. It is also helpful to predict the new POSs labels, which give us an online performance evaluation to see the status (i.e. goodness/badness) of new POSs and finding the flaws quickly, and reinforcing the positive points of the system. These rules represent *green*, *yellow* and *red lights* of the system, where can be used as a *monitoring dashboard*.

3. Clustering

Clustering is the process of grouping a set of physical or abstract items into classes of similar items (Han & Kamber, 2006) where the groups are meaningful, useful, or both (Tan, et al., 2005). A cluster is a collection of data items that are *similar (or related)* to one another within the same cluster and are *dissimilar (or unrelated)* to the objects in other clusters, so a cluster of data items can be treated collectively as one group and so may be considered as a form of data compression (Han & Kamber, 2006), which helps us easily annotate all the data items. The better or more distinct clustering has more similarity (or homogeneity) inside a group and more difference among various groups (Tan, et al., 2005), which is the final goal of a clustering process. There are so many clustering algorithms (Han & Kamber, 2006) (Tan, et al., 2005) (Xu & Wunsch, 2005) with different characteristics and applications. In this paper, clustering is applied as a data reduction technique to help us only consider and compare centroids instead of the entire data items. Another important issue is assigning meaningful business-related class labels to the clusters, so we need non-overlapping results. Therefore an appropriate algorithm for the purpose of this paper is a crisp prototype-based (centroid-based) one. Vector quantization (VQ) methods are good candidates to find a set of main vectors (centroids) to represent the entire data items (Wu & Yang, 2006). The k-means clustering algorithm is a famous batch-type vector quantization method, but its batch processing has the flaw of not converging to an optimal result, so combining vector quantization with competitive learning neural networks can improve the convergence to an optimal result (Wu & Yang, 2006), and this combination mentions as a more accurate technique in (Ha, 2007). Two famous variants of LVQ which are ULVQ, SOM and a new variant called ALVQ introduced in (Wu & Yang, 2006) and k-means, a famous batch-type vector quantization, are considered for clustering and their results are compared to each other to find a more accurate one.

3.1. Learning vector quantization (LVQ)

One of the most commonly used unsupervised clustering algorithms is the learning vector quantizer (LVQ) developed by Kohonen (Kohonen, 2001). While several versions of LVQ exist (Kohonen, 2001) (Wu & Yang, 2006), this subsection reviews and compares three versions called Unsupervised LVQ (ULVQ), Self-Organized Map (SOM), Alternative LVQ (ALVQ) and k-means as a batch-type of VQ is also compared.

LVQ has only one layer of neurons and each neuron has a weight vector and it represents a cluster. During training, the cluster unit whose weight vector is the *closest* to the current input pattern is acknowledged as the winner (competition phase). The corresponding weight vector and that of

neighboring units are then updated to better adjust the input pattern (learning phase). The *closeness* of an input pattern, p, to a weight vector is usually calculated using the Euclidean distance. Neuron's weight in learning vector quantization are updating according to the following general formula,

$$W_i(t) = W_i(t-1) + \eta(t)h_{i,k}(t)[X(t) - W_i(t-1)]\,, \tag{2}$$

where $\eta(t)$ is the decaying learning rate, and $h_{i,k}(t)$ is the neighborhood function to update the neighbor of the winning neuron (the *k* index shows the index of winner neuron) .

Table 2
Neighborhood functions for some LVQ variants

	Neighborhood Function, $h_{i,k}(t)$	Description	Reference
ULVQ	$\begin{cases}1 & if\ i=k\\ 0 & otherwise\end{cases}$	Winner-take-all function	(Engelbrecht, 2007)
SOM	$e^{-\frac{\|c_i-c_k\|}{2\sigma^2(t)}}$	The smooth Gaussian kernel function. $\|c_i - c_k\|$ is Euclidian distance between neuron_i and winner neuron_k position in net topology. $\sigma(t)$ is the kernel's width	(Engelbrecht, 2007)
ALVQ	$h_{i,k}'(t) \times e^{-\|X(t)-W_i(t-1)\|^2/\beta(t)}$ $\beta(t) = \frac{\sum_{i=1}^{c}\|W_i(t-1) - \bar{W}(t-1)\|}{c}$ $\bar{W}(t-1) = \frac{\sum_{i=1}^{c} W_i(t-1)}{c}$	The exponential function tries to measure the similarity between the input vector and the winner neuron and prevent outliers to attract weight vectors and move far away $\beta(t)$ normalizes the dissimilarity measure ($\|X(t) - W_i(t-1)\|^2$) $h_{i,k}'(t)$ is an alternative neighborhood function and in this paper it is winner-take-all	(Wu & Yang, 2006)

The neighborhood function is different for ULVQ, SOM and ALVQ as summarized in Table 2. In LVQ, weights are initialized to random values, sampled from a uniform distribution, or by taking some input patterns (chosen in this article) (Engelbrecht, 2007). There are some stopping conditions defined in (Engelbrecht, 2007), but for simplicity the stopping condition selected in this article is a maximum number of epochs to reach, which are 100 for this article. Each algorithm has its specific input parameters. In section 5, in the implementation result, these parameters are summarized. The learning rate (η) and the width of kernel for SOM are suggested in (Engelbrecht, 2007) and (Wu & Yang, 2006), and they are applied in this article.

3.2. Cluster evaluation indices

For most clustering algorithms, the number of clusters must be mentioned as an input data but there are no straight algorithms to find the number of true clusters. Besides, clustering is totally a subjective process and the data set can be partitioned differently for different applications (Jain, et al., 1999). There are also some *objective measures of pattern interestingness* (Han & Kamber, 2006), called cluster validation or evaluation indices, like Silhouette and Dunn and are described later. These quantitative indices help us assess the clustering result and find an appropriate one. As calculating these indices may take some time, a range for number of clusters must be considered.

3.2.1. Silhouette Index

The Silhouette index was introduced by Rousseeuw (Rousseeuw, 1987) and is reviewed in (Brun, et al., 2007) and (Bolshakova & Azuaje, 2003). For a given cluster, this method assigns *Silhouette width* to each input data item as a quality measure, and the final *Silhouette* is the average of the Silhouette width of all the points. If *x* is a data item in the cluster C_i, then the Silhouette width of *x* is defined by the ratio

$$s(x) = \frac{b(x)-a(x)}{\max[b(x),a(x)]}, \quad (3)$$

where $a(x)$ is the average distance between *x* and all other data items in its cluster, C_i,

$$a(x) = \frac{1}{n_i-1}\sum_{y\in C_i, y\neq x} d(x,y), \quad (4)$$

where n_i is the number of data items in cluster C_i, and $d(x,y)$ is the distance between two data items (or dissimilarity distance (Rousseeuw, 1987)). The $b(x)$ is the minimum of the average distances between *x* and the data items in other clusters,

$$b(x) = \min_{h=1,\ldots,K, h\neq k}[\frac{1}{n_h}\sum_{y\in C_h} d(x,y)], \quad (5)$$

where n_h is the number of data items in cluster C_h. Finally, the global Silhouette index is defined by

$$S = \frac{1}{K}\sum_{k=1}^{K}[\frac{1}{n_k}\sum_{x\in C_k} S(x)]. \quad (6)$$

For a given point *x*, its Silhouette width ranges from −1 to 1, which is the same as global Silhouette. The higher the Silhouette, the more compact and separated are the clusters (Rousseeuw, 1987). If the Silhouette is positive (b(x) > a(x)) shows the average outer clusters distance for point *x* is larger than the average inner cluster distance, so it is positioned in a good clustering result. If the value is zero (b(x) = a(x)), the point *x* is stated in a middle of two clusters with equal distance. If the value is negative (b(x) < a(x)), the point x is not in a proper cluster, and it could be moved to another cluster.

3.2.2. Dunn indices

The Dunn indices are reviewed in (Bolshakova & Azuaje, 2003) (Brun, et al., 2007). This collection of indices evaluates sets of clusters based on compactness and well-separation (Bolshakova & Azuaje, 2003). For any partition, $C = \{C_1, C_2, \ldots, C_k\}$, where C_i represents the i^{th} cluster of such partition, the Dunn's validation indices, *D*, is defined as:

$$D(C) = \frac{\min_{i,j=1,\ldots,k,\ i\neq j}\delta(C_i,C_j)}{\max_{i=1,\ldots,k}\Delta(C_i)}, \quad (7)$$

where $\delta(C_i, C_j)$ defines the distance between clusters C_i, C_j (intercluster distance), and $\Delta(C_i)$ represents the intracluster distance of cluster C_i (Bolshakova & Azuaje, 2003) or the size of the cluster C_i (Brun, et al., 2007), and *k* is the total number of clusters. The main goal of this measure is to maximize intercluster distances (linkage (Bolshakova & Azuaje, 2003)) or to minimize intracluster distances (diameter (Bolshakova & Azuaje, 2003)). Thus higher values of *D* correspond to better clustering result.

Table 3

Linkage and diameters used in Dunn (Brun, et al., 2007)

Intercluster Distances (linkages)	Single		Complete
	$\delta_1(C_i,C_j) = \min_{x\in C_i, y\in C_j}\{d(x,y)\}$		$\delta_2(C_i,C_j) = \max_{x\in C_i, y\in C_j}\{d(x,y)\}$
	Average		Centroid
	$\delta_3(C_i,C_j) = \frac{1}{\lvert C_i\rvert\lvert C_j\rvert}\sum_{x\in C_i, y\in C_j} d(x,y)$		$\delta_4(C_i,C_j) = d(\bar{C_i},\bar{C_j})$
Intracluster Distances (diameters)	Complete	Average	Centroid
	$\Delta_1(C_i) = \max_{x,y\in C_i}\{d(x,y)\}$	$\Delta_2(C_i) = \frac{1}{\lvert C_i\rvert(\lvert C_i\rvert-1)}\sum_{x,y\in C_i} d(x,y)$	$\Delta_3(C_i) = 2(\frac{\sum_{x\in C_i} d(x,\bar{C_i})}{\lvert C_i\rvert})$

$\bar{C_i}$ is the centroid of cluster C_i, and $|C_i|$ is the size of cluster C_i

There are several intercluster and intracluster distance measure functions defined in (Bolshakova & Azuaje, 2003) (Brun, et al., 2007). The ones used in this article are summarized in Table 3. Based on the defined linkages and diameters functions in Table 3, twelve Dunn indices are calculated and applied for every clustering result in this article.

4. Rule-based Classification

Rule-based classifications are a group of classification algorithms that produce "if … then …" rules which could be used for prediction and observing a behavioral or structural view in a more abstract level. The output rules for the model are represented in a disjunctive normal form, $\Re = (r_1 \vee r_2 \vee \ldots \vee r_k)$, where $\Re$ is called the *rule set* and r_i's are classification rules. Each classification rule can be expressed as $r_i : (condition_i) \rightarrow y_i$. The $condition_i$ has a conjunction of attributes in the form of $(A_1\ op\ v_1)\ \ (A_2\ op\ v_2)\ \ \ldots\ \ (A_n\ op\ v_n)$ where A_i is an attribute name and v_i is a literal of attribute A_i's types and *op* is a logical operator selected from $\{=, \neq, <, >, \leq, \geq\}$ set. y_i is called the *rule consequent* which will be the predicted class label. A rule *r covers* a record *x* if the condition of *r* matches the attributes of *x*. A rule set must have two important characteristics: *mutually exclusive* rules, and *exhaustive* rules. Mutually exclusive rules means that no two rules in $\Re$ cover the same record and exhaustive rules means there is a rule for each combination of attribute value (Tan, et al., 2005). These properties ensure that every record is covered only by a specific rule. The *sequential covering* algorithm is often used to extract rules directly from data. Rules are grown in a greedy way until a stop condition is met. The algorithm extracts the rules for one class at a time from the data set. In this way, a conjunction may be appended to the current rule if the new rule has higher evaluation-index value than the previous one. So there are some rule evaluations used during rule growing phase (Tan, et al., 2005). One of them is the FOIL information gain index which is used in RIPPER algorithm, and is as follows (Tan, et al., 2005):

$$FOIL's\ Information\ Gain = p_1 \times \left(log_2 \frac{p_1}{p_1+n_1} - log_2 \frac{p_0}{p_0+n_0}\right), \quad (8)$$

where p_1 is the new true positive covered examples and n_1 is the new negative covered examples of the new rule, and p_0 is the current true positive covered examples and n_0 is the current negative covered examples of the current rule. The higher value of FOIL's information gain shows a better rule result. There are two types of rule classifier algorithms: direct, and indirect (Tan, et al., 2005). The direct algorithms generate rules directly from the data sets like the RIPPER algorithm. The indirect algorithms generate rules from the result of other classification algorithms, mainly decision trees.

4.1. RIPPER Algorithm

The RIPPER algorithm is a commonly used rule induction algorithm and scales almost linearly with the number of training records (Tan, et al., 2005). Its pseudo code is shown in Algorithm 1.There are some characteristics of RIPPER algorithm that makes it a very good choice for rule induction. The reasons are:

- It can generate descriptive rules vs. neural networks that are black box.
- It has linear execution scalability to the number of training records (Tan, et al., 2005).
- It is a direct rule generator.
- It generates rules for classes with less distribution to more distribution and the class with the most members is considered as default class.

In decision tree using the impurity measures like Gini or Entropy for splitting, the majority class (class with most members) is considered first and most of the rules have consequent of major class. In behavioral analysis, the class with more than 50% of data items usually shows the common and normal behavior and classes with fewer members have some special characteristics and they are more interesting to be considered. Since RIPPER considers classes with fewer members first, it generates more desirable rules and it is more preferable than decision trees. The result in the implementation section also shows that this algorithm generates a very accurate result.

Let D be the training records with the form of (A_1, A_2, …, A_n) attributes
Let Y be the set of classes {y_1, y_2, …, y_k} sorted in an ascending frequency order and y_k with the most frequency is assumed as the default class
Let ℜ={} be the rule set, initialized with no rule
for each y ∈ Y-{y_k} **do**
Growing phase of rule r
while not covering negative records **do**
growing rule r over records with y class in a general-to-specific manner using FOIL'S information gain measure (8) to choose the best conjunct to be added into the rule antecedent
end
Pruning phase of rule r
pruning the rule r using formula (p-n)/(p+n), where p/n is the number of positive/negative examples in the validation set covered by the rule
Building the rule set phase, adding rule r to ℜ
ℜ → ℜ ⊕ r
Remove the training records from D that are covered by r
End

Algorithm 1 RIPPER rule-based classifier

5. Implementation results

In this section, the implementation result for a case study is described. For the case, the needed data is extracted from a private bank database. The permitted data has 30,524 POSs and 1,030,120 no of successful and settled transactions of purchase by only debit cards. Transactions are from 2008/05/13 to 2009/10/27 or 532 working days. Each transaction has three attributes (fields): received date, amount, and POS terminal id. The applied framework for data mining has three phases:

1. data preprocess
2. clustering, cluster analysis and result selection
3. classification and behavioral rules induction

5.1. Data preprocess

Table 4 summarizes the necessary steps of this phase. The first column is the description of the step, and the result column shows the outcome of the step.

Table 4
Data preprocess steps

Description	Result
Extracting successful and settled purchase transactions of POSs	1,030,120 records of transactions
Transactions are aggregated based on the terminal id, to calculate parameters introduced in section 2.2.1	25,553 records of R, F^*, M^*, D for each POS
Only POS terminals with more than 180 days duration (D ≥ 180) can be considered as an assessable POS (based on the bank's electronic-payment project manager's suggestion, less than this duration is too soon to evaluate the POS)	15,786 records of R, F^*, and M^* (62% of total)
Histograms of R, F^* and M^* (Fig 1) shows there are many POSs in the lower part of diagrams. The RF^*M^* score is calculated based on mapping Table 5 F^*, M^* and R, and POSs with RF^*M^* score lower than the average (9.009) are omitted	9,081 records of R, F^*, and M^* (36% of total, 57% of previous step)

Table 5 is the look-up table which maps the real value of each R, F^*, and M^*, to its equivalent score as discussed in section 2.1. Each cell shows a range of values, and if the related parameter's value according to the row header is placed in the range, its equivalent score is obtained by looking up the corresponding column-header's value. For example, if a data item has R=200, its corresponding score for R (R_{score}) is 2. Table 6 shows the look-up table for data items after the last preprocessing step of Table 4. The differences of boundaries in Table 5 and Table 6 show that the POSs with the lowest value with the three parameters are omitted in the last step of preprocessing. Table 6 will be applied in the next section for analyzing the clustering result. The reason for applying RF^*M^* scoring as a filtering mechanism is that all three parameters are participated for removing the worthless records, and since the RF^*M^* scoring is a meaningful and important paradigm in this article, this filtering is also meaningful.

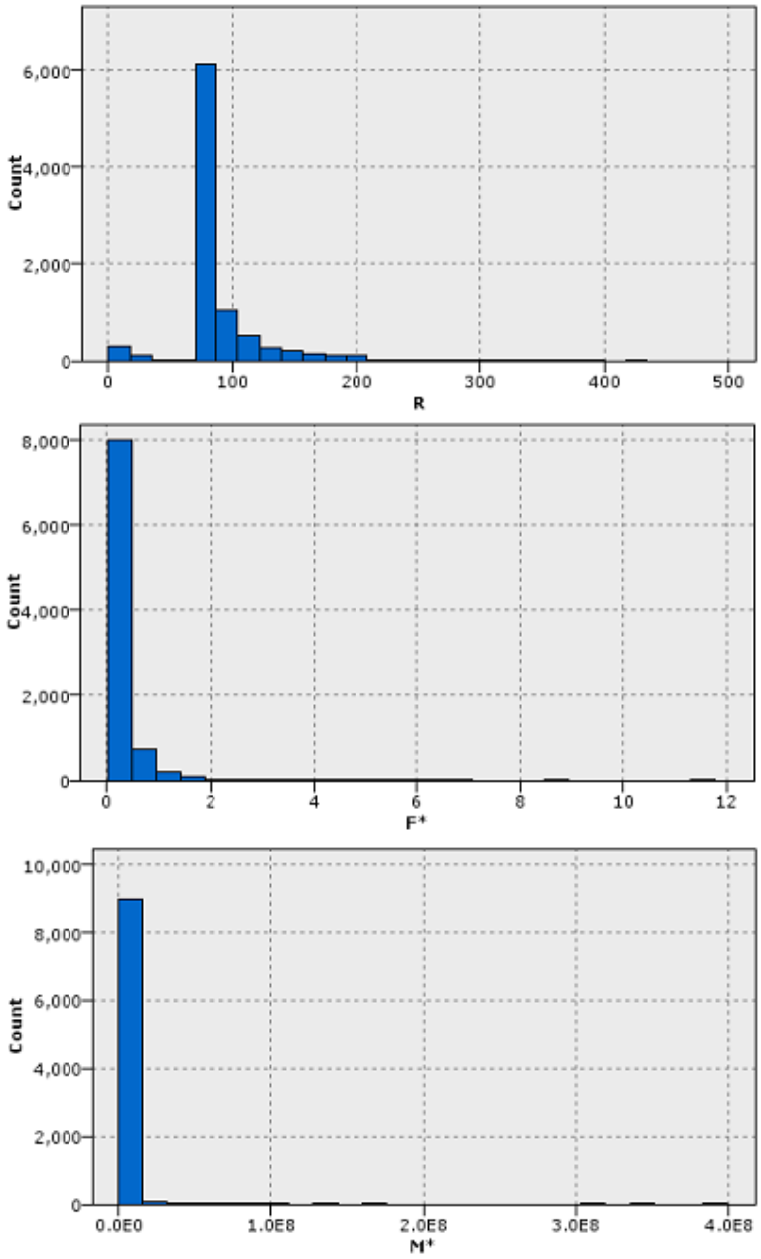

Fig 1. R, F^* and M^*

5.2. Clustering, cluster analysis and result selection

Now these 9,081 POSs are clustered using the four algorithms to find better results. The number of clusters is always debatable. In this article, the Silhouette index and twelve Dunn indices are calculated for clustering results of between two to twenty five clusters, which is depicted in Fig 2 for four algorithms. The results in Fig 2 show that ULVQ represents the most accurate results among the four algorithms, so the results of ULVQ are selected for more process. The input parameters of the aforementioned algorithms are summarized in Table 7. The best clustering result is for three clusters, but the result is affected by the outliers, where the histograms of Fig 1 shows there are some records with unusual values of M^* and F^*. After clustering result with 13 clusters, there is lower variation both in the Silhouette and Dunn indices, and in the number of items in clusters, which means we reach stability in clustering result in spite of increasing more clusters. So the result with 13 number of clusters are chosen, and then R, F^* and M^* scores (based on Table 6) and the RF^*M^* score are calculated for each cluster, summarized in Table 8.

Table 5
Mapping ranges of R, F^* and M^* to their scoring values (column header) applied in preprocessing

	Score				
	1	2	3	4	5
R	**[**561 **,** 209**)**	**[**209 **,** 115**)**	**[**115 **,** 80**)**	**[**80 **,** 72**)**	**[**72 **,** 0**]**
F^*	**[**0.0020 **,** 0.0150**)**	**[**0.0150 **,** 0.0396**)**	**[**0.0396 **,** 0.0859**)**	**[**0.0859 **,** 0.1990**)**	**[**0.1990 **,** 11.7689**]**
M^*	**[**11,000.0000 **,** 167,142.8571**)**	**[**167,142.8571 **,** 287,500.0000**)**	**[**287,500.0000 **,** 521,583.3333**)**	**[**521,583.3333 **,** 1,227,817.7677**)**	**[**1,227,817.7677 **,** 399,155,117.6471**]**

Table 6
Mapping ranges of R, F^* and M^* to their scoring values (column header) applied in analyzing final clustering result

	Score				
	1	2	3	4	5
R	**[**435 **,** 97**)**	**[**97 **,** 77**)**	**[**77 **,** 72**)**	**[**72 **,** 70**)**	**[**70 **,** 0**]**
F^*	**[**0.0049 **,** 0.0522**)**	**[**0.0522 **,** 0.0952**)**	**[**0.0952 **,** 0.1627**)**	**[**0.1627 **,** 0.3160**)**	**[**0.3160 **,** 11.7689**]**
M^*	**[**19,545.4545 **,** 242,702.7027**)**	**[**242,702.7027 **,** 418,823.5294**)**	**[**418,823.5294 **,** 783,823.5294**)**	**[**783,823.5294 **,** 1,677,835.0515**)**	**[**1,677,835.0515 **,** 399,155,117.6471**]**

Table 7
Input parameters for the clustering algorithms used in this article

	Number of clusters	η(t)	h(t)	σ(t)
ULVQ	The range between 2 to 25 number of clusters are chosen and with consideration of the thirteen clustering evaluation indices, to select the appropriate result	1/t	see Table 2	undefined
SOM		η(0)=1, η(t)= η(t-1)/t	see Table 2	e^{-t}
ALVQ		1/t	see Table 2, for $h_{i,k}^{'}(t)$ the winner-take –all is selected	undefined
k-means		undefined	undefined	undefined

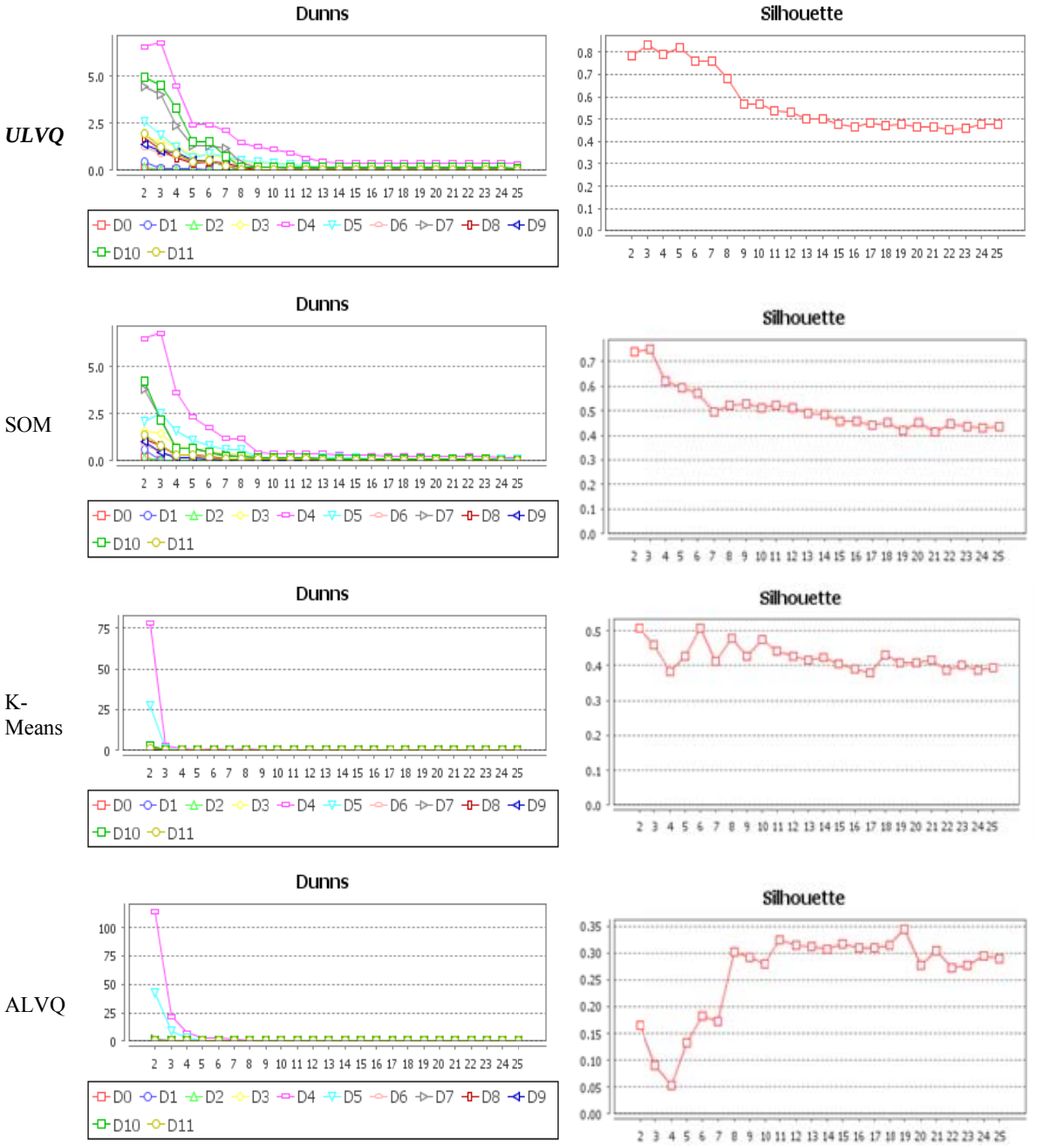

Fig 2. Silhouette and 12 Dunn indices results of four algorithms for clusters between 2 to 25

For each cluster a meaningful business-related label is assigned except for outliers, which are very special cases. For more clarification, in every row of Table 8, the main reason(s) of assigning the related label is(are) underlined. The outliers, which are rare, are records that have one or some attributes with unusual and faraway value (clusters 3, 4 and 11). The rare classes (high transactive, eager, rich transactive, lazy rich) are also special and have few data items that are considered special and exceptional (all of them are about 10% of 9081 POSs), so they are not worth of more processing. The results in Table 8 show three considerable clusters, which are 10, 6, and 12-13 with the most items.

Table 8
Final clustering result

No	Count	R	F*	M*	Grade	Score	Assigned Label	%
1	27	47.2593	4.7585	788,948.1	5 5 4	14	*High Transactive*	4%
2	341	66.7126	1.4783	580,034.6	5 5 4	14	*High Transactive*	
3	1	0	11.7689	686,001.0	5 5 3	13	Outlier (for F*)	-
4	3	71.3333	0.1004	352,282,241.6	4 3 5	12	Outliers (for M*)	-
5	362	9.8398	0.2459	736,167.8	5 4 3	12	*Eager*	4%
6	1906	73.1542	0.4055	496,639.0	3 5 3	11	*Transactive*	21%
7	68	71.0441	0.5657	9,432,476.9	4 5 5	10	*Rich Transactive*	0.7%
8	20	93.8	0.0814	46,860,550.8	2 2 5	9	*Lazy Rich*	0.8%
9	58	93.6034	0.0851	25,563,046.3	2 2 5	9	*Lazy Rich*	
10	4558	78.6542	0.0961	1,625,969.5	2 3 4	9	*Regular*	50%
11	4	108.5	0.0705	158,037,331.4	1 2 5	8	Outliers (for M*)	-
12	22	91.5	0.0504	79,667,469.5	2 1 5	8	*Inactive*	19%
13	1711	137.0175	0.133	1,012,103.7	1 3 4	8	*Inactive*	

Their descriptions are as follows:

- Clusters 12-13 with highest R (137 days) are an alarm of 19% inactive POSs.
- Cluster 10 with 50% POSs, in comparison to other clusters, shows a relative good current state (its score is 9), and as it has half of the POSs, it represents the common behavior of POS usage.
- Cluster 6 with 21% POSs is the most interesting cluster. Since there are more "transactive" POSs to "rich" ones (clusters 7, 8 and 9), it means that the bank must focus and plan to move POSs in regular cluster to "transactive" one. The M^* in this cluster is relatively low, so it is better not to charge them with fee to persuade them as an incentive for more transaction.

The cluster centers (centroid) are only a representative of clusters and they do not show *boundaries* of parameters in each cluster. To know better these three clusters based on the three behavioral parameters, in the next section some behavioral rules based on these three parameters are induced. These rules represent an abstract and meaningful view over the three aforementioned main clusters.

5.3. Classification and behavioral rules induction

For classification, the Weka (Weka 3: Data Mining Software in Java) software is chosen, which is a famous software in data mining and machine learning. Among the classification algorithms in Weka, the JRip is used which is the RIPPER algorithm's implementation. It is common to divide the data set into two partitions of α-percent as train and the remaining as test. The test partition is for evaluating the generated classification model's accuracy and α=66% is commonly chosen. After generating the model with training records, the test records' class labels are examined against the generated class label using the model for the test records. The percent of the correct classified testing records shows the accuracy of the model. The clustering phase resulted in three main classes, which are transactive, regular and inactive. Other classes can be ignored as they are outliers (exceptions) or they are rare and the generated rule for them is so specific. So the records of the three main clusters are given to JRip. The rules induced from training records are presented in Table 9. Total number of training records is 5,410 and total number of test records is 2,787. Correctly classified instances of test records are **91.5%,** which is a good accuracy of the induced rules.

Table 9
Behavioral Rules for the three main segments (Inactive, Transactive, and Regular)

	Rule				Tot Rec	Mis Rec
1	*if*	$R \geq 101$ and $M^* \leq 1{,}778{,}333.4$ and $F^* \leq 0.481$	*then*	Inactive	1097	9
2	*else if*	$R \geq 89$ and $M^* \leq 799{,}736.8$ and $F^* \leq 0.296$	*then*	Inactive	318	50
3	*else if*	$R \geq 113$ and $M^* \leq 3{,}400{,}000.0$	*then*	Inactive	207	13
4	*else if*	$F^* \geq 0.190$ and $M^* \leq 645{,}182.432$	*then*	Transactive	1539	65
5	*else if*	$F^* \geq 0.227$ and $M^* \leq 1{,}455{,}333.333$	*then*	Transactive	298	22
6	*else if*	$F^* \geq 0.327$	*then*	Transactive	137	34
7	*else*	Regular			4601	189

Tot Rec: Total number of records in the data set are covered by the rule in the row
Mis Rec: Number of records that are covered by the rule but with different class label (negative ones)

The above table has some interesting results:

- Based on the nature of RIPPER algorithm, the rules are ordered for considerable classes (inactive and transactive).
- The inactive class is an alarm and it is the lower limit of the system, so if the new POS is assigned to this class, it shows the dissatisfaction of the merchant from the system or the laziness of its customers for using POS.
- The first rule of inactive labels has the expression of $R \geq 101$, but the results from clustering shows R=137 in the centroid, and this sample shows the lower limit of R for inactive clusters, although the centroid shows other value, and this outlined the application and meaning of behavioral rule.
- The transactive class is interesting because it is the profitable POSs and we must persuade and keep them as loyal merchants. The lower limit of F^* also shows approximately 0.2 transactions per day for most of transactive ones, and it means during 10 days only 2 transactions happen over the POS, which is *not* a good and profitable limit. The M^* for transactive POSs shows 650,000 monetary unit per transactions which is a good and profitable limit.
- Another interesting result is that the rules partitioned inactive and transactive classes into three groups, and for each group the total number of records and boundaries of R, F^* and M^* are represented. This is a more detailed characteristic of subgroups of three main clusters, but we consider the rule with the most associated items.

6. Conclusion and Future Work

In this article a new RF^*M^* approach is defined for banking industry to analyze transactional behavior of POSs. The RF^*M^* scoring is applied for both preprocessing and clustering. Filtering records using RF^*M^* scoring is a proper way to apply three parameters together to remove the worthless records. The clustering algorithms are applied for data reduction to consider only the clusters' representations instead of the entire records. Thirteen clustering indices are measured to find appropriate number of clusters with more accuracy among four algorithms' results. The proper number of clusters has been chosen based on the no-further change in the results of the thirteen indices. Meaningful business-related labels have been assigned to clusters according to the RF^*M^* scoring of their centroids. Only records of three main class labels with their R, F^* and M^* parameters are entered to the RIPPER rule-based classifier to induce behavioral rules. The rules present a more abstract and meaningful behavioral and descriptive model of each cluster. The generated boundaries of R, F^* and M^* from the rules can be used for both prediction of new upcoming POSs' labels and assessment of current POSs'

profitability. They also show three subgroups in the two main clusters. A good future work is to apply the new RF^*M^* scoring model of this paper to generate new tracking transition path introduced in (Ha, 2007) to achieve numerical transitional path for merchants over time periods. This combination will be a quantitative long-term assessment of behaviors and a new mechanism for assigning labels to clusters.

Acknowledgement

The authors would like to thank the anonymous referees for their valuable comments on an earlier version of this work.

References

Bolshakova, N., & Azuaje, F. (2003). Cluster validation techniques for genome expression data. *Signal Processing* , 83(4), 64-80.

Brun, M., Sima, C., Hua, J., Lowey, J., Carroll, B., Suh, E., et al. (2007). Model-based evaluation of clustering validation measures. *Pattern Recognition* , 40(3), 807-824.

Cheng, C.-H., & Chen, Y.-S. (2009). Classifying the segmentation of customer value via RFM model and RS theory. *Expert Systems with Applications* , 36(3), 4176-4184.

Engelbrecht, A. P. (2007). Computational Intelligence. John Wiley & Sons.

Garland, R. (2002). Non-financial drivers of customer profitability in personal retail banking. Journal of Targeting, *Measurement and Analysis for Marketing* , 10(3), 233-248.

Ha, S. H. (2007). Applying knowledge engineering techniques to customer analysis in the service industry. *Advanced Engineering Informatics* , 21(3), 293-301.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On Clustering Validation Techniques. *Journal of Intelligent Information System* , 17(2-3), 107-143.

Han, J., & Kamber, M. (2006). Data Mining Concepts and Techniques. Morgan Kaufmann.

Hughes, A. M. (1994). Strategic Database Marketing. Chicago: Probus Publishing.

Humphrey, D. B., Kim, M., & Vale, B. (2001). Realizing the Gains from Electronic Payments: Costs, Pricing, and Payment Choice. *Journal of Money, Credit and Banking* , 33.

Jain, A., Murty, M., & Flynn, P. (1999). Data Clustering: A Review. *ACM Computing Surveys* , 31(3), 264-323.

Kohonen, T. (2001). Self- Organizing Maps (Third ed.). Springer.

Kotler, P., Wong, V., Saunders, J., & Armstrong, G. (2005). Principles of Marketing. Pearson Education.

Lughofer, E. (2008). Extensions of vector quantization for incremental clustering. *Pattern Recognition* , 41.

Romdhane, L., Fadhel, N., & Ayeb, B. (2010). An efficient approach for building customer profiles from business data. *Expert Systems with Applications*, 37(2), 1573-1585.

Rousseeuw, P. J. (1987). Silhouettes: a graphcal aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20.

Tan, P.-N., Steinbach, M., & Kumar, V. (2005). Introduction to Data Mining. Pearson Education.

Weka 3: Data Mining Software in Java. (n.d.). Retrieved from Weka 3: http://www.cs.waikato.ac.nz/ml/weka

Wu, K.-L., & Yang, M.-S. (2006). Alternative learning vector quantization. Pattern Recognition , 39.

Xu, R., & Wunsch, D. (2005). Survey of Clustering Algorithms. *IEEE Transactions on Neural Networks* , 16.

Yeh, I.-C., Yang, K.-J., & Ting, T.-M. (2009). Knowledge discovery on RFM model using Bernoulli sequence. *Expert Systems with Applications,* 36.