Proceedings of the World Congress on Engineering 2010 Vol I
WCE 2010, June 30 - July 2, 2010, London, U.K.

An Optimized Classification Model for Time-Interval Sequences

Chieh-Yuan Tsai, Chun-Ju Chien

Abstract—**Sequence classification is a popular data mining method to explore customer behavior by assigning the most probable class label to a given sequence. However, previous researches seldom discussed sequence classification problem related to time information. Without time information, two sequences with the same itemsets but different time-intervals will be classified as the same class. For this reason, this research presents a time-interval sequence classification methodology to help business managers understand their customer behaviors in depth. The proposed sequence classification method includes two main stages. The first stage is time-interval sequential pattern mining, which employs I-PrefixSpan algorithm to discover time-interval sequential patterns in the large database. The second stage is time-interval sequence classification method, which contains time-interval sequence classification model and particle swarm optimization procedure. A simple case is employed to show the performance of the proposed classification method. The experiment result indicates the proposed time-interval sequence classification method is feasible and efficient.**

Index Terms—**Time-interval sequential pattern mining, Sequence classification, Particle swarm optimization.**

I. Introduction

In current competitive and fast-changed business environment, if an enterprise can provide right products and right services at the right time to the right customers, the enterprise might have more chance to survive than others. Therefore, it is important for enterprises to know their customer behaviors in depth. Sequential pattern mining, one of the most popular data mining methods, receives many attentions in recent years. Since sequential pattern mining can discover frequent occurring patterns from large databases, it has become a critical approach to explore customer behaviors. For example, the sequential pattern mining can generate a sequential pattern such as "inkjet printer → ink cartridge → papers → ink cartridge refill kit." This pattern shows that customers who buy an inkjet printer will have strong probability to buy an ink cartridge, papers, and ink cartridge refill kit in order. With the help of this pattern, managers can send an ink cartridge refill kit promotion program to customers after customers bought an inkjet printer, an ink cartridge, and papers.

The researches related to sequential pattern mining mainly focus on mining interesting sequential patterns and detecting similar sequential patterns [1]–[3]. There is few researches discuss the issue of how to accurately classify an unseen sequence. The sequence classification is to assign the most probable class label to a given sequence by a generative model (or classifier). Huang et al. [4] employed support vector machine (SVM) to develop sequence classification model. Joung et al. [5] employed hidden Markov model (HMM), a statistical model in which the system being modeled is assumed to be a Markov process with unobserved state, to generate a sequence classification model. Bouchaffra and Tan [6] extended the traditional hidden Markov model named "structural hidden Markov model" by partitioning the set of observation sequences into classes of equivalences to enhance the accuracy for the sequence classification model. Bruyn et al. [7] integrated estimation, clustering, and classification into the traditional, three-step approach to make the result of sequence classification more reliable.

The above sequence classification researches mainly discuss about the biological protein sequence classification and text recognition. Seldom of them applied it to customer behavior prediction. In addition, the sequential patterns in these studies do not cover time-interval information. However, this drawback is especially serious when dealing with customer behavior data. For example, "notebook → (1 week) → printer" and "notebook → (1 year) → printer" are two sequential patterns with the same order but different time-intervals. It is clear that customers buy the printer after they bought notebook 1 week later and 1 year should not be viewed as the same behavior. If time-interval information is not considered in the two sequential patterns, managers might consider the two patterns as the same pattern of "notebook → printer".

To solve the above problems, this study developed an efficient classifier that can accurately predict the class label of the sequences with time-interval information. The proposed classifier is expected to help decision makers not only can clearly distinguish different types of customers' behavior over time but can propose more intelligent business strategies according the classification result.

II. Research Method

The framework of the proposed time-interval sequence classification model is presented in Fig. 1 which includes two stages. In the first stage, I-PrefixSpan algorithm [8] is used to generate time-interval sequential patterns for every class in the database, respectively. In the second stage, the similarity between each pair of time-interval sequences is calculated first. Next, the generated time-interval sequential pattern sets

Manuscript received February 22, 2009. Chieh-Yuan Tsai is with the Department of Industrial Engineering and Management, Yuan Ze University, Taiwan, R.O.C. (phone: +886-3-463- 8800 Ext. 2512; fax: +886-3-463-8907; e-mail: cytsai@saturn.yzu.edu.tw)

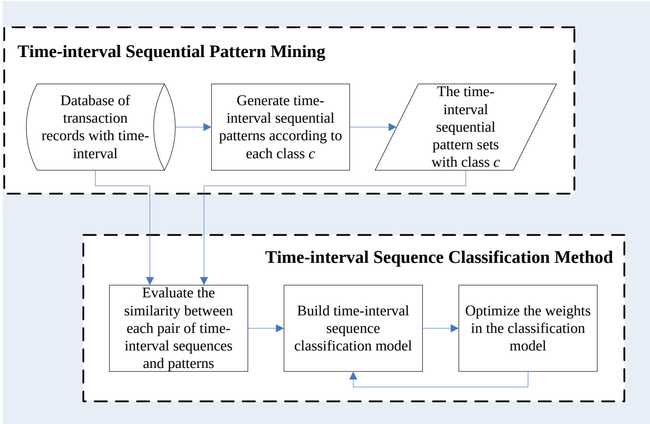

Fig. 1 Framework of the proposed time-interval sequence classification model

are employed to build the time-interval sequence classification model. Then the optimization method based on particle swarm optimization (PSO) is employed to update the weights included in the time-interval sequence classification model to maximize the accuracy of the classification result. Finally, when a new time-interval sequence inputted, the proposed classification model can accurately predict the class label of the time-interval sequence.

A. *Time-Interval Sequential Pattern Mining*

The transaction database is divided to n sub-databases according to the class that each sequence belongs to. Let the cth sub-database be presented as PH^c, where $c = 1, 2, \ldots, n$. After conducting time-interval sequential pattern mining (I-PrefixSpan algorithm) to PH^c, the frequent time-interval sequential pattern sets for class c, denoted as FP^c are found.The I-PrefixSpan algorithm, which is proposed by [8] is used to generate the time-interval sequential patterns from sequence data. It is extended from the well-known PrefixSpan algorithm. I-PrefixSpan is a projected-based algorithm whose steps for generating time-interval sequential patterns are shown in Fig. 2.

B. *Similarity Measure between Sequences and Patterns*

To facilitate the following classification model building process, the dissimilarity between a sequence in PH^c and a time-interval sequential pattern in FP^c, should be defined first. Two dissimilarity premises between a time-interval sequence and a time-interval sequential pattern are clarified. First, each time-interval occurs after an item was taken place but before the following item appeared. Therefore, the time-interval and the item occurred before it is treated as a unit when conducting the edit distance evaluation in the dynamic programming. Second, item comparison is conducted first when calculating the dissimilarity between a sequence and a pattern. Whenever items are the same in the sequence and the pattern, dissimilarity between the following time-intervals is then evaluated. If items in the sequence are not the same with the item in the pattern, the dissimilarity between the time-intervals appeared after the two items are not worth to discuss.

Base on the above concept, the cost of taking required edit operations is summarized as (1):

Algorithm I-PrefixSpan(α, l, $PH^c|_\alpha$)
Parameter: α : a time-interval sequential pattern; l : the length of α;
c : the class label where $c = 1, 2, \ldots, n$;
Method:
(1) Scan $PH^c|_\alpha$ one time.
(2) If $l = 0$, then find all frequent items in $PH^c|_\alpha$.
(3) For every frequent item f, append f to α as α'.
(4) Output all α'.
(5) If $l > 0$, then construct *TI_Table* by scanning all the transactions in $PH^c|_\alpha$.
(6) For every frequent cell *TI_Table*(I_i, f), append (I_i, f) to α as α'.
(7) Output all α'.
(8) For each α' construct α'-projected database $PH^c|_{\alpha'}$, and call I-PrefixSpan (α', l+1, $PH^c|_{\alpha'}$)

Fig. 2 The pseudo-code of the I-PrefixSpan algorithm

$$Cost_{p,q} = \begin{cases} 1, & \text{if Insertion or Deletion} \\ 1, & \text{if Substitution (with different items)} \\ 0 + r \times \dfrac{\left| f(I_p) - f(I_q) \right|}{f(I_m) - f(I_0) + 1}, & \\ & \text{if Substitution (with the same items)} \\ 0, & \text{if No Change} \end{cases} \quad (1)$$

where p means the pth unit in the time-interval sequential pattern, q means the qth unit in the sequence. The ratio r, defined by users, is the maximum degree that time-interval dissimilarity can affect the cost. $f(I_b)$ is the rank of the time-interval I_b and is defined as $f(I_b) = b$ where $b = 0, 1, \ldots, m$.

The cost for edit distance is the degree of dissimilarity between a time-interval sequence and a time-interval sequential pattern. Therefore, the similarity between each edit operation is defined as 1 minus the cost of each edit operation, which is defined as:

$$Sim_{p,q} = 1 - Cost_{p,q} \quad (2)$$

Based on (2), the total similarity between the whole time-interval sequence and the whole time-interval sequential pattern is the sum of all $Sim_{p,q}$ and divides by the value of the longest length between the sequence and the sequential pattern. The similarity of the ith time-interval sequential pattern and the jth sequence is summarized in (3) where $p = 1, 2, \ldots, l_s$ and $q = 1, 2, \ldots, l_p$, l_s denotes the length of the time-interval sequence and l_p denotes the length of the time-interval sequential pattern. This normalization operation is set to avoid the condition the longer sequences obtain higher similarity value with a specific pattern which might affects the fairness of similarity measure.

$$Sim = \frac{\sum_{p}^{l_s} \sum_{q}^{l_p} Sim_{p,q}}{\max(l_s, l_p)} \quad (3)$$

where $p = 1, 2, \ldots, l_s$ and $q = 1, 2, \ldots, l_p$.

C. *Framework of the Classification Method*

Similar to the framework of the classification model proposed by Exarchos *et al*. [9], which was originally applied to evaluate the biological sequence dataset, the proposed framework is designed to classify time-interval sequences. In the proposed classification method, all sequences

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$S \in \{PH^c | c = 1, 2, ..., n\}$ and all the frequent sequential patterns $F \in \{FP^c | c = 1, 2, ..., n\}$ are used as input to the classification model, and the output is a confusion matrix (CM) denoting the classification accuracy information. Let sequences in database PH^c be marked with class labels denoted as database $PH^c = \{S_j, c\}$, where S_j represents the *j*th sequence and *c* represents the class sequence belongs, with *n* is the number of classes and *ns* is the overall number of sequences in database PH^c, where $c = 1, 2, \ldots, n$. In addition, the number of time-interval sequential patterns in *F* is presented as q_c where $c = 1, 2, \ldots, n$. The classification method is divided into four steps, and they are described as follows:

Step 1: Calculation of the pattern score matrices. After the extraction of the time-interval sequential patterns, we use the sequences and the time-interval sequential pattern sets to construct the pattern score matrix for each class. The pattern score matrices are denoted as $\mathrm{PSM}^c = [y_{ckj}]$ where $c = 1, 2, \ldots, n$, $k = 1, 2, \ldots, q_c$. Symbol *c* denotes the *c*th PSM, *k* denotes the *k*th row of PSM^c, *j* means the sequence S_j. Every element in PSM^c, denoted as y_{ckj}, stands for a score which defines the similarity of the *k*th time-interval sequential pattern within class *c* between the sequences S_j. Thus, the size of a PSM is $q_c \times ns$.

Step 2: Update of the pattern score matrices. Each row of the pattern score matrix is multiplied by a parameter which represents the weight of a specific time-interval sequential pattern in the class. That is, y_{ckj} in the same row are multiplied by same weight. Therefore, each PSM^c is updated as:

$$row_c^k = w\_pattern_c^k \times row_c^k \tag{4}$$

where row_c^k denotes the *k*th row in the *c*th PSM, and the $w\_pattern_c^k$ is the weight of the *k*th time-interval sequential pattern in the *c*th pattern set. The weights are generated from an efficient particle swarm algorithm, which will be introduced in next section.

Step 3. Calculation of the class score matrix. The class score matrix, denoted as CSM, is derived from the updated PSM^c. Assume CSM = $[x_{cj}]$ where each x_{cj} is the element of CSM represents the score of similarity of the sequences S_j belonging to the *c*th class. As mentioned above, the row size of PSM^c depends on the number of time-interval sequential patterns the *c*th class contains. Therefore, the more time-interval sequential patterns in the class *c*, the higher score the *c*th row of CSM. In order to prevent such unfair situation, every cell of the CSM has to be normalized. The score of similarity of sequences S_j in the *c*th class in CSM is defined as:

$$x_{cj} = \frac{\sum_{k=1}^{q_c} y_{ckj}}{q_c} \tag{5}$$

where $c = 1, 2, \ldots, n$, $k = 1, 2, \ldots, q_c$, and $j = 1, 2, \ldots, ns$. (5) is the sum of the score of all the patterns in *c*th class for the sequences S_j divided by the number of patterns the *c*th class contained. Therefore, the size of the class scored matrix CSM is $n \times ns$.

Step 4. Update of the class score matrix. Each row of the class score matrix is multiplied by a parameter which represents the weight of a specific class. That is, x_{cj} in the same row is multiplied by the same weight. Thus, each row of the class score matrix is updated as:

$$row_c = w\_class_c \times row_c \tag{6}$$

where *c* denotes the *c*th row of CSM and the w_class_c is the weight of the *c*th class. The weights are also generated from particle swarm algorithm introduced in the next section.

For each sequence S_j, a class is predicted base on the updated CSM. The predicted class (pc_j) is determined after estimating the class which got the highest score for the sequence S_j: $pc_j = \max_{c\,=\,1,\,2,\ \ldots,\ n} (\mathrm{CSM}(c, j))$. Based on the real class rc_j and the predicted class pc_j of each sequence, the confusion matrix (CM) for all sequences in database is then calculated. The rows of the confusion matrix (CM) represent the predicted class while the columns of the CM represent the actual class. The diagonal of the confusion matrix denotes the number of the correctly classified sequences for each class. Therefore, *cor_cla*(CM) is the function that summarizes the diagonal elements of the confusion matrix CM. The larger the value of *cor_cla*(CM) is, the more accurate the classification result.

D. The Optimization Process for Classification Model

Two sets of weights, *w_class* for each row of the class score matrix (CSM) and *w_pattern* for each row of pattern score matrices (PSM) have been defined above. However, the values of the two sets of weights (*w_pattern* and *w_class*) might dramatically affect the accuracy of the proposed classification model. Therefore, the optimal values of the two sets of weights should be derived so that the best classification result can be obtained. This research employs particle swarm optimization (PSO) algorithm, a global optimization search approach, to find the best *w_class* and *w_pattern*. PSO algorithm not only is easy to code but also the run time is short.

The objective function of this optimization problem is then defined as:

$$\min\ f(\mathrm{CM}) = ns - cor\_cla(\mathrm{CM}) \tag{7}$$

where *ns* is the number of all sequences in database *PH*, and *cor_cla*(CM) is the number of correctly classified sequences in CM described in last section. Notes that *f*(CM) is the fitness function in the PSO algorithm. The minimal value of the objective function is 0 when all sequences are correctly classified (*ns* = *cor_cla*(CM)). In addition, in this research, a particle is represented as:

$$x_i = [\, w\_pattern_1^1, w\_pattern_1^2, \cdots, w\_pattern_1^{q_1}, w\_pattern_2^1, w\_pattern_2^2, \ldots, w\_pattern_2^{q_2}, \cdots, w\_pattern_n^1, w\_pattern_n^2, \cdots, w\_pattern_n^{q_n}, w\_class_1, w\_class_2, \cdots, w\_class_n \,].$$

where the total dimension of the particle is $(\sum_{c=1}^{n} q_c) + n$.

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE 1 THE DATASET

PH^c	PH		
	sid	S_i	c_i
PH^1	1	<(A, 3) (B, 5) (C, 25) (C, 35) (D, 72)>	1
	2	<(A, 5) (B, 7) (E, 41) (C, 70)>	1
	3	<(B, 4) (C, 6) (A, 37) (A, 65)>	1
	4	<(A, 10) (C, 49) (D, 52) (C, 106)>	1
	5	<(B, 2) (A, 27) (B, 35)>	1
	6	<(A, 1) (C, 36) (D, 61) (D, 70)>	1
	7	<(A, 2) (B, 7) (C ,20) (D, 25)>	1
	8	<(A, 3) (B, 15) (D, 17)>	1
PH^2	9	<(B, 7) (D, 21) (D, 44) (B, 53)>	2
	10	<(A, 6) (A, 13) (D, 25) (D, 46)>	2
	11	<(B, 8) (B, 32) (D, 43) (C, 79) (D, 85)>	2
	12	<(B, 1) (D, 8) (A, 22) (D ,58)>	2
	13	<(C, 3) (B, 17) (D, 29) (C, 41)>	2
	14	<(C, 12) (B, 18) (D, 23)>	2
	15	<(B, 12) (A, 21) (D, 23) (B 26) (D, 30)>	2
	16	<(B, 3) (B, 7) (C, 12) (D, 18)>	2
PH^3	17	<(C, 3) (B, 15) (E, 21)>	3
	18	<(C, 9) (B, 17) (A, 30) (E, 39)>	3
	19	<(A, 3) (B, 10) (C, 23) (E, 37)>	3
	20	<(C, 16) (A, 28) (E, 31) (A, 45)>	3
	21	<(B, 4) (C, 32) (B, 38) (E, 50) (A, 82)>	3
	22	<(B, 4) (C, 11) (E, 29) (A, 63)>	3
	23	<(A, 1) (C, 5) (E, 8) (A, 16)>	3
	24	<(C, 9) (B, 12) (E, 16) (A, 32)>	3

TABLE 2 TIME-INTERVAL SEQUENTIAL PATTERNS IN EACH CLASS

Time-interval sequential patterns		
FP^1	FP^2	FP^3
$A\rightarrow I_1\rightarrow B$	$A\rightarrow I_1\rightarrow D$	$A\rightarrow I_1\rightarrow E$
$A\rightarrow I_4\rightarrow C$	$A\rightarrow I_2\rightarrow D$	$B\rightarrow I_1\rightarrow E$
$A\rightarrow I_5\rightarrow D$	$A\rightarrow I_4\rightarrow D$	$B\rightarrow I_2\rightarrow A$
$B\rightarrow I_2\rightarrow C$	$B\rightarrow I_1\rightarrow C$	$B\rightarrow I_3\rightarrow E$
$C\rightarrow I_1\rightarrow D$	$B\rightarrow I_1\rightarrow D$	$B\rightarrow I_5\rightarrow A$
$C\rightarrow I_4\rightarrow D$	$B\rightarrow I_2\rightarrow D$	$C\rightarrow I_1\rightarrow B$
$A\rightarrow I_1\rightarrow B\rightarrow I_2\rightarrow C$	$B\rightarrow I_4\rightarrow D$	$C\rightarrow I_2\rightarrow E$
$A\rightarrow I_4\rightarrow C\rightarrow I_4\rightarrow D$	$B\rightarrow I_5\rightarrow C$	$C\rightarrow I_3\rightarrow A$
	$B\rightarrow I_5\rightarrow D$	$C\rightarrow I_5\rightarrow A$
	$C\rightarrow I_1\rightarrow D$	$E\rightarrow I_2\rightarrow A$
	$D\rightarrow I_1\rightarrow B$	$E\rightarrow I_4\rightarrow A$
	$D\rightarrow I_3\rightarrow D$	$C\rightarrow I_1\rightarrow B\rightarrow I_2\rightarrow A$
	$A\rightarrow I_2\rightarrow D\rightarrow I_3\rightarrow D$	$C\rightarrow I_2\rightarrow E\rightarrow I_4\rightarrow A$
	$B\rightarrow I_1\rightarrow C\rightarrow I_1\rightarrow D$	
	$B\rightarrow I_5\rightarrow C\rightarrow I_1\rightarrow D$	

TABLE 3 THE SET OF WEIGHTS OF PSM FOR FP^1.

FP^1	$w\_pattern_1^*$
$A\rightarrow I_1\rightarrow B$	1.1941
$A\rightarrow I_4\rightarrow C$	1.0806
$A\rightarrow I_5\rightarrow D$	0.9217
$B\rightarrow I_2\rightarrow C$	1.0010
$C\rightarrow I_1\rightarrow D$	1.1032
$C\rightarrow I_4\rightarrow D$	1.7247
$A\rightarrow I_1\rightarrow B\rightarrow I_2\rightarrow C$	1.1946
$A\rightarrow I_4\rightarrow C\rightarrow I_4\rightarrow D$	1.0987

TABLE 4 THE SET OF WEIGHTS OF PSM FOR FP^2.

FP^2	$w\_pattern_2^*$
$A\rightarrow I_1\rightarrow D$	0.7162
$A\rightarrow I_2\rightarrow D$	1.1293
$A\rightarrow I_4\rightarrow D$	0.9739
$B\rightarrow I_1\rightarrow C$	1.0901
$B\rightarrow I_1\rightarrow D$	1.1003
$B\rightarrow I_2\rightarrow D$	0.9891
$B\rightarrow I_4\rightarrow D$	1.0015
$B\rightarrow I_5\rightarrow C$	1.0857
$B\rightarrow I_5\rightarrow D$	0.8625
$C\rightarrow I_1\rightarrow D$	0.7379
$D\rightarrow I_1\rightarrow B$	1.3321
$D\rightarrow I_3\rightarrow D$	0.9968
$A\rightarrow I_2\rightarrow D\rightarrow I_3\rightarrow D$	1.0981
$B\rightarrow I_1\rightarrow C\rightarrow I_1\rightarrow D$	0.9603
$B\rightarrow I_5\rightarrow C\rightarrow I_1\rightarrow D$	0.9719

TABLE 5 THE SET OF WEIGHTS OF PSM FOR FP^5.

FP^3	$w\_pattern_3^*$
$A\rightarrow I_1\rightarrow E$	0.8701
$B\rightarrow I_1\rightarrow E$	1.0228
$B\rightarrow I_2\rightarrow A$	1.0129
$B\rightarrow I_3\rightarrow E$	0.9729
$B\rightarrow I_5\rightarrow A$	0.9866
$C\rightarrow I_1\rightarrow B$	1.0598
$C\rightarrow I_2\rightarrow E$	1.0049
$C\rightarrow I_3\rightarrow A$	1.1249
$C\rightarrow I_5\rightarrow A$	1.0071
$E\rightarrow I_2\rightarrow A$	0.9980
$E\rightarrow I_4\rightarrow A$	0.9218
$C\rightarrow I_1\rightarrow B\rightarrow I_2\rightarrow A$	1.0738
$C\rightarrow I_2\rightarrow E\rightarrow I_4\rightarrow A$	0.9677

III. EXPERIMENT RESULTS

A. Introduction to Dataset

Let the merchandise itemset be $I = \{A, B, C, D, E\}$ where each alphabet represents an item. The synthetic dataset includes twenty-four transaction records which imitates customer purchasing behaviors. Customers are divided into three classes ($n = 3$). That is, eight sequences belong to the same class. Table 1 show the dataset. Note the unit of time in datasets is a day. For example, customer sid 1 buys item *A* in day 3, buys item *B* in day 5, then buys *C* in day 25 and so on. In each class, six sequences are used as training data to create the sequence classification model and the remaining two sequences are used for testing. Hence, the training dataset PH_{train} consists of 18 sequences ($|PH_{train}|=18$) and the test dataset PH_{test} consists of 6 sequences ($|PH_{test}|=6$).

B. Generation of the Time-Interval Cassification Model

Any sequence in the synthetic dataset *PH* belongs to one of the three classes. Therefore, $PH = \{S_i, c_i\}$ with S_i being the sequence and c_i being the corresponding class. The synthetic dataset is divided to three sub-datasets when mining frequent time-interval sequential patterns by I-Prefixspan algorithm. Assume that the value of support is set as 2, and the set of time-intervals $TI = \{I_0, I_1, I_2, I_3, I_4, I_5\}$, where I_0: $t=0$, I_1: $0<t\le 10$, I_2: $10<t\le 20$, I_3: $20<t\le 30$, I_4: $30<t\le 40$, and I_5: $40<t\le\infty$. After conducting I-Prefixspan to each sub-dataset, the generated time-interval sequential pattern sets are denoted as FP^c where c = 1, 2, 3 and are shown in Table 2.

The PSO algorithm is employed to the optimization process. The parameters in the PSO such as population of particles, *pn*; the inertial constant, *w*; the cognitive parameter, c_1; and the social parameter, c_2, and the restriction of moving velocity of particles (v_{max}) are all set by user. The inertial constant, *w*, is usually set within the range 0.5 to 1. Therefore, *w* is set as 0.5 plus a random value divided to 2 in accordance with advice of the literatures [11] Further, the restriction of cognitive and social parameter c_1 and c_2 is $c_1+c_2\le 4$, for fast convergence, c_1 is set as 1.5, and c_2 is set as 1.5, too. And the population of particles, *pn*, is set as 20. However, in order to remain value of *w_pattern* and *w_class* positive and stable, velocity v_i of

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

particle p_i where i = 1, 2, …, 20, is restricted in the range

TABLE 6 THE SET OF WEIGHTS OF CSM.

row_i of CSM	w_class
row_1 (c_1)	1.0438
row_2 (c_2)	1.1897
row_3 (c_3)	0.9814

TABLE 7 THE OPTIMIZED CLASSIFICATION RESULT.

S_i	*Original Class* (c_i)	*Prediction Witout Optim.* (pc_i)	*Prediction With Optim.* (pc_i)
<(*A*, 3) (*B*, 5) (*C*, 25) (*C*, 35) (*D*, 72)>	1	1	1
<(*B*, 4) (*C*, 6) (*A*, 37) (*A*, 65)>	1	3	**1***
<(*A*, 10) (*C*, 49) (*D*, 52) (*C*, 106)>	1	1	1
<(*A*, 1) (*C*, 36) (*D*, 61) (*D*, 70)>	1	3	**1***
<(*A*, 2) (*B*, 7) (*C* ,20) (*D*, 25)>	1	1	1
<(*A*, 3) (*B*, 15) (*D*, 17)>	1	2	2
<(*B*, 7) (*D*, 21) (*D*, 44) (*B*, 53)>	2	3	**2***
<(*A*, 6) (*A*, 13) (*D*, 25) (*D*, 46)>	2	2	2
<(*B*, 8) (*B*, 32) (*D*, 43) (*C*, 79) (*D*, 85)>	2	2	2
<(*C*, 12) (*B*, 18) (*D*, 23)>	2	2	2
<(*B*, 12) (*A*, 21) (*D*, 23) (*B* 26) (*D*, 30)>	2	2	2
<(*B*, 3) (*B*, 7) (*C*, 12) (*D*, 18)>	2	1	1
<(*C*, 3) (*B*, 15) (*E*, 21)>	3	3	3
<(*C*, 9) (*B*, 12) (*E*, 16) (*A*, 32)>	3	3	3
<(*C*, 16) (*A*, 28) (*E*, 31) (*A*, 45)>	3	3	3
<(*B*, 4) (*C*, 32) (*B*, 38) (*E*, 50) (*A*, 82)>	3	3	3
<(*B*, 4) (*C*, 11) (*E*, 29) (*A*, 63)>	3	3	3
<(*C*, 9) (*B*, 17) (*A*, 30) (*E*, 39)>	3	3	3

[-2.5, 2.5], and is recalculated until the updated value of particle p_i is positive. Last, to avoid too extremely large value of variables of dimension of the particles, the dynamic range of variables of dimensions in each particle is restricted smaller than 5. Table 3 to Table 6 display the output of PSO optimization process.

Based on the updated PSM(s) and CSM, the predictions for each sequence can be derived, as it is shown in Table 7. According to Table 7, the sequence <(*B*, 4) (*C*, 6) (*A*, 37) (*A*, 65)>, <(*A*, 1) (*C*, 36) (*D*, 61) (*D*, 70)>, and <(*B*, 7) (*D*, 21) (*D*, 44) (*B*, 53)> were incorrectly classified without using optimization process are now correctly classified as their original class with using the PSO optimization process. Thus, the accuracy of the classification method is increased by 11.17 % (from 77.72 % without using optimization process to 88.89 % with using the PSO optimization process).

To confirm the fairness and reliability of classification result, 4-fold cross-validation is employed. The synthetic dataset is divided into four groups as sample sets, and each group contains six sequences which include two sequences from each class (c = 1, 2, 3). One of sample sets of the synthetic dataset is selected as testing data, the rest three sample sets are used as training data which are employed to build the classification model. The experiment is repeated for four times so that every group of sample sets is employed as testing data for the classification model. The classification result of the first fold is displayed as above. After the experiment, four classification results are obtained and shown in Table 8 and Table 9.

IV. CONCLUSION

Nowadays, how to seizing customer behavior and making proper business strategies to different types of customers are

TABLE 8 CLASSIFICATION ACCURACY OF TRAINING DATA.

Experiment Number	Without optimization	After optimization	Improvement
fold-1	72.22%	88.89%	16.67%
fold-2	61.11%	66.67%	5.56%
fold-3	55.56%	83.33%	27.77%
fold-4	61.11%	77.78%	16.67%
Average	62.50%	79.17%	16.67%

TABLE 9 CLASSIFICATION ACCURACY OF TESTING DATA.

Experiment Number	Without optimization	After optimization	Improvement
fold-1	83.33%	83.33%	0.00%
fold-2	33.33%	50.00%	16.67%
fold-3	50.00%	50.00%	0.00%
fold-4	66.67%	83.33%	16.66%
average	58.33%	66.67%	8.33%

important issues for enterprises. This research proposes a time-interval sequence classification method considering the time-interval information between items. With the proposed classification method, a new customer can be easily classified as one of previous known customer classes according to his/her time-interval sequence. Once the class of a customer is known, decision makers can develop different business strategies to different customers at the right time. The classification accuracy is greatly improved after applying the particle swarm optimization process. There are two major contributions of this research. First, a new similarity measure and evaluation approach for time-interval sequences have been proposed. Second, this research proposes an effective classification method involving particle swarm optimization process for time-interval sequences.

REFERENCES

[1] C. Y. Tsai, C. Y. Tsai, and P. W. Huang, "An association clustering algorithm for can-order policies in the joint replenishment problem," *International Journal of Production Economics*, vol. 117, 2009, pp .30-41.
[2] C. Y. Tsai and Y. C. Shieh, "A change detection method for sequential patterns," *Decision Support Systems*, vol. 46, 2009, pp. 501–511.
[3] C. S. Wang and A. J. T. Lee, "Mining inter-sequence patterns," *Expert Systems with Applications*, vol. 36, 2009, pp. 8649–8658.
[4] D. S. Huang, X. M. Zhao, G. B. Huang, and Y. M. Cheung, "Classifying protein sequences using hydropathy blocks," *Pattern Recognition*, vol. 39, 2006, pp. 2293–2300.
[5] J. G. O. S. J. Joung and B. T. Zhang, "Protein sequence-based risk classification for human papillomaviruses," *Computers in Biology and Medicine*, vol. 36, 2006, pp. 353–667.
[6] D. Bouchaffra and J. Tan, "Structural hidden Markov models using a relation of equivalence: application to automotive designs," *Data Mining and Knowledge Discovery*, vol. 12, 2006, pp. 79–96.
[7] A. D. Bruyn, J. C.Liechty, E. K. R. E.Huizingh, and G. L Lilien. "Offering online recommendations with minimum customer input through conjoint-based decision aids," *Marketing Science*, 2008, pp. 1–38.
[8] Y. L. Chen, M. C. Chiang, and M. T.Ko, "Discovering time-interval sequential patterns in sequence database," *Expert Systems with Applications*, vol. 25, 2003, pp. 343–354.
[9] T. P. Exarchos, M. G. Tsipouras, C. Papaloukas, and D. I. Fotiadis, "A two-stage methodology for sequence classification based on sequential pattern mining and optimization," *Data Mining and Engineering*, vol. 66, 2008, pp. 467–487.
[10] O. Arbell, G. M. Landau, and J. S. B. Mitchell, "Edit distance of run-length encoded strings," *Information Processing Letters*, vol. 83, 2002, pp. 307–314.

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[11] S. W. Lin and S. C. Chen, “PSOLDA: A particle swarm optimization approach for enhancing classification accuracy rate of linear discriminant analysis,” *Applied Soft Computing*, vol. 9, 2009, pp. 1008–1015.

ISBN: 978-988-17012-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)