International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **4**, April **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 2491–2511

CELLULAR LEARNING AUTOMATA FOR MINING CUSTOMER BEHAVIOUR IN SHOPPING ACTIVITY

Mansour Esmaeilpour[1], Vahideh Naderifar[1] and Zarina Shukur[2]

[1]Department of Computer Engineering
Hamedan Branch, Islamic Azad University
Hamedan, Iran
{ esmaeilpour; naderifar }@iauh.ac.ir

[2]Faculty of Information Science and Technology
National University of Malaysia
43600 Bangi, Selangor, Malaysia
zs@ftsm.ukm.my

Received January 2011; revised May 2011

Abstract. *Customer behavior mining in shopping activates is important from two perspectives namely, the perspective supplier of goods and the perspective of shop owners. Both groups want to know that their customers' interest in which goods and identify which sequence of the goods that are most popular. The latter group will even change the manner of arrangements of goods in their shops, manner of good order, and manner of warehousing and marketing. Through the information about the fact that customers have bought which products in sequence, suppliers of goods and shop owners could gain profits by providing suitable arrangements in the shelves. Sequence mining is essential for suppliers of goods and shop owners and will lead to an increase in annual profit. This research provides a way in sequences mining of the customer shopping. The research demonstrates that when dataset scanning is repeated several times, it could obtain sequences of the customer shopping in shorter running time. In addition to this, the research also provides a method of finding two-members and higher sequences by Cellular Learning Automata. Its cost is lower than the Apriori, FP-growth and FP-tree based on array, because of the number of total scan on the dataset. Test was done on an online basket data of costumer shopping from UCI Machine Learning and it is clear that the results of this research are much better than other works.*
Keywords: Customer activity, Data mining, Cellular learning automata, Shopping activity, Sequence pattern mining

1. **Introduction.** Study of frequency patterns has been one of the most significant and considerable subjects in previous researches in the field of data surveying. A great number of articles and scientific works have been published here and remarkable progresses are achieved. It is true that study of frequency patterns causes significant advances in the view of data analysis and has great effect in improving methodology and application of data surveying. Frequency patterns are a set of items, sequences, or infrastructures that can be repeated in a data set or what is determined by user as threshold limit, for example, purchasing behavior of the customer that shows how goods are purchased together. Several items (bread, milk, butter and honey) that are usually used in existing transactions or in transaction base of a store are some of the examples of a set of frequency items. In a sequence, apart from the items, uses of time intervals in a transaction are important. For example, that we purchase a computer, that we purchase a desk for the computer, and that we purchase game CDs, are accounted as a sequence in dataset of a store, which indicates that four out of five of the customers purchase computer at first,

and then purchase a desk. Similarly, four out of five of the customers buy bread at a supermarket and then buy milk sequentially. An infrastructure may include different kinds of structures like sub-graph, sub-tree and sub-network, in which it is possible to mix with sub-sequences sets of items. If a structure repeats in a graphic base indistinctly, it is called frequency structure patterns. Context of frequency pattern has a key role in association patterns study, correlation and for most of data interrelationships [25]. Thus, frequency pattern study is one of the significant tasks in data surveying for which hundreds of articles have been published about study of frequency algorithms in the past decade. Now, it will be studied efficient and measurable sub-methods of frequency pattern. Concept of frequency items was used for studying information on transaction base for the first time.

Assume that $i = \{i_1, i_2, \cdots, i_m\}$ is set of all existing items in a transaction base like D. Set of item K items of α that includes K-item of set i will be frequency if and only if the existing transactions in D included in α were not less than $(\theta|\mathrm{D}|$. Here θ is a determined threshold limitation by the user, which is called minimum supporting and $|\mathrm{D}|$ is number of existing transactions in D. θ is between 0 and 1 that shows a percent of D transactions, which included α. Included α means that a transaction has all existing items in α.

Frequency patterns are set of items, sequences, or infrastructures that could be repeated in a data set [21] or what is determined by user as threshold limit; for example purchasing behavior of the customer that shows how goods are purchased together.

Look at the following example. Consider the data base shown in Table 1 over the set of item:

$$\mathrm{I} = \{\mathrm{bread}, \mathrm{milk}, \mathrm{butter}, \mathrm{honey}\}$$

Table 1. Shows the transaction dataset D

$\mathrm{T_{id}}$	X
100	{bread, milk, butter}
200	{bread, milk}
300	{butter, honey}
400	{milk, butter}

All D sequence items with at least the support of one number and numbers of their repeats are presented in Table 2.

Table 2. Shows the item sets and their support in D

Itemset	Cover	Support	Support
{ }	{100, 200, 300, 400}	4	100%
{bread}	{100, 200}	2	50%
{milk}	{100, 200, 400}	3	75%
{butter}	{100, 300, 400}	3	75%
{honey}	{300}	1	25%
{bread, milk}	{100, 200}	2	50%
{milk, butter}	{100, 400}	2	50%
{butter, honey}	{300}	1	25%
{bread, honey, milk}	{100}	1	25%

If $\theta = 50\%$ is in all the problems, it means that the threshold limit is considered to be 50%, and all items with less than 50% frequency will be omitted from Table 3. And thus, the search for frequency patterns will take place in the above items by using appropriate patterns.

TABLE 3. An example from frequency pattern with min-support 50%

Itemset	Cover	Support	Support
{ }	{100, 200, 300, 400}	4	100%
{bread}	{100, 400}	2	50%
{milk}	{100, 300, 400}	3	75%
{butter}	{200, 400}	2	50%
{honey}	{100, 300}	2	50%

In Section 2, literature review will be demonstrated. Then, in Section 3 the training method that is CLA is explained. In Section 4 the demonstration of the steps taken as well as the implementation is discussed. The experiment of this approach on several online datasets which aim at the representation of operational power of the proposed model is presented in Section 5. The result will be compared with against methods. Section 6 concludes the work.

2. **Literature Review.** Studying of frequency patterns was first developed by Agrawal in 1993 followed with extended ideas of his work by [1] as discussed in the introduction section of this article. In almost all cases, the goal of studying of frequency pattern in those works had been motivated to analysis shopping basket within the framework of association rules of study. Results of those studies demonstrate that customers, who buy special goods, will have desires to buy some other goods. So, stores' owners could only set the goods in the shelves according to needs of the customers. Agrawal found an interesting specification, called Apriori in frequency patterns set for the first time in 1994 [2]. According to this specification, a K item set is a frequency just when all its subsets are frequency. The main result of this work is a non-frequency of super patterns of a frequency pattern that will make it possible for removal of a non-frequency K item-set when studying $(K + 1)$ items for algorithm exploration. So that all frequency of item-sets can be found by the first scanning to find one item-set, then using the frequency of one item-set to make all frequency of two item-sets. To have confidence of these frequency, each of the two item-sets of candidates of items that are derived from the first item-sets, must be scanned one more time. Therefore, these repetition processes help to distinguish whether these two item-sets are frequency items or non-frequency items. After determining the outcome of the two frequency sets of items, a similar application can frequently be made to the two-item-sets which in turn are used to make three-item-sets. These candidates are checked in the dataset whether it is frequency or non-frequency. This process will continue to extract all frequency items existing in the transaction base till no more frequency of k item-sets can be made relevant any k items. This is the essence of Apriori algorithm.

Following the discovery of Apriori algorithm, a voluminous of article has been written with efforts of wide studies which were done to improve of its efficiency and popularization. Although in most of the problems and samples, Apriori method can cause significant decrease in size of the produced candidates by principle of "non-frequency of super pattern or a non- frequency pattern" there are always two problems exist in Apriori. These problems can be mentioned as follows:

1. Generation significant number of candidates in the framework of abundant candidate.
2. Continues exploration of the transaction of dataset in order to adjust the produced candidates with the transaction for distinguishing frequency or non-frequency.

According the above statement, the disadvantages to find 100 elements of sequence is that we must for example; produce 2^{100} candidates to apply Apriori principle. It means that time complexity is 2^n and the number of dataset that can be scanned $(n+1)$ times, so that n is all of the items which are available in the dataset [5]. When big dataset always needs more scans of dataset and become exorbitant; and generates vast candidates for which counting and saving each of them would very hard, boring, and time consuming. To solve such problems, some of the algorithms were used to optimize Apriori algorithm in decreasing transaction based scan and decreasing number of the candidates. The following works have underlined about optimized algorithms as follows:

In 1995, Park has provided a technique, named hashing technique for optimizing Apriori function [6]. This algorithm is commonly known as Direct Hashing & Pruning (DHP). The algorithm could minimize production of abundant number of candidates, which is less than that of Apriori, but this algorithm needs several scanning on the dataset.

Savasere has provided Partition algorithm to improve number of scans on the dataset [7]. In this technique, a large transaction of dataset is divided into number of parts, for which each part could be kept in main memory. Here, total dataset will be scanned at most two times. In the first scanning, the parts will be read and are placed in the main memory one by one and frequency patterns can be extracted from each part by helping of Apriori with ratio of the threshold limit. In the second scanning, this algorithm merges all related frequent item-sets of each part together. Then the supports of all this item-sets are computed during second scanning. For example, if the threshold limit was 10 for total of the dataset, this dataset will be divided into 10 parts and each part could have its own threshold limit. If some parts have threshold equal 1, it means that counting a lot of these patterns in the second scanning may be very expensive.

Sampling is another kind of optimized patterns that was proposed by Toivonen [6]. It performs at most two scanning through the dataset. This algorithm, instead of direct exploration of dataset, it picks sample of dataset and then finds frequent pattern in that sample. The sample could be as little as sufficiently in most rare cases, that algorithm may not be able to find all frequent pattern. In this case, the missing pattern may be found during second scanning on dataset. Studies based on this algorithm demonstrate that it is quicker than Apriori and Partition.

The last optimized method of Apriori that has been registered was in 2001 [8]. This algorithm calculates maximum numbers of selected patterns in different sizes (without repetition) by upper bound formula. For instance, if set L has 6 items, it calculates 13 patterns with 3 elements without repetition, 6 patterns with 4 elements without repetition and 1 pattern with 5 elements without repetition by upper bound formula. In this method, there is just one scan on dataset and number of candidates will be minimized and prevented from further production; but there are several disadvantages. The method presents just maximum numbers of selected patterns in different sizes, but does not demonstrate that these patterns constituted of what items.

As it is considered, breadth methods (according to candidates production) such as Apriori and its optimization patterns, they are not appropriate algorithms to find frequency patterns because they generate a lot of candidates or scanning dataset several times, and they are very expensive and time consuming.

Han invented a method, named FP-Growth that works according to in-depth method (without generation candidates) [9]. This algorithm explores all of frequency item-sets without generation candidate. This data structure is used by FP-Tree (frequent pattern tree). The method was based on a divide-and-conquer. In this method, the goal is to explore all frequency items in the transaction with once scanned and ordered and the items in the dataset base are assumed on descending order. The transaction base is compacted in the framework of a frequency pattern tree (FP-tree); and the tree will be created as follow. First the tree is created with the root node. Each transaction in the dataset should start from the root table and a branch is created for each transaction. Each node has a number which shows path of number of items in transaction that share with that node. Single items and their numbers that acounted are stored in the header table in decreasing order of their frequency. After this step, conditional FP-tree of the said pattern is made and exploration operation is applied recursively.

This application can be shown in the example of FP-Growth with min-supp=50% in Figure 1:

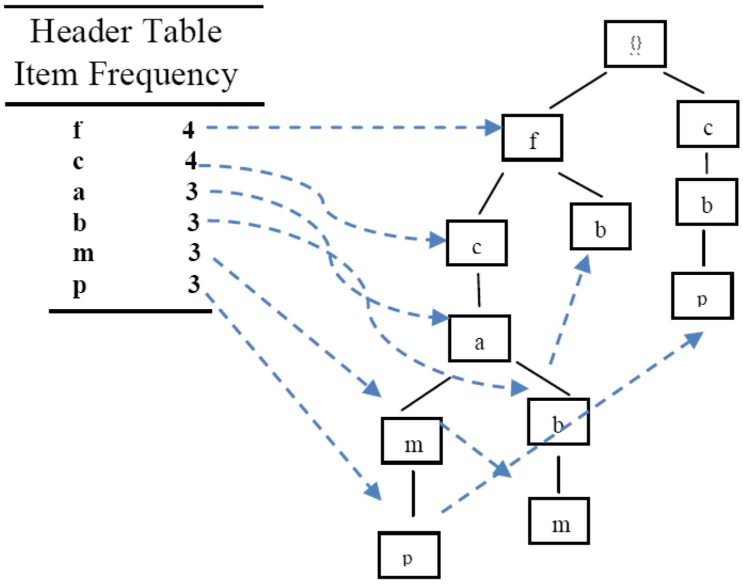


FIGURE 1. An example of FP-growth

TABLE 4. An example from dataset of FP-growth

T_{ID}	Items Bought	Frequent Item
100	{a,b,c,f,l,m,o}	{f,c,a,b,m}
200	{f,a,c,d,g,i,m,p}	{f,c,a,m,p}
300	{b,f,h,i,o}	{f,b}
400	{b,c,k,s,p}	{c,b,p}
500	{a,f,c,e,l,p,m,n}	{f,c,a,m,p}

TABLE 5. An example from frequency pattern with FP-growth method

Items	Conditional Pattern-base	Conditional FP-Tree	All Frequent Pattern
p	{(fcam:2), (cb:1)}	{(c:3)}|p	cp
m	{(fca:2), (fcab:1)}	{(f:3, c:3, a:3)}|m	fm, cm, am, fcm, fam, cam, fcam
b	{(fca:1), (f:1),(c:1)}	Empty	——
a	{(fc:3)}	{(f:3, c:3)}|a	fa, ca, fca
c	{(f:3)}	{(f:3)}|c	fc
f	Empty	Empty	——

Studies about efficiency of this algorithm have shown that this method will significantly decrease searching time and has the following advantages.

1. Data compact (construct FP-Tree)
2. Without generation of candidates
3. Quicker than Apriori
4. Two scans will be done on dataset

FP-Growth has also several disadvantages as follows:

1. FP-Growth may not fit on the main memory, when dataset is sparse, so size of FP-tree will be very high and a memory has not capacity of it.
2. Construct tree: FP-Tree is very expensive and needs a lot of time for constructing the tree, particularly if support was high.

To solve the above disadvantages, several methods are presented to improve FP-Growth, such as infrastructure exploring methods hyper structure mining (H-Mine), which was developed by Pei [11]. This method is used to find frequency patterns of exploring an infrastructure. Costs of production are less than FP-Growth; H-Min can use all available main memory space if necessary. H-Min operates in different data with high efficiency, even in macro and massive datasets. When data set was compressed, it would produce frequency patters as part of exploration of FP-tree processing. H-Min could not place large size of itself in the memory and may not be good idea to solve density data because of its limit space.

Grahne & Zhu have provided simplification method on the basis of array [10]. The main task of FP-Growth in the first scanning is, to find one item-set and in the second scanning is to create FP-Tree. Eighty percent (80%) of CPU time is spent for exploration of FP-tree which is very high. Here is a question; "Can we reduce exploration time to get all frequent pattern?" By using that simplification method on the basis of array, exploration time could be decreased. In FP-Growth, two scans were necessary. In the first scanning it was assumed to find one item-set with high repetition in transaction base and order this item-set in the main memory descending. In second one it was assumed to construct FP-tree and after that to start exploration and finding frequency patterns. This method is very expensive and time of the research is very long especially for sparse data base. To solve this problem, simplification method of the basis of array in comparison with that of FP-Growth which could decrease exploration times was provided. In this method two scans should be done on dataset, such as FP-Growth, in the first scanning which searches one item-set in the dataset and order them based on descending order like FP-Growth. In the second scanning, it creates the array while building FP-Tree. At first, each cell of array includes the numbers accounted of all the two item-sets for instance, cell A [e, f] is the counter for item-set {e, f}, and cell A [c, g] is the counter for item-set {c, g}, and so forth. In FP-Growth, to find each sequence, we should do scanning recursively from head table, but in the array, instead of scanning of the tree for each item such as i, we search the linked list that is started by i to get all frequent items.

Simplification method on the basis of array is more efficient, particularly in sparse data set but mini patterns base on FP-Tree are very big and bushy when data set is dense. FP-Tree is compressed and for each item in FP-Tree, exploration will be done quickly. Although counting the elements in the array takes more time in high density data sets, using the array needs time. Using array in dense data set may not be good idea; however, in sparse data sets, using mini method in the array is a good idea for traversal time. This algorithm could improve exploration time by two scans toward other algorithms in depth method. Mini method on the array could be accounted as the last effort to find optimized algorithm in depth methods (which is without generation of candidates). Another research that has been done on data mining techniques to predict customer relationship management was [28]. Many data mining techniques are studied, for example, survival analysis [18], SVM [30], decision tree [37], clustering [32], neural network, random forest [33], generic algorithm [34] and fuzzy system [36]. However, many of them are for prediction the customer relationship but the purpose of this research is customer purchase activity sequence mining and will be dealt in this article to provide "real-world" application.

3. **Background of the Cellular Learning Automata.** Tsypkin shown a model to solve the problems of determination of optimal parameter and apply hill climbing techniques [12]. Tsetlin started working on learning Automat at the same time [13]. The concept of Learning Automat was proposed by Tsetlin for the first time in 1973. Other researches introduced regarding the problems were finding an optimal action between permitted actions in stochastic automata [14]. However, most of attempts in learning automata have been done by Tsetlin. Varshavski and Vorontsova [15] introduced learning automata with a variable structure. Thereafter, automat updates its number of actions for which the result is the reduction of number states compression of deterministic Automata. The first attempts in this case were done by fu [16,17].

Learning Automata can be used as applications of parameter estimation, Pattern recognition, and Game theory applications. The applications of Learning Automata can be pattern recognition, graph partitioning and path planning [22]. Learning Automata can be imagined as an object with limited sets of actions. Learning Automata randomly chooses one of the actions among its sets of actions and sends it to environment. Automata can update its action of probability based on responses of an environment and the procedure that is repeated [23]. A learning Automata is composed of two parts:

1. A stochastic Automata with number of limited actions and a stochastic environment.
2. Learning algorithm which automata will learn optimal action by using that action.

Each action selected by potential environment is assessed and the answer is given to a learning Automata. Learning Automata uses this answer and selects its action for the next stage. Figure 2 shows the relationship between learning automata and environment [31].

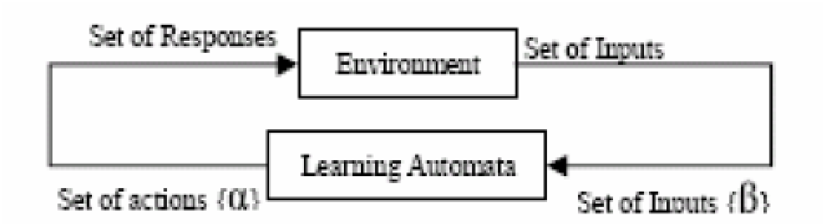


FIGURE 2. Environment and learning automata and relationship between them [31]

3.1. **Environment.** The formulation of environment can be derived by the following formula: E={α, β, c} where $\alpha = \{\alpha_1, \cdots, \alpha_m\}$ is a set of inputs $\beta = \{\beta_1, \cdots, \beta_m\}$ is a set of outputs and $c = \{c_1, c_2, \cdots, c_m\}$ is a set of penalty probabilities [15].

Furthermore, the formula of environment can be divided into the following three models:

- When β has two value, environment is called P-Model. $\beta_i = 1$ as penalty and $\beta_i = 0$ as reward.
- In the case of environment of Q-model, $\beta(n)$ is finite output of set with more than two values between $[0, 1]$.
- In S-Model, $\beta(n)$ is a continuous random variable within the range $[0, 1]$.

c_i is set of probability of an undesirable answer. In a stationary environment, values c_i remain unchanged while in non stationary environment, these values change over time. Learning Automata are classified into two groups with variable structures or fixed structure. We will describe Learning automata structure in the following statement when it is classified as variable structures.

In following it will be described variable learning automata structure.

3.2. **Stochastic learning automata.** Stochastic learning automata is shown with quintuple $\{\alpha, \beta, F, G, \phi\}$ which $\alpha = \{\alpha_1 \cdots \alpha_m\}$ is the set of automata actions, $\beta = \{\beta_1 \cdots \beta_m\}$ is the set of Automata inputs, F is production function and Automata new state. G is an output function that maps the current state and input into the current output. ϕ is the set of automata internal state. The set of α contains Automata action and the Automata chooses an action between its action and sends it to environment. β determined the set of Automata inputs. F and G map current state of Automata input into next output. As noted above, stochastic Learning Automata are divided into two group, fixed-structure Automata and variable-structure Automata. In fixed structure, automata is fixed probability of Automata action but in variable structure, Automata is updated probability of Automata action and each repetition is based on environment response without delay for which that update is done by Learning algorithm.

3.3. **Stochastic automata with variable structures.** Variable learning automata is shown with quadruple $\{\alpha, \beta, P, T\}$ which $\alpha = \{\alpha_1, \cdots, \alpha_m\}$ is a set of Automata actions, $\beta = \{\beta_1, \cdots, \beta_m\}$ is a set of Automata inputs, $p = \{p_1, \cdots, p_m\}$ is probability sequence for selection of each one of the action and Learning algorithm can be shown as follows:

$$P(n+1) = T[\alpha(n), \beta(n), p(n)] \tag{1}$$

3.4. **P-model leaning automata.** In the early activity, automata (r is the number of automata action), all probability of actions in any forms of automata would be equal. For an r-action Automata, the probability of action is given by $p_i(n) = 1/r$ with which in each repetition will change and update the value of this probability (based on reward or penalty). In this kind of Automata, if action α_i is selected between the other actions in this stage then $P_i(\alpha_i)$ receives the desirable answer and its probability will increase and the opposite probabilities will decrease. However, for undesirable answers, when the probabilities of $P_i(n)$ decrease the rest of the probabilities will increase. Anyway, changes are made in such a way that the sum of $P_i(n)$ is equal to one. This assumption has been shown below for both desirable answer and undesirable answer formula of learning algorithm in variable learning automata in P-Model [19]. The following algorithm is an example of linear learning algorithms in variable automata.

Desirable answer

$$\begin{array}{c} P_i(n+1) = P_i(n) + a[1 - P_i(n)] \\ P_j(n+1) = (1-a)P_j(n) \quad \forall j, j \neq i \end{array} \tag{2}$$

Undesirable answer

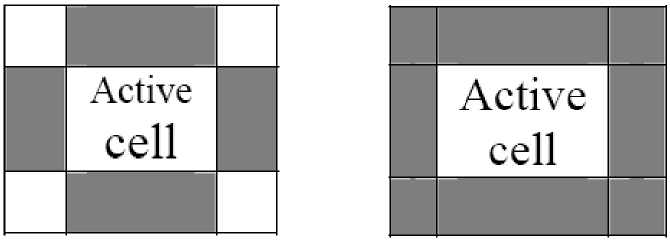
$$\begin{array}{c} P_i(n+1) = (1-b)P_i(n) \\ P_j(n+1) = (b/r - 1) + (1-b)P_j(n) \quad \forall j, j \neq i \end{array} \tag{3}$$

where $\underline{a}$ in above formula is reward parameter and $\underline{b}$ is penalty parameter and $\underline{r}$ is the number of actions.

Three states can be considered regarding the values of $\underline{a}$ and $\underline{b}$.

- $\mathrm{L_{RP}}$ (Linear Reward Penalty) when $a = b$.
- $\mathrm{L_{R\&P}}$ (Linear Reward Epsilon Penalty) when b is lower than a ($a \gg b$).
- $\mathrm{L_{RI}}$ (Linear Reward Inaction) when b equals zero ($b = 0$).

3.5. **Cellular automata.** Cellular Automata is a mathematic model to the system in which number of simple components cooperates to produce complex patterns. In fact, a cellular automaton is discrete dynamical system and communication between its cells is limited which is based on local interaction. CA composes of lattice of cells and set of rules [35]. Each square is called cell with each cell having two states that are black and white colors. The rules of cellular automaton determine how the states changes. In 1940's cellular automata was proposed by Von Neumann. After a short time it was suggested to adjust complex system manner by Ulam. Figure 3 are examples of well-known neighbor of CA as follow:



(a) (b)

FIGURE 3. Shows the Moore and Von Neumann neighborhoods

For each cell, there is a set of cells around active cell called neighbors. In Figure 3(a) the "neighbors" of a cell are eight squares interacted and in Figure 3(b), it is declared as Neighbor=(up, down, left, right). Cellular Automata is based on neighbor's behavior and past experience. The neighbor is defined as a set of cells around the activity of the cell such as for example with distance of two or less than two. Cellular Automata is based on some of the local rules and the rules of Automata can usually be designed by the user. In individual cell is affected by its local neighbor's rules. New state is created according to these local rules and its neighbors states. An automaton is updated based on local rules. The value of neighbors will impact the determination of new state of each cell [38]. Formally, a Cellular Automata can be defined as follows:

A d-dimensional Cellular Automata is a structure of $A = (Z^d, \phi, N, F)$ where

1. Z^d is a lattice of d-tuples of integer number of which this lattice could consist finite lattice, infinite lattice or semi-finite.
2. $\Phi = \{1, \cdots, m\}$ is a finite set of states.
3. $N = \{x_1, x_2, \cdots, x_m\}$ is a finite subset of Z^d called the neighborhood vector, ($x_i \in Z^d$).
4. F is the local rule of the cellular automata.

This rule can be defined by the users.

The rules cellular automaton determine how the states change and how the set of cells are neighbors with each other of which this rule can be defined by the users. Figure 4 is a sample of rules.

In this sample, there are eight rules because there are eight possible ways to set the states of the cell's three neighbors.

3.6. **Cellular automata characters.** Cellular automata, as a reinforcement learning has the following characters:

1. Cellular automat is discrete space.
2. Time goes discrete.
3. Each cells consists a number of limited state.

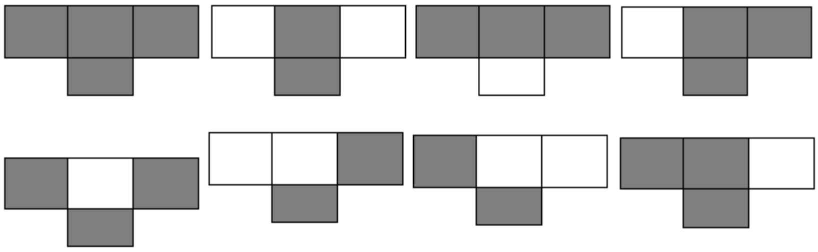


FIGURE 4. How create neighborhood of several cells

4. All the cells are in the same position.
5. All the cells are updated at the same time.
6. The rule in each site depends on the value of the site around its neighbors.
7. The rule for new value of each site just depends on value of limited number of previous states.

One of the big problems of Cellular Automata is determination final form rules which is needed for special user. On the other hands, all of considered rules are not permitted in Cellular Automata.

3.7. **Cellular leaning automata.** Cellular Automata is only unable to solve most of the problems collectively [38].

Regarding the weakness of Cellular Automata with combination of two recent models (Learning Automata and Cellular Automata), a new model called Cellular Learning Automata was created. It's a powerful mathematical model for many decentralized problems and phenomena. The main idea of CLA, which is a subclass of stochastic CA, is using Learning Automata to adjust the state transition probability of stochastic CA. A CLA is a CA in which Learning Automaton one or more than one is assigned to every cell. There is a rule that CLA operates under it. Those rules and the actions that are selected by the neighbors' LA determine reinforcement signal to LA resided in each cell. According to those rules and the previous states of neighbors, new transaction of dataset will give reward or penalty. Giving penalty or reward will update CLA structure. A CLA is formally defined as follows.

A d-dimensional cellular Learning Automata is a structure $CLA = \{Z^d, \phi, A, N, F\}$ where

1. Z^d is a lattice of d-tuples of integer number which this lattice could consist finite lattice ,infinite lattice or semi-finite.
2. $\Phi = \{1, \cdots, m\}$ is a finite set of states.
3. A is collection of learning automat (LA) each of which is assigned to one cell of the CLA. Each cell can have a LA or more than one.
4. $N = \{x_1, x_2, \cdots, x_m\}$ is a finite subset of Z^d called the neighborhood vector ($x_i \in Z^d$).
5. F is the local rule of the cellular automata. This rule can be defined by users.

The operation of Cellular Learning Automata could be described in the following statement.

Each Learning Automata in CLA chooses an action between its actions. Selection of action may be chosen based on random or pervious observation. Action selected causes movement of the Learning Automaton from one cell to another. The rule of Cellular Automata determines the reinforcement signal to each Learning Automaton residing in that cell as in Figure 5. Actions of Learning Automata in every active cell receive reward or penalty based on current Learning Automata of actions within LA and rules of neighbors.

Receiving reward or penalty updates internal structure of LA. Rewarding or giving penalties continues until the system arrives to a sustain state. Finally, each learning automata updates its action of probability vector based on supplied reinforcement signal and the action that are selected. This process continues until the desired result is obtained. The operations of updating Learning Automata structure in CLA are performed by a Learning algorithm.

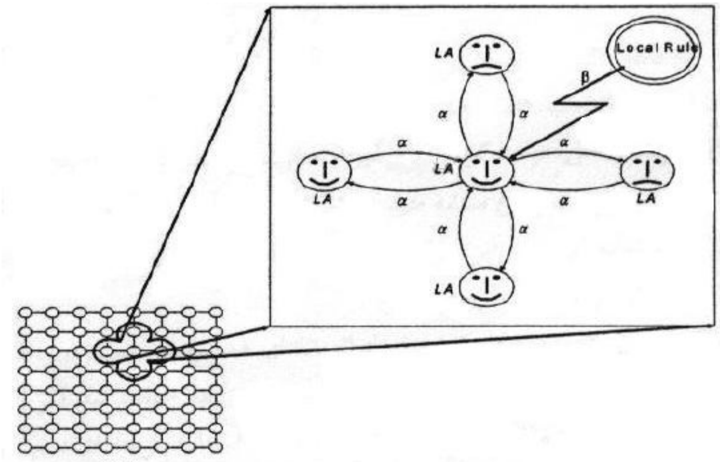


FIGURE 5. Cellular learning automata [35]

3.8. **Uniform learning automata.** A Cellular Learning Automata is called uniform if all cells are the same neighbor function, rule, Learning Automat, otherwise Learning Automata called non-uniform [38].

3.9. **Regular learning automata.** A Cellular Learning Automata is called regular if all cells (active cell and its neighbors) have the regular order. Regular order means, a one-dimensional or two-dimensional orders that is used by Von Neumann and Moore. Regular Learning Automata has used different applications so far. But some applications need model with no limitation of a regular lattice such as subsequence mining. Because in such network, nodes are completely point random and could not be considered a regular structure for that. This type of Automata is called irregular cellular learning automata. The only difference among them is how their neighbors are defined.

3.10. **Irregular learning automata.** Irregular cellular learning automata are a kind of CLA with only irregularity of the main different. ICLA doesn't have a restriction of lattice cells. ICLA is an undirected graph in which each vertex shows a cell which is resided with learning automata. The learning automaton determines its action based on its action of probability vector. Like CLA, there is a rule that the ICLA operate under it. The rule and action is selected by learning Automata of neighbors. LA will send a reinforcement signal for a leaning automat which is resided in an active cell. An ICLA is formally defined below:

A d-dimensional cellular Learning Automata is a structure $A = (G < E, \langle V \rangle, \phi, A, F)$ where

- G is an undirected graph which V is set of vertex and E is set of edge
- ϕ is set of finite states
- A is set of Learning Automat resided in each cell of Celloular Learning Atomata
- F is local rule of irregular cellular learning Automata in each vertex.

4. **Applying CLA in Customer Behavior Mining.** We will propose an algorithm to solve most of the problem of frequency pattern in Cellular Learning Automata. Our focus will be on algorithm of irregular uniform which has been used for Cellular Learning Automata. Uniform Cellular Learning Automata is used because each of the cells under study has the same neighborhood function. Applications of Local rules on Learning Automata, and irregular Cellular Learning Automata are used because the nodes are at quite random points and there could not be a special order considered such as Von Neumann and Moore. According to the definition of Cellular Learning Automata, Cellular Learning Automata is a CLA which has one or more than one Learning Automata in each cell. Learning Automata of each cell chooses one action among its actions. In this research, Learning algorithm of L_{RI} is used for all learning Automata. In each cell of CLA, Cellular Learning Automata is the same as items which is available in each row of the dataset. If each cell or cells of Cellular Learning Automata are repeated more than the min-support and the amount of its reinforcement arrive as much as threshold, they would be sent to the output as a sequence with higher repetition. Otherwise those cells would be reinforced and receive reward of Learning Automata which are actions Chosen that depends on the cell and its neighbors in the previous step.

- **Governing law on how to neighboring**

There are two groups of neighboring known as Semi-neighborhood and Reinforced and these neighborhoods are explained below.

A. **Semi-neighborhood**

Consider situation where i and$j$; j and k; k and i are neighbors but i, j, k are not nieghbors then there is semi-neighborhood that are established between them. According to Figure 6.

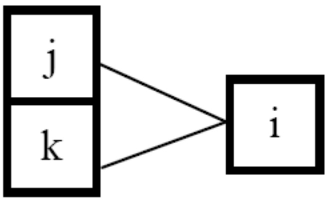


Figure 6. Shows the semi-neighborhood

According to Figure 7, if there are the adages from 1 to 3 and 2 to 3 then (1 and 3), (2 and 3) and (1, 2) are neighbors but 1, 2, 3 are not neighbor because they not connected with each other. But In Figure 8 can say 1, 2, 3 are neighbor with each other.

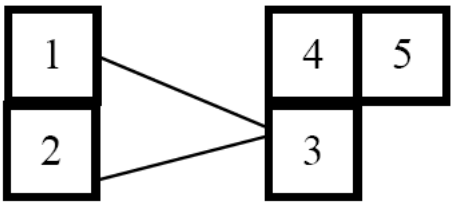


Figure 7. Shows the semi-neighborhood

B. **Reinforced neighbors**

All i neighbors are neighbors for k and j, if there is a connection between i neighbors to j and k. According to Figure 9, (1, 2, 3) are neighbors and 2, 4 are also neighbors

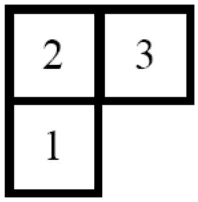


FIGURE 8. Shows the neighborhoods

because they are connected with each other but 1, 2, 3, 4 are not neighbors because they are not connected with each other. When 4, 1 and 4, 3 become neighbors as a confection adage has it from 4 to 3 and 4 to 1 separately according Figure 9.

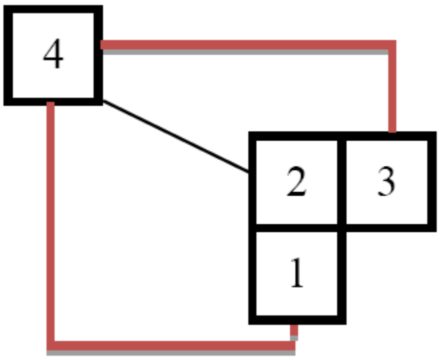


FIGURE 9. Shows reinforced neighbor

Here is where a proposed method with Cellular Learning Automata will be illustrated using above neighborhood rules.

5. **Proposed Method with Cellular Learning Automata.** As noted in the previous chapter, Cellular Learning automata are lattice of cells and each cell has one Learning Automata or more than one Learning Automata. Learning Automat in which cells uses L_{R-I} of methods of algorithm, would not have any penalty. In addition, irregular Cellular Leaning Automata and uniform Cellular Learning Automata are used as follow

1. Each transaction of data base is shown in neighborhood. If in subsequent transactions repeat pervious transactions or subset of them, then neighbors will be strengthened and each time of repetition they get reward of leaning automat in each cell and if reinforcement rate is more than threshold, they are sent to output as a sequence with high frequent.
2. Thereafter will be updated basis on new transaction (rows) of dataset. The first reinforcement cells are done and thereafter neighbors are updated.
3. Stage 1 and 2 would continue until the whole transaction of dataset be read and performed on Cellular Learning Automata and sequences are extracted.

The numerical example below is clarifying the explanation given above.

We assume reward rate is 0.5 ($a = 0.5$) and min-support is 2. This means each cell is repeated 2 times or more than 2 times and would be stronger than they are strengthened and be sent to output as a sequence with high frequency for reinforcement rate.

- Rows of the dataset are read one by one and are performed in Cellular Learning Automata. Regarding one transaction of dataset in row of T1, (1, 2, 3) production

Table 6. Shows the data transactions

T_{ID}	SEQUENCE
T1	1, 2, 3
T2	2, 4
T3	2, 3
T4	1, 2, 3
T5	1, 3
T6	2, 3
T7	1, 3
T8	1, 2, 3, 5
T9	1, 2, 3

have been purchased together as a result, 1, 2, 3 cells are neighbors with each other according Figure 10.

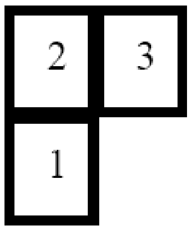


Figure 10. Shows the sample of the neighborhood

- T2 row are read then (2, 4) production have been purchased together. When Cellular Learning Automata reads a transaction of dataset to find neighboring, the sequence is investigated in all CLA's cells. If the entered sequence contained items that have already been neighbors previously in the Cellular Learning Automata, then they will be reinforced while the other remaining will be wakened. Reinforced cells will stay together and weakened cells will stay far away from each other. This process will be repeated for each row of dataset to make batch of cells together which each batch of cells together will create higher repeated sequence. The Amount is determined as threshold that is the same as min-support. If amount of reinforcements is more than threshold, it means that those sequences can be considered as output. Finally neighborhood would be updated based on new transaction (rows) of data base. In the example bellow, there is no neighborhood between 2 and 4 which then must create this neighboring according to Figure 11. Min-support is selected by the users and is dependent on iteration of items in each dataset

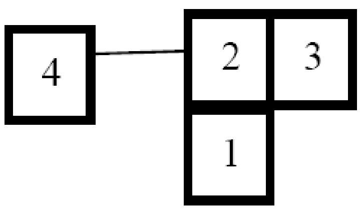


Figure 11. Shows the sample of the neighborhood

- As shown in the figure above, cell 4 is connected only to cell 2 to create neighboring effect with cell 2 but not with cell 1 and cell 3. T3 in third row can clearly be read because this neighborhood is already existed (2, 3) then both cells get reward and because their reinforcement they are more than threshold rate, as a result they would be sent to the output as a 2 item sequence.

Finally, this work are repeated continuously until all row of dataset can clearly be read and would be created for new reinforcements and neighbors to find and send sequence with high frequency and deliver them to output. More frequency patterns are as follow for above example

$$\{\{2,3\},\{1,2\},\{1,3\},\{1,2,3\}\}$$

The Cellular Learning Automata approch for Frequent Pattern Mining algorithm as bellow:

```
CLA-FPM Algorithm
Input:I€Item transaction dataset, Threshold, G (V, E);
Output: subset of Items;
  Begin
    Construct an irregular CLA isomorphic;
    For each row on dataset do
      Each cell chooses one of its actions according to dataset row;
        If cells are neighbors then
          Reward actions have been chosen by Automata Ai;
        If probability of actions (pi)>=Threshold then
          Write (cells which are neighbors);
        End If
        End IF
      Update neighbors;
    End For
End CLA-FPM Algorithm
```

FIGURE 12. Shows the CLA-FPM algorithm

6. **Experimental Result.** In our proposed method in chapter four, research methods of Cellular Learning Automata had been discussed. Use of Learning Automata that attribute to each node had been explained. In additions, it had been shown that in each stage, some of the items were imported to the random graph. Furthermore, the learning Automaton was assumed to be active and would update its probability of actions by importing each item to the graph. Moreover, the process was assumed to continue up to when all of the subsets of the customers shopping set were examined. Finally, the method was applied on aggregate data that are based on on-line dataset of two firms namely, Mushroom dataset and Belgian customers shopping basket dataset.

In this chapter, we will discuss the underline factor of our research, the way how data are collected and analyzed as well as how the detailed implementations of these data are employed. The rest of this chapter is organized as follows: part one will discuss the data collection process in Section 1, followed by Section 2 that deals with the simulation software in the second part that follows.

6.1. **Data collection.** The dataset that consisted files of Belgian Customers shopping basket that contains transactions of a Belgian retail store was obtained. This would include the following dataset: namely Mushroom dataset which consist a high dense dataset and Belgian Customers shopping basket dataset which consists low dense dataset. This method was applied on online dataset of Mushroom and Belgian customers shopping basket dataset to examine the effect of our study. Mushroom dataset has 22 fields and 4018 records that states properties of mushroom in different modes and was obtained from the UCI Machine Learning Dataset [26]. Belgian Customers shopping basket dataset has 16450 fields and 41373 records. This file contains transactions from a Belgian retail store [27]. Mushroom dataset is high dense dataset and Belgian Customers shopping basket dataset is low dense. These data had been fed into the programmer so that the resulting output yield the desired running time. They were designed to be an enumeration of the states and properties of mushroom and customer shopping basket dataset within the UCI Machine Learning Datasets in different modes. This method was implemented for frequency pattern of mining, a set with n members. To implement the simulation program, the process has been used from programming language of Microsoft Visual Basic.net 2008 express edition with 2GB RAM, CPU 2.0 GHz and windows XP SP/2 Operating System. The results are displayed in Figures 14 and 15 in comparison with other methods. This work was done with different supports and demonstrates better results and efficiency of the suggested method.

In this section all models selected and the proposed methods are investigated from the viewpoints of operation on the data, generating the outputs and creating the number of steps, and then reviewing the running time of the proposed method.

To obtain the frequent patterns in Apriori method, the process started from one-item sets and continued with their combination to two-item sets and above. Consequently, this caused a cost of 2^n to obtain frequency pattern in which n denoted the total number of items in the dataset.

The second method was FP-Growth in which the corresponding tree is generated and scanned to obtain the frequency patterns. Since the scanning process is done through the generated tree from leaves to root, the costs of tree generation and scanning have to be added to the total cost of FP-Growth. The cost of FP-Growth will be b^d, where b is the branch factor of each node in the tree and d is depth of the tree. The performance cost of this method worsens in the files with low density of iterative characteristics and reaches to 2^{n-1} in the worst case.

The third method discussed in this research was FP-Tree with Array. This method (as has been shown in chapter two which examined one-item and two-item sets, put them in an array, and entered them in a tree after finding and classification of item sets. Due to the tree structure, the cost of generation and scanning in this method is high; however, it acts better than the former method when the files have low density of iterative characteristics. The cost of this method will increase up to 2^{n-1} (n refers to the total number of item in the data file) for the files with high density of iterative characteristics. Two scans should be done on dataset, such as FP-Growth. In the first scanning, it is search frequency data sets from FP-Tree and in the second scanning; it made the array at the time of constructing FP-Tree. After producing the array from the tree, the process will be to start exploration and finding frequency patterns from the array. In order use the tree, the cost of constructing and scanning in this method will be high.

The proposed method acts row by row and does not need any tree or other specific data structure. Finding frequency patterns is dependent on the number of items in each row, because to find the frequency patterns in each row is simply to compare them with the previous rows and then the frequency patterns will be extracted with navigation on one the data file. This reduces the runtime and requires less space. Run time is 2^T in the worst case in which T is the number of items in each row rather than n as the total number of items in the previous methods. This causes a significant reduction in the runtime of the proposed method in comparison with the other approaches.

All of the above mentioned methods are categorized in Table 7 and Figure 13 where n is the number of all items in the data file, T is the number of items in each row, b is branch factor in each node that created tree, and d is the depth of created tree.

TABLE 7. The comparison of the studied methods and the proposed method

row	method	Runtime (seconds)	Runtime in the worst case (seconds)
1	Apriori	2^n	2^n
2	FP-Growth	b^d	2^{n-1}
3	FP-Tree with Array	b^d	2^{n-1}
4	CLA-FPM	2^T	$2^{n/2}$

where n is total of item set of dataset, b is tree branch factor, d is depth of tree and T is total item sets in each rows.

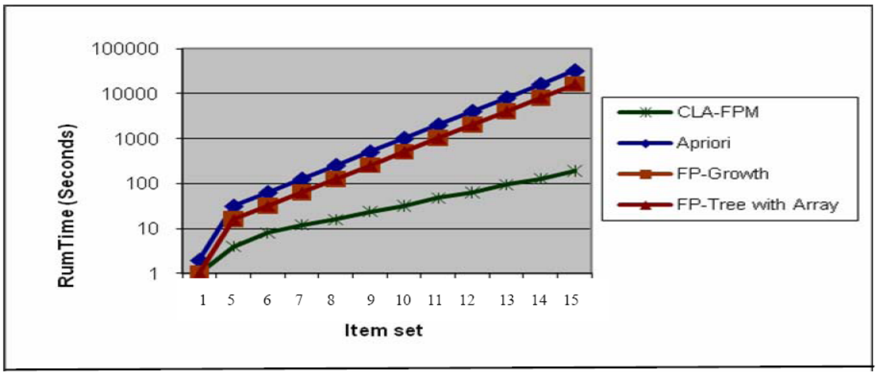


FIGURE 13. The comparison of the studied methods and the proposed method in worst case

6.2. **Comparison the other method with the proposed method according to experimental results.** The Figures 14 and 15 are related to obtained results of implementing software on proposed method in which X and Y axes refer to min-support and run time, respectively.

Figure 14 compares the other methods with the proposed method in the mushroom dataset. Since there are many similar iterative characteristics of the mushrooms, the X axis is varied from 600 to 1000 in Figure 14. In Figure 15 the other methods are compared with the proposed method in the basket dataset and min-support is considered between 5 and 100.

In all cases the higher the min-support, the less the runtime, however, the proposed method needs the least time to find the frequent patterns and it has the highest speed in comparison with the other methods. Min-support is selected by user and is dependent on iteration of items in each dataset. The Figures 14 and 15 are related to obtained results of implementing software on proposed method in which X and Y axes refer to min-support and run time, respectively.

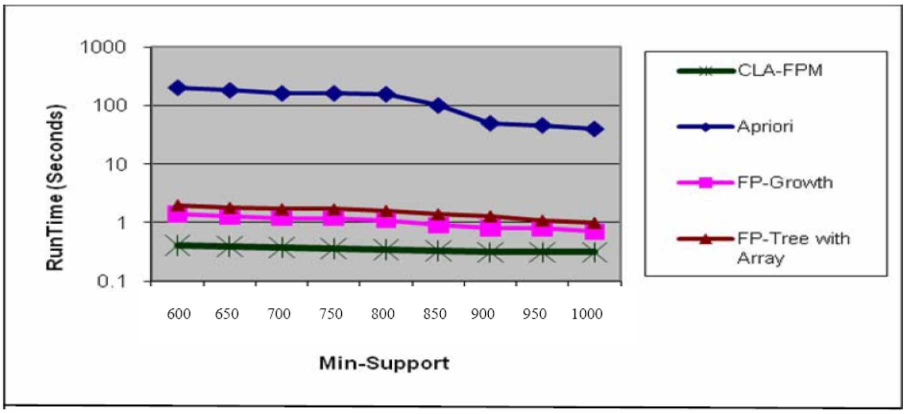


Figure 14. Shows comparison the other method with the proposed method in the mushroom dataset

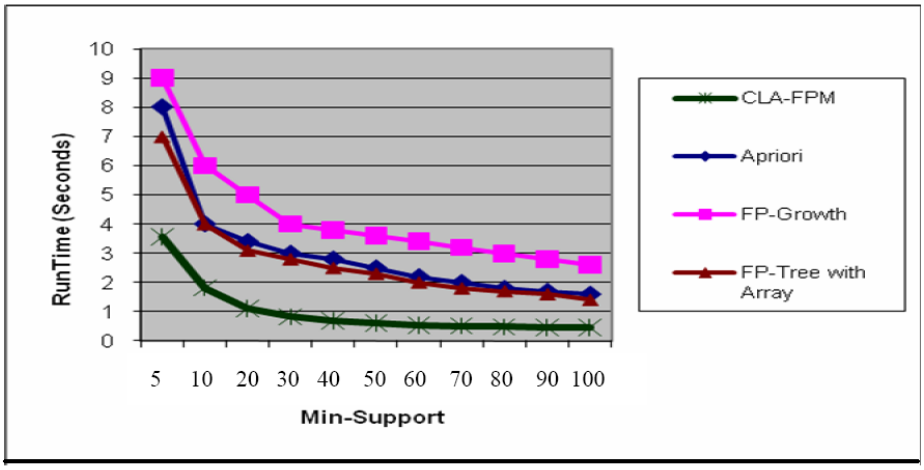


Figure 15. Shows comparison the other method with the proposed method in the Basket dataset

7. **Conclusion.** In this research, we have investigated the detailed patterns of frequency of cellular Learning Automata in modern data mining literature. Frequency patterns help shop owners to order goods in their shops in the most appropriate manner which was our goal of achievement. The frequency patterns are the labels of goods which are bought by different customers. One of the most important issues in this process was to find the frequency patterns in the least possible time which was possible with the application of Cellular Learning Automat (CLA) as investigated in this research. CLA is one of the many data mining tools that are capable of classifying future works based on previous historical records and had in detail been discussed in Section 3. Our research method of proposal is discussed in Section 4. Since CLA classifies the frequency patterns faster than other comparable methods based on neighborhood classification and send them to generate output for decision making process, it is concluded that the proposed method of CLA in our work achieves less time to gain frequency patterns than any other methods.

The other target of this research was decreasing the run time to obtain frequency patterns. In this research, we tried to find higher frequency in less possible time. The number of items in the file will be deleted with increasing min support, as a result, the size of file decreases. In this research, run time is very important because run time causes to increase decision time for items arrangement especially in online shops. This issue is very remarkable when dataset is big. However, in other methods runtime will be increase when data base is big, In order to investigate the ability of our proposed method to achieve this target and, two aggregate-data-based studies namely, basket dataset and mushroom dataset were investigated. The results of these studies demonstrated that our proposed method has the least running time in comparison with other related works.

Advantages of this method

1. Finding frequency patterns in the least possible time compared to other related methods.
2. Goods classification in terms of shop-keeping and warehouses to increase the financial benefits.
3. Less confusion for costumers in order to find desired goods.
4. Performing any kind of file with any structure on proposed method.
5. Using this model in on-line shops and reordering goods in online shops.

Many applications can be suggested for this research. Some of them have been mentioned above and other applications of this method can be noted as follows.

1. Classification of desktop icons in personal computers.
2. Classification of the interest in social networks.

Some of the attractive cases in which this research can be applied nowadays are taken into consideration such as for example, to classify the behavior of users in social networks, to find their interests for classification of the tools, access method, etc.

To extend this research to the required and interested applications of most of software companies, and on-line Internet services are of the future works. For this reason, we believe that several solutions can be suggested to achieve the classification of the interests, access method, tools, software, etc. in order to increase more user's friendly of the systems.

In this study, cellular learning automata method was examined for sequence pattern of mining in a set of sequence data. This method could assist in a great extent to suppliers of goods or shop owners in finding sequence of the customers shopping. It will be demonstrated that by this method, can extract two-members, three-members, $\cdots$, n-members that have had the most iteration and provide them to the shop owners for decision making and accurate arrangements of goods in the shelves. It was compared with other methods and its results show higher efficiency of the proposed method than the other ones. Finding of the categories leads to increase of satisfaction of the customers in shopping, increase of the stores profits and decrease of the customer's confusion in finding their necessary products.

REFERENCES

[1] R. Agrawal, T. Imielinski and A. Swami, Mining association rules between sets of items in large datasets, *Proc. of the ACM-SIGMOD International Conference on Management of Data*, Washington, DC, pp.207-216, 1993.

[2] R. Agrawal and R. Srikant, Fast algorithms for mining association rules, *Proc. Int. Conf. Very Large Data Bases*, Santiago, Chile, pp.487-499, 1994.

[3] H. Mannila, H. Toivonen and A. I. Verkamo, Efficient algorithms for discovering association rules, *KDD*, Seattle, WA, pp.181-192, 1994.

[4] K. S. Narendra and M. A. L. Thathachar, *Learning Automata: An Introduction*, Prentice Hall, 1989.

[5] J. S. Park, M.-S. Chen and P. S. Yu, An effective hash based algorithm for mining association rules, *SIGMOD*, San Jose, CA, pp.175-186, 1995.

[6] H. Toivonen, Sampling large datasets for association rules, *Proc. of the International Conference on Very Large Data Bases*, Bombay, India, pp.134-145, 1996.

[7] A. Savasere, E. Omiecinski and S. Navathe, An efficient algorithm for mining association rules in large datasets, *Proc. of the 1995 International Conference on Very Large Data Bases*, Zurich, Switzerland, pp.432-443, 1995.

[8] F. Geerts, B. Goethals and J. Bussche, A tight upper bound on the number of candidate patterns, *Proc. of the International Conference on Data Mining*, San Jose, CA, pp.155-162, 2001.
[9] J. Han, J. Pei and Y. Yin, Mining frequent patterns without candidate generation, *Proc. of the ACM-SIGMOD International Conference on Management of Data*, Dallas, TX, pp.1-12, 2000.
[10] G. Grahne and J. Zhu, Efficiently using prefix-trees in mining frequent, *Proc. of the ICDM'03 International Workshop on Frequent Itemset Mining Implementations*, Melbourne, FL, pp.123-132, 2003.
[11] J. Pei, *Pattern-Growth Methods for Frequent Pattern Mining*, Ph.D. Thesis, 2002.
[12] Y. Z. Tsypkin, *Adaptation and Learning in Automatic Systems*, Academic Press, New York, 1971.
[13] M. L. Tsetlin, *Automaton Theory and Modeling of Biological Systems*, Academic Press, New York, 1973.
[14] K. S. Narendra and R. Viswanathan, Learning models using stochastic automata, *Proc. of the International Conference of Cybernetics and Society*, Washington DC, 1972.
[15] V. I. Varshavski and I. P. Vorontsova, On the behavior of stochastic automata with variable structure, *Avtomatika & Telemechanika*, vol.24, pp.353-360, 1963.
[16] K. S. Fu and T. J. Li, On stochastic automata and languages, *Information Science*, vol.1, no.4, pp.403-419, 1969.
[17] K. S. Fu, Stochastic automata as models of learning systems, *A Prelude to Neural Networks: Adaptive and Learning Systems*, pp.393-431, 1994.
[18] J. Lu, Predicting customer churn in the telecommunications industry an application of survival analysis techniques, *Proc. of the 27th Annual SAS Users Group International Conference*, SAS Institute Inc, Cary, NC, 2002.
[19] M. Alipour and M. R. Meybodi, Solving traveling salesman problem using distributed learning automata, *Proc. of the 10th Annual CSI Computer Conference*, Computer Engineering Dep. Iran Telecommunication Research Center, Tehran, Iran, pp.271-280, 2005.
[20] K. S. Fu and G. J. McMurtry, A study of stochastic automata as models of adaptive and learning controllers, *IEEE Transactions on Automatic Control*, vol.11, pp.379-387, 1966.
[21] C.-Y. Chen, S.-W. Shyue and C.-J. Chang, Association rule mining for evaluation of regional environmets: Case study of Dapeng bay, Taiwan, *International Journal of Innovative Computing, Information and Control*, vol.6, no.8, pp.3425-3436, 2010.
[22] M. R. Meybodi and M. R. Kharazmi, Cellular learning automata and its application to image processing, *Journal of Amirkabir*, vol.14, no.56A, pp.1101-1126, 2004.
[23] H. Beigy and M. R. Meybodi, A mathematical framework for cellular learning automata, *Advances on Complex Systems*, vol.7, no.3, pp.1-25, 2004.
[24] M. R. Meybodi and F. Mehdipour, VLSI placement using cellular learning automata, *Proc. of the 8th Annual International Computer Society of Iran Computer Conference*, Mashhad, Iran, pp.195-203, 2003.
[25] C.-J. Chu, V. S. Tseng and T. Liang, Mining temporal rare utility item sets in large database using relative utility thresholds, *International Journal of Innovative Computing, Information and Control*, vol.4, no.11, pp.2775-2792, 2008.
[26] *http://www.ics.uci.edu/∼mlearn/MLRepository.html, 1987.*
[27] *http://www.fimi.cs.helsinki.fi/data/retail.dat, 1991.*
[28] E. W. T. Ngai, X. Li and D. C. K. Chau, Application of data mining techniques in customer relationship management: A literature review and classification, *Expert Systems with Applications*, vol.36, pp.2592-2602, 2009.
[29] A. Tsoularis, C. Kambhampati and K. Warwick, Path planning of robots in noisy workspaces using learning automata, *Proc. of the IEEE International Symposium of Intelligent Control*, Chicago, pp.560-564, 1993.
[30] C. Archaux, A. Martin and A. Khenchaf, An SVM based churn detector in prepaid mobile telephony, *International Conference on Information and Communication Technologies*, Damas, Syrie, pp.19-23, 2004.
[31] M. R. Meybodi, H. Beigy and M. Taherkhani, Cellular learning automata and its applications, *Journal of Science and Technology*, University of Sharif, no.25, pp.54-77, 2004.
[32] I. Bose and X. Chen, Hybrid models using unsupervised clustering for prediction of customer churn, *Journal of Organizational Computing and Electronic Commerce*, vol.19, pp.133-151, 2009.
[33] Y. Xie, X. Li, E. W. T. Ngai and W. Ying, Customer churn prediction using improved balanced random forests, *Expert Systems with Applications*, vol.36, pp.5445-5449, 2009.

[34] C. Pendharkar, Genetic algorithm based neural network approaches for predicting churn in cellular wireless network services, *Expert Systems with Applications*, vol.36, pp.6714-6720, 2009.

[35] H. Beigy and M. R. Meybodi, Cellular learning automata with multiple learning automata in each cell and its applications, *IEEE Transaction on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol.40, no.1, pp.54-66, 2010.

[36] S.-F. Lin and Y.-C. Cheng, Two-strategy reinforcement evolutionary algorithm using data-mining based crossover strategy with TSK-type fuzzy controllers, *International Journal of Innovative Computing, Information and Control*, vol.6, no.9, pp.3863-3885, 2010.

[37] A. Lemmens and C. Croux, Bagging and boosting classification trees to predict churn, *Journal of Marketing Research*, vol.43, no.2, pp.276-286, 2006.

[38] M. Esnaashari and M. R. Meybodi, Irregular cellular learning automata and its application to clustering in sensor networks, *Proc. of the 15th Conference on Electrical Engineering*, Vol on Communication, Telecommunication Research Center, Tehran, Iran, 2007.