A Framework for Clustering Categorical Time-Evolving Data

Fuyuan Cao, Jiye Liang, Liang Bai, Xingwang Zhao, and Chuangyin Dang, *Senior Member, IEEE*

***Abstract*—A fundamental assumption often made in unsupervised learning is that the problem is static, i.e., the description of the classes does not change with time. However, many practical clustering tasks involve changing environments. It is hence recognized that the methods and techniques to analyze the evolving trends for changing environments are of increasing interest and importance. Although the problem of clustering numerical time-evolving data is well-explored, the problem of clustering categorical time-evolving data remains as a challenging issue. In this paper, we propose a generalized clustering framework for categorical time-evolving data, which is composed of three algorithms: a drifting-concept detecting algorithm that detects the difference between the current sliding window and the last sliding window, a data-labeling algorithm that decides the most-appropriate cluster label for each object of the current sliding window based on the clustering results of the last sliding window, and a cluster-relationship-analysis algorithm that analyzes the relationship between clustering results at different time stamps. The time-complexity analysis indicates that these proposed algorithms are effective for large datasets. Experiments on a real dataset show that the proposed framework not only accurately detects the drifting concepts but also attains clustering results of better quality. Furthermore, compared with the other framework, the proposed one needs fewer parameters, which is favorable for specific applications.**

***Index Terms*—Categorical time-evolving data, clusters relationship analysis, data labeling, drifting-concept detecting.**

I. Introduction

MANY real applications, such as network-traffic monitoring, the stock market, credit card fraud detection, and web click streams, generate continuously arriving data, which are known as data streams [1]. A data stream is a real-time, continuous, ordered (implicitly by arrival time or explicitly by time-stamps) sequence of items. For data-stream applications, it is impossible to control the order in which items arrive, and the volume of data is usually too large to be stored on permanent devices or to be scanned thoroughly more than once. Moreover, the concept of interest may depend on some hidden context, not given explicitly in the form of predictive features. In other words, the concepts, which we try to learn from those data, drift with time. For example, the buying preferences of customers may change with time, depending on the current day of the week, availability of alternatives, discounting rate, etc. As the concepts behind the data evolve with time, the underlying clusters may also change considerably with time. Performing clustering on the entire time-evolving data not only decreases the quality of clusters but also disregards the expectations of users that usually require recent clustering results. It is hence recognized that the methods and techniques to analyze the evolving trends in fast data streams have become very important in recent years [2].

The problem of clustering time-evolving data in the numerical domain has been explored in the literature [3]–[13]. However, there exist many categorical data with drifting concepts in real world. For example, buying records of customers, web logs that record the browsing history of users, or web documents often evolve with time. The existing work on clustering categorical data focuses on doing clustering on the entire dataset and do not take into consideration the drifting concepts. Thus, it is desired to devise an efficient method that is able to cluster the categorial time-evolving data.

In the categorical domain, Nasraoui *et al.* [14] presented a complete framework and findings in mining web-usage patterns from Web log files of a real website that has all the challenging aspects of real-life web-usage mining, including evolving user profiles and external data describing an ontology of the web content. Chen *et al.* [15] proposed a framework to perform clustering on the categorical time-evolving data. The framework detects the drifting concepts at different sliding windows, generates the clustering results based on the current concept, and shows the relationship between clustering results by the visualization. However, this framework needs to set many system parameters, which may increase the difficulty for different applications.

Rough-set theory, which was introduced by Pawlak [16], is a kind of machine-learning technology for categorical data table with information uncertainty [17], [18]. In recent years, rough-set theory has attracted much attention in the clustering and outlier-detection literature. Parmar *et al.* [19] proposed a new algorithm min-min-roughness (MMR) to cluster categorical data based on rough-set theory, which has the ability to handle the uncertainty in the clustering process. By the notion of rough membership function in rough-set theory, Jiang *et al.* [20], [21]

Manuscript received July 17, 2009; revised January 31, 2010; accepted April 9, 2010. Date of publication May 20, 2010; date of current version September 29, 2010. This work was supported by the National Key Basic Research and Development Program of China (973) under Grant 2007CB311002, the National Natural Science Foundation of China under Grant 60773133, Grant 70971080, and Grant 60875040, the High Technology Research and Development Program of China under Grant 2007AA01Z165, the Doctor Authorization Foundation of the Ministry of Education under Grant 200801080006, and the Natural Science Foundation of Shanxi under Grant 2008011038 and Grant 2010021016-2.

F. Cao, J. Liang, L. Bai, and X. Zhao are with the School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China (e-mail: cfy@sxu.edu.cn; ljy@sxu.edu.cn; sxbailiang@126.com; zhaoxw84@163.com).

C. Dang is with the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong (e-mail: mecdang@cityu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TFUZZ.2010.2050891

defined the rough outlier factor for outlier detection. Chen and Wang [22] presented an improved clustering algorithm, which is based on rough-set and Shannon's entropy theory. Based on the neighborhood rough-set model, an initialization method for the k-means algorithm was presented [23]. Especially, the rough membership function in rough-set theory represents a vague concept and can induce a fuzzy set [24]. As rough sets and fuzzy sets have been proved to be powerful mathematical tools to deal with uncertainty, combining rough sets with fuzzy sets has become an important research topic [25]–[28].

In this paper, a framework to perform clustering on the categorical time-evolving data is proposed. In particular, this framework is independent of clustering algorithms (in other words, any categorical clustering algorithm can be utilized). The proposed framework can be summarized as follows: Based on the rough membership function and the sliding-window technique [3], [5], [6], [8], the distance between two concepts (i.e., two sliding windows) is defined, and then, a drifting-concept detecting algorithm (DCDA) is proposed. If the distance is larger than some threshold, the current sliding window will perform reclustering to capture the recent concept. In contrast, if the concept is steady, each object of the current window will be allocated into the corresponding proper cluster according to the similarity between it and the clustering results of last sliding window, which is named as a data-labeling algorithm (DLA). Moreover, a cluster-relationship-analysis algorithm (CRAA) is proposed, which can explain the drifting concepts by analyzing the relationship between clustering results at different time-stamps, and capture the time-evolving trend in the dataset. The time-complexity analysis indicates that the proposed algorithms are effective for large datasets. Experiments on a real dataset show that the proposed algorithms not only accurately detect the drifting concepts but also attain clustering results of better quality. Furthermore, compared with Chen's framework [15], the proposed one needs fewer parameters, which is favorable for practical applications.

The outline of the rest of this paper is as follows. In Section II, some basic concepts of rough-set theory are reviewed, and the problem of the categorical time-evolving is formulated. In Section III, the distance between two concepts is defined based on the rough membership function, the DCDA and the DLA are proposed, and the corresponding time complexity is also analyzed. In Section IV, based on the idea of the distance between two concepts, the distance between clustering results of different sliding windows is given, the CRAA is described, and the corresponding time complexity is analyzed as well. Experimental studies on a real dataset are conducted in Section V. This paper concludes with some remarks in Section VI.

II. Preliminaries

In Section II-A, several basic concepts are reviewed, including indiscernibility relations, lower and upper approximations, and rough membership functions [16]. After that, the formal description of clustering the categorical time-evolving data follows in Section II-B.

A. Some Basic Concepts of Rough-Set Theory

As we know, the structural data are stored in a table, where each row (tuple) represents facts about an object. Data in the real world are prevalently described by categorical attributes. More formally, a categorical data table can be defined as a quadruple $IS = (U, A, V, f)$, where

U—a nonempty set of objects, which is called the universe;

A—a nonempty set of attributes;

V—the union of all attribute domains, i.e., $V = \bigcup_{a \in A} V_a$, where V_a is the domain of attribute a and is finite and unordered;

$f : U \times A \to V$—a mapping, which is called an information function, such that for any $x \in U$ and $a \in A$, $f(x, a) \in V_a$.

For any attribute subset $P \subseteq A$, a binary relation $\mathrm{IND}(P)$, which is called indiscernibility relation, is defined as

$$\mathrm{IND}(P) = \{(x, y) \in U \times U | \forall a \in P, f(x, a) = f(y, a)\}. \tag{1}$$

It is obvious that $\mathrm{IND}(P)$ is an equivalence relation on U and $\mathrm{IND}(P) = \bigcap_{a \in P} \mathrm{IND}(\{a\})$.

Given $P \subseteq A$, the relation $\mathrm{IND}(P)$ induces a partition of U, which is denoted by $U/\mathrm{IND}(P) = \{[x]_P | x \in U\}$, where $[x]_P$ denotes the equivalence class determined by x with respect to P, i.e., $[x]_P = \{y \in U | (x, y) \in \mathrm{IND}(P)\}$.

As follows, we give the definitions of a lower approximation and an upper approximation in rough-set theory.

For any given categorical data table $IS = (U, A, V, f)$, with $P \subseteq A$ and $X \subseteq U$, one can define a lower approximation of X in U and an upper approximation of X in U by

$$\underline{P}X = \{x | [x]_P \subseteq X\} \tag{2}$$

and

$$\overline{P}X = \{x | [x]_P \bigcap X \neq \emptyset\} \tag{3}$$

where $\underline{P}X$ is a set of objects that belong to X with certainty, while $\overline{P}X$ is a set of objects that possibly belong to X.

The set $BN_P(X) = \overline{P}X - \underline{P}X$ is called the P-boundary region of X and consists of those objects that we cannot decisively classify into X on the basis of knowledge in P. The set $U - \overline{P}X$ is called the P-outside region of X and consists of those objects that can be with certainty classified as not belonging to X.

In classical set theory, an element either belongs to a set, or it does not. The corresponding membership function is the characteristic function of the set, i.e., the function takes values 1 and 0, respectively. In the case of rough sets, the notion of membership is different.

Definition 1 [16]: Let $IS = (U, A, V, f)$ be a categorical data table, with $P \subseteq A$ and $X \subseteq U$. The rough membership function $\mu_X^P : U \to [0, 1]$ is defined as

$$\mu_X^P(x) = \frac{|[x]_P \bigcap X|}{|[x]_P|}. \tag{4}$$

The rough membership function quantifies the degree of relative overlap between the set X and the equivalence class $[x]_P$ to which x belongs. Obviously, the rough membership function takes values between 0 and 1. Therefore, the rough membership

function represents a vague concept and can induce a fuzzy set F_X^P of U, which is given by $F_X^P = \{(x, \mu_X^P(x)) | x \in U\}$.

B. Problem Description of the Categorical Time-Evolving Data

Similarly, a categorical time-evolving data can also be stored in a table. More formally, a categorical time-evolving data table can be formulated as a quintuple $TIS = (U, A, V, f, t)$, where

U—a nonempty set of objects, which is called the universe;

A—a nonempty set of attributes;

V—the union of all attribute domains, i.e., $V = \bigcup_{a \in A} V_a$, where V_a is a set of attribute values for a, which is the domain of attribute a and is finite and unordered;

$f : U \times A \times t \to V$—a mapping, which is called an information function, such that for any $x \in U$ and $a \in A$, $f(x, a, t) \in V_a$, where t is the arriving time of object x.

Suppose that the sliding-window size N is given; then, the TIS is separated into several continuous subset $S^{T_i} (1 \le i \le \lfloor \frac{U}{N} \rfloor)$, where the number of objects in each S^{T_i} is N. The superscript number T_i is the identification number of the sliding window and is also called the time stamp in this paper. For example, the first N objects in TIS are located in the first subset S^{T_1}.

III. Drifting-Concept Detecting

In this section, based on the rough membership function and the sliding-window technique, the distance between two concepts, i.e., the difference between the current subset S^{T_j} and the last subset S^{T_i}, is defined. If the difference is large enough, the T_jth sliding window will be considered as a concept-drifting window, and S^{T_j} will perform reclustering. In contrast, each object of the current window S^{T_j} will be allocated into the corresponding proper cluster according to the clustering results of S^{T_i} by the data-labeling technique. Based on the foregoing discussion, a DCDA and a DLA are presented, and the corresponding time complexity is analyzed as well.

A. Distance Between Two Concepts

Definition 2: Let $TIS = (U, A, V, f, t)$ be a categorical time-evolving data table and $S^{T_i}, S^{T_j} \subseteq U$, where $S^{T_i} \bigcap S^{T_j} = \emptyset$ and $S^{[T_i,T_j]} = S^{T_i} \bigcup S^{T_j}$. The difference measure between S^{T_i}, and S^{T_j} with respect to A is defined as

$$d_A(S^{T_i}, S^{T_j}) = \frac{1}{|A|} \sum_{a \in A} d_{\{a\}}(S^{T_i}, S^{T_j}) = \frac{\sum_{a \in A} \sum_{x \in S^{[T_i,T_j]}} \left| \mu_{S^{T_i}}^{\{a\}}(x) - \mu_{S^{T_j}}^{\{a\}}(x) \right|}{|S^{[T_i,T_j]}||A|}. \tag{5}$$

If $S^{[T_i,T_j]}/\mathrm{IND}(\{a\}) = \{X | X = \{u\}, u \in S^{[T_i,T_j]}\}$, where $a \in A$, then $d_{\{a\}}(S^{T_i}, S^{T_j})$ achieves its maximum value 1.

If $S^{[T_i,T_j]}/\mathrm{IND}(\{a\}) = \{X | X = S^{[T_i,T_j]}\}$, where $a \in A$, then $d_{\{a\}}(S^{T_i}, S^{T_j})$ achieves its minimum value 0.

For $d_A(S^{T_i}, S^{T_j})$, it is easy to prove the following properties.

Property 1: Let $TIS = (U, A, V, f, t)$ be a categorical time-evolving data table. For any $S^{T_i}, S^{T_j}, S^{T_k} \subseteq U$, where $S^{T_i} \bigcap S^{T_j} \bigcap S^{T_k} = \emptyset$, we have the following:

1) *Symmetry:* $d_A(S^{T_i}, S^{T_j}) = d_A(S^{T_j}, S^{T_i})$;
2) *Nonnegativity:* $d_A(S^{T_i}, S^{T_j}) \ge 0$;
3) *Triangle inequality:* $d_A(S^{T_i}, S^{T_j}) + d_A(S^{T_j}, S^{T_k}) \ge d_A(S^{T_i}, S^{T_k})$.

Property 1 shows that the difference measure d_A is a distance metric.

Property 2: Let $TIS = (U, A, V, f, t)$ be a categorical time-evolving data table, $a \in A$ and $S^{T_i}, S^{T_j} \subseteq U$, where $S^{T_i} \bigcap S^{T_j} = \emptyset$ and $S^{[T_i,T_j]} = S^{T_i} \bigcup S^{T_j}$.

1) If $x \in \underline{\{a\}} S^{T_i} \bigcup \underline{\{a\}} S^{T_j}$, then $|\mu_{S^{T_i}}^{\{a\}}(x) - \mu_{S^{T_j}}^{\{a\}}(x)| = 1$.
2) If $x \in S^{[T_i,T_j]} - (\underline{\{a\}} S^{T_i} \bigcup \underline{\{a\}} S^{T_j})$, then $|\mu_{S^{T_i}}^{\{a\}}(x) - \mu_{S^{T_j}}^{\{a\}}(x)| = 0$.
3) If $\overline{\{a\}} S^{T_i} \bigcap \overline{\{a\}} S^{T_j} = \emptyset$ and $x \in S^{[T_i,T_j]}$, then $|\mu_{S^{T_i}}^{\{a\}}(x) - \mu_{S^{T_j}}^{\{a\}}(x)| = 1$.
4) If $x \in BN_{\{a\}} S^{T_i} \bigcup BN_{\{a\}} S^{T_j}$, then $0 < |\mu_{S^{T_i}}^{\{a\}}(x) - \mu_{S^{T_j}}^{\{a\}}(x)| < 1$.

Property 3: Letting $TIS = (U, A, V, f, t)$ be a categorical time-evolving data table and $S^{T_i}, S^{T_j} \subseteq U$, where $S^{T_i} \bigcap S^{T_j} = \emptyset$ and $S^{[T_i,T_j]} = S^{T_i} \bigcup S^{T_j}$, then

$$d_A(S^{T_i}, S^{T_j}) = \frac{\sum_{a \in A} \sum_{Y \in c_m^{\{a\}}} \| Y \bigcap S^{T_i} | - | Y \bigcap S^{T_j} \|}{|S^{[T_i,T_j]}||A|} \tag{6}$$

where $c_m^{\{a\}} = S^{[T_i,T_j]}/\mathrm{IND}(\{a\})$.

Proof: We have that

$$\begin{aligned} & d_A(S^{T_i}, S^{T_j}) \\ &= \frac{\sum_{a \in A} \sum_{x \in S^{[T_i,T_j]}} |\mu_{S^{T_i}}^{\{a\}}(x) - \mu_{S^{T_j}}^{\{a\}}(x)|}{|S^{[T_i,T_j]}||A|} \\ &= \frac{\sum_{a \in A} \sum_{x \in S^{[T_i,T_j]}} \left| \frac{|[x]_{\{a\}} \bigcap S^{T_i}|}{|[x]_{\{a\}}|} - \frac{|[x]_{\{a\}} \bigcap S^{T_j}|}{|[x]_{\{a\}}|} \right|}{|S^{[T_i,T_j]}||A|} \\ &= \frac{\sum_{a \in A} \sum_{x \in S^{[T_i,T_j]}} \| [x]_{\{a\}} \bigcap S^{T_i}| - |[x]_{\{a\}} \bigcap S^{T_j}| |}{|S^{[T_i,T_j]}||A||[x]_{\{a\}}|} \\ &= \frac{\sum_{a \in A} \sum_{Y \in c_m^{\{a\}}} \sum_{x \in Y} \| [x]_{\{a\}} \bigcap S^{T_i}| - |[x]_{\{a\}} \bigcap S^{T_j}| |}{|S^{[T_i,T_j]}||A||[x]_{\{a\}}|} \\ &= \frac{\sum_{a \in A} \sum_{Y \in c_m^{\{a\}}} |[x]_{\{a\}}| \| [x]_{\{a\}} \bigcap S^{T_i}| - |[x]_{\{a\}} \bigcap S^{T_j}| |}{|S^{[T_i,T_j]}||A||[x]_{\{a\}}|} \\ &= \frac{\sum_{a \in A} \sum_{Y \in c_m^{\{a\}}} \| Y \bigcap S^{T_i}| - |Y \bigcap S^{T_j}| |}{|S^{[T_i,T_j]}||A|}. \end{aligned}$$

■

Example 1: A categorical time-evolving example dataset is given in Table I.

In Table I, $U = \{x_1, x_2, \ldots, x_{20}\}$ is the universe, and $A = \{A_1, A_2, A_3\}$ is the attribute set. Suppose that the size of sliding window is $N = 5$; we have $S^{T_1} = \{x_1, x_2, \ldots, x_5\}$, $S^{T_2} =$

TABLE I
CATEGORICAL TIME-EVOLVING EXAMPLE DATASET

Object	A_1	A_2	A_3
x_1	A	M	C
x_2	Y	E	P
x_3	X	E	P
x_4	Y	M	P
x_5	A	M	D
x_6	A	M	C
x_7	X	M	P
x_8	A	M	D
x_9	Y	M	P
x_{10}	A	M	C
x_{11}	B	E	G
x_{12}	X	M	P
x_{13}	B	E	D
x_{14}	Y	M	P
x_{15}	B	F	D
x_{16}	Y	M	P
x_{17}	X	M	P
x_{18}	Z	N	T
x_{19}	X	M	P
x_{20}	Y	M	P

$\{x_6, x_7, \ldots, x_{10}\}$, $S^{T_3} = \{x_{11}, x_{12}, \ldots, x_{15}\}$, and $S^{T_4} = \{x_{16}, x_{17}, \ldots, x_{20}\}$. In the following, the computational process of the distance between S^{T_1} and S^{T_2} is described.

By calculating, one can have

$$S^{[T_1,T_2]}/\text{IND}(\{A_1\}) = \{\{x_1, x_5, x_6, x_8, x_{10}\}, \{x_2, x_4, x_9\}, \{x_3, x_7\}\}$$

$$S^{[T_1,T_2]}/\text{IND}(\{A_2\}) = \{\{x_1, x_4, \ldots, x_{10}\}, \{x_2, x_3\}\}$$

$$S^{[T_1,T_2]}/\text{IND}(\{A_3\}) = \{\{x_1, x_6, x_{10}\}, \{x_2, x_3, x_4, x_7, x_9\}, \{x_5, x_8\}\}.$$

According to Property 3, we have

$$d_{\{A_1\}}(S^{T_1}, S^{T_2}) = \frac{1}{5}$$

$$d_{\{A_2\}}(S^{T_1}, S^{T_2}) = \frac{2}{5}$$

$$d_{\{A_3\}}(S^{T_1}, S^{T_2}) = \frac{1}{5}.$$

Therefore, we have

$$d_A(S^{T_1}, S^{T_2}) = \frac{4}{15}.$$

In a similar way, we have

$$d_A(S^{T_2}, S^{T_3}) = \frac{8}{15}, \qquad d_A(S^{T_3}, S^{T_4}) = \frac{9}{15}.$$

Then, it is easy to see that

$$d_A(S^{T_1}, S^{T_2}) < d_A(S^{T_2}, S^{T_3}) < d_A(S^{T_3}, S^{T_4}).$$

TABLE II
DCDA

```
Input: TIS = (U, A, V, f, t), N and θ, where N is
       the size of sliding window and θ is the
       specified threshold value;
Begin
    Drifting-Window=∅;
    For i = 1 to ⌊|U|/N⌋ − 1
        If Distance-Between-Concepts(S^{T_i}, S^{T_{i+1}}, A) ≥ θ
            Drifting-Window=Drifting-Window⋃{i + 1};
        End;
    End;
End;
Output: Drifting-Window.
```

TABLE III
COMPUTATIONAL PROCESS OF THE DISTANCE BETWEEN TWO CONCEPTS

```
Function Distance-Between-Concepts(S^{T_i}, S^{T_j}, A).
Begin
    S^{[T_i,T_j]} = S^{T_i} ⋃ S^{T_j};
    distance = 0;
    For p = 1 to |A|
        S^{[T_i,T_j]}/IND({a_p}) = {c_1, c_2, ···, c_{m_p}}, a_p ∈ A;
        For j = 1 to m_p
          distance = distance + ||c_j ⋂ S^{T_i}| − |c_j ⋂ S^{T_j}||;
        End;
    End;
    Return distance/(|A||S^{[T_i,T_j]}|) ;
End;
```

Suppose that the concept-drifting threshold is set to 0.5; then, T_3 and T_4 are considered as two concept-drifting windows. Therefore, S^{T_3} and S^{T_4} are going to perform reclustering.

B. Drifting-Concept Detecting Algorithm

Based on the distance between two sliding windows or two concepts, the $DCDA$ is presented in Table II. In addition, the computational process of the distance between two concepts is described in Table III.

The runtime complexity of the $DCDA$ is analyzed as follows. The runtime complexity to compute the distance between sliding windows is $O((|S^{[T_i,T_{i+1}]}| + m_p)|A|) = O(|S^{[T_i,T_{i+1}]}\|A|)$. Therefore, the whole computational cost of the $DCDA$ is $O(\lfloor\frac{|U|}{N}\rfloor|S^{[T_i,T_{i+1}]}\|A|) = O(\lfloor\frac{|U|}{N}\rfloor 2N|A|) = O(|U\|A|)$, where U is the universe, $|A|$ is the number of attributes, N is the size of sliding window, and m_p is the number of distinct categorical values with respect to attribute $a_p \in A$. Based on the above analysis, the time complexity of the $DCDA$ is linear with respect to the number of the objects in U.

C. Data-Labeling Algorithm

The goal of clustering is to allocate every data object into an appropriate cluster. For the current sliding window, if the stable concept remains, the clustering results of the current sliding window can be obtained by data-labeling technique. In other

TABLE IV
SIMILARITY BETWEEN EACH OBJECT OF S^{T_2} AND EACH CLUSTER OF S^{T_1}

	x_6	x_7	x_8	x_9	x_{10}
$c_1^{T_1}$	0.8333	0.3333	0.8333	0.3333	0.8333
$c_2^{T_1}$	0.1111	0.5556	0.1111	0.6667	0.1111

words, based on the similarity between an unlabeled data object and a cluster, each data object in the current sliding window can be allocated to the cluster in the last sliding window with the maximal similarity. Note that after executing the data labeling, the labeled data point just obtains a cluster label instead of being really added to the cluster.

Definition 3: Let $TIS = (U, A, V, f, t)$ be a categorical time-evolving data table and $S^{T_i}, S^{T_j} \subseteq U$, where $S^{T_i} \bigcap S^{T_j} = \emptyset$ and $S^{[T_i,T_j]} = S^{T_i} \bigcup S^{T_j}$. Suppose that a prior clustering result $C^{T_i} = \{c_1^{T_i}, c_2^{T_i}, \ldots, c_{k_{T_i}}^{T_i}\}$ is given on S^{T_i}, where $c_m^{T_i}$, $1 \le m \le k_{T_i}$ is the mth cluster. For any unlabeled object $x \in S^{T_j}$, the similarity between x and the cluster $c_m^{T_i}$ with respect to A is defined as

$$\mathrm{Sim}_A\left(x, c_m^{T_i}\right) = \frac{1}{|A|} \sum_{a \in A} \frac{|[x]_{\{a\}}^{S^{T_i}} \bigcap c_m^{T_i}|}{|c_m^{T_i}|} \tag{7}$$

where $[x]_{\{a\}}^{S^{T_i}}$ denotes the equivalence class determined by x with respect to a in the universe S^{T_i}, i.e., $[x]_{\{a\}}^{S^{T_i}} = \{u \in S^{T_i} | f(u,a) = f(x,a)\}$.

Obviously, we have $0 \le \mathrm{Sim}_A(x, c_m) \le 1$.

Example 2 (Continued from Example 1): Since $d_A(S^{T_1}, S^{T_2}) \le 0.5$, we need to decide the most-appropriate cluster label for each object of S^{T_2}. Suppose that the clustering results of S^{T_1} are $C^{T_1} = \{c_1^{T_1}, c_2^{T_1}\}$, where $c_1^{T_1} = \{x_1, x_5\}$, and $c_2^{T_1} = \{x_2, x_3, x_4\}$. According to Definition 3, the similarity between each object of S^{T_2} and each cluster of S^{T_1} is shown in Table IV.

From Table IV, we obtain that $c_1^{T_2} = \{x_6, x_8, x_{10}\}$ and $c_2^{T_2} = \{x_7, x_9\}$. The pseudocode of the algorithm to label unlabeled categorical data is described in Table V.

The runtime complexity of the DLA is analyzed as follows. The runtime complexity to compute the similarity between an arbitrary object and a cluster is $O(|S^{T_j}||A|)$. Therefore, the total computational cost of the proposed algorithm is $O(|S^{T_i}||A||S^{T_j}|k_{T_i})$. Based on the above analysis, the time complexity on the data-labeling phase is linear with respect to the number of the objects in the unlabeled dataset S^{T_j}, i.e., the size of the sliding window.

TABLE V
DLA

```
Function Data-Labeling(S^{T_i}, S^{T_j}, A).
Begin
    Generate a partition C^{T_i} = {c_1^{T_i}, c_2^{T_i}, ..., c_{k_{T_i}}^{T_i}} of
    S^{T_i} with respect to A by calling the corresponding
    categorical clustering algorithm;
    For j' = 1 to |S^{T_j}|
        For i' = 1 to k_{T_i}
            Calculate the similarity Sim_A(x_{j'}^{T_j}, c_{i'}^{T_i})
            according to Definition 3, where x_{j'}^{T_j}
            is the j'th object in S^{T_j}.
        End;
        Give label L to x_{j'}^{T_j}, where
        L = arg max_{m=1,...,k_{T_i}} {Sim_A(x_{j'}^{T_j}, c_m^{T_i})};
    End;
    Return C^{T_j} = {c_1^{T_j}, c_2^{T_j}, ..., c_{k_{T_i}}^{T_j}};
End;
```

IV. CLUSTER-RELATIONSHIP ANALYSIS

After performing clustering on the entire dataset where the drifting concept is considered, several clustering results with time stamps are obtained. Each clustering result is generated from one concept that persists over a period of time. In order to analyze the relationship between clusters, the distance between clusters and the representative of a cluster are defined. Furthermore, a visualizing algorithm that tries to present the evolving trend of clustering results is proposed.

A. Distance Between Two Clusters

In order to present the relationship between clusters at different time-stamps, the distance between two clusters is defined. The computation of the distance between two clusters is similar to that of two concepts.

Definition 4: Let $TIS = (U, A, V, f, t)$ be a categorical time-evolving data table and $S^{T_i}, S^{T_j} \subseteq U$, where $S^{T_i} \bigcap S^{T_j} = \emptyset$ and $S^{[T_i,T_j]} = S^{T_i} \bigcup S^{T_j}$. Suppose that clustering results $C^{T_i} = \{c_1^{T_i}, c_2^{T_i}, \ldots, c_{k_{T_i}}^{T_i}\}$ and $C^{T_j} = \{c_1^{T_j}, c_2^{T_j}, \ldots, c_{k_{T_j}}^{T_j}\}$ are given on S^{T_i} and S^{T_j}, respectively. The distance between $c_{i'}^{T_i}$ and $c_{j'}^{T_j}$ with respect to A is defined as

$$d_A\left(c_{i'}^{T_i}, c_{j'}^{T_j}\right) = \frac{\sum_{a \in A} \sum_{x \in C^{[T_i,T_j]}} |\mu_{c_{i'}^{T_i}}^{\{a\}}(x) - \mu_{c_{j'}^{T_j}}^{\{a\}}(x)|}{|C^{[T_i,T_j]}||A|} \tag{8}$$

where $C^{[T_i,T_j]} = c_{i'}^{T_i} \bigcup c_{j'}^{T_j}$, $1 \le i' \le k_{T_i}$, $1 \le j' \le k_{T_j}$.

Example 3 (Continued from Example 2): From Example 2, we have obtained that $c_1^{T_1} = \{x_1, x_5\}$, $c_2^{T_1} = \{x_2, x_3, x_4\}$, $c_1^{T_2} = \{x_6, x_8, x_{10}\}$, and $c_2^{T_2} = \{x_7, x_9\}$. In Example 1, T_3 and T_4 are considered as concept-drifting windows, and we suppose that the reclustering results of S^{T_3} and S^{T_4} are $C^{T_3} = \{c_1^{T_3}, c_2^{T_3}\}$ and $C^{T_4} = \{c_1^{T_4}, c_2^{T_4}\}$, where $c_1^{T_3} = \{x_{11}, x_{13}, x_{15}\}$, $c_2^{T_3} = \{x_{12}, x_{14}\}$, $c_1^{T_4} = \{x_{16}, x_{17}, x_{19}, x_{20}\}$, and $c_2^{T_4} = \{x_{18}\}$. The distances of the clustering results between S^{T_i} and $S^{T_{i+1}}$ $(1 \le i \le 3)$ are shown in Table VI, respectively.

B. Visualizing the Evolving Clusters

To facilitate the observation of the evolving clusters, the representative of a cluster is necessary. First, we review the mode of a set [29]. Let $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ be a set of n objects in which each object x_i is represented as $[x_{i1}, x_{i2}, \ldots, x_{im}]$, where m is the number of categorical attributes. A mode of $\mathbf{X}$ is a vector $Q = [q_1, q_2, \ldots, q_m]$ that minimizes $D(\mathbf{X}, Q) =$

TABLE VI
DISTANCES OF CLUSTERS BETWEEN THE CLUSTERING RESULTS AT DIFFERENT TIME-STAMPS

	$c_1^{T_1}=\{x_1,x_5\}$	$c_2^{T_1}=\{x_2,x_3,x_4\}$
$c_1^{T_2}=\{x_6,x_8,x_{10}\}$	0.2000	0.8889
$c_2^{T_2}=\{x_7,x_9\}$	0.6667	0.3333
	$c_1^{T_2}=\{x_6,x_8,x_{10}\}$	$c_2^{T_2}=\{x_7,x_9\}$
$c_1^{T_3}=\{x_{11},x_{13},x_{15}\}$	0.8889	1
$c_2^{T_3}=\{x_{12},x_{14}\}$	0.7333	0
	$c_1^{T_3}=\{x_{11},x_{13},x_{15}\}$	$c_2^{T_3}=\{x_{12},x_{14}\}$
$c_1^{T_4}=\{x_{16},x_{17},x_{19},x_{20}\}$	1	0.3333
$c_2^{T_4}=\{x_{18}\}$	1	1

TABLE VII
REPRESENTATIVE OF EACH CLUSTER AT DIFFERENT TIME-STAMPS

Cluster	Representative
$c_1^1=\{x_1,x_5\}$	$R(c_1^1)=\{A,M,C\}$
$c_2^1=\{x_2,x_3,x_4\}$	$R(c_2^1)=\{Y,E,P\}$
$c_1^2=\{x_6,x_8,x_{10}\}$	$R(c_1^2)=\{A,M,C\}$
$c_2^2=\{x_7,x_9\}$	$R(c_2^2)=\{X,M,P\}$
$c_1^3=\{x_{11},x_{13},x_{15}\}$	$R(c_1^3)=\{B,E,D\}$
$c_2^3=\{x_{12},x_{14}\}$	$R(c_2^3)=\{X,M,P\}$
$c_1^4=\{x_{16},x_{17},x_{19},x_{20}\}$	$R(c_1^4)=\{Y,M,P\}$
$c_2^4=\{x_{18}\}$	$R(c_2^4)=\{Z,N,T\}$

$\sum_{i=1}^{n}\sum_{j=1}^{m} d(x_{ij},q_j)$, where

$$d(x_{ij},q_j)=\begin{cases}0, & x_{ij}=q_j\\ 1, & \text{otherwise.}\end{cases}$$

In other words, $q_i(1\le i\le m)$ is the most-frequent value in $\mathbf{X}$ with respect to the ith attribute such that the vector Q is a mode. Here, Q is not necessarily an object of $\mathbf{X}$. However, "mode" mainly focuses on the intracluster similarity and does not take the intercluster similarity into account. In the following, the representative of a cluster is defined, which considers both the intracluster similarity and the intercluster similarity.

Definition 5: Let $TIS=(U,A,V,f,t)$ be a categorical time-evolving data table and $S^T\subseteq U$. Suppose that clustering results $C^T=\{c_1^T,c_2^T,\ldots,c_{k_T}^T\}$ are given on S^T. The representative of a cluster $c_i^T\in C^T$ is defined as

$$R(c_i^T)=\left\{q_j|q_j=\arg\max_{q_{j'}\in V_{a_j}} m'_{a_j}\times\omega'_{a_j}, j=1,2,\ldots,|A|\right\} \tag{9}$$

where

$$m'_{a_j}=\frac{|\{x|f(x,a_j)=q_{j'},x\in c_i^T\}|}{|c_i^T|}$$

and

$$\omega'_{a_j}=\frac{|\{x\in c_i^T|f(x,a_j)=q_{j'}\}|}{|\{x\in S^T|f(x,a_j)=q_{j'}\}|}$$

Example 4 (Continued from Example 3): According to Definition 5, the representative of each cluster is shown in Table VII.

TABLE VIII
VISUALIZING ALGORITHM

```
Procedure Visualizing(S^{T_i}, S^{T_j}, A, γ), where γ is
the specified threshold value;
Begin
    Obtain clustering results C^{T_i} = {c_1^{T_i}, c_2^{T_i}, ..., c_{k_{T_i}}^{T_i}} and
    C^{T_j} = {c_1^{T_j}, c_2^{T_j}, ..., c_{k_{T_j}}^{T_j}} with respect to A;
    For i' = 1 to k_{T_i}
      Generate R(c_{i'}^{T_i}) according to Definition 5;
    End;
    For j' = 1 to k_{T_j}
      Generate R(c_{j'}^{T_j}) according to Definition 5;
    End;
    For i' = 1 to k_{T_i}
      For j' = 1 to k_{T_j}
        If d_A(c_{i'}^{T_i}, c_{j'}^{T_j}) ≤ γ
          Connect c_{i'}^{T_i}, c_{j'}^{T_j} with line;
        End;
      End;
    End;
End;
```

Based on the above ideas, a visualizing algorithm of cluster-relationship analysis is described in Table VIII.

The runtime complexity of the visualizing algorithm is analyzed as follows. The total computational cost of the $CRAA$ is $O(|S^{T_i}\|A|k_{T_i}+|S^{T_j}\|A|k_{T_j}+k^{T_i}k^{T_j}|S^{T_i}\bigcup S^{T_j}\|A|)=O(k^{T_i}k^{T_j}|S^{T_i}\bigcup S^{T_j}\|A|)$.

Suppose that threshold value $\gamma=0.2$. Fig. 1 shows the evolving process between clusters at different time stamps.

In Fig. 1, the horizontal direction is the time axis. The blue and red circles in a column indicate the different clustering results at the same time stamp. Note that the size of each circle represents the number of objects in the clustering results. The content in each circle is the "representative" of each cluster. In addition, we use lines to link the similar clusters. If we mark all cluster relations in the visualization, the lines are too numerous to show the clusters evolving clearly. Therefore, a user-specified threshold is used to prune the unimportant relationship when the distance between clusters is too large.

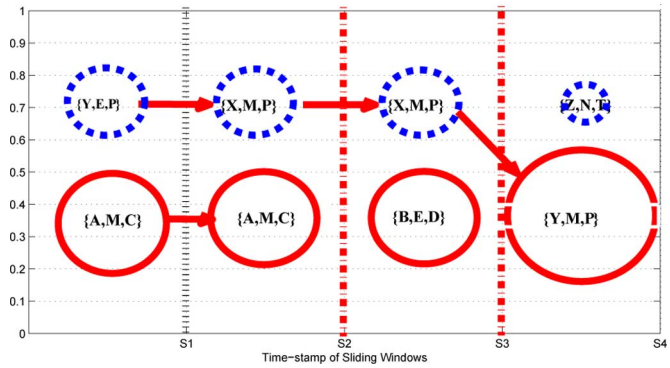


Fig. 1. Visualization of the evolving clusters.

V. Experimental Analysis

In this section, we demonstrate the performance of the proposed framework on clustering categorical time-evolving data by a thorough experimental study on the real dataset. In Section V-A, the test environment and the dataset used are described. The comparison of different frameworks is presented in Section V-B. In Section V-C, the evolving processes of clustering results at different time stamps are visualized on the real dataset.

A. Test Environment and Dataset

All of our experiments are conducted on a PC with an Intel Pentium D (2.8 G) processor with 1 GB memory and the Windows XP SP3 professional operating system. In all experiments, the k-modes [29] clustering algorithm is chosen to do the initial clustering and reclustering on the datasets. As the k-modes algorithm is dependent on the selection of initial cluster centers, we utilize an initialization method, which was proposed in [33], to obtain initial cluster centers before executing the k-modes.

The KDD-CUP'99 network-intrusion-detection stream dataset [30], which has been used earlier to evaluate several stream-clustering algorithms and DCDAs, is used in our study. The network-intrusion-detection dataset consists of a series of transmission control protocol (TCP) connection records from two weeks of LAN traffic managed by the Lincoln Laboratories at the Massachusetts Institute of Technology. Each record can either correspond to a normal connection or an intrusion (or attack). The attacks fall into 22 types, such as buffer-overflow, guess-passwd, neptune, portsweep, rootkit, smurf, spy, etc. As a result, the data contain a total of 23 classes including the class for "normal connection." In the following experiments, all different 22 attack-types are seen as "attack." We utilize the class label which indicates that the record is a normal connection or an attack to identify the drifting concept. Most of the connections in this dataset are normal, but occasionally, there could be a burst of attacks at certain times. One of the objectives in the intrusion-detection system is to detect the changes of connections from normal to a burst of attacks or from the attacks back to normal, and those changes naturally correspond to a drifting concept. Therefore, this dataset is time-evolving data and is suitable for evaluating our algorithms.

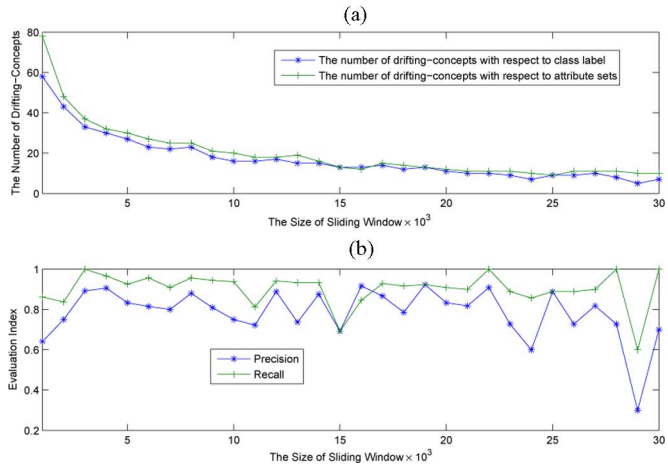


Fig. 2. Number of drifting concepts, precision, and recall varying with the size of sliding window. (a) The variations of the number of drifting-concepts on KDD-CUP'99 data set varying the size of sliding window. (b) The variations of the Precision and Recall on the KDD-CUP'99 data set varying the size of the sliding window.

We utilize the 10% subset version, which is provided from the KDD-CUP'99 website for our experiments. In this dataset, there are 494 021 records, and each record contains 42 attributes (class label is included), such as the duration of the connection, the number of data bytes transmitted from source to destination (and *vice versa*), the percentile of connections that have "SYN" errors, the number of "root" accesses, etc. Also, 34 attributes are continuous. We adopt uniform quantization on those numerical attributes where each attribute is quantized into five categorical values.

B. Evaluation on Accuracy

1) Drifting-Concept Detecting: In order to evaluate the effectiveness of $DCDA$, the following two evaluation indexes, i.e., precision and recall, are employed in this experiment. Suppose that a dataset and the size of sliding window are given. In order to define the two kinds of evaluation indexes, the following quantities are needed:

a—the number of drifting concepts with respect to class label;

b—the number of drifting concepts with respect to attribute sets (not including class label);

c—the number of drifting concepts that are correctly detected by attribute sets (not including class label).

The precision and recall are defined as

$$\text{Precision} = \frac{c}{b} \tag{10}$$

and

$$\text{Recall} = \frac{c}{a} \tag{11}$$

respectively.

A drifting concept is recognized if the characteristic of the current window is very different from that of the last window. Setting the size of sliding window and a proper threshold value is very important for detecting the concept drifting. If the dataset varies dramatically, one can set smaller sliding-window size to capture the frequent drifting concepts. In contrast, if the dataset

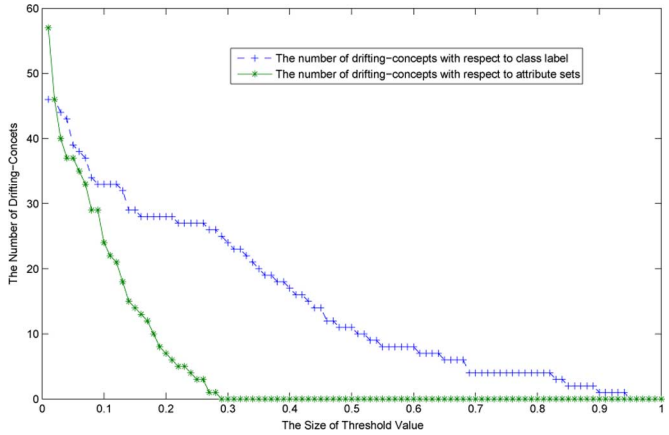


Fig. 3. Number of drifting concepts varying with the size of threshold value.

TABLE IX
PRECISION AND RECALL ON THE KDD-CUP'99 DATASET

Sliding Window Size	Precision	Recall
N=1000	0.3571	0.1720
N=2000	0.4000	0.2791
N=3000	0.3750	0.2727
N=4000	0.5000	0.3667

is stable, the size of sliding window is able to be set larger in order to save the execution time. In Fig. 2(a), the variations of the number of drifting concepts with respect to class label and attribute sets are shown, respectively. Precision and recall, with respect to attribute sets, are presented in Fig. 2(b). In this experiment, the threshold value between two concepts with respect to attribute sets is set to 0.05, the threshold value between two concepts with respect to class label is set to 0.1, and the size of sliding window is from 1000 to 30 000 with step length of 1000.

From Fig. 2(a), it is clear that the number of drifting concepts decreases with the increasing of sliding window size. In Fig. 2(b), one can find that the precision and recall are insensitive to the size of sliding window.

Furthermore, the number of drifting concepts varying with the size of threshold value is also analyzed. Fig. 3 shows the number of drifting concepts varying with the size of threshold value with respect to class label and attribute sets, respectively. In this experiment, suppose that the size of sliding window is set to 3000 and that the size of threshold is from 0.01 to 1 with step length of 0.01.

From Fig. 3, one can find that the variance ratio of the number of drifting concepts with respect to attribute sets is greater than that of the class label. To make the number of drifting concepts with respect to class label as close to that of attribute sets as possible, the threshold value with respect to class label should be greater than that of attribute sets. In the practical application, a user may choose a proper threshold by the prior knowledge and specific requirement.

In addition, we ran Chen's clustering framework, and experimental results are shown in Table IX. In the experiment, the outlier threshold is set to 0.1, and the cluster-variation threshold is set to 0.1, and the cluster-difference threshold is set to 0.5. The threshold value between two concepts with respect to class label is set to 0.1.

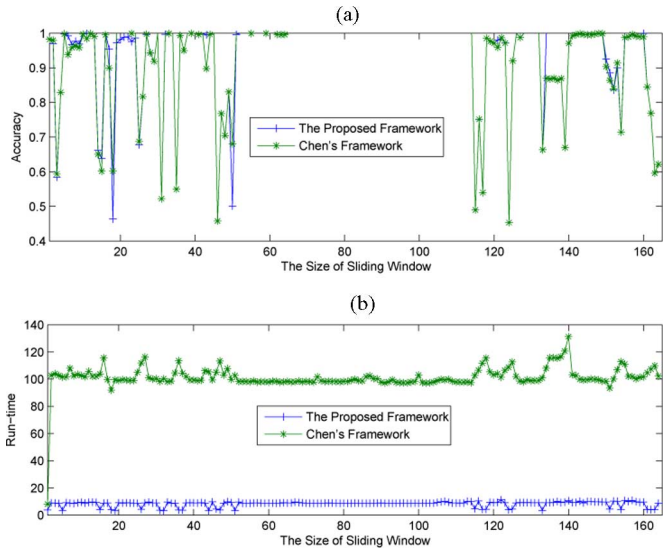


Fig. 4. Comparison of the two different frameworks on KDD-CUP dataset. (a) Comparison of AC with two different framework. (b) Comparison of run-time with two different framework.

From Table IX and Fig. 2(b), one can find that the $DCDA$ is obviously superior to Chen's method.

2) Clustering Results: In order to evaluate the effectiveness of the proposed framework in clustering, clustering accuracy is defined as

$$\text{Accuracy} = \frac{\sum_{i=1}^{k} a_i}{|U|} \tag{12}$$

where k is the number of classes of the dataset, a_i is the number of objects that are correctly assigned to the ith $(1 \leq i \leq k)$ class, and U is the universe.

Fig. 4 shows the comparison of accuracy and run-time between the proposed framework and Chen's framework on each sliding window. In this experiment, the threshold value θ between two concepts with respect to attribute sets is set to 0.1. The outlier threshold is set to 0.1, the cluster-variation threshold is set to 0.1, and the cluster-difference threshold is set to 0.5.

From Fig. 4(a), one can obtain that the accuracy of the proposed framework is greater than that of Chen's framework on 144 sliding windows among the 164. From Fig. 4(b), it is clear that the run-time of the proposed framework is obviously less than that of Chen's. Furthermore, compared with Chen's framework, the proposed one needs fewer parameters.

C. Trend Analysis

1) Method to Determine the Number of Clusters: One of the major problems in cluster analysis is the determination of the number of clusters, which is a basic input for most clustering algorithms. Chen and Liu [34] proposed an entropy-based categorical clustering algorithm, i.e., agglomerative categorical clustering with entropy criterion (ACE), to determine the number of clusters. The experimental results show that the ACE can effectively identify the significant clustering structures. However, the time complexity of the ACE is $O(n^2)$, which prevents

TABLE X
CLUSTERING RESULTS WITH THE DIFFERENT WINDOW SIZE ON THE KDD-CUP'99 DATASET

	$R(c_1^1)$	$R(c_2^1)$	$R(c_1^2)$	$R(c_2^2)$	$R(c_3^2)$	$R(c_1^3)$	$R(c_2^3)$	$R(c_1^4)$	$R(c_2^4)$	$R(c_1^5)$	$R(c_2^5)$	$R(c_3^5)$	$R(c_4^5)$	$R(c_1^6)$	$R(c_2^6)$
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	3	2	2	1	2	2	2	2	2	2	2
3	20	39	20	39	9	20	55	12	20	20	39	20	20	20	55
4	5	7	5	7	5	5	2	5	5	5	7	3	2	5	3
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	1	0	1	0	0	1	0	0	1	1	0	1	0	1	0
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
25	3	3	3	3	3	3	4	3	3	3	3	4	4	3	4
26	3	3	3	3	3	3	4	3	3	3	3	3	4	3	4
27	3	4	3	4	3	3	3	3	3	3	4	3	3	3	3
28	3	4	3	4	3	3	3	3	3	3	4	3	3	3	3
29	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
30	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
31	3	3	3	3	4	3	3	3	4	3	3	4	3	3	3
32	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
33	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
34	4	4	4	4	3	4	4	4	3	4	4	4	4	4	3
35	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
36	3	3	3	3	4	3	3	4	3	3	3	4	3	3	3
37	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
38	3	3	3	3	3	3	4	3	3	3	3	4	4	3	3
39	3	3	3	3	3	3	4	3	3	3	3	3	3	3	4
40	3	4	3	4	3	3	3	3	3	3	4	3	3	3	3
41	3	4	3	4	3	3	3	3	3	3	4	3	3	3	3

it from working directly on large datasets, where n is the number of objects. In general, the number of clusters on a dataset is between 2 and $\sqrt{n}$ [31], [32]. Based on the foregoing discussion, the ACE can be improved further. Thus, the input dataset can be partitioned into $\sqrt{n}$ small subclusters by the k-modes clustering algorithm in the first phase, and then, the subclusters are continuously merged based on incremental entropy [34], as proposed by Chen and Liu, in a hierarchical manner, in the second phase. However, the k-modes algorithm is likely to obtain different clustering results with different initial cluster centers, which makes it important to start with a reasonable initial partition in order to achieve high-quality clustering solutions. Therefore, we utilize an initialization method, which was proposed in [33], to obtain initial cluster centers before executing the k-modes in the first phase. The improved ACE clustering algorithm can effectively detect the number of clusters in categorical data, and the corresponding time complexity is dropped to $O(n^{3/2})$ as well. In order to detect the number of clusters at different timestamps, the improved ACE clustering algorithm is employed on the KDD-CUP'99 set. In this experiment, let us suppose that the size of sliding window is 3000 and that the number of clusters at different sliding windows is shown in Fig. 5.

In Fig. 5, the number of clusters drops to 1 in the range 52–114 and 134–149 because the records are the same in those sliding windows.

2) Visualizing Clustering Results for Trend Analysis: Trend analysis is very important to predict the future development. Fig. 6 shows the relationship between clusters at different timestamps. In this experiment, we choose the first 30 000 objects of KDD-CUP'99 set as the test set. Suppose that the size of sliding window is set to 3000 and that the threshold value γ of the distance between two clusters is 0.1. If the distance between two

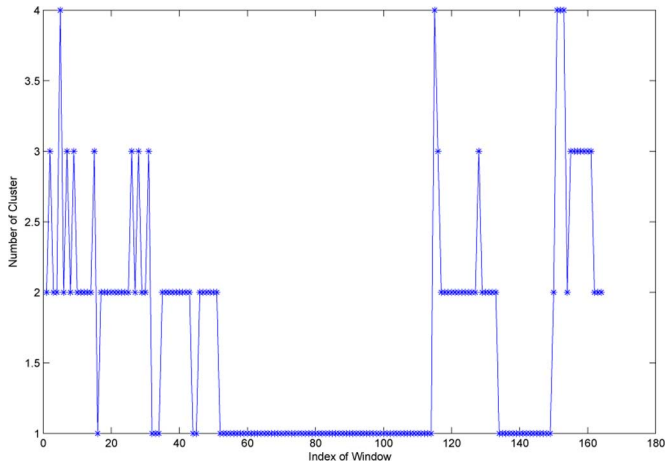


Fig. 5. Numbers of clusters with the time-stamps sliding window.

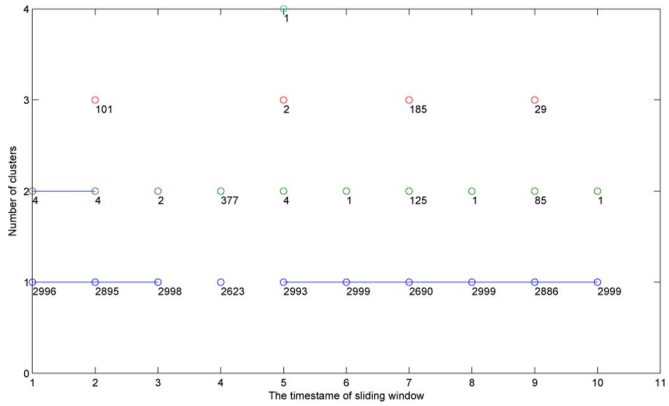


Fig. 6. Visualization of the evolving clusters on the first 30 000 objects of KDD-CUP'99 dataset.

clusters is less than 0.1, two points are lined together. Furthermore, the representative of each cluster is given in Table X (only including the first six sliding windows). Note that several circles in a column indicate the clustering results of the current sliding window. The number near each circle represents the number of objects of each cluster.

From Fig. 6 and Table X, the evolving processes of clusters at different time-stamps can easily be seen. For example, sliding windows 4 and 5 are considered as two drifting concepts, which can be explained by the fact that the first 7793 objects are normal, and then, 3695 attacks come after the normal connections. In addition, 34 attributes are continuous on the KDD-CUP dataset. Each value of numerical attributes in Table X corresponds to one of the values, which are obtained by the uniform quantization method on each continuous attribute. The values of the rest of the attributes in Table X correspond to one of the domain values, which are replaced with numeric values.

VI. Conclusion

In this paper, we have proposed a framework to perform clustering on the categorical time-evolving data. Based on sliding-window techniques and the distance between two concepts, drifting concepts can be obtained at different sliding windows. A data-labeling method has been proposed based on the similarity between an object and a cluster. In order to observe the evolving process of clustering results at different sliding windows, a visualizing method has been presented. The time-complexity analysis of the proposed algorithms indicates that the proposed framework is efficient and scalable at handling a large dataset. The proposed framework has been demonstrated on a real dataset. As evidenced by the empirical results, the proposed framework not only is able to detect the drifting concepts accurately but can also provide high-quality clustering results. In addition, users can easily track some evolving trends from the clusters by the visualizing method, which could be interesting to users. Compared with Chen's framework, the proposed one needs fewer parameters, which is favorable for specific applications.

Acknowledgment

The authors are very grateful to the editors and reviewers for their valuable comments and suggestions.

References

[1] B. Babcock, S. Babu, M. Dater, and R. Motwanti, "Models and Issues in data stream systems," in *Proc. PODS*, 2002, pp. 1–16.
[2] T. W. Liao, "Clustering of time series data—A survey," *Pattern Recognit.*, vol. 38, no. 11, pp. 1857–1874, 2005.
[3] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in *Proc. Very Large Data Bases Conf.*, Sep. 2003, pp. 81–92.
[4] F. Cao, M. Ester, Q. Qian, and A. Zhou, "Density-based clustering over an evolving data streams with noise," in *Proc. SIAM Conf.*, 2006, pp. 328–339.
[5] D. Chakrabarti, R. Kumar, and A. Tomkins, "Evolutionary clustering," in *Proc. ACM SIGKDD. Knowl. Discov. Data Mining*, 2006, pp. 554–560.
[6] Y. Chi, X-D. Song, D-Y. Zhou, K. Hino, and B.L. Tseng, "Evolutionary spectral clustering by incorporating temporal smoothness," in *Proc. ACM SIGKDD*, 2007, pp. 153–162.
[7] B.-R. Dai, J.-W. Huang, M.-Y. Yeh, and M.-S. Chen, "Adaptive clustering for multiple evolving steams," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 9, pp. 1166–1180, Sep. 2006.
[8] M. M. Gaber and P. S. Yu, "Detection and classification of changes in evolving data streams," *Int. J. Inf. Technol. Decis. Making*, vol. 5, no. 4, pp. 659–670, 2006.
[9] O. Nasraoui and C. Rojas, "Robust clustering for tracking noisy evolving data streams," in *Proc. SIAM Conf., Data Mining*, 2006, pp. 618–622.
[10] M. Y. Yeh, B. R. Dai, and M. S. Chen, "Clustering over multiple evolving streams by events and correlations," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 10, pp. 1349–1362, Oct. 2007.
[11] C. C. Aggarwal, J. Han. J. Wang, and P. S. Yu, "A framework for projected clustering of high dimensional data streams," in *Proc. Very Large Data Bases Conf.*, 2004, pp. 852–863.
[12] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan, "Clustering data streams," in *Proc. Symp. Found. Comput. Sci.*, Nov. 2000, pp. 359–366.
[13] L. O'Callaghan, N. Mishra, A. Meyerson, S. Guha, and R. Motwani, "Streaming-data algorithms for high quality clustering," in *Proc. Int. Conf. Data Eng.*, 2002, p. 685.
[14] O. Nasraoui, M. Soliman, E. Saka, A. Badia, and R. Germain, "A web usage mining framework for mining evolving user profiles in dynamic web sites," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 2, pp. 202–215, Feb. 2008.
[15] H.-L. Chen, M.-S. Chen, and S.-C. Lin, "Catching the trend: A framework for clustering concept-drifting categorical data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 5, pp. 652–665, Mar. 2009.
[16] Z. Pawlak, "Rough sets," *Int. J. Comput. Inf. Sci.*, vol. 38, no. 11, pp. 341–356, 1982.

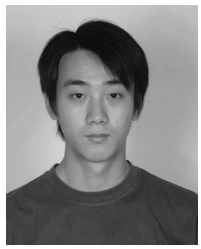
[17] J. Y. Liang, J. H. Wang, and Y. H. Qian, "A new measure of uncertainty based on knowledge granulation for rough sets," *Inf. Sci.*, vol. 179, no. 4, pp. 458–470, 2009.
[18] Y. H. Qian, J. Y. Liang, W. Pedrycz, and C. Y. Dang, "Positive approximation: An accelerator for attribute reduction in rough set theory," *Artif. Intell.*, vol. 174, no. 9–10, pp. 597–618, 2010.
[19] D. Parmar, T. Wu, and J. Blackhurst, "MMR: An algorithm for clustering data using rough set theory," *Data Knowl. Eng.*, vol. 63, no. 3, pp. 897–893, 2007.
[20] F. Jiang, Y. F. Sui, and C. G. Cao, "A rough set approach to outlier detection," *Int. J. General Syst.*, vol. 37, no. 5, pp. 519–536, 2008.
[21] F. Jiang, Y. F. Sui, and C. G. Cao, "Some issues about outlier detection in rough set theory," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 4680–4687, 2009.
[22] C. B. Chen and L. Y. Wang, "Rough set-based clustering with refinement using Shannon's entropy theory," *Comput. Math. Appl.*, vol. 52, pp. 1563–1576, 2006.
[23] F. Y. Cao, J. Y. Liang, and G. Jiang, "An initialization method for the k-Means algorithm using neighborhood model," *Comput. Math. Appl.*, vol. 58, no. 3, pp. 474–483, 2009.
[24] J. Y. Liang, K. S. Chin, C. Y. Dang, and C. M. Y. Richard, "A new method for measuring uncertainty and fuzziness in rough set theory," *Int. J. General Syst.*, vol. 31, no. 4, pp. 331–342, 2002.
[25] R. Jensen and Q. Shen, "New approaches to fuzzy-rough feature selection," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 4, pp. 824–838, Aug. 2009.
[26] Y. J. Yang and C. Hinde, "A new extension of fuzzy sets using rough sets: R-fuzzy sets," *Inf. Sci.*, vol. 180, no. 3, pp. 354–365, 2010.
[27] E. C. C. Tsang, D. G. Chen, D. S. Yeung, X. Z. Wang, and J. W. T. Lee, "Attributes reduction using fuzzy rough sets," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 5, pp. 1130–1141, Oct. 2008.
[28] X. D Liu, W. Pedrycz, T. Y. Chai, and M. L. Song, "The development of fuzzy rough sets with the use of structures and algebras of axiomatic fuzzy sets," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 3, pp. 443–462, Mar. 2009.
[29] Z. X. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining Knowl. Discov.*, vol. 2, no. 3, pp. 283–304, 1998.
[30] UCI Machine Learning Repository. (2010). [Online]. Available: http://www.ics.uci.edu/mlearn/MLRepository.html
[31] J. C. Bezdek and N. R. Pal, "Some new indexes of cluster validity," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 3, pp. 301–315, Jun. 1998.
[32] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 370–379, Aug. 1995.
[33] F. Y. Cao, J. Y. Liang, and L. Bai, "A new initialization method for categorical data clustering," *Expert Syst. Appl.*, vol. 36, no. 7, pp. 10223–10228, 2009.
[34] K. Chen and L. Liu, "The "best K" for entropy-based categorical clustering," in *Proc. Int. Conf. Sci. Statist. Database Manage*, 2005, pp. 253–262.

Fuyuan Cao received the B.E. and M.S. degrees in computer science in 1998 and 2004, respectively, from Shanxi University, Taiyuan, China, where he is currently working toward the Ph.D degree with the School of Computer and Information Technology.

His research interests include data mining and machine learning.

Jiye Liang received the M.S. and Ph.D degrees from Xi'an Jiaotong University, Xi'an, China, in 1990 and 2001, respectively.

He is currently a Professor with the School of Computer and Information Technology and the Key Laboratory of Computational Intelligence and Chinese Information Processing of the Ministry of Education, Shanxi University, Taiyuan, China. He has authored or coauthored more than 70 journal papers in his research fields. His current research interests include computational intelligence, granular computing, data mining, and knowledge discovery.



Liang Bai is currently working toward the Ph.D. degree with the School of Computer and Information Technology, Shanxi University, Taiyuan, China.

His research interests are in the areas of machine learning.

Xingwang Zhao is currently working toward the M.S. degree with the School of Computer and Information Technology, Shanxi University, Taiyuan, China.

His research interests are in the areas of machine learning.

Chuangyin Dang (SM'03) received the Ph.D degree in operations research/economics from the University of Tilburg, Tilburg, the Netherlands, in 1991 and the M.S. degree in applied mathematics from Xidian University, Xi'an, China, in 1986.

He is currently an Associate Professor with the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong. His research interests include computational intelligence and optimization theory and technology.