An Optimization Model for Clustering Categorical Data Streams with Drifting Concepts

Liang Bai, Xueqi Cheng, *Member, IEEE*, Jiye Liang, and Huawei Shen

Abstract—There is always a lack of a cluster validity function and optimization strategy to find out clusters and catch the evolution trend of cluster structures on a categorical data stream. Therefore, this paper presents an optimization model for clustering categorical data streams. In the model, a cluster validity function is proposed as the objective function to evaluate the effectiveness of the clustering model while each new input data subset is flowing. It simultaneously considers the certainty of the clustering model and the continuity with the last clustering model in the clustering process. An iterative optimization algorithm is proposed to solve an optimal solution of the objective function with some constraints. Furthermore, we strictly derive a detection index for drifting concepts from the optimization model. We propose a detection method that integrates the detection index and the optimization model to catch the evolution trend of cluster structures on a categorical data stream. The new method can effectively avoid ignoring the effect of the clustering validity on the detection result. Finally, using the experimental studies on several real data sets, we illustrate the effectiveness of the proposed algorithm in clustering categorical data streams, compared with existing data-streams clustering algorithms.

Index Terms—Cluster analysis, optimization model, iterative algorithm, categorical data stream, drifting-concept detection

1 Introduction

Cluster analysis is a branch in statistical multivariate analysis and unsupervised machine learning. The goal of clustering is to group a set of objects into clusters so that the objects in the same cluster have high similarity but are very dissimilar with objects in other clusters [1]. To tackle this problem, various types of clustering algorithms have been proposed in the literature (e.g., [2] and references therein).

Recently, increasing attention has been paid to analyzing cluster structures in data streams, since this task is of great practical relevance in many real applications, such as network-traffic monitoring, stock market analysis, credit card fraud detection analysis, and web click stream analysis. Unfortunately, conventional clustering techniques meet several challenges while clustering data streams. First, data objects are observed sequentially on a data stream. The data generating model often changes as the data are streaming. 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. This phenomenon is called the "concept drifting" [3], [4]. However, conventional clustering techniques assume that the cluster structures do not change with time. They focus on clustering the entire data set and do not take the drifting concepts into consideration. Such process not only decreases the quality of clusters but also disregards the expectations of users that usually require recent clustering results. Furthermore, with advances in data storage technology and the wide deployment of sensor systems and Internet based continuous-query applications, the volume of the data stream is huge. Storing and taking the entire data set is very expensive. Therefore, the techniques for effectively and efficiently clustering data streams are required.

The problem of clustering data streams in the numerical domain has been well-explored in the literature [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. However, the data streams contain not only numerical data but also categorical data. For example, buying records of customers, web logs that record the browsing history of users, or web documents often evolve with time. The lack of intuitive geometric properties for categorical data imposes several difficulties on clustering this kind of data [3], [15]. For example, since the domains of categorical attributes are unordered, the distance functions for numerical data fail to capture resemblance between categorical data objects. Furthermore, for numerical data, the representative of a cluster is often defined as the mean of objects in the cluster. However, it is infeasible to compute the mean for categorical values. These imply that the techniques used in clustering numerical data are not suitable for categorical data. Therefore, it is widely recognized that designing clustering techniques to directly tackle data streams in the categorical domain is very important for many applications.

Currently, several clustering frameworks for categorical data streams have been reported [3], [4], [16], [17], [18]. In [17], the Coolcat algorithm [17] used the information entropy to describe cluster structures and determine the assignment of new input data objects. However, the algorithm did not

- *L. Bai is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and with the School of Computer and Information Technology, Shanxi University, Taiyuan, Shanxi 030006, China. E-mail: sxbailiang@hotmail.com.*
- *X. Cheng and H. Shen are with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China. E-mail: {cxq, shenhuawei}@ict.ac.cn.*
- *J. Liang is with the School of Computer and Information Technology, Shanxi University, Taiyuan, Shanxi 030006, China. E-mail: ljy@sxu.edu.cn.*

Manuscript received 30 Sept. 2015; revised 2 Mar. 2016; accepted 19 July 2016. Date of publication 26 July 2016; date of current version 3 Oct. 2016. Recommended for acceptance by S. Yan.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2016.2594068

consider the "concept drifting". In [18], Chen et al. used the Hierarchical Entropy Tree structure (HE-Tree) to capture the entropy characteristics of clusters and applied the change of the best number of clusters to detect the change of the cluster structures in the data stream. In [3], Chen et al. defined the N-Nodeset Importance Representative (abbreviated as NNIR) to reflect characteristics of clusters, and used the number of outliers and the numbers of objects in each clusters to detect the drifting concepts in the data stream. In [4], Cao et al. proposed a distance between concepts, based on the rough membership function, to detect the drifting concepts in the data stream.

However, there are two main issues in these above algorithms. The one is that these algorithms only use the similarity between objects and clusters to once determine the cluster labels of objects, for new input data subsets. Due to the lack of the validity criterion and optimization strategy, they do not adjust or optimize the clustering result. The other is that there is a lack of relevance between the clustering objectives and drifting-concept detection indices in these algorithms. That maybe lead to ignoring the impact of the effectiveness of the clustering results on the drifting-concept detection. For a new input data subset, if its clustering result is poor, the drifting-concept detection result maybe un-true. Thus, users need a detection method based on an optimization model to enhance the reliability of the detection results. To get rid of these deficiencies, we will build an optimization model to solve the clustering problem for categorical data streams. The major contributions are as follows:

- We construct an optimization model for clustering categorical data streams. In the model, a new validity function is defined as the optimization objective function. On each new input data subset, minimizing it with some constraints aims to finding out the new clustering model which has good certainty for cluster representatives and continuity with the last clustering model. An iterative optimization algorithm is proposed to solve the optimization problem.
- We strictly derive a detection index for drifting concepts from the optimization model. A detection method is proposed which integrates the detection index and the optimization model to catch the evolution trend of cluster structures on a categorical data stream. It can effectively avoid ignoring the effect of the clustering validity on the detection result of drifting concepts.
- The performance of the proposed clustering algorithm is investigated by using real data sets.

This paper is organized as follows: In Section 2, we review the notation of categorical data and the clustering model for static categorical data. Section 3 presents an optimization clustering model for categorical data streams. Section 4 illustrates the performance of the proposed model. Finally, a concluding remark is given in Section 5.

2 Preliminaries

2.1 Categorical Data

Huang et al. [19] provided the notation of categorical data which was introduced as follows: Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be a set of n objects and $A = \{a_1, a_2, \ldots, a_m\}$ be a set of m attributes which are used to describe X. Each attribute a_j describes a domain of values, denoted by D_{a_j}, associated with a defined semantic and a data type. Here, only consider two general data types, numerical and categorical, and assume other types used in database systems can be mapped to one of the two types. The domains of attributes associated with the two types are called numerical and categorical, respectively. A numerical domain consists of real numbers. A domain D_{a_j} is defined as categorical if it is finite and unordered, i.e., $D_{a_j} = \{a_j^{(1)}, a_j^{(2)}, \ldots, a_j^{(n_j)}\}$ where n_j is the number of categories of attribute a_j for $1 \le j \le m$. For any $1 \le p \le q \le n_j$, either $a_j^{(p)} = a_j^{(q)}$ or $a_j^{(p)} \ne a_j^{(q)}$. For $1 \le i \le n$, an object $\mathbf{x}_i \in X$ is represented as a vector $[x_{i1}, x_{i2}, \ldots, x_{im}]$, where $x_{ij} \in D_{a_j}$, for $1 \le j \le m$. If each attribute in A is categorical, X is called a categorical data set.

If X is a data stream, each object has an arriving time. Let $S = \{S^1, S^2, \ldots, S^T\}$ be a partition of X, according to a sliding window, where $\bigcup_{p=1}^{T} S^p = X$, $S^p \cap S^q = \emptyset$, for $1 \le p \ne q \le T$. A sliding window includes an input data subset S^p of X in a time interval. There is no overlap between time intervals. When clustering the data stream, users are more interested in the cluster structure in a time range than the entire data set. The main symbols used in this paper are summarized in Table 1.

2.2 Clustering Optimization Model for Static Categorical Data

For static categorical data sets, there are three well-known clustering objective functions: the k-modes objective function [19], the category utility function [20] and the information entropy function [17]. Many algorithms have been developed to use these validity functions as objective functions and find their (local) optimal solutions. The representative algorithms include the k-modes-type and their variant algorithms [19], [21], [22], [23], the Cobweb algorithm based on the category utility function [24] and the information entropy-based algorithms [17], [25]. In the papers [26], [27], we proposed a generalized objective function and optimization problem for static categorical data to analyze the generality and difference of the three types of optimization models. They are special cases of the generalized optimization model.

The generalized optimization model is written as follows:

$$\min_{W,V} F_g(W,V) = \sum_{l=1}^{k}\sum_{i=1}^{n} w_{li} d_g(\mathbf{x}_i, c_l) + \alpha \sum_{l=1}^{k}\sum_{i=1}^{n} w_{li} \sum_{j=1}^{m}\sum_{q=1}^{n_j} \left(v_{lj_q}\right)^2, \tag{1}$$

subject to

$$\begin{cases} w_{li} \in \{0,1\}, \sum_{l=1}^{k} w_{li} = 1, 1 < \sum_{i=1}^{n} w_{li} < n, \\ v_{lj_q} \in [0,1], \sum_{q=1}^{n_j} v_{lj_q} = 1, \end{cases} \tag{2}$$

where

- $W = [w_{li}]$ is a k-by-n $\{0,1\}$ matrix, w_{li} indicates whether $\mathbf{x}_i$ belongs to the lth cluster, $w_{li} = 1$ if $\mathbf{x}_i$ belongs to the lth cluster and 0 otherwise.

TABLE 1
Description of the Main Symbols Used in This Paper

Symbol	Description
X	A data set (stream)
n	The number of objects in X
m	The number of attributes in X
a_j	The jth attribute in X
D_{a_j}	The domain of a_j
n_j	The number of values in a_j
$a_j^{(q)}$	The qth value of a_j
S^p	The data subset in the pth sliding window
$\|S^p\|$	The number of objects in S^p
T	The number of the sliding windows
$\mathbf{x}_i$ ($\mathbf{x}_i^p$)	The ith data object (on S^p)
c_l (c_l^p)	The lth cluster (on S^p)
w_{li} (w_{li}^p)	The membership degree of $\mathbf{x}_i$ to c_l (after S^p is input)
W (W^p)	The membership matrix (on S^p)
v_{lj_q} ($v_{lj_q}^p$)	The representability of $a_j^{(q)}$ to c_l (after S^p is input)
$\mathbf{v}_l$ ($\mathbf{v}_l^p$)	The representation of c_l (after S^p is input)
V (V^p)	The clustering model (after S^p is input)
$\|c_l\|$ ($\|c_l^p\|$)	The number of objects in c_l (on S^p)
$\|c_{lj_q}\|$ ($\|c_{lj_q}^p\|$)	The number of objects with $a_j^{(q)}$ in c_l (on S^p)
k (k^p)	The number of clusters (on S^p)
t (t^p)	The number of clustering iterations (on S^p)
$d_g(.,.)$	The dissimilarity measure between an object and a cluster
$F_g(.,.)$	The objective function for clustering a static data set or re-clustering the data subset in a window
$M(.,.)$	The objective function for clustering the data subset in a window
$\alpha, \beta, \varepsilon$	The parameters in the optimization model
$\Omega(.,.)$	The detection index for drifting concepts

- $V = [v_{lj_q}]$ is a k-by-$\sum_{j=1}^m n_j$ matrix. v_{lj_q} is the representability of the qth categorical value of the jth attribute in the lth cluster, for $1 \le l \le k, 1 \le j \le m$, $1 \le q \le n_j$. The larger v_{lj_q} is, the more representability the categorical value $a_j^{(q)}$ has in the lth cluster. For each cluster ($1 \le l \le k$), we use $\mathbf{v}_l$ (which is the lth row of V) to summarize and characterize the lth cluster. For a categorical data set, V can be seen as the clustering model. It is used to predict the likelihood of an unseen object being a cluster member. If V has good predictive ability, it is thought to be good.
- $\sum_{l=1}^k \sum_{i=1}^n w_{li} d_g(\mathbf{x}_i, c_l)$ is the sum of the within-cluster dispersions that we want to minimize. $d_g(\mathbf{x}_i, c_l)$ is a dissimilarity measure between the object $\mathbf{x}_i$ and the lth cluster c_l defined as follows:

$$d_g(\mathbf{x}_i, c_l) = \sum_{j=1}^m \phi_{a_j}(\mathbf{x}_i, c_l),$$

with $\phi_{a_j}(\mathbf{x}_i, c_l) = 1 - v_{lj_q}$, *if* $x_{ij} = a_j^{(q)}, 1 \le q \le n_j$. Here, $\phi_{a_j}(\mathbf{x}_i, c_l)$ depends on v_{lj_q}, which is the representability of $a_j^{(q)}$ in the lth cluster. The larger v_{lj_q} is, the more representability $a_j^{(q)}$ has in the lth cluster, the smaller the dissimilarity between $\mathbf{x}_i$ and c_l in the attribute a_j is. When the representability of $a_j^{(q)}$ is 1, $\phi_{a_j}(\mathbf{x}_i, c_l) = 0$.

- $\sum_{l=1}^k \sum_{i=1}^n w_{li} \sum_{j=1}^m \sum_{q=1}^{n_j} (v_{lj_q})^2$ is used to stimulate more categorical values to contribute to the identification of clusters. When v_{lj_q} are the same for $1 \le q \le n_j$, the term achieves its minimum value given by $\min \sum_{q=1}^{n_j} (v_{lj_q})^2 = \frac{1}{n_j}$. If only one of the v_{lj_q} for $1 \le q \le n_j$ is nonzero, the term achieves the maximum value, i.e., $\max \sum_{q=1}^{n_j} (v_{lj_q})^2 = 1$. The smaller the term is, the more categories the weights are assigned to. While giving W, we wish to minimize it to make more categorical values identify clusters.
- $\alpha (\ge 0)$ is a parameter which is used to balance which part plays a more important role in the minimization process of (1). The larger α is, the more the last term contributes in the optimization process and the smoother or fuzzier of the resulting V is. However, the values of α should not be too large. The reason is that when α is very large so that each v_{lj_q} is close to $1/n_j$.

We minimize F_g by iteratively updating W and V. When V is given, W is computed by

$$\hat{w}_{li} = \begin{cases} 1, \ if \quad d_g(\mathbf{x}_i, \mathbf{v}_l) + \alpha \sum_{j=1}^m \sum_{q=1}^{n_j} (v_{lj_q})^2 \le \\ \quad d_g(\mathbf{x}_i, \mathbf{v}_h) + \alpha \sum_{j=1}^m \sum_{q=1}^{n_j} (v_{hj_q})^2, 1 \le h \le k, \\ 0, \ otherwise, \end{cases} \tag{3}$$

for $1 \le l \le k, 1 \le i \le n$.

When W is given, V is computed by

$$\hat{v}_{lj_q} = \frac{1}{2\alpha} \frac{|c_{lj_q}|}{|c_l|} + \frac{2\alpha - 1}{2 n_j \alpha}, \tag{4}$$

where $|c_l| = \sum_{i=1}^n w_{li}$ and $|c_{lj_q}| = \sum_{i=1, \mathbf{x}_{ij} = a_j^{(q)}}^n w_{li}$ for $1 \le l \le k$, $1 \le j \le m$, $1 \le q \le n_j$. According to the computing formula of V, we see that the v_{lj_q} value is proportional to the relative frequency of $a_j^{(q)}$ in the lth cluster, for $1 \le l \le k, 1 \le j \le m$ and $1 \le q \le n_j$.

Algorithm 1.

Input: X, initial V, α, k
Output: W, V
$F_g = 0$;
while $F_g \ne F_g'$ **do**
  $F_g' = F_g$;
  Given V, compute W by (3);
  Given W, compute V by (4);
  Compute the function F_g value;

The clustering algorithm is formalized in Algorithm 1. The algorithm can obtain a local optimal solution in the finite iterations, which had been proved in [26]. Before implementing it, we need to input three parameters: α, initial V and the number of clusters k on S^p. For the parameter α, we have analyzed its effect on evaluating the clustering results, seen in [27]. The experimental analysis showed that the objective function F_g is robust in evaluating the clustering results when the α value is a certain value, e.g., $\alpha \le 200$ on the tested data sets. However,

the α value should not be too large, otherwise the representability of each categorical value to clusters will be not recognized. For setting initial V and the number of clusters k, there are several methods for clustering categorical data proposed in [17], [28], [29], [30], [31]. In the papers [28], [30], we defined the density of a categorical value and integrated the simple matching distance to evaluate the representability of an object to a cluster. We selected the first k objects with higher density and separation as the representatives of k clusters. Based on these representatives, we applied the simple matching distance to assign each object into its nearest cluster and obtain an initial partition. While giving the initial partition, we can obtain an initial V, according to Eq. (4). In the paper [31], we provided a method of simultaneously obtaining the initial partition and the number of clusters. We continue using the representability of objects to evaluate the number of clusters. We thought that if the real number of clusters is k, the $k+1$th selected object will be representative of the same cluster as one of the first k selected objects. In the case, the representability of the first k selected objects should be much larger than that of the $k+1$th selected object. Therefore, we determined the number of clusters by analyzing the change curve of the representability.

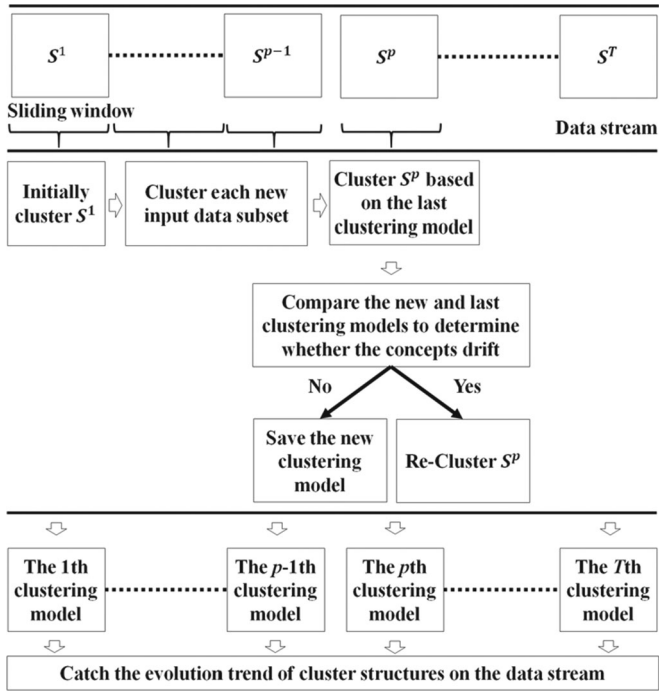


Fig. 1. The flowchart of the clustering framework for a categorical data stream.

3 Clustering Optimization Model for Categorical Data Streams

3.1 The Clustering Framework

In order to cluster categorical data streams and detect the drifting concepts, we apply a clustering framework based on the sliding window technique. The sliding window technique conveniently eliminates the outdated records and only saves the clustering models, which is utilized in several previous works on clustering time-evolving data [3], [4]. Therefore, based on the technique, we can cluster the latest data objects in the current window and catch the evolution trend of cluster structures on the data stream. The entire framework of performing clustering on a categorical data stream is shown in Fig. 1.

According to Fig. 1, we see that the clustering of a categorical data stream needs to address the following three problems:

- How to initially cluster the first input data subset and re-cluster a data subset?
- How to cluster a new input data subset based on the last clustering model?
- How to detect the drifting concepts on a data stream?

The first problem can be solved by Algorithm 1. In the following, we will investigate the second and third problems. We will extend the static clustering model in Section 2.2 to construct an optimization model for categorical data streams. The new optimization model will provide a clustering criterion and strategy to obtain a new clustering model with good certainty for cluster representatives and continuity with the last clustering model while a new input data subset is inputting. Furthermore, we will make use of the optimization model to enhance the credibility of the drifting-concept detection. For a new input data subset, we can produce a number of partitions. The different partition results maybe lead to the different detection results of drifting concepts. The poorer the partition result is, the more unreliable the detection result is. Therefore, we need the optimization model to find out the optimal partition result for the new input data subset. If the concepts from the optimal result still drift, it is thought that the change of the cluster structures is obvious.

3.2 The Optimization Clustering Algorithm

To cluster a new input data subset based on the last clustering model, we need to consider more terms in the objective function and optimization problem, compared to that of static categorical data. While building a new objective function for categorical data streams, we consider not only the physical meaning of the clustering validity on the new input data subset but also that of the drifting-concept detection. The meaning of clustering validity will be introduced in the following. The detection meaning of drifting concepts will be analyzed in Section 3.3.

Let S^p be the pth data subset input from a data stream, $\mathbf{x}_i^p \in S^p$ be a data object for $1 \leq i \leq |S^p|$, V^{p-1} be the last cluster model which has been known, W^p be the membership matrix of S^p, V^p be the cluster model after inputting S^p, k^p be the number of clusters on S^p. While clustering a new input data subset, we first assume that the concepts do not drift. Thus, we set the number of clusters is equal to that of the last window.

The objective function is defined as follows:

$$M(W^p, V^p) = F_g(W^p, V^{p-1}) + F_g(W^p, V^p) + D(V^p, V^{p-1}), \tag{5}$$

where

$$\begin{cases} F_g(W^p, V^{p-1}) = \sum_{l=1}^{k^p}\sum_{i=1}^{|S^p|} w_{li}^p d_g(\mathbf{x}_i^p, c_l^{p-1}) \\ \qquad\qquad + \alpha \sum_{l=1}^{k^p}\sum_{i=1}^{|S^p|} w_{li}^p \sum_{j=1}^{m}\sum_{q=1}^{n_j}\left(v_{ljq}^{p-1}\right)^2, \\ F_g(W^p, V^p) = \sum_{l=1}^{k^p}\sum_{i=1}^{|S^p|} w_{li}^p d_g(\mathbf{x}_i^p, c_l^p) \\ \qquad\qquad + \alpha \sum_{l=1}^{k^p}\sum_{i=1}^{|S^p|} w_{li}^p \sum_{j=1}^{m}\sum_{q=1}^{n_j}\left(v_{ljq}^{p}\right)^2, \\ D(V^p, V^{p-1}) = \beta \sum_{l=1}^{k^p}\sum_{i=1}^{|S^p|} w_{li}^p \sum_{j=1}^{m}\sum_{q=1}^{n_j}\left(v_{ljq}^{p} - v_{ljq}^{p-1}\right)^2. \end{cases} \tag{6}$$

The objective function is composed of the three terms. In the following, we illustrate the roles of these terms in clustering data streams:

- The first term $F_g(W^p, V^{p-1})$ is to measure the effectiveness of the clustering result while the last clustering model is used to represent the new input data subset. Many existing algorithms for clustering data streams only assume the last clustering model can effectively represent the new input data subset. They measure the similarity between objects and clusters to once determine the cluster labels of objects. However, the clustering model evolves as the new data objects are flowing. The last clustering model can not completely reflect the characteristics of the new data subset. This indicates that only using the term is not enough to cluster data streams. Thus, we need to consider other factors.
- The second term $F_g(W^p, V^p)$ is to measure the effectiveness of the clustering result while the new clustering model is used to represent the new input data subset. The new clustering model evolves from the last clustering model. It is used to reflect the characteristics of the historical and new data subsets. If an algorithm only uses the term as the objective function to cluster the new data subset, it becomes a clustering algorithm for static data. The obtained clustering model only represents the new data subset and ignores the continuity with the last clustering model. Thus, we need to combine the first and second terms.
- The third term $D(V^p, V^{p-1})$ is to measure the difference between the new and last clustering models. In the process of clustering a new input data subset, we first assume that the last clustering model can partly represent the new data subset. Thus, we wish the smaller the $D(V^p, V^{p-1})$ value, the better. If the difference is very large, the change of the cluster structures may be obvious. This indicates that the concepts suddenly drift and the last clustering model is unavailable in clustering the new data subset. In the case, we need to re-cluster the new data subset, independently of the last clustering model. Therefore, we integrate the three terms and wish obtaining a new clustering model which has simultaneously good certainty for cluster representatives and continuity with the last clustering model. $\beta(\geq 0)$ is a parameter which is used to control the role of the third term in the minimization process of (5).

In the new algorithm, we transform the problem of clustering new input data subsets into the following optimization problem:

$$\min_{W^p, V^p} M(W^p, V^p), \tag{7}$$

subject to

$$\begin{cases} w_{li}^p \in \{0,1\}, \sum_{l=1}^{k^p} w_{li}^p = 1, 1 < \sum_{i=1}^{n} w_{li}^p < |S^p|, \\ v_{ljq}^p \in [0,1], \sum_{q=1}^{n_j} v_{ljq}^p = 1. \end{cases} \tag{8}$$

While a new data subset is inputting, we hope to obtain a new clustering model which is simultaneously effective for cluster representatives and close to the last clustering model, by minimizing the objective function with the constraints. The obtained clustering model will be used to not only reflect the characteristics of clusters but also catch the evolution trend of cluster structures (which will be discussed in Section 3.3). In the following, we will introduce how to solve the optimization problem.

We use an iterative method to solve the optimization problem. That is, the problem is solved by iteratively solving the following two minimization subproblems:

Problem P_1. Fix $V^p = \hat{V}^p$, solve $\min_{W^p} M(W^p, \hat{V}^p)$;

Problem P_2. Fix $W^p = \hat{W}^p$, solve $\min_{V^p} M(\hat{W}^p, V^p)$.

To solve the above two subproblems, we calculate the updating formulas of W^p and V^p according to the following two theorems:

Theorem 1. *Let $\hat{V}^p$ be fixed and consider the problem:*

$$\min_{W^p} M(W^p, \hat{V}^p) \text{ subject to } (8).$$

The minimizer $\hat{W}^p$ is given by

$$\hat{w}_{li}^p = \begin{cases} 1, & if \quad \theta_{li} \leq \theta_{hi}, 1 \leq h \leq k^p, \\ 0, & otherwise, \end{cases} \tag{9}$$

where

$$\theta_{li} = d_g(\mathbf{x}_i^p, c_l^{p-1}) + \alpha \sum_{j=1}^{m}\sum_{q=1}^{n_j}\left(v_{ljq}^{p-1}\right)^2 + d_g(\mathbf{x}_i^p, c_l^p) + \alpha \sum_{j=1}^{m}\sum_{q=1}^{n_j}\left(v_{ljq}^{p}\right)^2 + \beta \sum_{j=1}^{m}\sum_{q=1}^{n_j}\left(v_{ljq}^{p} - v_{ljq}^{p-1}\right)^2,$$

for $1 \leq l \leq k^p, 1 \leq i \leq |S^p|$.

Proof. For a given $\hat{V}^p$, all the inner sums of the quantity

$$M(W^p, \hat{V}^p) = \sum_{i=1}^{n}\sum_{l=1}^{k^p} w_{li}^p \theta_{li},$$

are independent. Minimizing the quantity is equivalent to minimizing each inner sum. We write the ith inner

sum ($1 \le i \le |S^p|$) as

$$\varphi_i = \sum_{l=1}^{k^p} w_{li}^p \theta_{li}.$$

When $w_{li}^p = 1$, we have $w_{hi}^p = 0, 1 \le h \le k^p, l \ne h$ and $\varphi_i = \theta_{li}$. It is clear that φ_i is minimized iff θ_{li} is minimal for $1 \le l \le k^p$. The result follows. □

Theorem 2. *Let $\hat{W}^p$ be fixed and consider the problem:*

$$\min_{V^p} M(\hat{W}^p, V^p) \text{ subject to (8)}.$$

The minimizer $\hat{V}^p$ is given by

$$\hat{v}_{ljq}^p = \frac{1}{2(\alpha+\beta)} \frac{|c_{ljq}^p|}{|c_l^p|} + \frac{\beta}{\alpha+\beta} v_{ljq}^{p-1} + \frac{2\alpha-1}{2n_j(\alpha+\beta)}, \quad (10)$$

for $1 \le l \le k^p, 1 \le j \le m, 1 \le q \le n_j$.

Proof. Let

$$\vartheta_{lj} = \sum_{q=1}^{n_j} \Big[|c_{ljq}^p| \Big(1 - v_{ljq}^{p-1}\Big) + \alpha |c_l^p| \Big(v_{ljq}^{p-1}\Big)^2 + |c_{ljq}^p| (1 - v_{ljq}^p\Big) + \alpha |c_l^p| \Big(v_{ljq}^p\Big)^2 + \beta \Big(v_{ljq}^p - v_{ljq}^{p-1}\Big)^2 \Big],$$

for $1 \le l \le k^p$ and $1 \le j \le m$. Then

$$M(\hat{W}^p, V^p) = \sum_{l=1}^{k^p} \sum_{j=1}^{m} \vartheta_{lj},$$

where $|c_l^p| = \sum_{i=1}^{|S^p|} w_{li}^p$ and $|c_{ljq}^p| = \sum_{i=1, x_{ij}=a_j^{(q)}}^{|S^p|} w_{li}^p (1 \le q \le n_j)$ are constants for fixed W^p. Thus, minimizing the objective function is equivalent to minimizing ϑ_{lj}.

Since ϑ_{lj} is a strictly convex function, the well-known K-K-T necessary optimization condition is also sufficient. Therefore, $\hat{\mathbf{v}}_{lj}^p$ is an optimal solution if and only if there exists $\hat{\lambda}$ together with $\hat{\mathbf{v}}_{lj}^p$ satisfying the following system of equations:

$$\begin{aligned} &\nabla_{\mathbf{v}_{lj}^p} \tilde{\vartheta}_{l,j}(\mathbf{v}_{lj}^p, \lambda) = 0, \\ &\sum_{q=1}^{n_j} v_{ljq}^p = 1, \end{aligned} \quad (11)$$

where $\mathbf{v}_{lj} = \{v_{lj_1}, v_{lj_2}, \ldots, v_{lj_{n_j}}\}$ and

$$\begin{aligned} \tilde{\vartheta}_{lj}(\mathbf{v}_{lj}^p, \lambda) = \sum_{q=1}^{n_j} \Big[&|c_{ljq}^p| \Big(1 - v_{ljq}^{p-1}\Big) + \alpha |c_l^p| \Big(v_{ljq}^{p-1}\Big)^2 \\ &+ |c_{ljq}^p| \Big(1 - v_{ljq}^p\Big) + \alpha |c_l^p| \Big(v_{ljq}^p\Big)^2 \\ &+ \beta \Big(v_{ljq}^p - v_{ljq}^{p-1}\Big)^2 \Big] + \lambda \Big(\sum_{q=1}^{n_j} v_{ljq}^p - 1 \Big). \end{aligned} \quad (12)$$

We have

$$\frac{\partial \tilde{\theta}_{l,j}(\mathbf{v}_{lj}^p, \lambda)}{\partial v_{ljq}^p} = 2\alpha |c_l| v_{ljq}^p - |c_{lj_q}| + 2\beta |c_l| v_{ljq}^{p-1} + \lambda, \quad (13)$$

for $1 \le q \le n_j$.

From the above equations, we obtain the optimal solution

$$\hat{v}_{ljq}^p = \frac{1}{2(\alpha+\beta)} \frac{|c_{ljq}^p|}{|c_l^p|} + \frac{\beta}{\alpha+\beta} v_{ljq}^{p-1} + \frac{2\alpha-1}{2n_j(\alpha+\beta)}.$$

This completes the proof. □

According to the computing formula of V^p, we see that the v_{ljq}^p value is proportional to the v_{ljq}^{p-1} value and the relative frequency of $a_j^{(q)}$ in the lth cluster, for $1 \le l \le k^p$, $1 \le j \le m$ and $1 \le q \le n_j$. Besides, we also know that the representability of $a_j^{(q)}$ in a cluster is proportional to its relative frequency in the cluster, according to the static clustering optimization model. Therefore, we see that the new clustering model V^p can effectively describe the cluster characteristics of the historical and new data subsets.

For clustering the new input data subset, the proposed algorithm iteratively updates W^p and V^p by the two theorems, until the objective function M value does not change. The algorithm is formalized in Algorithm 2. The iterative number of the proposed algorithm is finite, which can be proved as in Theorem 3 below.

Algorithm 2.

Input: S^p, V^{p-1}, α, β
Output: W^p, V^p
Initially set $V^p = V^{p-1}$, $M = 0$;
while $M \ne M'$ **do**
  $M' = M$;
  Given V^p, compute W^p by Theorem 1;
  Given W^p, compute V^p by Theorem 2;
  Compute the function M value;

Theorem 3. *The proposed algorithm converges to a local minimal solution in a finite number of iterations.*

Proof. Let y be the number of all the possible partitions on the pth data subset S^p. Each partition can be represented by a membership matrix W^p. If two partitions are different, their membership matrices are also different, otherwise, they are identical. We note that y is finite, given the data subset S^p and the number of clusters k^p. Therefore, there are a finite number of W^p on the data subset. While applying the proposed algorithm to cluster S^p, we obtain a series of W^p, i.e., $W_1^p, W_2^p, \ldots, W_{t^p}^p$. According to Theorems 1 and 2, we know that the sequence $M(\cdot,\cdot)$ generated by the proposed algorithm is strictly decreasing. Thus, these membership matrices have the following relationships: $M(W_1^p, V_1^p) > M(W_2^p, V_2^p) > \ldots > M(W_{t^p}^p, V_{t^p}^p)$. We assume that the number of iterations t^p is more than $y+1$. That indicates that there are at last two of the same membership matrices in the sequence, i.e., $W_i^p = W_j^p, 1 \le i \ne j \le t^p$. For W_i^p and W_j^p, we have the minimizers V_i^p and V_j^p, according to Theorem 2, respectively. It is clear that $V_i^p = V_j^p$ since $W_i^p = W_j^p$. Therefore, we obtain $M(W_i^p, V_i^p) = M(W_j^p, V_i^p) = M(W_j^p, V_j^p)$. If the value of the function M is not decreasing, the algorithm stops. Therefore, the number of iterations t^p is not more than $y+1$. Hence, t^p is a finite number. □

The time complexity of the algorithm is $O(|S^p|k^p t^p \sum_{j=1}^{m} n_j)$ operations, where k^p is the number of clusters and t^p is the number of iterations. As for the storage, we need $O(|S^p|m + |S^p|k^p + 2k^p \sum_{j=1}^{m} n_j)$ space to hold the set of new input objects, the membership matrix W^p, the last and new clustering models V^{p-1} and V^p. Due to the linear time and space complexities with the number of objects, attributes or clusters, the proposed algorithm is very suited to deal with large data sets.

3.3 The Drifting-Concept Detection

After clustering new input data subsets, we need to analyze the change situation between the new and last clustering models, in order to determine whether the drifting concepts occur. While the concepts have drifted, the last clustering model V^{p-1} is not used to participate in the construction of the new clustering model V^p. In the case, we need to re-cluster the new input data subset, independently of V^{p-1}. In the paper, we consider the following two factors to find out the drifting concepts:

- The distribution variation between the last and new clustering models for cluster representatives;
- The certainty variation between the last and new clustering models for cluster representatives.

The first factor can be measured by a function

$$DV(V^p, V^{p-1}) = \sum_{l=1}^{k^p} \frac{|c_l^p|}{|S^p|} \sum_{j=1}^{m} \sum_{q=1}^{n_j} \left(v_{ljq}^p - v_{ljq}^{p-1} \right)^2,$$

which is a part of the objective function M. The larger the DV value is, the more the two clustering models are different. If the difference reaches up to a certain extent, the "concept drifting" may happen.

However, only considering the first factor may lead to ignoring the change direction between the two clustering models. The distribution variation may reduce or enhance the uncertainty of the clustering model for cluster representatives. If the uncertainty is reduced, it is thought that the concepts do not drift, although the distribution variation is large. We use the second term in the objective function F_g to measure the certainty of clustering model, i.e., $\sum_{l=1}^{k} |c_l| \sum_{j=1}^{m} \sum_{q=1}^{n_j} \left(v_{ljq} \right)^2$. The larger the term value is, the more the certainty of the clustering model for cluster representatives is. Here, what requires explanation is why we wish to minimize the term in F_g but maximize it in the drifting-concept detection. While clustering a data set, we wish to maximize the certainty of the clustering model for cluster representatives on the condition that the clustering model can objectively reflect the characteristics of each cluster possibly as well. Using the function F_g to find the best clustering result on a data set is like a dynamic game between W and V. The game scenario is described as follows: When V is given, it is wished that W can make each object belong to a cluster whose $\mathbf{v}_l$ has the best representative to the object. That can enhance the purity within clusters. When W is given, V should not blindly overestimate the purity within clusters but stimulate more categorical values to contribute to the identification of clusters and effectively avoid losing information. When V is obtained by Eq. (4) and used in F_g, we have

$$F_g(W, V) = nm + n \sum_{j=1}^{m} \frac{2\alpha - 1}{n_j} - \alpha \sum_{l=1}^{k} |c_l| \sum_{j=1}^{m} \sum_{q=1}^{n_j} \left(v_{ljq} \right)^2.$$

According to the above equation, We can see that minimizing $F_g(W, V)$ can realize maximizing the certainty of clustering model for cluster representatives. Therefore, we consider the certainty variation between the last and new clustering models in determining the drifting concepts, i.e.,

$$CV(V^p, V^{p-1}) = \sum_{l=1}^{k^p} \frac{|c_l^p|}{|S^p|} \sum_{j=1}^{m} \sum_{q=1}^{n_j} \left(\left(v_{ljq}^{p-1} \right)^2 - \left(v_{ljq}^p \right)^2 \right).$$

The larger the function value is, the more the uncertainty of the clustering models is enhanced.

Here, we integrate the two factors to define a detection criterion (index) for the drifting concepts, which is described as

$$\Omega(V^p, V^{p-1}) = \frac{2}{\gamma_1 + \gamma_2} \left[\gamma_1 DV(V^p, V^{p-1}) + \gamma_2 CV(V^p, V^{p-1}) \right], \tag{14}$$

where γ_1 and γ_2 are two weights. The larger the $\Omega(V^p, V^{p-1})$ value is, the more possibly the drifting concepts occur. If V^p is the same as V^{p-1}, $\Omega(V^p, V^{p-1})$ is zero. Here, we need to set a threshold ε. If $\Omega(V^p, V^{p-1})$ is larger than ε, the concepts are thought to drift.

However, we see that the $\Omega(V^p, V^{p-1})$ value depends on the effectiveness of V^p and the setting of γ_1 and γ_2. If the obtained V^p is poor, it is very unreliable while the $\Omega(V^p, V^{p-1})$ value is used to determine whether the concepts drift. Therefore, we need to find out a good V^p to judge the drifting concepts. The detection method is described as

$$if \quad \min_{V^p} \Omega(V^p, V^{p-1}) > \varepsilon, \tag{15}$$

it is thought that the "concept drifting" happens.

Next, we need to answer a question: Is the obtained V^p effective by Algorithm 2? When V^p is obtained by Eq. (10) and used in the objective function M, we have

$$\begin{aligned} &M(W^p, V^p) \\ &= \sum_{l=1}^{k^p} |c_l^p| \sum_{j=1}^{m} \sum_{q=1}^{n_j} \left[\frac{|c_{ljq}^p|}{|c_l^p|} \left(2 - v_{ljq}^{p-1} - v_{ljq}^p \right) + \alpha \left(v_{ljq}^{p-1} \right)^2 \right. \\ &\quad \left. + \alpha \left(v_{ljq}^p \right)^2 + \beta \left(v_{ljq}^{p-1} - v_{ljq}^p \right)^2 \right] \\ &= \sum_{l=1}^{k^p} |c_l^p| \sum_{j=1}^{m} \sum_{q=1}^{n_j} \left[(\alpha + \beta) \left(v_{ljq}^{p-1} - v_{ljq}^p \right)^2 + 2\beta \left(\left(v_{ljq}^{p-1} \right)^2 \right. \right. \\ &\quad \left. \left. - \left(v_{ljq}^p \right)^2 \right) \right] + 2m|S^p| + 2|S^p| \sum_{j=1}^{m} \frac{2\alpha - 1}{n_j}. \end{aligned}$$

According to the above equations, we can see that the term

$$\begin{aligned} &\sum_{l=1}^{k^p} |c_l^p| \sum_{j=1}^{m} \sum_{q=1}^{n_j} \left[(\alpha + \beta) \left(v_{ljq}^{p-1} - v_{ljq}^p \right)^2 \right. \\ &\quad \left. + 2\beta \left(\left(v_{ljq}^{p-1} \right)^2 - \left(v_{ljq}^p \right)^2 \right) \right], \end{aligned} \tag{16}$$

is the integration of the two detection factors for drifting concepts. We know that minimizing the objective function M is equal to minimizing Eq. (16). The smaller the objective

function M value is, the more the clustering model V^p for cluster representatives is certain and continuous with the last clustering model V^{p-1}. Thus, we fix $\gamma_1 = \alpha + \beta$ and $\gamma_2 = 2\beta$. In this case,

$$\Omega(V^p, V^{p-1}) = \sum_{l=1}^{k^p} \frac{|c_l^p|}{|S^p|} \left[\frac{2(\alpha+\beta)}{\alpha+3\beta} \sum_{j=1}^{m} \sum_{q=1}^{n_j} \left(v_{ljq}^p - v_{ljq}^{p-1} \right)^2 + \frac{4\beta}{\alpha+3\beta} \left(\sum_{j=1}^{m} \sum_{q=1}^{n_j} \left(v_{ljq}^{p-1} \right)^2 - \sum_{j=1}^{m} \sum_{q=1}^{n_j} \left(v_{ljq}^{p} \right)^2 \right) \right]. \tag{17}$$

According to Eq. (17), we see that The best V^p can be obtained to minimize $\Omega(V^p, V^{p-1})$ while minimizing the objective function M. Since the detection results based on the optimization model fully consider the clustering validity and continuity on the new input data subset, they are reliable and valid.

While the concepts have drifted, we need to re-cluster the new input data subset. In this case, we only use the function F_g as the objective function. The optimization problem becomes the static clustering problem which is described as follows:

$$\min_{W^p, V^p} F_g(W^p, V^p) \text{ subject to } (8). \tag{18}$$

We use Algorithm 1 to re-cluster the new input data subset.

3.4 Overall Implementation

We integrate the proposed clustering optimization algorithm and drifting-concept detection method to cluster categorical data streams. The overall implementation is described in Algorithm 3.

Algorithm 3.

Input: X, α, β, ε
Output: **Clustering** $= \{W^1, W^2, \ldots, W^T\}$,
Clustering models $= \{V^1, V^2, \ldots, V^T\}$,
Drift $= \{\Omega^1, \Omega^2, \ldots, \Omega^T\}$
Estimate the number of clusters k^1 and the initial V^1 on S^1;
$[W^1, V^1]$ = **Algorithm 1**(S^1, V^1, α);
$\Omega^1 = 0$;
For $p = 2 : T$ **do**
  $k^p = k^{p-1}$;
  $[W^p, V^p]$ = **Algorithm 2**(S^p, V^{p-1}, α, β);
  $\Omega^p = \Omega(V^p, V^{p-1})$;
  if $\Omega^p > \varepsilon$ **then**
    Re-estimate the number of clusters k^p and the initial V^p on S^p;
    $[W^p, V^p]$ = **Algorithm 1**(S^p, V^p, α);

The time complexity of the proposed algorithm is $O(\sum_{p=1}^{T} k^p t^p |S^p| (\sum_{j=1}^{m} n_j))$ which is linear with the number of objects, clusters or attribute values in the data stream X. In clustering each of new input data subsets, the algorithm only need save the objects in the current sliding window and the obtained clustering models. The outdated objects can be deleted. Thus, its average space complexity is $O(n/T + \sum_{p=1}^{T} k^p \sum_{j=1}^{m} n_j)$. Therefore, the proposed algorithm can efficiently cluster large-scale data streams.

In the algorithm, we need to set the three parameters α, β and ε. Since the parameters α and β are used in the clustering optimization model, their values may affect the performance of the proposed algorithm. However, the appropriate setting of α and β depends on the domain knowledge of the data sets and the users' subjective understanding, it is difficult to determine which values are the best for them. In the following experiments, we set $\alpha = \beta = 1/2$. In the case,

$$v_{ljq}^p = \frac{1}{2} \left(\frac{|c_{ljq}|}{|c_l^p|} + v_{ljq}^{p-1} \right),$$

and

$$\Omega(V^p, V^{p-1}) = DV(V^p, V^{p-1}) + CV(V^p, V^{p-1}).$$

In such setting, v_{ljq}^p indicates the new clustering model V^p is described by the historical and current cluster information with the same weights. $\Omega(V^p, V^{p-1})$ means that the two influencing factors in the detection criterion of drifting concepts are thought to be equally important. The setting of the parameter ε is also relevant to the domain knowledge of the data sets. While we lack the domain knowledge of a data set in a practical application, we need to test the Ω values between several windows and estimate the ε value. If the Ω value in a window is significantly greater than other windows, the Ω value is used to set ε.

4 Experimental Analysis

We present four experiments to evaluate the performance of the proposed algorithm. The first experiment is to test the clustering accuracy of the proposed algorithm on categorical data streams. The second experiment is to test the effectiveness of the proposed algorithm in detecting the drifting concepts. The third experiment is to test the scalability of the proposed algorithm. The final experiment is to test the effect of the parameters on the effectiveness of the proposed algorithm. In these experiments, we compare the proposed algorithm with other clustering algorithms for categorical data streams proposed by Chen et al. [3] and Cao et al. [4], respectively. These algorithms are tested on four data sets including Letters, DNA, Nursery and KDD-CUP'99 which can be downloaded from the UCI Machine Learning Repository. These data sets are described as follows:

Letters Data. The data set contains character image features of 26 capital letters in the English alphabet. We take data objects with similar looking alphabets, 'B', 'E' and 'F' alphabets from this data set. There are 2,309 data objects (766 'B', 768 'E' and 775 'F') described by 16 attributes which are finite-integer values and seen as categorical attributes in the experiments.

DNA Data. The data set contains 3,190 splice-junctions points with 60 categorical attributes on a DNA sequence at which "superfluous" DNA is removed during the process of protein creation in higher organisms. The data set is partitioned into three classes (767 'EI', 768 'IE' and 1,655 'Neither').

Nursery Data. Nursery Database was derived from a hierarchical decision model originally developed to rank applications for nursery schools. The data set contains 12,960 records with eight categorical attributes. It has five classes (4,320 "not recom", 2 "recommend", 328 "very recom", 4,266 "priority" and 4,044 "spec prior"). In the experiments, we only consider the "not recom", "priority" and "spec prior" classes.

TABLE 2
The Cluster-Distribution Information on Each of Nine Data Subsets

	1	2	3	4	5	6	7	8	9
Class 1	200	350	400	450	150	150	100	50	100
Class 2	200	150	150	100	350	400	450	100	150
Class 3	200	100	50	50	100	50	50	450	350

KDD-CUP'99 Data. The Network data set was used as a test data stream for The Third International Knowledge Discovery and Data Mining Tools Competition. The data set contains 494,021 records, each having a time stamp. The records are classified into 23 classes. One class indicates the normal connection and other 22 classes are network attack types. Each record is described by 41 attributes, in which 34 attributes are continuous and seven are categorical. We used uniform quantization to convert these continuous attributes into discrete values, each attribute with five categories. We also aggregated 22 attack classes into one general attack class.

Each of Chen's and Cao's algorithms provides the detection method of drifting concepts and clustering method for new input data subsets, but does not propose a clustering algorithm for initially clustering the first window or re-clustering data subsets while the concepts are drifting. They only used one of the classical clustering algorithms in these cases. To ensure that the comparisons are in a uniform environmental condition, we employ Algorithm 1 for re-clustering or initially clustering while using these algorithms to cluster a data stream.

4.1 Clustering Accuracy Evaluation

To test the effectiveness of the proposed algorithm for categorical data streams, we first select the data sets Letters, DNA and Nursery. We randomly sample nine data subsets with different cluster distributions from each of the data sets. These cluster-distribution information of sampled subsets on the three data sets is shown in Table 2. For a data set, we select each of the nine subsets as the last window and other subsets as new windows in turn. We assume that the cluster information in the last window is known and the cluster information in the new windows is unknown. We use the clustering algorithms to cluster the objects from new windows according to the cluster information in the last window. We compare the three clustering algorithms on these data streams. To evaluate the performance of clustering algorithms in the experiment, we consider the three validity measures [32]: 1) accuracy (AC), 2) precision (PE) and 3) recall (RE). Let X be a data set, $C = \{C_1, C_2, \ldots, C_k\}$ be a clustering result of X, $P = \{P_1, P_2, \ldots, P_{k'}\}$ be a partition of the original classes in X, n_{ij} be the number of common objects of groups C_i and P_j: $n_{ij} = |C_i \cap P_j|$, b_i be the number of objects in C_i, d_j be the number of objects in P_j. These validity measures are defined as $AC = \frac{1}{n}\sum_{i=1}^{k}\max_{j=1}^{k'} n_{ij}$, $PE = \frac{1}{k}\sum_{i=1}^{k}\frac{\max_{j=1}^{k'} n_{ij}}{b_j}$, and $RE = \frac{1}{k}\sum_{i=1}^{k}\frac{\max_{i=1}^{k} n_{ij}}{d_j}$.

To ensure that the comparisons are in a uniform environmental condition, we set that the number of clusters is equal to the true number of classes on each of the given data sets. The comparison results are shown in Table 3. According to the validity measure values, we see that the performance of Chen's and Cao's algorithms strongly depends on the cluster distributions. The proposed algorithm is more effective and stable than the other two algorithms on clustering the data streams.

Furthermore, we randomly produce 10 data streams on each of the Letters, DNA, Nursery and KDD-CUP'99 data sets. Each data stream includes 12 windows, one of which has a random cluster distribution. For Letters, DNA and Nursery, we set window size as 600. For KDD-CUP'99, we test three window sizes (1,000, 3,000 and 5,000). We compare the clustering effectiveness of the three algorithms on these data streams. The testing results are shown in Table 4. According to these tables, we see that the proposed algorithm is more effective and robust than the other two algorithms in clustering categorical data streams.

4.2 Accuracy Evaluation of Drifting-Concept Detection

We test the proposed detection method of drifting concepts on the KDD-CUP'99 data stream with different window sizes (1,000, 3,000, and 5,000). We compare the proposed method with the outlier and cluster-size variation method proposed by Chen et al. [3] and the data distribution method proposed by Cao et al. [4]. In the data stream, the drifting concepts are thought to happen, if network connections are from normal to a burst of attacks or from the attacks back to normal. Given a window size, we can apply the class labels to identify the drifting concepts. For a window, if the number of connections changes is at last 10 percent of the window size, compared to the last window, it is thought to have the drifting concepts. We use a vector $Dx = [dx_1, dx_2, \ldots, dx_T]$ to save the status of each window, where $dx_p = 1$ if the drifting concepts occur in the pth window, otherwise, $dx_p = 0$, for $1 \le p \le T$. For the first window, we set $dx_1 = 0$. While applying a detection method to the data stream, we can also obtain a status vector of windows $Dx' = [dx'_1, dx'_2, \ldots, dx'_T]$. To evaluate the performance of a detection method in the experiment, we consider the three validity measures: 1) precision (PD), 2) recall (RD) and 3) Euclidean distance (ED), which are defined as

$$PD = \frac{|\{dx_p == 1 \wedge dx'_p == 1, 1 \le p \le T\}|}{|\{dx'_p == 1, 1 \le p \le T\}|},$$

$$RD = \frac{|\{dx_p == 1 \wedge dx'_p == 1, 1 \le p \le T\}|}{|\{dx_p == 1, 1 \le p \le T\}|},$$

and

$$ED = \sqrt{\|Dx - Dx'\|^2}.$$

The first two measures are from the paper [3]. The higher their values are, the better the performance of the detection method is. Here, we add Euclidean distance which is used to judge the dissimilarity between the real window statuses and the window statuses recognized by the detection method. If $ED = \sqrt{\|Dx - Dx'\|^2}$ is very small, the performance of the detection method is very well. Before using the three methods to detect the drifting concepts, we need to set some parameters. For Chen's method, we set the outlier threshold as 0.1, the cluster variation threshold as 0.1 and the cluster difference threshold as 0.5, according to the

TABLE 3
The Results of Different Clustering Algorithms on the Three Data Sets

Last window	Index	Letters data			DNA data			Nursery data		
		Chen's algorithm	Cao's algorithm	Our algorithm	Chen's algorithm	Cao's algorithm	Our algorithm	Chen's algorithm	Cao's algorithm	Our algorithm
200 from class 1	AC	0.9115	0.8675	0.8888	0.8867	0.8042	0.9375	0.8746	0.8656	0.8802
200 from class 2	PE	0.9104	0.8876	0.8863	0.9111	0.8551	0.9381	0.8609	0.8537	0.8657
200 from class 3	RE	0.9110	0.8617	0.8887	0.8720	0.7496	0.9326	0.8671	0.8605	0.8704
350 from Class 1	AC	0.8231	0.8735	0.8860	0.5621	0.7915	0.9265	0.8946	0.4367	0.8956
150 from Class 2	PE	0.8788	0.8918	0.8876	0.8000	0.8443	0.9274	0.8922	0.6859	0.8846
100 from Class 3	RE	0.8280	0.8685	0.8870	0.5493	0.7513	0.9221	0.8761	0.6305	0.8819
400 from Class 1	AC	0.8048	0.8688	0.8815	0.6610	0.7917	0.9238	0.8531	0.3958	0.8663
150 from Class 2	PE	0.8648	0.8874	0.8836	0.8134	0.8441	0.9289	0.8394	0.6912	0.8522
50 from Class 3	RE	0.8074	0.8673	0.8822	0.6412	0.7587	0.9167	0.8346	0.5000	0.8507
450 from Class 1	AC	0.7733	0.8413	0.8842	0.6054	0.7904	0.9138	0.8763	0.4063	0.8813
100 from Class 2	PE	0.8485	0.8653	0.8868	0.7976	0.8424	0.9177	0.8645	0.4063	0.8663
50 from Class 3	RE	0.7872	0.8438	0.8828	0.5890	0.7575	0.9071	0.8584	1.0000	0.8661
150 from Class 1	AC	0.8965	0.8610	0.8875	0.4860	0.7969	0.9344	0.8667	0.3750	0.8646
350 from Class 2	PE	0.9122	0.8798	0.8927	0.7921	0.8485	0.9378	0.8588	0.3750	0.8567
100 from Class 3	RE	0.8884	0.8512	0.8827	0.4594	0.7571	0.9281	0.8618	1.0000	0.8553
150 from Class 1	AC	0.8952	0.8610	0.8896	0.5654	0.7844	0.9067	0.8877	0.3750	0.8769
400 from Class 2	PE	0.9106	0.8730	0.8975	0.7923	0.8393	0.9138	0.8834	0.3750	0.8726
50 from Class 3	RE	0.8871	0.8539	0.8835	0.5380	0.7513	0.8981	0.8860	1.0000	0.8714
100 from Class 1	AC	0.8633	0.8638	0.8790	0.4496	0.7877	0.9217	0.8763	0.3854	0.8629
450 from Class 2	PE	0.8869	0.8713	0.8866	0.7836	0.8417	0.9274	0.8731	0.3854	0.8594
50 from Class 3	RE	0.8590	0.8563	0.8782	0.4383	0.7555	0.9152	0.8762	1.0000	0.8604
50 from Class 1	AC	0.7994	0.8571	0.8771	0.8819	0.8258	0.9233	0.8298	0.4063	0.8648
100 from Class 2	PE	0.8172	0.8675	0.8672	0.8664	0.8726	0.9095	0.8252	0.4063	0.8379
450 from Class 3	RE	0.8278	0.8441	0.8757	0.8922	0.7341	0.9218	0.8423	1.0000	0.8568
100 from Class 1	AC	0.8292	0.8565	0.8883	0.7900	0.8148	0.9302	0.8590	0.4892	0.8917
150 from Class 2	PE	0.8354	0.8666	0.8917	0.8256	0.8670	0.9256	0.8553	0.7150	0.8707
350 from Class 3	RE	0.8484	0.8428	0.8842	0.8183	0.7388	0.9275	0.8688	0.6311	0.8809

suggestion of the paper [3]. For Cao's method, we set the window-distance threshold as 0.1, according to the suggestion of the paper [4]. For our method, we set the threshold ε as 1. On the data stream, we test the number of clusters k as 2, 5, 10 and 15 for the three methods.

The comparison results are shown in Table 5. We first analyze the effectiveness of these methods while $k = 2$. When the window size is 1,000, the number of real exception windows is 28. Our method can correctly find out 23 exception windows which is the most among the three methods. However, the proposed method also wrongly recognized several normal windows as exceptions. Thus, the ED value of our method is slightly less than that of Cao's method. While the window size is set to 3,000 or 5,000, our method can correctly find out all the exception windows and have the low error recognition rates. According to PD, RD and ED, we can also see that the proposed method is better than other methods with these window sizes. In the experiment, Chen's method wrongly recognized many normal windows as exceptions. The main reason is that the method uses a cluster-size variation index to judge the drifting concepts. The index is very sensitive to the clustering result in the window. In many cases, the changes of cluster sizes do not necessarily indicate the emergence of new clusters or the disappearance of old clusters. As the number of clusters increases, we find that the detection results obtained by Cao's method are constant. This reason is that Cao's method uses the difference of data distributions between two windows to judge whether concepts drift and does not consider the difference of the cluster structures. According to Table 5, we see that the effectiveness of Chen's method is better than Cao's method, while $k > 10$. This indicates that Chen's method is suitable to deal with data

TABLE 4
The Results of Different Clustering Algorithms on the Data Streams with Random Cluster Distributions

Data set	Window size	Index	Chen's algorithm	Cao's algorithm	Our algorithm
Letters	600	AC	0.7931±0.0017	0.7821±0.0017	0.8202±0.0008
		PE	0.7909±0.0016	0.7714±0.0011	0.8045±0.0008
		RE	0.7366±0.0032	0.7072±0.0053	0.7393±0.0025
DNA	600	AC	0.7077±0.0015	0.7852±0.0005	0.9105±0.0002
		PE	0.7868±0.0013	0.7882±0.0007	0.8721±0.0007
		RE	0.5729±0.0041	0.6880±0.0029	0.9015±0.0007
Nursery	600	AC	0.8175±0.0008	0.6028±0.0013	0.8313±0.0003
		PE	0.8078±0.0011	0.6329±0.0016	0.8187±0.0006
		RE	0.6947±0.0002	0.9178±0.0063	0.7235±0.0006
KDD-CUP'99	1000	AC	0.8577±0.0013	0.8656±0.0019	0.9225±0.0002
		PE	0.8397±0.0022	0.8441±0.0014	0.8857±0.0003
		RE	0.6524±0.0100	0.7114±0.0139	0.8144±0.0053
	3000	AC	0.8629±0.0009	0.8689±0.0015	0.9240±0.0002
		PE	0.8331±0.0007	0.8457±0.0010	0.8923±0.0003
		RE	0.6732±0.0092	0.6707±0.0169	0.8045±0.0043
	5000	AC	0.8458±0.0015	0.8599±0.0023	0.9128±0.0003
		PE	0.8321±0.0014	0.8277±0.0018	0.8772±0.0008
		RE	0.6530±0.0124	0.6784±0.0138	0.8188±0.0027

TABLE 5
The Results of Different Detection Methods for Drifting Concepts on the KDD-CUP'99 Data Stream

# of clusters	Window size	Chen's method	Chen's method	Cao's method	Our method
k=2	1000	PD	21/429	10/24	23/49
		RD	21/28	10/28	23/28
		ED	20.2731	5.6569	5.5678
	3000	PD	13/137	7/14	15/22
		RD	13/15	7/15	15/15
		ED	11.225	3.873	2.6458
	5000	PD	8/79	3/11	12/18
		RD	8/12	3/12	12/12
		ED	8.6603	4.1231	2.4495
k=5	1000	PD	18/63	10/24	19/33
		RD	18/28	10/28	19/28
		ED	7.4162	5.6569	4.7958
	3000	PD	12/30	7/14	14/16
		RD	12/15	7/15	14/15
		ED	4.5826	3.873	1.7321
	5000	PD	7/21	3/11	11/16
		RD	7/12	3/12	11/12
		ED	4.3589	4.1231	2.4495
k=10	1000	PD	18/36	10/24	19/29
		RD	18/28	10/28	19/28
		ED	5.2915	5.6569	4.3589
	3000	PD	12/18	7/14	14/16
		RD	12/15	7/15	15/15
		ED	3.0000	3.8730	1.7321
	5000	PD	6/12	3/11	10/16
		RD	6/12	3/12	10/12
		ED	3.4641	4.1231	2.8284
k=15	1000	PD	15/38	10/24	19/29
		RD	15/28	10/28	19/28
		ED	6.0000	5.6569	4.3589
	3000	PD	11/19	7/14	14/17
		RD	11/15	7/15	12/15
		ED	3.4641	3.8730	2.0000
	5000	PD	6/11	3/11	9/14
		RD	6/12	3/12	9/12
		ED	3.3166	4.1231	2.8284

sets with more clusters. We also see that the proposed method is better than other methods while setting these numbers of clusters.

4.3 Scalability Analysis

In the scalability analysis, we test the three algorithms on the the KDD-CUP'99 data stream. The window size is set to 1,000. The computational results were performed by using a machine with an Intel i7-4710MQ and 16 GB RAM. The computational times of algorithms are plotted with respect to the number of objects, attributes and clusters, while the other corresponding parameters are fixed. All of the experiments were repeated ten times and the average computational times were depicted. The comparison results are shown in Tables 6, 7, and 8.

Table 6 shows the computational times against the numbers of objects, while the number of attributes is 41 and the number of clusters is 2. Table 7 shows the computational times against the numbers of attributes, while the number of clusters is 2 and the number of objects is 100,000. Table 8 shows the computational times against the numbers of clusters, while the number of attributes is 41 and the number of objects is 100,000. According to the tables, the proposed algorithm requires more computational times than other algorithms. It is an expected outcome, since it needs more than one iteration for searching an optimal solution. The proposed algorithm considers the clustering problem of the data subset in a new window as an "iterative learning" but other algorithms see it as a "data labeling" which only need one iteration. Fortunately, the convergence speed of the proposed

TABLE 7
Computational Times (Seconds) of Clustering Algorithms for Different Numbers of Attributes

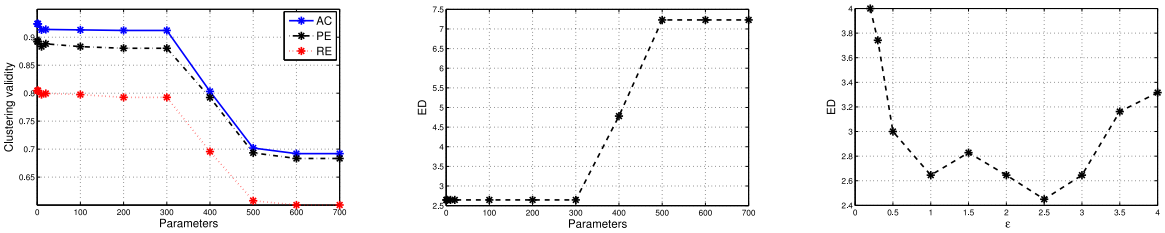
m	Chen's algorithm	Cao's algorithm	Our algorithm
10	11.42	7.77	35.47
20	21.54	15.12	69.03
30	32.10	22.49	99.05
40	42.59	30.02	161.81

TABLE 6
Computational Times (Seconds) of Clustering Algorithms for Different Numbers of Objects

n	Chen's algorithm	Cao's algorithm	Our algorithm
100,000	42.68	29.48	160.60
200,000	95.59	69.94	330.25
300,000	157.32	117.96	511.61
400,000	230.56	189.42	701.83

TABLE 8
Computational Times (Seconds) of Clustering Algorithms for Different Numbers of Clusters

k	Chen's algorithm	Cao's algorithm	Our algorithm
5	45.22	29.85	169.12
10	46.65	33.32	256.88
15	47.29	33.66	371.69
20	48.36	34.91	458.69



(a) The AC, PE and RE values with respect to different α and β values (b) The ED values with respect to different α and β values (c) The ED values with respect to different ε values

Fig. 2. The effect of parameters on the KDD-CUP'99 data stream.

algorithm is rather fast. The number of iterations is about between 5 and 10 in the experiment. Although it needs several iterations, it is still scalable. It can cluster categorical objects efficiently, since it has the linear-time complexity with the number of objects, attributes, clusters or iterations.

4.4 Parameters Analysis

We test the effect of the parameters α, β and ε on clustering data streams and detecting the drifting concepts. The test is carried out on the KDD-CUP'99 data stream with the window size as 3,000. In Section 3.4, we analyze the recommended setting of α and β, i.e., $\alpha = \beta = 1/2$. Setting α equal to β aims to balancing the significance of several terms in the objective function M and the two influencing factors in the detection index Ω. Therefore, we set α and β to the same values in the following.

First, we select several values of the parameters α and β to cluster the data stream. In Fig. 2a, we show the AC, PE and RE values of the clustering results against the α and β values. We see that if the parameters are less than a certain value, the changes of the AC, PE and RE values are not obvious as the parameter values increase. This illustrates that the proposed algorithm is robust in clustering the data streams when the the α and β values are small. However, while the parameter values continue to grow, the AC, PE and RE values drop sharply. This indicates that clusters can not be recognized when the parameter values are very large.

Furthermore, we analyze the effect of the parameters α and β on the detection of drifting concepts. Fig. 2b show the ED values of the detection results against the α and β values, while setting $\varepsilon = 1$. We see that the experimental result is similar to Fig. 2a. If the α and β values are very large, the clustering results are very bad, which inevitably leads to poor detection results.

Finally, we test the effect of the parameter ε on detecting drifting concepts. Fig. 2c show the ED values against the ε values, while fixing $\alpha = \beta = 1/2$. We see that the proposed method has the good detection results of drifting concepts on the data stream, while selecting the ε value in the interval $[0.5, 3]$.

5 Conclusions

In this paper, we have presented an optimization model for clustering categorical data streams. In the model, an objective function is proposed which simultaneously considers the clustering validity on new sliding windows and difference of cluster structures between windows. An iterative algorithm is developed to minimize the objective function with some constraints. Furthermore, a validity index for the drifting-concept detection is derived from the optimization model. We take use of the validity index and the optimization model to catch the evolution trend of cluster structures on data streams. Finally, we tested the performance of the proposed algorithm in the experiments. The experimental results have shown that the proposed algorithm is effective in clustering the categorical data streams and the detection results based on the proposed method are reliable.

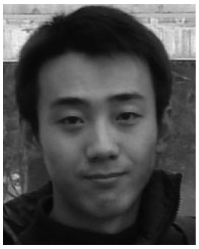
Acknowledgments

The authors are very grateful to the editors and reviewers for their valuable comments and suggestions. This work is supported by the National Natural Science Foundation of China (Nos. 61305073, 61432011, 61472400, 61573229, U1435212), the National Key Basic Research and Development Program of China (973)(Nos. 2013CB329404, 2014CB340400), the Foundation of Doctoral Program Research of Ministry of Education of China(No. 20131401120001), the Technology Research Development Projects of Shanxi (No. 2015021100), and Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi (No. 2014104, 2015107).

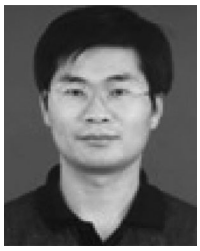
References

[1] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2001.
[2] A. Jain and R. Dubes, *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice Hall, 1988.
[3] H. Chen, M. Chen, and S. Lin, "Catching the trend: A framework for clustering concept-drifting categorical data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 5, pp. 652–665, May 2009.
[4] F. Cao, J. Liang, and L. Bai, "A framework for clustering categorical time-evolving data," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 5, pp. 872–885, Oct. 2010.
[5] C. Aggarwal, J. Han, J. Wang, and P. Yu, "A framework for clustering evolving data streams," in *Proc. 29th Int. Conf. Very Large Data Bases*, 2003, pp. 81–92.
[6] F. Cao, M. Ester, Q. Qian, and A. Zhou, "Density-based clustering over an evolving data streams with noise," in *Proc. SIAM Conf. Data Mining*, 2006, pp. 328–339.
[7] D. Chakrabarti, R. Kumar, and A. Tomkins, "Evolutionary clustering," in *Proc. 12th ACM SIGKDD. Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 554–560.
[8] Y. Chi, X.-D. Song, D.-Y. Zhou, K. Hino, and B. Tseng, "Evolutionary spectral clustering by incorporating temporal smoothness," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 153–162.
[9] 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.
[10] M. Gaber and P. Yu, "Detection and classification of changes in evolving data streams," *Int. J. Inf. Technol. Decision Making*, vol. 5, no. 4, pp. 659–670, 2006.
[11] S. Guha, N. Mishra, R. Motwani, and L. O. Callaghan, "Clustering data streams," in *Proc. Symp. Found. Comput. Sci.*, 2000, pp. 359–366.

[12] S. Ho and H. Wechsler, "A martingale framework for detecting changes in data streams by testing exchangeability," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 20, pp. 2113–2127, Dec. 2010.
[13] O. Nasraoui and C. Rojas, "Robust clustering for tracking noisy evolving data streams," in *Proc. 6th SIAM Conf. Data Mining*, 2006, pp. 618–622.
[14] M. Yeh, B. Dai, and M. Chen, "Clustering over multiple evolving streams by events and correlations," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 10, pp. 1349–1362, Oct. 2007.
[15] E. Cesario, G. Manco, and R. Ortale, "Top-down parameter-free clustering of high-dimensional categorical data," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 12, pp. 1607–1624, Dec. 2007.
[16] C. Aggarwal and P. Yu, "On clustering massive text and categorical data streams," *Knowl. Inf. Syst.*, vol. 24, pp. 171–196, 2010.
[17] D. Barbara, Y. Li, and J. Couto, "COOLCAT: An entropy-based algorithm for categorical clustering," in *Proc. 11th Int. Conf. Inf. Knowl. Manage.*, 2002, pp. 582–589.
[18] K. Chen and L. Liu, "HE-tree: A framework for detecting changes in clustering structure for categorical data streams," *Int. J. Very Large Data Bases*, vol. 18, no. 5, pp. 1241–1260, 2009.
[19] Z. Huang, "A fast clustering algorithm to cluster very large categorical data sets in data mining," in *Proc. SIGMOD Workshop Res. Issues Data Mining Knowl. Discovery*, 1997, pp. 1–8.
[20] M. A. Gluck and J. E. Corter, "Information uncertainty and the utility categories," in *Proc. 7th Annu. Conf. Cogn. Sci. Soc.*, 1985, pp. 283–287.
[21] Z. He, S. Deng, and X. Xu, "Improving K-modes algorithm considering frequencies of attribute values in mode," in *Proc. Comput. Intell. Security*, 2005, pp. 157–162.
[22] O. San, V. Huynh, and Y. Nakamori, "An alternative extension of the K-means algorithm for clustering categorical data," *Pattern Recognition*, vol. 14, no. 2, pp. 241–247, 2004.
[23] M. Ng, M. J. Li, Z. X. Huang, and Z. He, "On the impact of dissimilarity measure in K-modes clustering algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 503–507, Mar. 2007.
[24] D. Fisher, "Knowledge acquisition via incremental conceptual clustering," *Mach. Learn.*, vol. MH-2, no. 2, pp. 139–172, 1987.
[25] P. Andritsos, P. Tsaparas, R. Miller, and K. Sevcik, "LIMBO: Scalable clustering of categorical data," in *Proc. 9th Int. Conf. Extending Database Tech.*, 2004, pp. 123–146.
[26] L. Bai, J. Liang, C. Dang, and F. Cao, "The impact of cluster representatives on the convergence of the K-modes type clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1509–1522, Jun. 2013.
[27] L. Bai and J. Liang, "Cluster validity functions for categorical data: A solution-space perspective," *Data Mining Knowl. Discovery*, vol. 29, pp. 1560–1597, 2014, doi: 10.1007/s10618-014-0387-5.
[28] F. Cao, J. Liang, and L. Bai, "A new initialization method for categorical data clustering," *Expert Syst. Appl.*, vol. 36, no. 7, pp. 10223–10228, 2009.
[29] S. Wu, Q. Jiang, and J. Z. Huang, "A new initialization method for clustering categorical data," in *Proc. 11th Pacific-Asia Conf. Adv. Knowl. Discovery Data Mining*, 2007, pp. 972–980.
[30] L. Bai, J. Liang, C. Dang, and F. Cao, "A cluster centers initialization method for clustering categorical data," *Expert Syst. Appl.*, vol. 39, no. 9, pp. 8022–8029, 2012.
[31] L. Bai, J. Liang, and C. Dang, "An initialization method to simultaneously find initial cluster centers and the number of clusters for clustering categorical data," *Knowl.-Based Syst.*, vol. 24, no. 6, pp. 785–795, 2011.
[32] Y. Yang, "An evaluation of statistical approaches to text categorization," *J. Inf. Retrieval*, vol. 1, no. 1/2, pp. 67–88, 2004.



Liang Bai received the PhD degree in computer science from Shanxi University, in 2012. He is currently an associate professor in the School of Computer and Information Technology, Shanxi University, and a postdoctoral worker in the Institute of Computing Technology, Chinese Academy of Sciences. His research interest includes the areas of cluster analysis. He has published several journal papers in his research fields, including in the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *Data Mining and Knowledge Discovery*, the *IEEE Transactions on Fuzzy Systems*, the *Pattern Recognition*, the *Information Sciences*, and so on. He received the Excellent Doctorial Thesis Award from the Chinese Association for Artificial Intelligence (2014).



Xueqi Cheng is a professor in the Institute of Computing Technology, Chinese Academy of Sciences (ICT-CAS), and the director of the Research Center of Web Data Science & Engineering (WDSE), ICT-CAS. His main research interests include network science, web search and data mining, big data processing and distributed computing architecture, and so on. He has published more than 100 publications in prestigious journals and conferences, including the *IEEE Transactions on Information Theory*, the *IEEE Transactions on Knowledge and Data Engineering*, the *Journal of Statistics Mechanics: Theory and Experiment, Physical Review E.*, ACM SIGIR, WWW, ACM CIKM, WSDM, IJCAI, ICDM, and so on. He has won the Best Paper Award in CIKM (2011) and the Best Student Paper Award in SIGIR (2012). He is currently serving on the editorial board of the *Journal of Computer Science and Technology*, the *Journal of Computer*, and so on. He received the China Youth Science and Technology Award (2011), the Young Scientist Award of Chinese Academy of Sciences (2010), CVIC Software Engineering Award (2008), the Second prize for the National Science and Technology Progress (2004), and so on. He is a member of the IEEE.

Jiye Liang received the MS and PhD degrees from Xi'an Jiaotong University, Xi'an, China, in 1990 and 2001, respectively. He is currently a professor in 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. His current research interests include computational intelligence, rough set theory, granular computing, and so on. He has published more than 70 journal papers in his research fields, including in *Artificial Intelligence*, the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the *IEEE Transactions on Knowledge and Data Engineering*, the *IEEE Transactions on Fuzzy Systems*, *Data Mining and Knowledge Discovery*, the *IEEE Transactions on Systems, Man, and Cybernetics*, and so on.

Huawei Shen received the BE degree in electronic information from the Xi'an Institute of Posts and Telecommunications, China, in 2003, and the PhD degree in information security from the Institute of Computing Technology, Chinese Academy of Sciences (CAS), China, in 2010. He is currently an associate professor in the Institute of Computing Technology, CAS. His major research interests include network science, information recommendation, user behaviour analysis, machine learning, and social network. He has published more than 20 papers in prestigious journals and top international conferences, including in *Physical Review E*, the *Journal of Statistical Mechanics*, *Physica A*, WWW, CIKM, and IJCAI. He is a member of the Association of Innovation Promotion for Youth of CAS. He received the Top 100 Doctoral Thesis Award of CAS in 2011and the Grand Scholarship of the President of CAS in 2010.