RCRDE: A method for Reducing the Rate of Re-Clustering, using Replicated Data Eliminate algorithm

*Fateme Rashidi
Corresponding author;
Department of computer engineering, Payame Noor University, Po Box 19395-3697
Tehran, Iran

Arash Ghorbannia Delavar
Department of computer engineering, Payame Noor University, Po Box 19395-3697
Tehran, Iran

Fateme heidari Soureshjani
Department of computer engineering, Payame Noor University, Po Box 19395-3697
Tehran, Iran

Ali Broumandnia
Department of computer engineering, Tehran Azad University, south branch

ABSTRACT
In this paper is explored a way to reduce the rate of re-clustering and speed up the clustering process on categorical time-evolving data. This method introduces two algorithms RDE (Replicated Data Elimination) and RCRDE. The RDE algorithm removes the successive surveys of replicated data and considers counters to keep this data. Hence the number of created windows via the sliding window technique is limited and this leads to decrease the number of implementations of clustering algorithm. The RCRDE algorithm based on MARDL (MAximal Resemblance Data Labeling) framework decides about re-clustering implementation or modification of previous clustering results. The presented method is independent of clustering algorithm's type and any kind of categorical clustering algorithm can be used.
According to the results obtained on different data sets, this method performs well in practice and facilitates the clustering implementation on categorical data. Also, this method can be utilized to cluster a very large categorical static database with higher quality than previous work.

Keywords
Categorical time-evolving data, clustering, data labeling, drifting-concept detecting.

1. INTRODUCTION
The data streams due to evolving the large number of applications turn to a major problem. Therefore clustering techniques considerably is surveyed in order to explore and analysis extracting information from streams [1]. One challenging problem in the field of the clustering data streams is that the concept of interest may be related to some hidden context, have not been clearly in default features. This means that user tries to understand the concept of data witch change with time [3],[2]. For example, the customer's preferences and buying patterns may change with time, related to the current day of the week, availability of alternatives, discounting rate, etc. When the data stream is changes with time, the result of clustering on the data must be modified. In overall, in the performance of clustering on the time-evolving data, the quality of cluster should be preserved and the demands of users also be considered, which requires adjustment of clustering results.

Clustering on the time-evolving data in numerical domains has been evaluated in several studies [1], [4], [6], [7], [22], [24], [23], [25]. Yet there are many categorical data sets that this issue has not been widely resolved in them. For example, the browsing history of users is stored by web logs or documents in various applications are changing with time.
Most of the works done in the context of clustering categorical data focuses on doing clustering on the entire data set and do not pay attention to the drifting concepts problem. Hence an effective method for clustering on the categorical time-evolving data requires to be addressed.
Gaber et al in [4] proposed an algorithm that utilizes clustering results to identify the drifting concepts in the numerical domain. In this strategy an online clustering algorithm is performed at any time frame that used a distance threshold technique for assigning new points to existing clusters. Also in this method several numerical characteristics such as the mean size of clusters and the mean and standard deviation of cluster centers are used to represent clustering. However this method due to complexity of explanation numerical characteristics of clusters in categorical domain is impractical.
A similar approach with [4] for clustering time-evolving data in the categorical domain is proposed in [5] which named MARDL framework. In this framework for detecting the drifting concepts, the sliding windows technique is used. In this approach, drifting concepts is described by analyzing the relationship between the clustering results at different times in sliding windows. Then the clustering result based on current concept is created. The implementation of MARDL on a categorical data set with twenty data points is shown in Figure.1.
This paper focuses on a method for implementing clustering on the categorical time-evolving data. The main goal of this method is to reduce the execution time of clustering on categorical time-evolving data relative to MARDL framework, which its re-clustering rate is high .RCRDE method consists of four parts: a Replicated Data Elimination (RDE) algorithm that analyzes the data received from input in order to create a sliding window without repetitive data, a data-labeling algorithm that determines the Nearest cluster

label for each data point of the current sliding window based on the last clustering results, a categorical data clustering algorithm that can be any categorical data clustering algorithm, and RCRDE algorithm that recognize the difference of cluster distributions between generated clustering result in current sliding window and last sliding window**.**

[23], [24], [25]. First, in a study by Aggarwal et al [1] the problem of data stream over time is discussed and CluStream model for clustering on the data stream in the various times is proposed. In this model, clustering process is composed of an online component which periodically stores information in terms of micro-clusters at snapshots in time that is used by an

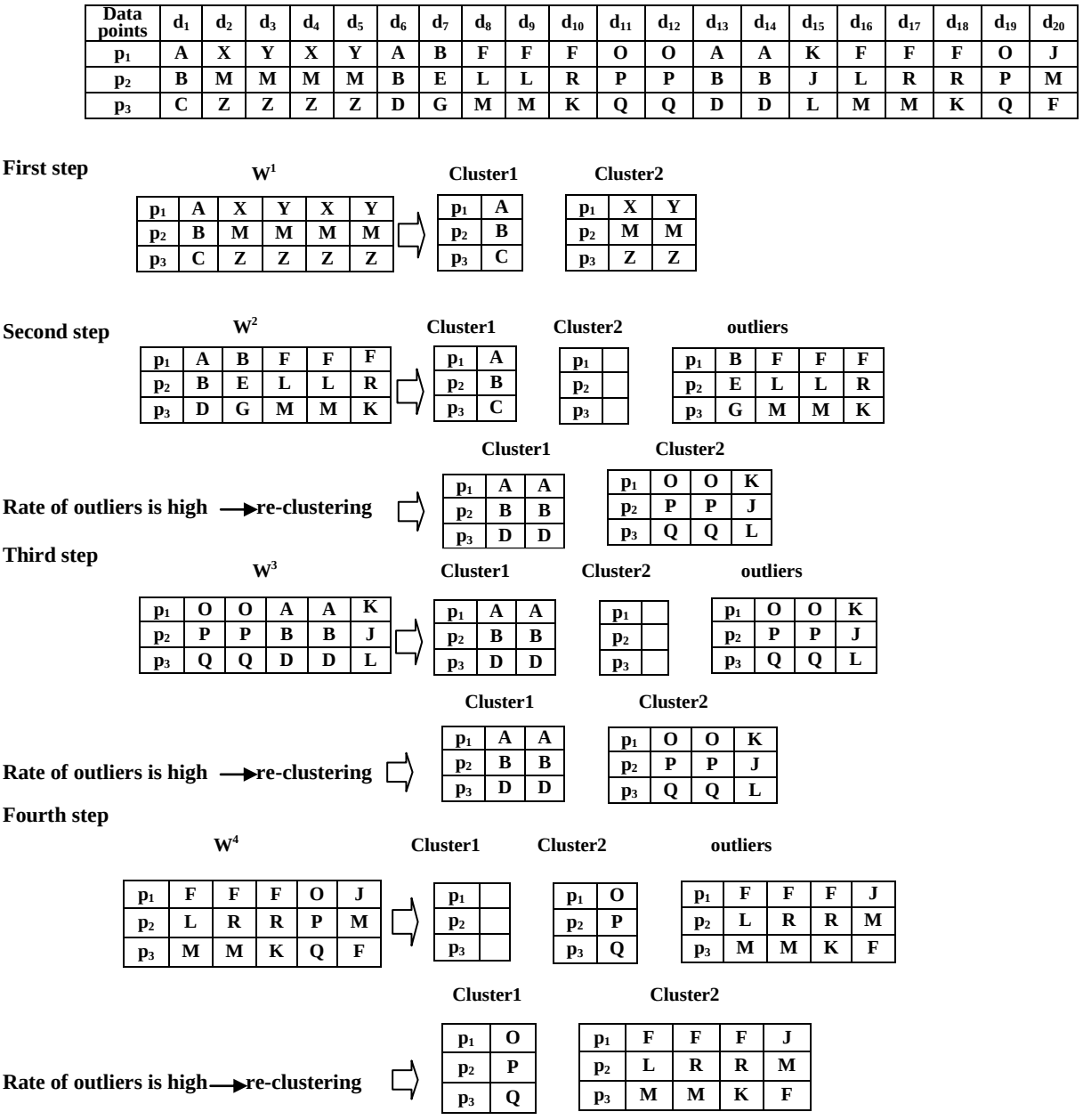


Fig.1: Implement of MARDL framework on the example categorical data set.

The description of this paper is as follows: a review of related works (Section.2), define RCRDE method and its related formulas and algorithms (Section.3). RCRDE method performance is studied on data sets (Section.4) and conclusions (Section.5).

2. RELATED WORK

Clustering on the time-evolving data in numerical domains has been evaluated in several studies [1], [4], [6], [7], [22], offline component which is dependent on user-defined inputs such as the time horizon or the number of clusters. This process provides flexibility to an analyst in an environment, which is changing with time. Cao et al in [6] proposed an approach for discovering clusters of arbitrary shape in an evolving data stream. Based on concept of density a core-micro-cluster synopsis is designed to summarize the clusters instead of the snapshots with constant number of micro-clustering.

Chakrabarti et al in [7], presented an evolutionary clustering framework, which generates effective clustering results for numerical domain that change over time. In this framework, the clustering should be of high quality at any time. yet, it must ensure that clustering results do not change in any time step in Compare with previous time stamp. Also, in this framework evolutionary version of two algorithms K-means and agglomerative hierarchical clustering is considered. The framework presented in [4], measures online deviation of the clustering results to detect changes in data distribution over time, then an offline voting-based classification algorithm links each change with a previously encountered event.

As is mentioned earlier in this paper, implementation of clustering on numerical time-evolving data has been widely studied, But there is not enough attention to the issue of clustering categorical time evolving data. In general, the issue of clustering categorical data was discussed for the first time in [8] by Han et al, which cluster is proposed based on approaches for clustering related categorical data using association rules, and clustering related transactions data.

Gibson et al [9] considered the problem of clustering categorical domain as a type of nonlinear dynamical systems. In this approach the categorical data set can be clustered when the dynamical system converged. Huang et al [10] extended the K-means algorithm for clustering on categorical data, called K-modes. Based on this algorithm, several algorithms were created for different applications presented such as fuzzy k-modes [11], initial points refinement [12].

In ROCK algorithm presented by Guha et al [13] instead of using distance metric for measuring similarity between data points, a concept of links was used to measure similarity, because algorithms that employed distances metric between points for clustering are not suitable for categorical data. In other words, ROCK is an agglomerative hierarchical clustering, which each point is considered as a distinct cluster. Then the clusters that are nearest neighbors merged until the number of clusters equal to the number of clusters is determined.

The CACTUS [14] is an algorithm based on summarization for clustering on categorical data. CLICK [15] implements the clustering on the categorical domain based on a search for k-partite maximal groups and to ensure complete search using a selective vertical method. In this algorithm, the categorical data set as k-partite graphs is considered, which means each cluster is corresponding to a k-partite. Both COOLCAT [16] and LIMBO [17] algorithms are created based on statistics. COOLCAT places data points into the clusters where it minimizes the expected entropy of the clusters, While in LIMBO algorithm, the Information Bottleneck method to the problem of clustering categorical data in the large data set is used.

The problem of clustering categorical time-evolving data was first addressed by Chen et al [5]. Chen et al in [18] extended a categorical cluster representative technique, named NIR, for clustering on very large categorical database. This technique was used in order to represent clusters by distribution of the attribute values. Then they proposed MARDL framework to allocate each data point into the appropriate cluster. Based on this approach, presented a framework to perform clustering on categorical time-evolving data in [5]. The framework using DCD (Drifting Concept Detecting) algorithm, detects the drifting concepts at different sliding windows, and produces the clustering result based on current concept.

In current paper is presented a new method for clustering categorical time-evolving data and conduces the running time is lower than MARDL framework.

3. RCRDE method:

For Problem description, a data set of the categorical data with concept drifting as a set of points is considered (each point has q attributes). Using the sliding windows technique [19], the points are placed in the windows with size N. Each window W^T has a time stamp T and at different time stamps, various clustering results are formed. $C^{(Ti,Tj)}$ is considered the clustering results from T_i to T_j. M is the number of data points that is received from data points set, in order to create a window with size N. One counter is used in order to keep the number of repetitions per data point. RDE algorithm to eliminate the repetitive data points in M is employed. Then, data points with its counters are entered into the sliding windows and in the next step initial clustering is performed on them. RCRDE method is independent of clustering algorithms type. In other words, this method is not affected by choosing any categorical clustering algorithms. In next time stamp, the next window is created to help the RDE algorithm, then Temporal clustering result on this window is obtained by data labeling. Chen et al proposed data labeling technique in [20] and they modified this technique to detect outliers in any window in [5].

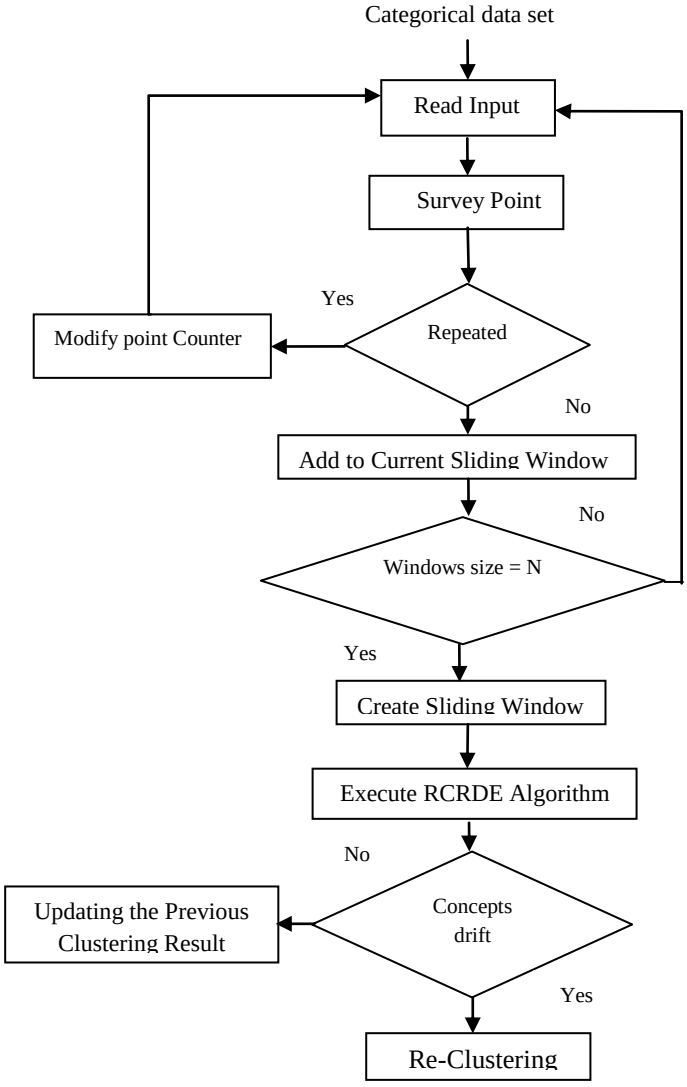


Fig.2: The flowchart of RE and RCRDE algorithm.

The goal of data labeling is to identify the appropriate cluster label in the last clustering result for each data point in the current window. Data points that do not belong to any cluster in the previous clustering results are considered as outlier. In

next step of RCRDE method, the previous clustering results and the temporal clustering results generated by RDE algorithm and data labeling technique were compared with each other. If a drifting concept has been detected, re-clustering would be performed, otherwise the previous results will be updated. The flowchart of clustering algorithm on the categorical data in this method using the RDE and RCRDE algorithm are shown in Figure.2. In RCRDE method, repetitive data point would be checked by RDE algorithm when data point is received from input. If the data point is repeated, its counter's value would be increased, otherwise is placed inside the current sliding window. M is considered equal to the number of data points is received from input, until a window is made with size N without repetitive data points. The M value depends on the repetition rate of data points, the higher repetition rate of data points, accompanied by the larger M value and vice versa. For example, if D be the number of data points in the original data set, the best situation in RCRDE method for the data point clustering occurs when M is equal to D and the worst situation occurs when M is equal to N. In other words, in the worst situation there is no repeated point. If P is considered as probability of re-clustering in each sliding window, according to definitions in [5], P is equal to:

P=P1+P2

$$\sum_{M=N}^{D}\frac{N}{M}=N\sum_{M=N}^{D}\frac{1}{M}=N\left(\sum_{M=1}^{D}\frac{1}{M}-\sum_{M=1}^{N}\frac{1}{M}\right)$$
$$=N(lnD+\ \gamma+\ \varepsilon k-lnN-\gamma-\ \varepsilon N)$$
$$=N(lnD-lnN+\varepsilon k-\varepsilon N)$$

According to [5] drifting concept occurs when large number of outliers in the temporal clustering is found or a large number of clusters in compare with its previous status have been changed in the rate of data points. In [5] for detecting these changes, a threshold θ as outlier threshold is considered, if the ratio of outliers in the sliding window is larger than θ, the clustering results will change.

Also a double-threshold method to compare the clusters with its previous status is employed in [5]. ∈ as cluster variation threshold decides ratio of data points in a cluster is changed or not. Cluster is changed if the ratio of data points exceeds from cluster variation threshold. And η is considered as the cluster difference threshold, in which is large number of clusters has changed, in this case, the drifting concept has occurred in the current sliding window.

Data points	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}	d_{16}	d_{17}	d_{18}	d_{19}	d_{20}
p_1	A	X	Y	X	Y	A	B	F	F	F	O	O	A	A	K	F	F	F	O	J
p_2	B	M	M	M	M	B	E	L	L	R	P	P	B	B	J	L	R	R	P	M
p_3	C	Z	Z	Z	Z	D	G	M	M	K	Q	Q	D	D	L	M	M	K	Q	F

W1

Counter	1	2	2	1	1
p_1	A	X	Y	A	B
p_2	B	M	M	B	E
p_3	C	Z	Z	D	G

Cluster1

Counter	1	1	1
p_1	A	A	B
p_2	B	B	E
p_3	C	D	G

Cluster2

Counter	2	2
p_1	X	Y
p_2	M	M
p_3	Z	Z

Fig.3. Initial clustering is performed on the example categorical

Which P1 is probability of having a large number of outliers in the current window that identified by data labeling technique and P2 is probability of changing a large number of clusters in compare with the previous clustering results. In MARDL in per time stamp N data points are clustered, while in RCRDE method M data points are clustered. So the number of windows in RCRDE method compared with previous methods is reduced. In other words, in this algorithm for M data points received from input, which N of them is non-repetitive, one window is created, while in MARDL, M/N windows is created for the same input, as follows:

$$f(x)=\frac{(1\times P)+\left(1\times(1-P)\right)}{\left(\frac{M}{N}\times P\right)+\left(\frac{M}{N}\times(1-P)\right)}=\frac{1}{\frac{M}{N}}=\frac{N}{M}$$

Function f(x) calculates the probability of re-clustering on a window. This function indicates re-clustering rate is reduced in RCRDE method compared with MARDL framework. Considering the repetition rate in the data points, as follows:

$if\ M=N\rightarrow f(x)=\frac{N}{M}=\frac{N}{N}=1$

$if\ M=N+1\ \rightarrow f(x)=\frac{N}{M}=\frac{N}{N+1}$

.

.

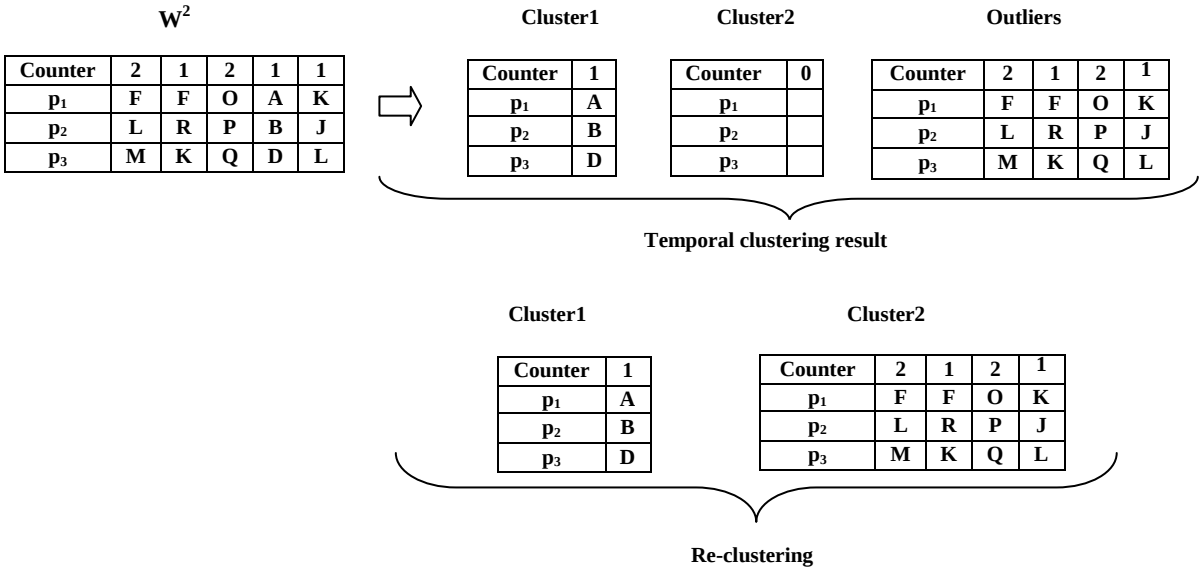
$if\ M=D, D=kN\ \rightarrow f(x)=\frac{N}{M}=\frac{N}{kN}$

In RCRDE method using of eliminating the repetitive data points in each window and reducing the number of created windows, the drifting concept is defined as follows:

$$\begin{cases} concept\ drifting\ occeres;\ if\ \dfrac{number\ of\ outlier}{M}>\ \theta \\ concept\ drifting\ occeres;\quad if\ \dfrac{\sum_{i=1}^{k^{(t,t-1)}} d\left(c_i^{(t,t-1)},c_i'^{t}\right)}{k^{(t,t-1)}}>\eta, \\ where\ d\left(c_i^{(t,t-1)},c_i'^{t}\right)= \\ \begin{cases} 1, if\ \left|\dfrac{\sum_{j=1}^{mi^{(t,t-1)}} counter(j)}{\sum_{x=1}^{k^{(t,t-1)}}\sum_{j=1}^{m_x^{k^{(t,t-1)}}} counter(j)}-\dfrac{\sum_{j=1}^{mi^{t}} counter(j)}{\sum_{x=1}^{k^{(t,t-1)}}\sum_{j=1}^{m_x^{t}} counter(j)}\right|>\epsilon \\ 0,\quad otherwise \end{cases} \\ do\ not\ occeres,\quad otherwise \end{cases}$$

Example.1: In this example, is implemented RCRDE method on the same dataset presented in Figure.1. This set is composed of twenty data points d_1, d2, d20. Each of these data points are composed of three attributes $p_{i\ (}1<i<3_{)}$ and the sliding windows size N=5 and the number of clusters k=2.

Then data labeling is performed on W2. In data labeling using of NIR tables, temporary clustering results on the current sliding window is created and points that do not belong to any cluster are considered as outlier.



Counter	2	1	2	1	1
p_1	F	F	O	A	K
p_2	L	R	P	B	J
p_3	M	K	Q	D	L

Counter	1
p_1	A
p_2	B
p_3	D

Counter	0
p_1	
p_2	
p_3	

Counter	2	1	2	1
p_1	F	F	O	K
p_2	L	R	P	J
p_3	M	K	Q	L

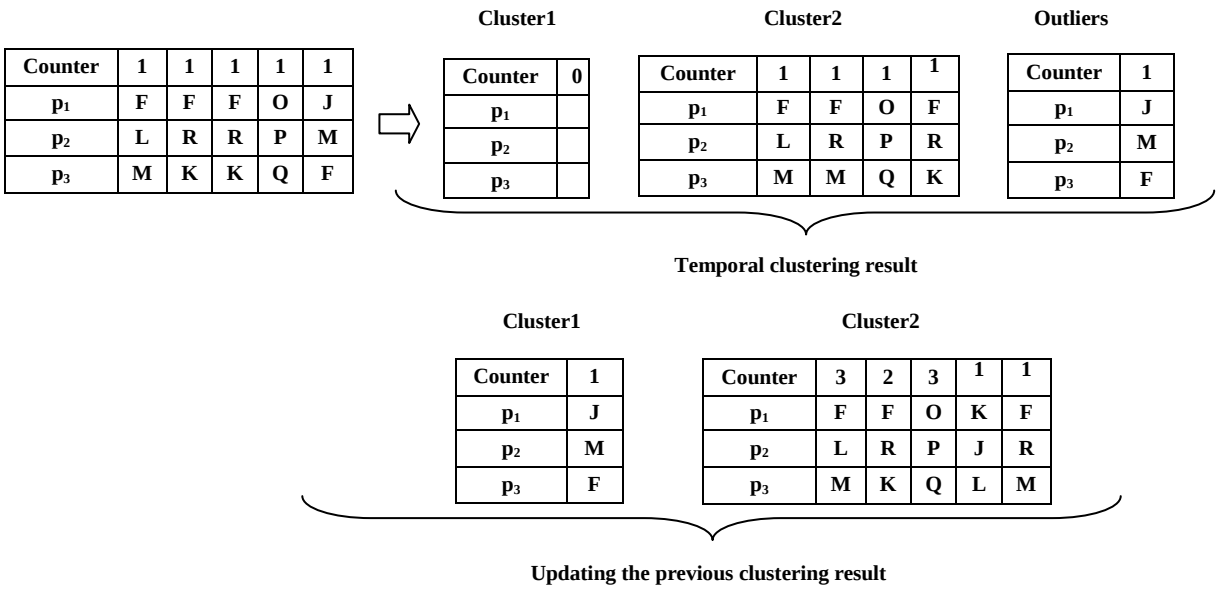
Counter	1
p_1	A
p_2	B
p_3	D

Counter	2	1	2	1
p_1	F	F	O	K
p_2	L	R	P	J
p_3	M	K	Q	L

Fig.4: Clustering on the next sliding window of example data set.

Using the RDE algorithm, sliding window W1 is created without the repetitive points. Each data point has a counter r_i, ($1< i< m_j$, m is the number of members per cluster and j is the number of clusters, 1<j <k) that keeps the number of its repetition. W1 using the initial clustering algorithm is clustered. The results of initial clustering on W1 are shown in the Figure.3. In the next step sliding window W2 is created using RDE algorithm.

NIR table, where the importance of each attribute of data point in the cluster is located, related to each cluster is generated according to the formulas presented in [19].The NIR tables are utilized to detect each data points in the current sliding window belong to which cluster, in the last clustering result. In this example, the threshold considered as: in temporary clustering results, if half of the data points is outlier



Counter	1	1	1	1	1
p_1	F	F	F	O	J
p_2	L	R	R	P	M
p_3	M	K	K	Q	F

Counter	0
p_1	
p_2	
p_3	

Counter	1	1	1	1
p_1	F	F	O	F
p_2	L	R	P	R
p_3	M	M	Q	K

Counter	1
p_1	J
p_2	M
p_3	F

Counter	1
p_1	J
p_2	M
p_3	F

Counter	3	2	3	1	1
p_1	F	F	O	K	F
p_2	L	R	P	J	R
p_3	M	K	Q	L	M

Fig.5: The final clustering result on the example data set.

or more than half of the clusters is change, re-clustering implements. In W2, due to the high rate of outliers, re-clustering is needed. Clustering on the next sliding window of data set is shown in Figure.4. In W^3, the rate of clusters changed and rate of outliers is less than the thresholds, so the previous clustering results are updated and re-clustering is not necessary. Clustering result on the rest of data set is shown in Figure.5. With the implementation of RCRDE method on the data set of this example, re-clustering occurs only once, while re-clustering needs to be perform 3 times using MARDL framework (Figure.1).

RDE algorithm for analyzing the data received from input is created. If data point is repeated, this algorithm would increase data point's counter, Otherwise it would put in the sliding window. In this method the number of received data from the input, depends on the repetition rate (The higher repetition rate, to form a window, the more data is read from the input). The RDE algorithm is shown in Figure.6.

Algorithm 1. *Repetition eliminate (D)*

WHILE *there is next data point in D*
Read a data point from input
IF *it's a repeated data point*
Add to data point counter
ELSE
Put it into current sliding window
n=n+1
END
IF *n==N*
RETURN *sliding window*
Break
END
END

Fig 6: the RE algorithm for eliminating of replicated data

On the window created by the RDE algorithm, data labeling is performed and the temporary clustering results generated and entered into the next step of RCRDE algorithm. In the RCRDE algorithm (Figure.7), the cluster distributions between the last clustering result and the temporal clustering results is compared and initial clustering algorithm is called if necessary.

4. EVALUATION RESULT

The result of implementation of RCRDE clustering method on categorical time- evolving data is shown in Figure.8. In this implementation is used EM algorithm, which is proposed in [22], to cluster data.

A synthetic data set was used in these experiments. To evaluate the performance of RCRDE method, several data sets with various data sizes has been used (Figure.8, part I). In this comparison, the fixed number of clusters with the same dimensional was employed in various clustering result.

According to the results shown in part.I of Fig.8, re-clustering rates in RCRDE method is less than re-clustering rates in MARDL.

Of course, the rate of re-clustering in RCRDE method depends on the amount of repetition in input data. In part II of Figure.8, several data sets with various percent of repetition in M has been used to evaluate the performance of this method (with fixed data size and fixed number of clusters). This survey illustrates the effect of replicated data on clustering rate. According to this results, more rate of repetition in the data points, which is received from input in order to create a sliding window, accompanied by the less rate of re-clustering in RCRDE method in compare with MARDL.

Algorithm2. *RCRDE* ($C^{(t,\ t-1),}$ W^T)

CALL *Data labeling* ($C^{(t,\ t-1),}$ W^T) *"data labeling returned number of outliers and* C^t *in the current sliding window"*

Number of changed cluster=0

FOR *all clusters* $c_i^{(t,\ t-1)}$ *in* $C^{(t,\ t-1)}$

IF $\left| \frac{\sum_{j=1}^{mi^{(t,t-1)}} counter(j)}{\sum_{x=1}^{k^{(t,t-1)}} \sum_{j=1}^{mx^{(t,t-1)}} counter(j)} - \frac{\sum_{j=1}^{mi^{t}} counter(j)}{\sum_{x=1}^{k^{(t,t-1)}} \sum_{j=1}^{mx^{t}} counter(j)} \right| > \epsilon$

Number of changed cluster ++

END

END

IF *(sum of outliers counters /M >*θ*)* ***OR*** *(Number of changed /*$k^{(t,t-1)}$ *>*η*)*

Do re-clustering on the W^T

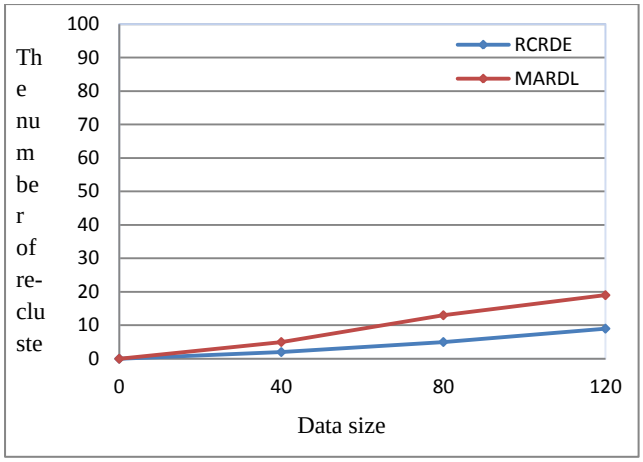
ELSE

Update $C^{(t,\ t-1)}$ *and NIR tables*

END

Fig 7: the RCRDE algorithm

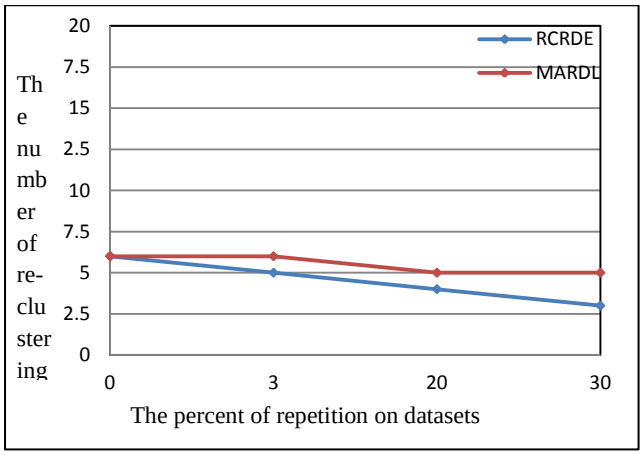
In part III of Figure.8 execution time of clustering in RCRDE method and MARDL framework on these data sets is shown. Considering this chart, the difference in repetition rates did not change the clustering execution time in the MARDL framework and this time is approximately equal in these datasets. In the first dataset, which has a lower repetition rate, the clustering execution time in this method and the MARDL framework is almost near together and in the last dataset, that contains more repetition rate, the clustering execution time in this approach compared to MARDL has further reduced.



(I)

Table1: Results of the performing clustering on D1, D2 and D3 , θ=0.5, η=0.5, Є=0.3, k=2, N=5

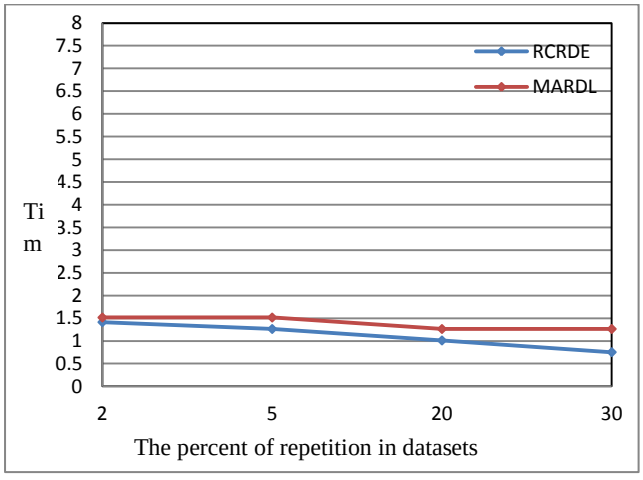
Data size	Rate of re-clustering	
	MARDL	RCRDE
40	5	2
80	13	5
120	19	9



(II)

Table.2: Results of the number of re-clustering on different percent of repetition in datasets D=80, K=2, N=5, θ=0.5, η=0.5, Є=0.3

The percent of repetition in D dataset	Rate of re-clustering	
	MARDL	RCRDE
3	6	5
20	5	4
30	5	3



(III)

Table.3: Results of the performing clustering on different percent of repetition in datasets D=80, K=2, N=5, θ=0.5, η=0.5, Є=0.3

The percent of repetition in D dataset	The execution time of clustering	
	MARDL	RCRDE
5	1.51666	1.2639
20	1.2639	1.0111
30	1.2648	0.7533

Fig 8: (I). Comparison of the rate of re-clustering between RCRDE and MARDL in three data set with different size (II). The number of re-clustering between RCRDE and MARDL on data sets with different percent of repetition and fix data size (III). Execution time comparison between RCRDE and MARDL on several data sets whit different percent of repetition

5. CONCLUSIONS

In this study is proposed a method for clustering on time-evolving data in the categorical domains with the purpose of reducing execution time of clustering on this data. This method tries to improve the MARDL framework used to determining sliding windows without the repetitive data. For

this reason, is presented RDE algorithm and RCRDE algorithms. The RDE algorithm is created to remove the successive surveys of repetitive in clustering process on categorical time-evolving data. The RCRDE algorithm, which using results generated by RDE algorithm and data labeling technique, decides to implement the re-clustering on data points.

According to the results of experiments, the algorithm performs well in practice and not only reduces the number of sliding windows, but also speeds up the clustering implementation on categorical data. This method has a significant impact on datasets that have high repetition rate.

6. REFERENCES

[1] C. C.Aggarwal, J. Han, J.Wang, and P. S.Yu, "A framework for clustering evolving data streams," Proceedings of the 29th International Conference Very Large Data Bases (VLDB), Sep. 2003, pp. 81–92.

[2] G. Widmer and M. Kubat, "Learning in the Presence of Concept Drift and Hidden Contexts," Machine Learning, April. 1996, pp. 69 – 101.

[3] H. Wang, W. Fan, P.S. Yun, J. Han, "Mining Concept-Drifting Data Streams Using Ensemble Classifiers," Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining, 2003, pp . 226 – 235.

[4] M.M. Gaber , P.S. Yu, "Detection and Classification of Changes in Evolving Data Streams," International Journal of Information Technology and Decision Making, 2006, pp. 659-670.

[5] H.-L. Chen, M.-S. Chen, S.-C. Lin, "Catching the Trend: A Framework for Clustering Concept-Drifting Categorical Data," IEEE Transactions on Knowledge and Data Engineering, May. 2009, pp.652-665.

[6] F. Cao, M. Ester, W. Qian, A. Zhou, "Density-Based Clustering over an Evolving Data Stream with Noise," Proceedings of the 6th SIAM Conference on Data Mining (SDM), 2006, pp. 326-337.

[7] D. Chakrabarti, R. Kumar, and A. Tomkins, "Evolutionary Clustering," Proc. 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006, pp. 554-560.

[8] E. H. Han, G. Karypis, V. Kumar, B. Mobasher, "Clustering Based on Association Rule Hyper graphs," Proceedings of ACM SIGMOD Workshop Research Issues in Data Mining and Knowledge Discovery (DMKD), 1997.

[9] D. Gibson, J.M. Kleinberg, P. Raghavan, "Clustering Categorical Data: An Approach Based on Dynamical Systems," VLDB Journal ,vol. 8, nos. 3-4, 2000, pp. 222-236.

[10] Z. Huang, "Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values," Data Mining and Knowledge Discovery, 1998, pp. 283-304.

[11] Z. Huang, M.K. Ng, "A Fuzzy k-Modes Algorithm for Clustering Categorical Data," IEEE Transactions on Fuzzy Systems, 1999, pp. 446 – 452.

[12] Y. Sun, Q. Zhu, and Z. Chen, "An Iterative Initial-Points Refinement Algorithm for Categorical Data Clustering," Pattern Recognition Letters, vol. 23, no. 7, May. 2002, pp. 875–884.

[13] S. Guha, R. Rastogi, and K. Shim, "ROCK: A Robust Clustering
Algorithm for Categorical Attributes," Proceedings of the 15th international conference on Data Engineering (ICDE), 2000, pp. 345-366.

[14] V. Ganti, J. Gehrke, R. Ramakrishnan, "CACTUS-Clustering Categorical Data Using Summaries," Proccedings of the ACM SIGKDD international conference on Knowledge discovery and data mining,1999, pp. 73-83.

[15] M.J. Zaki and M. Peters, "Clicks: Mining Subspace Clusters in Categorical Data via k-Partite Maximal Cliques," Proceedings of the 21st international conference on Data Engineering, April. 2005.

[16] D. Barbara, Y. Li, J. Couto, "Coolcat: An Entropy-Based Algorithm for Categorical Clustering," Proceedings of the eleventh International conference on Information and knowledge management (CIKM), 2002, pp. 582-589.

[17] P. Andritsos, P. Tsaparas, R.J. Miller, K.C. Sevcik, "Limbo: Scalable Clustering of Categorical Data," Proceedings of the 9th International Conference on Extending Database Technology (EDBT), March. 2004, pp. 123-146.

[18] H.-L. Chen, K.-T. Chuang, M.-S. Chen, "On Data Labeling for Clustering Categorical Data," IEEE Transactions on Knowledge and Data Engineering , November. 2008, pp. 1458-1471.

[19] A. Zhou, F. Cao, W. Qian, C. Jin, "Tracking Clusters in Evolving Data Streams over Sliding Windows," J. Knowledge and Information Systems, May. 2008, pp. 181-214.

[20] H.-L. Chen, K.-T. Chuang, M.-S. Chen, "Labeling Unclustered Categorical Data into Clusters Based on the Important Attribute Values," Proc. 15th IEEE International Conference on Data Mining (ICDM), 2005, pp. 27-30.

[21] A.P. Dempster, N.M. Laird, D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," J. Royal Statistical Soc., 1977, pp .1-38.

[22] B.-R. Dai, J.-W. Huang, M.-Y. Yeh, M.-S. Chen, "Adaptive Clustering for Multiple Evolving Streams," IEEE Transactions on Knowledge and Data Engineering, Sept. 2006, pp. 1166-1180.

[23] O. Nasraoui, C. Rojas, "Robust Clustering for Tracking Noisy Evolving Data Streams," Proceedings of the 6th SIAM Conference on Data Mining (SDM), 2006, pp .618-622.

[24] Y. Chi, X. Song, D. Zhou, K. Hino, B.L. Tseng, "Evolutionary Spectral Clustering by Incorporating Temporal Smoothness," Proc. 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007, pp. 153-162.

[25] M.-Y. Yeh, B.-R. Dai, M.-S. Chen, "Clustering over Multiple Evolving Streams by Events and Correlations," IEEE Transactions on Knowledge and Data Engineering, Oct. 2007, pp. 1349-1362.