An Algorithm for Clustering Categorical Data With Set-Valued Features

Fuyuan Cao, Joshua Zhexue Huang, Jiye Liang, Xingwang Zhao, Yinfeng Meng, Kai Feng, and Yuhua Qian

***Abstract*— In data mining, objects are often represented by a set of features, where each feature of an object has only one value. However, in reality, some features can take on multiple values, for instance, a person with several job titles, hobbies, and email addresses. These features can be referred to as set-valued features and are often treated with dummy features when using existing data mining algorithms to analyze data with set-valued features. In this paper, we propose an SV-*k*-modes algorithm that clusters categorical data with set-valued features. In this algorithm, a distance function is defined between two objects with set-valued features, and a set-valued mode representation of cluster centers is proposed. We develop a heuristic method to update cluster centers in the iterative clustering process and an initialization algorithm to select the initial cluster centers. The convergence and complexity of the SV-*k*-modes algorithm are analyzed. Experiments are conducted on both synthetic data and real data from five different applications. The experimental results have shown that the SV-*k*-modes algorithm performs better when clustering real data than do three other categorical clustering algorithms and that the algorithm is scalable to large data.**

***Index Terms*— Categorical data set-valued feature, set-valued modes, SV-*k*-modes algorithm.**

I. Introduction

A COMMON data representation model in data analysis and mining describes a set of n objects $\{x_1, x_2, \cdots, x_n\}$ by a set of m features $\{A_1, A_2, \cdots, A_m\}$. In this model [1], a data set X is represented as a table or matrix in which each row is a particular object and each column is a feature whose value for an object is a single value. This data matrix is used as input to most data mining algorithms. However, this data representation is oversimplified. In real applications, features can have multiple values for an object, for instance, a person with several job titles and hobbies. Such data are widespread in questionnaire, banking, insurance, telecommunication, retail, and medical databases.

TABLE I
Example of Data With Set-Valued Features

ID	Name	Sex	⋯	Title	Hobby
1	John	M		{CEO, Prof.}	{Sport, Music}
2	Tom	M		{CEO, Chair}	{Reading, Sport}
⋯	⋯	⋯	⋯	⋯	⋯
n	Katty	F		{Prof., Chair}	{Traveling, Music}

A more general data representation in real-world applications is illustrated in the example in Table I.

Without loss of generality, the data in Table I can be formulated as follows. Let $\mathbf{X} = \{X_1, X_2, \cdots, X_n\}$ be a set of n objects described by a set of m features $\{A_1, A_2, \cdots, A_m\}$. Let $V^j (1 \le j \le m)$ denote a set of categorical values for A_j in $\mathbf{X}$. Suppose that $V_{X_i}^{A_j}$ is a nonempty finite set of values of A_j for object X_i. If $V_{X_i}^{A_j} \subseteq V^j$, A_j is called a set-valued feature and X_i is called a set-valued object. In the traditional representation, A_j has a single value from V^j for each object and is a single-valued feature, which is a special case of set-valued features.

To analyze a set of set-valued objects in Table I, the commonly used method uses dummy categorical features that are created to represent set-valued features. Each unique value of a set-valued feature is made a dummy feature whose value is 1 if an object has that value in the set-valued feature; otherwise, 0 is assigned to the dummy feature for that object. Although dummy features simplify the representation of set-valued features and enable classification or clustering algorithms to be used to analyze set-valued objects, this treatment may result in the fragmentation of semantic information because a single feature is divided into many features. In addition, as the number of set-valued features increases in a data set, many distance measures become meaningless [2]. Although some distance measures can make a large difference for the 0/1 coding, the coding method may generate meaningless cluster centers in certain clustering algorithms.

Giannotti *et al.* [2] proposed a transactional k-means algorithm (Trk-means) with the Jaccard distance to cluster set-valued objects but omitted analysis of the convergence of the algorithm. Guha *et al.* [3] presented a ROCK algorithm, which is an agglomerative hierarchical method unscalable to large data. It is also difficult to obtain the interpretable cluster representatives from hierarchical clustering results.

Manuscript received August 4, 2016; revised February 22, 2017 and October 22, 2017; accepted November 1, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61573229, Grant 61473194, Grant 61432011, Grant 61773247, Grant 61603230, and Grant U1435212, in part by the Natural Science Foundation of Shanxi Province under Grant 2015011048, and in part by the Shanxi Scholarship Council of China under Grant 2016-003. *(Corresponding author: Joshua Zhexue Huang.)*

F. Cao, J. Liang, X. Zhao, Y. Meng, K. Feng, and Y. Qian are with the Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China (e-mail: cfy@sxu.edu.cn; ljy@sxu.edu.cn; zhaoxw84@163.com; mengyf@sxu.edu.cn; fengkai@sxu.edu.cn; jinchengqyh@sxu.edu.cn).

J. Z. Huang is with the College of Computer Sciences and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: zx.huang@szu.edu.cn).

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/TNNLS.2017.2770167

In this paper, we propose a set-valued k-modes (SV-k-modes) algorithm to cluster categorical data with set-valued features. This algorithm takes a data set with set-valued features in the format shown in Table I as input data. A set-valued object is defined to represent the center of a cluster as *set-valued modes*, and a new cluster center update method is developed to search for the cluster center from the given set of set-valued categorical objects by minimizing the sum of the distance between the objects and the cluster center. Based on the new cluster center representation and the center update method, the SV-k-modes algorithm is developed to extend the clustering process of the k-modes algorithm to cluster categorical data with set-valued features. To speed up the search for a new cluster center, we propose a heuristic method to construct new cluster centers in each iteration of the clustering process, which can significantly reduce the search time for new cluster centers. A method for the selection of the initial cluster centers is also developed to improve the clustering performance of the SV-k-modes algorithm. Experiments are conducted on both synthetic data and real data from five different applications. The experimental results have shown that the SV-k-modes algorithm performs better at clustering the real data than do three other categorical clustering algorithms and is scalable to large data.

The remainder of this paper is organized as follows. Section II presents the SV-k-modes algorithm. Section III presents a heuristic method to construct new cluster centers. In Section IV, a method for selecting initial cluster centers is given. In Section V, we show the experimental results on real data sets from five different applications. Section VI shows the results of a scalability test of the SV-k-modes algorithm using synthetic data sets. Some related work is reviewed in Section VII. Conclusions about this paper are given in Section VIII.

II. SV-k-Mode Clustering

In this section, we present the SV-k-modes algorithm, which uses the k-means clustering process [4] to cluster categorical data with set-valued features. Given a set of initial cluster centers, the k-means clustering process iterates in two steps: 1) allocating objects into clusters according to a distance measure and 2) updating cluster centers according to the new allocation of objects in the clusters. In the SV-k-modes algorithm, the Jaccard coefficient [5] is used as the distance measure between two set-valued objects. k set-valued objects, called set-valued k modes, are used as representatives of k cluster centers. Given a cluster of set-valued objects, the cluster center is found by minimizing the sum of the distances between objects and the cluster center. As with the k-means algorithm, the SV-k-modes algorithm converges to a local minimum after a number of iterations.

A. Distance Measure Between Two Set-Valued Objects

Definition 1: [6] Let X and Y be two nonempty finite sets. The dissimilarity measure between X and Y is defined as

$$d(X, Y) = 1 - \frac{|X \cap Y|}{|X \cup Y|}$$

$d(X, Y)$ is a distance measure that satisfies the following properties [6].

1) Non-negativity: $d(X, Y) \geq 0$ and $d(X, X) = 0$.
2) Symmetry: $d(X, Y) = d(Y, X)$.
3) Triangle inequality: $d(X, Y) + d(Y, Z) \geq d(X, Z)$.

$d(\cdot, \cdot)$ is a generalization of the simple matching distance measure that is used in the k-modes algorithm to cluster categorical data with single-valued features. Clearly, $0 \leq d(\cdot, \cdot) \leq 1$.

Given two set-valued objects X_i and X_j described by m set-valued features $\{A_1, A_2, \ldots, A_m\}$, the dissimilarity measure between the two objects is defined as

$$D_m(X_i, X_j) = \sum_{s=1}^{m} d\big(V_{X_i}^{A_s}, V_{X_j}^{A_s}\big). \tag{1}$$

$D_m(X_i, X_j)$ is a distance measure that satisfies the following properties.

1) Non-negativity: $D_m(X_i, X_j) \geq 0$ and $D_m(X_i, X_i) = 0$.
2) Symmetry: $D_m(X_i, X_j) = D_m(X_j, X_i)$.
3) Triangle inequality: $D_m(X_i, X_j) + D_m(X_j, X_k) \geq D_m(X_i, X_k)$.

Proof: Properties 1) and 2) can be verified directly by the definition of $D_m(X_i, X_j)$.

Property 3) can be proved by mathematical induction as follows.

When $m = 1$, by property 3) of $d(\cdot, \cdot)$

$$\begin{aligned} D_1(X_i, X_j) + D_1(X_j, X_k) &= d\big(V_{X_i}^{A_1}, V_{X_j}^{A_1}\big) + d\big(V_{X_j}^{A_1}, V_{X_k}^{A_1}\big) \\ &\geq d\big(V_{X_i}^{A_1}, V_{X_k}^{A_1}\big) \\ &= D_1(X_i, X_k). \end{aligned}$$

For any positive integer $m \geq 2$, we assume that $D_{m-1}(X_i, X_j) + D_{m-1}(X_j, X_k) \geq D_{m-1}(X_i, X_k)$. We prove that $D_m(\cdot, \cdot)$ satisfies property 3).

Using the triangle inequality properties of $D_{m-1}(\cdot, \cdot)$ and $d(\cdot, \cdot)$, we have that

$$\begin{aligned} &D_m(X_i, X_j) + D_m(X_j, X_k) \\ &\quad = \sum_{s=1}^{m} d\big(V_{X_i}^{A_s}, V_{X_j}^{A_s}\big) + \sum_{s=1}^{m} d\big(V_{X_j}^{A_s}, V_{X_k}^{A_s}\big) \\ &\quad = \Bigg(\sum_{s=1}^{m-1} d\big(V_{X_i}^{A_s}, V_{X_j}^{A_s}\big) + \sum_{s=1}^{m-1} d\big(V_{X_j}^{A_s}, V_{X_k}^{A_s}\big) \\ &\qquad + \big(d\big(V_{X_i}^{A_m}, V_{X_j}^{A_m}\big) + d\big(V_{X_j}^{A_m}, V_{X_k}^{A_m}\big)\big) \\ &\quad = (D_{m-1}(X_i, X_j) + D_{m-1}(X_j, X_k)) \\ &\qquad + \big(d\big(V_{X_i}^{A_m}, V_{X_j}^{A_m}\big) + d\big(V_{X_j}^{A_m}, V_{X_k}^{A_m}\big)\big) \\ &\quad \geq D_{m-1}(X_i, X_k) + \big(d\big(V_{X_i}^{A_m}, V_{X_j}^{A_m}\big) + d\big(V_{X_j}^{A_m}, V_{X_k}^{A_m}\big)\big) \\ &\quad \geq D_{m-1}(X_i, X_k) + d\big(V_{X_i}^{A_m}, V_{X_k}^{A_m}\big) \\ &\quad = D_m(X_i, X_k). \end{aligned}$$

■

B. Set-Valued Modes as Cluster Centers

Given a set of set-valued objects $\mathbf{X}$ with m set-valued features, the center of $\mathbf{X}$ is defined as follows.

Definition 2: Let $\mathbf{X} = \{X_1, X_2, \cdots, X_n\}$ be a set of n objects with m set-valued features, and let Q be a set-valued

object with the same m set-valued features. Q is the set-valued modes or the center of $\mathbf{X}$ if Q minimizes the following function:

$$F(\mathbf{X}, Q) = \sum_{i=1}^{n} D_m(X_i, Q) \tag{2}$$

where $X_i \in \mathbf{X}$ and $D_m(X_i, Q)$ is the distance between X_i and Q as defined in (1). Here, Q is not necessarily an object of $\mathbf{X}$.

If the features of $\mathbf{X}$ are single valued, Q is the modes of $\mathbf{X}$ [7]. In a general case, if $\mathbf{X}$ has set-valued features, Q is called the set-valued modes of $\mathbf{X}$.

To minimize $F(\mathbf{X}, Q)$, we can separately minimize the sum of the distance between the objects and the center in feature A_j ($j \in \{1, 2, \cdots, m\}$) to search for the set value Q_{A_j}, i.e., minimizing $\sum_{i=1}^{n} d(V_{X_i}^{A_j}, Q_{A_j})$ to find Q_{A_j}. Supposing that A_j has r_j' distinct categorical values, the number of categorical values that Q_{A_j} can have is between 1 and r_j'. If the size of Q_{A_j} is r_j, the number of possible sets for Q_{A_j} is $C_{r_j'}^{r_j}$. Therefore, the total number of possible sets for Q_{A_j} to choose from is $\sum_{r_j=1}^{r_j'} C_{r_j'}^{r_j}$. With an exhaustive search method, we can traverse each of $\sum_{r_j=1}^{r_j'} C_{r_j'}^{r_j}$ unique combinations to find a Q_{A_j} that minimizes $\sum_{i=1}^{n} d(V_{X_i}^{A_j}, Q_{A_j})$. This global optimization algorithm, global algorithm of finding set-valued modes (GAFSM), for finding set-valued modes is described in Algorithm 1.

Algorithm 1 *GAFSM* Algorithm

Input:
- $\mathbf{X} = \{X_1, X_2, \cdots, X_n\}$: the set of n set-valued objects;
- m : the number of features;

Output: The set-valued modes Q;
Method:
$Q = \emptyset$;
for $j = 1$ *to* m **do**
  Generate a set $Q^j = \{Q_{A_j}^1, Q_{A_j}^2, \cdots, Q_{A_j}^{2^{|V^j|}-1}\}$ of $\sum_{r_j=1}^{r_j'} C_{r_j'}^{r_j}$ combinations in V^j by binomial theorem;
  for $i = 1$ *to* $2^{|V^j|} - 1$ **do**
    $TempValue = \infty$;
    $TempQ = \emptyset$;
    Compute $F_i = F(\mathbf{X}, Q_{A_j}^i)$ according to (2);
    if $F_i \le TempValue$ **then**
      $TempValue = F_i$;
      $TempQ = Q_{A_j}^i$;
    end if
  end for
  $Q \leftarrow TempQ$;
end for
return Q;

C. SV-k-Modes Algorithm

Given (1) as the distance measure between objects with set-valued features, the SV-k-modes algorithm for clustering a set of set-valued objects $\mathbf{X} = \{X_1, X_2, \cdots, X_n\}$ into $k(\ll n)$ clusters minimizes the following objective function:

$$F'(W, Q) = \sum_{l=1}^{k} \sum_{i=1}^{n} \omega_{li} D_m(X_i, Q_l)$$

subject to

$$\omega_{li} \in \{0, 1\}, \quad 1 \le l \le k, \ 1 \le i \le n \tag{3}$$

$$\sum_{l=1}^{k} \omega_{li} = 1, \quad 1 \le i \le n \tag{4}$$

$$0 < \sum_{i=1}^{n} \omega_{li} < n, \quad 1 \le l \le k \tag{5}$$

where $W = [\omega_{li}]$ is a k-by-n $\{0, 1\}$ matrix in which $\omega_{li} = 1$ indicates that object X_i is allocated to cluster l and $Q = [Q_1, Q_2, \cdots, Q_k]$, where $Q_l \in Q$ is the set-valued modes of cluster l with m set-valued features.

$F'(W, Q)$ can be solved with an iterative process for solving two subproblems iteratively until the process converges. The first step is to fix $Q = Q^t$ at iteration t and solve the reduced problem $F'(W, Q^t)$ with (1) to find W^t that minimizes $F'(W, Q^t)$. The second step is to fix W^t and solve the reduced problem $F'(W^t, Q)$ using algorithm $GAFSM$ for finding the Q^{t+1} that minimizes $F'(W^t, Q)$. The SV-k-modes algorithm is given in Algorithm 2.

Algorithm 2 SV-k-Modes Algorithm

Input:
- $\mathbf{X}$: a set of n set-valued objects;
- k : the number of clusters;

Output: $\{C_1, C_2, \cdots, C_k\}$, a set of k clusters;
Method:
Step 1. Randomly choose k objects as $Q^{(1)}$. Determine $W^{(1)}$ such that $F'(W, Q^{(1)})$ is minimized with (1). Set $t = 1$.
Step 2. Determine $Q^{(t+1)}$ such that $F'(W^{(t)}, Q^{(t+1)})$ is minimized with *Algorithm* 1. If $F'(W^{(t)}, Q^{(t+1)}) = F'(W^{(t)}, Q^{(t)})$, then stop; otherwise, goto step 3.
Step 3. Determine $W^{(t+1)}$ such that $F'(W^{(t+1)}, Q^{(t+1)})$ is minimized. If $F'(W^{(t+1)}, Q^{(t+1)}) = F'(W^{(t)}, Q^{(t+1)})$, then stop; otherwise, set $t = t + 1$ and goto step 2.

The computational complexity of the SV-k-modes algorithm is analyzed as follows.

1) The computational complexity for the calculation of the distance between two objects on feature A_j is $\mathcal{O}(|V^j|)$. The computational complexity of the calculation of the distance between two objects in m features is $\mathcal{O}(m \times |V'|)$, where $|V'| = \max\{|V^j||1 \le j \le m\}$.
2) Updating cluster centers. The main goal of updating cluster centers is to find the set-valued modes in each cluster according to the partition matrix W. The computational complexity for this step is $\mathcal{O}(km \times 2^{|V'|})$, where $|V'| = \max\{|V^j||1 \le j \le m\}$.

If the clustering process needs t iterations to converge, the total computational complexity of the SV-k-modes algorithm

is $\mathcal{O}(nmtk \times 2^{|V'|})$, where $|V'| = \max\{|V^j||1 \le j \le m\}$. It is obvious that the time complexity of the proposed algorithm increases linearly as the number of objects, features, or clusters increases.

Theorem 1: The SV-k-modes algorithm converges to a local minimal solution in a finite number of iterations.

Proof: We note that the number of possible values for the center of a cluster is $N = \Pi_{j=1}^{m} \sum_{r_j=1}^{|V^j|} C_{|V^j|}^{r_j}$, where $|V^j|$ is the number of unique values in A_j and $C_{|V^j|}^{r_j}$ is the number of combinations in choosing r_j values from a set of $|V^j|$ values. When dividing a data set into k clusters, the number of possible partitions is finite. We can show that each possible partition only occurs once in the clustering process. Let W^h be a partition at iteration h. We can obtain the Q^h that depends on W^h.

Suppose that $W^{h_1} = W^{h_2}$, where h_1 and h_2 are two different iterations, i.e., $h_1 \neq h_2$. If Q^{h_1} and Q^{h_2} are obtained from W^{h_1} and W^{h_2}, respectively, then $Q^{h_1} = Q^{h_2}$ because $W^{h_1} = W^{h_2}$. Therefore, we have

$$F'(W^{h_1}, Q^{h_1}) = F'(W^{h_2}, Q^{h_2}).$$

However, the value of the objective function $F'(\cdot, \cdot)$ generated by the SV-k-modes algorithm is strictly decreasing. h_1 and h_2 must be two consecutive iterations in which the clustering result is no longer changing and the clustering process converges. Therefore, the SV-k-modes algorithm converges in a finite number of iterations. ■

III. Heuristic Method for Updating Cluster Centers

The *GAFSM* algorithm for finding cluster centers is not efficient if the cluster is large and if the number of unique values in the set-valued features is large. In this section, we propose a heuristic method that is used to update the cluster centers in the *SV-k*-modes clustering process. This method constructs a cluster center Q with a subset of values in V^j with the highest frequency in the cluster of objects $\mathbf{X}$, and this Q results in a small value of (2). This heuristic method increases the efficiency of updating cluster centers in the *SV-k*-modes algorithm.

Definition 3: Let $V^j = \{q_1^j, q_2^j, \cdots, q_{r_j'}^j\}$ be r_j' distinct values of A_j appearing in the cluster of objects $\mathbf{X}$, and let S^j be a subset of V^j. The probability-based frequency of S^j is defined as

$$f(S^j) = \frac{1}{n} \sum_{i=1}^{n} \nu\left(S^j, V_{X_i}^{A_j}\right) \tag{6}$$

where n is the number of objects in $\mathbf{X}$ and

$$\nu\left(S^j, V_{X_i}^{A_j}\right) = \begin{cases} \dfrac{|S^j|}{|V_{X_i}^{A_j}|}, & \text{if } S^j \subseteq V_{X_i}^{A_j}. \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Theorem 2: Let $\mathbf{X}$ be a set of n set-valued objects, and let A_j be a feature with a value set $V^j = \{q_1^j, q_2^j, \cdots, q_{r_j'}^j\}$. Suppose that $Q_{A_j} = \{q_1^j\}$ is a subset of V^j. Q_{A_j} minimizes $F(\mathbf{X}, Q_{A_j})$ of (2) if $f(\{q_1^j\}) \ge f(\{q_t^j\})$, where $t \in \{2, 3 \cdots, r_j'\}$.

Proof: To minimize $F(\mathbf{X}, Q_{A_j}) = \sum_{i=1}^{n}(1 - (|V_{X_i}^{A_j} \cap Q_{A_j}|/|V_{X_i}^{A_j} \cup Q_{A_j}|)) = n - \sum_{i=1}^{n}(|V_{X_i}^{A_j} \cap Q_{A_j}|/|V_{X_i}^{A_j} \cup Q_{A_j}|)$, we only need to maximize $\sum_{i=1}^{n}(|V_{X_i}^{A_j} \cap Q_{A_j}|/|V_{X_i}^{A_j} \cup Q_{A_j}|)$. With (6), we have

$$\begin{aligned} \sum_{i=1}^{n} \frac{\left|V_{X_i}^{A_j} \cap Q_{A_j}\right|}{\left|V_{X_i}^{A_j} \cup Q_{A_j}\right|} &= \sum_{i=1}^{n} \frac{\left|V_{X_i}^{A_j} \cap \{q_1^j\}\right|}{\left|V_{X_i}^{A_j} \cup \{q_1^j\}\right|} \\ &= \sum_{i=1}^{n} \frac{\left|V_{X_i}^{A_j} \cap \{q_1^j\}\right|}{\left|V_{X_i}^{A_j}\right|} \\ &= nf\left(\{q_1^j\}\right). \end{aligned}$$

Given $Q'_{A_j} = \{q_t^j\} \neq Q_{A_j}$, we have

$$\sum_{i=1}^{n} \frac{\left|V_{X_i}^{A_j} \cap Q'_{A_j}\right|}{\left|V_{X_i}^{A_j} \cup Q'_{A_j}\right|} = nf\left(\{q_t^j\}\right).$$

Because $f(\{q_1^j\}) \ge f(\{q_t^j\})$, $F(\mathbf{X}, Q_{A_j}) \le F(\mathbf{X}, Q'_{A_j})$. ■

Lemma 1: Let A and B be two finite sets, and let $B = \{q_1, q_2, \cdots, q_n\}$. We have

$$|A \cap B| = |A \cap \{q_1\}| + |A \cap \{q_2\}| + \cdots + |A \cap \{q_n\}|. \tag{8}$$

Proof: By the inclusion–exclusion principle [8], we have

$$\begin{aligned} |A \cap B| &= |A \cap \{q_1, q_2, \cdots, q_n\}| \\ &= |(A \cap \{q_1\}) \cup (A \cap \{q_2\}) \cup \cdots \cup (A \cap \{q_n\})| \\ &= |(A \cap \{q_1\})| + |(A \cap \{q_2\})| + \cdots + |(A \cap \{q_n\})| \\ &\quad + (-1)^{2-1} \sum_{h=1}^{n} \sum_{t>h}^{n} |(A \cap \{q_h\}) \cap (A \cap \{q_t\})| + \cdots \\ &\quad + (-1)^{n-1} |(A \cap \{q_1\}) \cap (A \cap \{q_2\}) \cap \cdots \\ &\quad \times \cap (A \cap \{q_n\})|. \end{aligned}$$

Because $q_1 \neq q_2 \neq \cdots \neq q_n$, we have $(A \cap \{q_h\}) \cap (A \cap \{q_t\}) = \emptyset$ for $1 \le h, t \le n$, and $h \neq t$. Thus, we have

$$|A \cap B| = |A \cap \{q_1\}| + |A \cap \{q_2\}| + \cdots + |A \cap \{q_n\}|.$$

■

Theorem 3: Let $\mathbf{X}$ be a set of n set-valued objects, and let A_j be feature with the value set $V^j = \{q_1^j, q_2^j, \cdots, q_{r_j'}^j\}$. Suppose that $Q_{A_j} = \{q_1^j, q_2^j, \cdots, q_{r_j}^j\}$ is a subset of V^j and that $Q_{A_j} \subseteq V_{X_i}^{A_j}$. Q_{A_j} minimizes $F(\mathbf{X}, Q_{A_j})$ if $f(\{q_1^j\}) \ge f(\{q_2^j\}) \ge \cdots \ge f(\{q_{r_j}^j\}) > f(\{q_{r_j+1}^j\}) \ge \cdots \ge f(\{q_{r_j'}^j\})$.

Proof:

To minimize $F(\mathbf{X}, Q_{A_j}) = \sum_{i=1}^{n}(1 - (|V_{X_i}^{A_j} \cap Q_{A_j}|/|V_{X_i}^{A_j} \cup Q_{A_j}|)) = n - \sum_{i=1}^{n}(|V_{X_i}^{A_j} \cap Q_{A_j}|/|V_{X_i}^{A_j} \cup Q_{A_j}|)$, we only need to maximize

$\sum_{i=1}^{n}(|V_{X_i}^{A_j} \cap Q_{A_j}|/|V_{X_i}^{A_j} \cup Q_{A_j}|)$. With Lemma 1 and (6), we have

$$\begin{aligned}
&\sum_{i=1}^{n} \frac{\left|V_{X_i}^{A_j} \cap Q_{A_j}\right|}{\left|V_{X_i}^{A_j} \cup Q_{A_j}\right|} \\
&= \sum_{i=1}^{n} \frac{\left|V_{X_i}^{A_j} \cap \{q_1^j\}\right| + \left|V_{X_i}^{A_j} \cap \{q_2^j\}\right| + \cdots + \left|V_{X_i}^{A_j} \cap \{q_{r_j}^j\}\right|}{\left|V_{X_i}^{A_j} \cup Q_{A_j}\right|} \\
&= \sum_{i=1}^{n} \frac{\left|V_{X_i}^{A_j} \cap \{q_1^j\}\right| + \left|V_{X_i}^{A_j} \cap \{q_2^j\}\right| + \cdots + \left|V_{X_i}^{A_j} \cap \{q_{r_j}^j\}\right|}{\left|V_{X_i}^{A_j}\right|} \\
&= n\big(f(\{q_1^j\}) + f(\{q_2^j\}) + \cdots + f(\{q_{r_j}^j\})\big).
\end{aligned}$$

Given $Q'_{A_j} = \{q_{s1}^j, q_{s2}^j, \cdots, q_{sr_j}^j\} \neq Q_{A_j}$ and $Q'_{A_j} \subseteq V_{X_i}^{A_j}$, we have

$$\begin{aligned}
\sum_{i=1}^{n} \frac{\left|V_{X_i}^{A_j} \cap Q'_{A_j}\right|}{\left|V_{X_i}^{A_j} \cup Q'_{A_j}\right|} = n\big(&f(\{q_{s1}^j v\}) + f(\{q_{s2}^j\}) \\
&+ \cdots + f(\{q_{sr_j}^j\})\big).
\end{aligned}$$

Because $f(\{q_1^j\}) + f(\{q_2^j\}) + \cdots + f(\{q_{r_j}^j\}) > f(\{q_{s1}^j\}) + f(\{q_{s2}^j\}) + \cdots + f(\{q_{sr_j}^j\})$, $F(\mathbf{X}, Q_{A_j}) < F(\mathbf{X}, Q'_{A_j})$. ■

Using Definition 2 and Theorems 2 and 3, we can construct Q_{A_j} from the set of r'_j unique values in feature A_j. We first compute the frequency $f(\{q_h^j\})$ of all categorical values in feature A_j from cluster $\mathbf{X}$ and rank the categorical values in descending order of $f(\{q_h^j\})$ in set $V^j = \{q_1^j, q_2^j, \cdots, q_{r'_j}^j\}$. Assume that Q_{A_j} has r_j values. We consider three situations to construct Q_{A_j}.

1) When $r_j = 1$, we choose the most frequent categorical value $\{q_1^j\}$ for Q_{A_j} according to Theorem 2. If there is more than one most frequent categorical value, we randomly choose one value for Q_{A_j}.
2) When $r_j = r'_j$, we choose all categorical values in A_j for Q_{A_j} as the center of the cluster.
3) When $1 < r_j < r'_j$, we have the following three cases.
   Case 1: $f(\{q_1^j\}) \geq f(\{q_2^j\}) \geq \cdots \geq f(\{q_{r_j}^j\}) > f(\{q_{r_j+1}^j\}) \geq \cdots \geq f(\{q_{r'_j}^j\})$. We choose the first r_j most frequent categorical values for Q_{A_j} according to Theorem 3.
   Case 2: $f(\{q_1^j\}) \geq f(\{q_2^j\}) \geq \cdots > f(\{q_{r_j}^j\}) = f(\{q_{r_j+1}^j\}) > \cdots \geq f(\{q_{r'_j}^j\})$. We first choose the first $r_j - 1$ most frequent values $Q' = \{q_1^j, q_2^j, \cdots, q_{r_j-1}^j\}$ as part of values for Q_{A_j}. If $\sum_{i=1}^{r_j-1} f(\{q_i^j, q_{r_j}^j\}) > \sum_{i=1}^{r_j-1} f(\{q_i^j, q_{r_j+1}^j\})$, we choose $\{q_{r_j}^j\}$ as the r_jth value for Q_{A_j}, i.e., $Q_{A_j} = \{q_{r_j}^j\} \cup Q'$. If $\sum_{i=1}^{r_j-1} f(\{q_i^j, q_{r_j}^j\}) < \sum_{i=1}^{r_j-1} f(\{q_i^j, q_{r_j+1}^j\})$, we choose $Q_{A_j} = \{q_{r_j+1}^j\} \cup Q'$. If $\sum_{i=1}^{r_j-1} f(\{q_i^j, q_{r_j}^j\}) = \sum_{i=1}^{r_j-1} f(\{q_i^j, q_{r_j+1}^j\})$, we choose either $Q_{A_j} = \{q_{r_j}^j\} \cup Q'$ or $Q_{A_j} = \{q_{r_j+1}^j\} \cup Q'$.
   Case 3: $f(\{q_1^j\}) \geq f(\{q_2^j\}) \geq \cdots > f(\{q_{r_j-p'}^j\}) = \cdots = f(\{q_{r_j}^j\}) = f(\{q_{r_j+1}^j\}) = \cdots = f(\{q_{r_j+p}^j\}) > f(\{q_{r_j+p+1}^j\}) \geq \cdots \geq f(\{q_{r'_j}^j\})$, where p' and p are two integers. We choose the first $(r_j - p' - 1)$ most frequent categorical values as $Q' = \{q_1^j, q_2^j, \cdots, q_{r_j-p'-1}^j\}$. Assume that Q^j is the set of all combinations of $p'+1$ categorical values from the next $p'+p+1$ categorical values. Let Q_s be a combination in Q^j that produces the largest sum of frequencies, i.e., $\sum_{i=1}^{r_j-p'-1} f(\{q_i^j\} \cup Q_s) \geq \sum_{i=1}^{r_j-p'-1} f(\{q_i^j\} \cup Q_t)$, where Q_t is any combination in Q^j and $Q_t \neq Q_s$. We choose Q_s as the remaining values for Q_{A_j}, i.e., $Q_{A_j} = Q_s \cup Q'$.

The cluster center Q_{A_j} constructed with the above methods from a given set of set-valued objects $\mathbf{X}$ results in a smaller value of $F(\mathbf{X}, Q_{A_j})$ of (2). Therefore, we use this heuristic method to update the set-valued modes in the set-valued k-modes clustering process, which is more efficient than the exhaustive method for transversing all possible values to update the set-valued modes. To further reduce the computational complexity of updating the set-valued modes, we fix the number of categorical values in Q_{A_j} as $r_j = round(\sum_{i=1}^{n}(|V_{X_i}^{A_j}|/n))$. For $r_j = 1$, this case is equivalent to the k-modes algorithm. The algorithm using the heuristic method to update the set-valued modes is given in Algorithm 3. The name of the algorithm, *HAFSM*, is the abbreviation for heuristic algorithm of finding set-valued modes.

IV. Method for Obtaining Initial Cluster Centers

Because the SV-k-modes algorithm is sensitive to the initial cluster centers, the choice of appropriate initial cluster centers has a direct impact on the final clustering result. In this section, we propose an algorithm for selecting the initial cluster centers for the SV-k-modes algorithm. This algorithm is an extension of a previous initialization method [9] used to obtain the initial cluster centers for the k-modes algorithm.

Definition 4: Let $\mathbf{X} = \{X_1, X_2, \cdots, X_n\}$ be a set of n set-valued objects with m set-valued features. For any $X_i \in \mathbf{X}$, the density of X_i is defined as

$$Dens(X_i) = \frac{1}{n} \sum_{j=1}^{m} \sum_{p=1}^{n} \frac{\left|V_{X_i}^{A_j} \cap V_{X_p}^{A_j}\right|}{\left|V_{X_i}^{A_j} \cup V_{X_p}^{A_j}\right|}. \tag{9}$$

$Dens(X_i)$ is a measure of the number of objects in the neighborhood of X_i. A larger value of $Dens(X_i)$ denotes a higher number of objects in the neighborhood of X_i. With this measure, we can select the objects with large values of $Dens(X_i)$ as the candidates for the initial cluster centers. Among the candidates, we compute the mutual distances of these objects and select the candidates with large mutual distances as the initial cluster centers. The generating initial cluster centers algorithm (GICCA) is shown in Algorithm 4.

In finding a set of initial cluster centers, finding the first initial cluster center has a computational complexity of $\mathcal{O}(n^2 m|V'|)$, where $|V'| = \max\{|V^j||1 \leq j \leq m\}$.

Algorithm 3 *HAFSM*

Input:
- $\mathbf{X} = \{X_1, X_2, \cdots, X_n\}$: the set of n set-valued objects;
- m : the number of features;

Output: The set-valued modes Q;

Method:

$Q = \emptyset$;
for $j = 1$ *to* m **do**
  Obtain the domain values V^j of the jth feature;
  Compute the probability-based frequency of each domain value according to (6) and arrange the categorical values of $V^j = \{q_1^j, q_2^j, \cdots, q_{|V^j|}^j\}$ in descending order of the probability-based frequency;
  $r = round(\sum_{i=1}^{n}(|V_{X_i}^{A_j}|/n))$;
  if $r = 1$ **then**
    Select the most frequent value as the jth component of Q; if there is more than one most frequent categorical value, we arbitrarily select one value as the jth component of Q;
  end if
  if $r > 1$ and $f(\{q_r^j\}) > f(\{q_{r+1}^j\})$ **then**
    Select the first r values $\{q_1^j, q_2^j, \cdots, q_r^j\}$ as the jth component of Q;
  end if
  if $r > 1$ and $f(\{q_{r-p'}^j\}) = f(\{q_{r-p'+1}^j\}) = \cdots = f(\{q_r^j\}) = \cdots = f(\{q_{r+p}^j\})$ **then**
    Select $p' + 1$ values from $\{q_{r-p'}^j, q_{r-p'+1}^j, \cdots, q_r^j, q_{r+p}^j\}$, generate $c_{r-p'+1+p}^{p'+1}$ types of combinations by the binomial theorem.
    Take the first $r-p'-1$ most frequent categorical values as Q';
    For each combination, form $r - p' - 1$ pairs with each value of Q';
    Compute the sum of the probability-based frequencies of $r - p' - 1$ pairs for each combination using (6).
    Take the combination with the largest sum and Q' as the jth component of Q;
  end if
end for
return Q;

Algorithm 4 *GICCA*

Input:
- $\mathbf{X}$: a set of n set-valued objects;
- k : the number of clusters desired;

Output: k objects;

Method:

Step 1: $Centers = \emptyset$;
Step 2: For each $X_i \in \mathbf{X}$, calculate the $Dens(X_i)$, $Centers = Centers \cup \{X_{i_1}\}$, where X_{i_1} satisfies $Dens(X_{i_1}) = \max\{Dens(X_i)|1 \le i \le n\}$, and the first cluster center is selected;
Step 3: Find the second cluster center, $Centers = Centers \cup \{X_{i_2}\}$, where X_{i_2} satisfies $D_m(X_{i_2}, X_m) \times Dens(X_{i_2}) = \max\{D_m(X_i, X_m) \times Dens(X_i)|X_m \in Centers, 1 \le i \le n\}$;
Step 4: If $|Centers| < k$, then goto Step 5; otherwise, goto Step 6;
Step 5: For any $X_i \in \mathbf{X}$, $Centers = Centers \cup \{X_{i_3}\}$, where X_{i_3} satisfies $D_m(X_{i_3}, X_m) \times Dens(X_{i_3}) = \max\{\min_{X_m \in Centers}\{D_m(X_i, X_m) \times Dens(X_i)\}|X_i \in \mathbf{X}\}$, goto Step 4;
Step 6: Return $Centers$;

Finding the remaining initial cluster centers has a computational complexity of $\mathcal{O}(nmk|V'|)$, where $|V'| = \max\{|V^j||1 \le j \le m\}$. The total computational complexity of *GICCA* is $\mathcal{O}(n^2m|V'|)$, where $|V'| = \max\{|V^j||1 \le j \le m\}$.

Example 1: $\mathbf{X}$ has four objects, X_1, X_2, X_3, and X_4, each described by one feature A_1, where $V_{X_1}^{A_1} = \{a, b, e\}$, $V_{X_2}^{A_1} = \{a, d, e\}$, $V_{X_3}^{A_1} = \{a, b, c, d\}$, and $V_{X_4}^{A_1} = \{a, b, c\}$. We suppose that $\mathbf{X}$ can be divided into two clusters. The two initial cluster centers can be computed as follows. According to Definition IV, we have that

$$\begin{aligned} &Dens(X_1) \\ &= \frac{1}{4}\left(\frac{|\{a,b,e\} \cap \{a,b,e\}|}{|\{a,b,e\} \cup \{a,b,e\}|} + \frac{|\{a,b,e\} \cap \{a,d,e\}|}{|\{a,b,e\} \cup \{a,d,e\}|}\right. \\ &\quad \left. + \frac{|\{a,b,e\} \cap \{a,b,c,d\}|}{|\{a,b,e\} \cup \{a,b,c,d\}|} + \frac{|\{a,b,e\} \cap \{a,b,c\}|}{|\{a,b,e\} \cup \{a,b,c\}|}\right) \\ &= \frac{1}{4}\left(1 + \frac{2}{4} + \frac{2}{5} + \frac{2}{4}\right) \\ &= \frac{48}{80}. \end{aligned}$$

Similarly, we can obtain $Dens(X_2) = (42/80)$, $Dens(X_3) = (51/80)$ and $Dens(X_4) = (49/80)$.

Therefore, X_3 can be taken as the first initial cluster center. For the second initial cluster center, we have that

$$\begin{aligned} D_m(X_3, X_1) \times Dens(X_1) &= \frac{3}{5} \times \frac{48}{80} = \frac{150}{400} = 0.3600 \\ D_m(X_3, X_2) \times Dens(X_2) &= \frac{3}{5} \times \frac{42}{80} = \frac{126}{400} = 0.3150 \\ D_m(X_3, X_4) \times Dens(X_4) &= \frac{1}{4} \times \frac{49}{80} = \frac{49}{320} = 0.1531. \end{aligned}$$

Thus, X_1 is taken as the second initial cluster center.

V. Experiments on Real Data

In this section, we present experiment results on five real data sets from different applications to show the effectiveness and efficiency of the SV-k-modes algorithm. We first discuss the preprocessing methods that are used to convert the real data into the set-valued representation. Then, we present five external indices used for evaluating clustering algorithms. Finally, we show the comparison results for the SV-k-modes algorithm against other algorithms on the five real data sets.

A. Data Preprocessing

The five publicly available real data sets are not in the set-valued representation, and the SV-k-modes algorithm cannot directly cluster these data sets in their original formats.

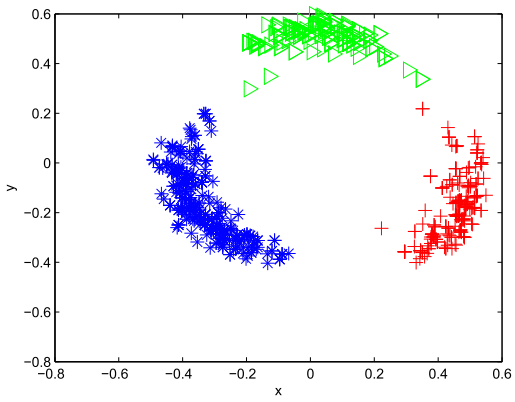


Fig. 1. Distribution of Market Basket data.

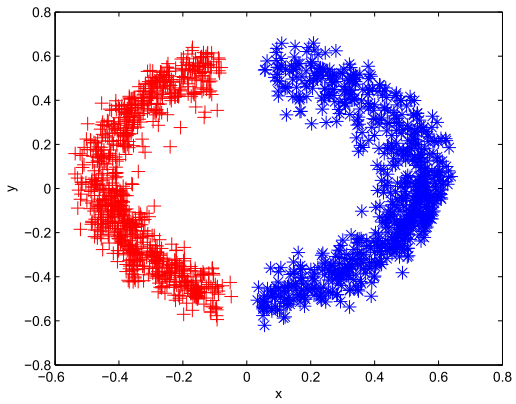


Fig. 2. Distribution of Microsoft Web data.

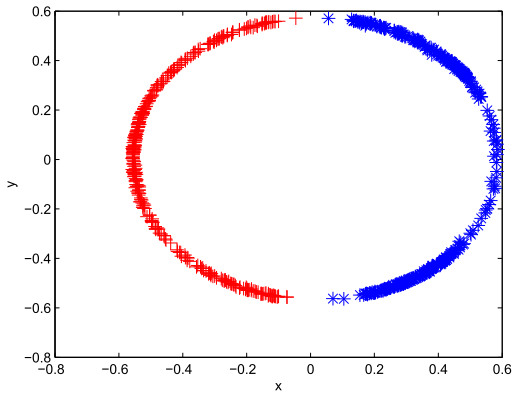


Fig. 3. Distribution of the Alibaba data.

We conducted a series of data preprocessing steps on each data set to convert them into the set-valued representation. These preprocessing steps on each data set are elaborated upon below.

1) Musk Data: Musk data used for drug activity prediction [10] were downloaded from University of California, Irvine (UCI) [11]. The Musk data are given in two data sets: Musk1 and Musk2. We only used Musk1, which contains a set of molecules. Each molecule has different shapes or conformations, which are described by 166 numerical features. Each shape or conformation of a molecule is represented as a single record in the data set. Some molecules have only one shape record, whereas some have as many as 1044 records. Therefore, each molecule can be treated as a set-valued object. The molecules in the Musk data were labeled by human experts into two classes: *musks* and *nonmusks*. In the experiment, we considered that the Musk data could be clustered into two clusters.

2) Market Basket Data: Market Basket data were downloaded from a website.[1] These data have been frequently used to evaluate association rule algorithms. The Market Basket data set contains transactions of 1001 customers, each having at most seven transaction records. The transaction records have four attributes or features: Customer_Id, Transaction_Time, Product_Name, and Product_Id. We deleted Transaction_Time because the time value was the same for all records. Product_Id and Product_Name represent the same feature; therefore, we only kept Product_ID. After removing Transaction_Time and Product_Name from the data set, we converted the Market Basket data into a set of 1001 set-valued objects (customers) with one set-valued feature: Product_ID. Then, we used the set-valued distance in (1) to compute the mutual distances between customers and the multidimensional scaling technique [12] to compute two coordinate values x and y for each customer from the mutual distance matrix. Fig. 1 shows the distribution of the 1001 customers in the 2-D space. We can observe that the 1001 customers can be divided into 3 clusters. In the experiments, we considered the number of clusters in this data set as 3.

3) Microsoft Web Data: The Microsoft Web data were downloaded from UCI [11]. The data set records the areas (Vroots) of www.microsoft.com where users visited during a week in February 1998. Each record has two features: User_Id and Vroots. If a user visited several areas of the website, the user had several records. Therefore, users are set-valued objects, and Vroots is a set-valued feature. Similarly, we computed the distance matrix of the users using (1), and we computed two coordinates x and y of the users from the distance matrix using the multidimensional scaling technique. Fig. 2 shows the distribution of users in the 2-D space. We can see that this data set has two clusters.

4) Alibaba Data: These data were downloaded from a website.[2] This data set was provided by Alibaba for a big data competition in 2014. The Alibaba data set contains records of User_Id, Time, Action_type, and Brand_Id. Each record describes a user who visited a brand at a given time and took a specific action. In these data, we only considered the User_Id and Brand_Id features, i.e., each user was interested in a particular set of brands. Therefore, users are set-valued objects, and Brand is a set-valued feature. With the same technique, we plotted the distribution of the users in a 2-D space as shown in Fig. 3. We can see that the Alibaba data set has two clusters.

5) MovieLens Data: The MovieLens data were downloaded from the MovieLens website.[3] The data contain four tables about rating information, user information, movie information,

[1] http://www.datatang.com/datares/go.aspx?dataid=613168

[2] http://102.alibaba.com/competition/addDiscovery/index.htm

[3] http://grouplens.org/data sets/movielens/

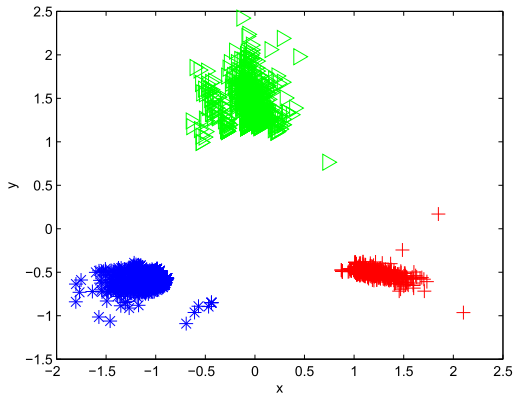


Fig. 4. Distribution of MovieLens data.

TABLE II
SUMMARY OF REAL SET-VALUED DATA SETS

Data set	Objects	Features	Records	k
Musk1	92	166	476	2
Market Basket	703	2	4921	3
Microsoft Web	962	2	9857	2
Alibaba	745	2	159661	2
MovieLens	2306	6	2306	3

and tag information. We only used the first three tables. The data were divided into three different sizes: MovieLens 100 k, MovieLens 1 M, and MovieLens 10 M. We chose the MovieLens 1 M data to evaluate the SV-k-modes algorithm.

The rating table contains 6040 users, who rated approximately 3900 movies. The rating was on a 5-star scale. Each user has at least 20 rating records. The rating table has 1 000 209 rating records with four features: User_Id, Movie_Id, Rating, and Timestamp. The movie table has three features: Movie_Id, Title, and Genres. Movie_Id and Genres have a one-to-many relationship, i.e., each movie has several genre values.

The user table has User_Id and other demographic features, such as Gender, Age, Occupation, and Zip-code, which are categorical features. Age contains seven categories corresponding to age ranges. Occupation has 21 distinct categorical values.

The rating table was first joined with the movie table on Movie_Id. Then, the joined table was further joined with the user table on User_Id to create a final table with eight features: User_Id, Gender, Age, Occupation, Zip-code, Genres, Rating, and Timestamp. Among them, User_Id, Gender, Age, Occupation, Zip-code, and Timestamp are single-valued features, and Genres and Rating are set-valued features. The final table possesses 6040 set-valued objects (users). In the experiments, Zip-code and Timestamp were removed because they took on too many different values. We took a sample of 2306 objects and computed their 2-D coordinates. Fig. 4 shows the distribution of the 2306 objects. We can observe that the sample data have three clusters.

The final data sets from the five real data sets are listed in Table II. These set-valued data sets were used in the experiments to evaluate the SV-k-modes algorithm.

TABLE III
CONTINGENCY TABLE

	C_1	C_2	$\cdots$	$C_{k'}$	$Sums$
P_1	n_{11}	n_{12}	$\cdots$	$n_{1k'}$	p_1
P_2	n_{21}	n_{22}	$\cdots$	$n_{2k'}$	p_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
P_k	n_{k1}	n_{k2}	$\cdots$	$n_{kk'}$	p_k
$Sums$	c_1	c_2	$\cdots$	$c_{k'}$	n

B. Measures of Clustering Results

Five measures were used to evaluate the clustering results: 1) the adjusted Rand index (ARI) [13]; 2) the normalized mutual information (NMI) [14]; 3) accuracy (ac); 4) precision (PE); and 5) recall (RE). These measures are defined as follows.

Let $\mathbf{X}$ be a categorical set-valued data set, let $C = \{C_1, C_2, \cdots, C'_k\}$ be the set of clusters of $\mathbf{X}$ generated by a clustering algorithm, and let $P = \{P_1, P_2, \cdots, P_k\}$ be the set of the true classes of $\mathbf{X}$. The intersections of clusters and classes are summarized in the contingency table shown in Table III, where n_{ij} denotes the number of objects in common between P_i and C_j: $n_{ij} = |P_i \cap C_j|$. p_i and c_j are the numbers of objects in P_i and C_j, respectively.

The five evaluation measures are calculated from the contingency table as follows:

$$ARI = \frac{\sum_{ij} C_{n_{ij}}^2 - \left[\sum_i C_{p_i}^2 \sum_j C_{c_j}^2\right]/C_n^2}{\frac{1}{2}\left[\sum_i C_{p_i}^2 + \sum_j C_{c_j}^2\right] - \left[\sum_i C_{p_i}^2 \sum_j C_{c_j}^2\right]/C_n^2}$$

$$NMI = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k'} n_{ij} log\left(\frac{n_{ij}n}{p_i c_j}\right)}{\sqrt{\sum_{i=1}^{k} p_i log\left(\frac{p_i}{n}\right) \sum_{j=1}^{k'} c_j log\left(\frac{c_j}{n}\right)}}$$

$$ac = \frac{1}{n} \max_{j_1 \, j_2 \cdots j_k \in S} \sum_{i=1}^{k} n_{ij_i}$$

$$PE = \frac{1}{k}\sum_{i=1}^{k} \frac{n_{ij_i^*}}{p_i}$$

$$RE = \frac{1}{k'}\sum_{i=1}^{k'} \frac{n_{ij_i^*}}{c_i}$$

where $n_{1j_1^*} + n_{2j_2^*} + \cdots + n_{kj_k^*} = \max_{j_1 \, j_2 \cdots j_k \in S} \sum_{i=1}^{k} n_{ij_i}$ $(j_1^* j_2^* \cdots j_k^* \in S)$ and $S = \{j_1 j_2 \cdots j_k | j_1, j_2, \cdots, j_k \in \{1, 2, \cdots, k\}, j_i \neq j_t \text{ for } i \neq t\}$ is a set of all permutations of $1, 2, \cdots, k$. In these experiments, we let $k = k'$, i.e., the number of clusters to be found was equal to the number of classes in the data set. Larger values of ARI, NMI, ac, PE, and RE indicate better clustering results.

TABLE IV
COMPARISON RESULTS OF *GAFSM* AND *HAFSM* ON MARKET BASKET DATA

	AC	PE	RE	ARI	NMI
SV-k-modes+*GICCA*+*GAFSM*	0.8990	0.9345	0.8781	0.7044	0.7021
SV-k-modes+*GICCA*+*HAFSM*	0.9232	0.9504	0.9101	0.7672	0.7663
SV-k-modes+*Random*+*GAFSM*	0.7950 $\pm$ 0.1426	0.8113 $\pm$ 0.1348	0.7589 $\pm$ 0.1629	0.5751 $\pm$ 0.2779	0.5912 $\pm$ 0.2319
SV-k-modes+*Random*+*HAFSM*	0.8752 $\pm$ 0.1173	0.8879 $\pm$ 0.1117	0.8592 $\pm$ 0.1306	0.7155$\pm$ 0.2166	0.7123 $\pm$ 0.1745

TABLE V
RUNTIME OF THE SV-k-MODES ALGORITHM WITH *GAFSM* AND *HAFSM* ON MARKET BASKET DATA

	Run-time (Second)
SV-k-modes +*Random*+*GAFSM*	1.2227 $\times 10^5$ $\pm$ 3604
SV-k-modes +*Random*+*HAFSM*	3.7637 $\pm$ 1.5379

C. Comparisons of Two Cluster Center Update Algorithms: GAFSM and HAFSM

In this section, we show the performance comparison results of the cluster center update algorithms *GAFSM* and *HAFSM* used in the SV-k-modes algorithm. Because the exhaustive search algorithm *GAFSM* is very time-consuming, we only used it to cluster the Market Basket data. Table IV shows the results of the clustering performance of the SV-k-modes algorithm with two initial cluster center selection methods, *Random* and *GICCA*, and two cluster center update methods, *GAFSM* and *HAFSM*. In this experiment, each combination of the SV-k-modes algorithm was run 20 times on the data set, except the combination with *GICCA* because the clustering results of the SV-k-modes algorithm with *GICCA* are unique. The values of the five performance measures in Table IV are the mean values and standard deviations of 20 results.

From Table IV, we can see that the combination of SV-k-modes+*GICCA*+*HAFSM* achieved the best performance. Comparing the combinations with *GAFSM* and *HAFSM*, we can see that the SV-k-modes algorithm with *HAFSM* performed much better than the SV-k-modes algorithm with *GAFSM*. This indicates that the *HAFSM* cluster center update method is better than the *GAFSM* cluster center update method. This may imply that cluster centers constructed with *HAFSM* are better representatives of clusters than are the cluster centers found by *GAFSM*.

Comparing the random initial cluster center selection method with *GICCA*, we can see that *GICCA* improved the clustering performance significantly. This indicates that *GICCA* is a necessary step in the SV-k-modes algorithm.

Table V shows the execution time of the SV-k-modes algorithm with the two update methods. We can see that the SV-k-modes algorithm with *GAFSM* was very slow. It took approximately 34 hours to produce one clustering result from the Market Basket data. This is not acceptable in real applications. Therefore, we did not use *GAFSM* in the other experiments. On the other hand, we can see that the SV-k-modes algorithm with *HAFSM* only took a few seconds to produce a clustering result from the same data set. *HAFSM* speeds up the SV-k-modes process tremendously and is a key step in the SV-k-modes algorithm.

Fig. 5 shows the relationship between the accuracy ac of the clustering results and the values of the objective function minimized by the SV-k-modes algorithm with *GAFSM*. From the 20 results, we can see that the clustering accuracy ac is negatively related to the objective function value F'. Specifically, a smaller objective function value F' results in a higher clustering accuracy ac. However, this relationship is not deterministic. Certain larger values of F' result in high clustering accuracy. This may be why *HAFSM* can produce more accurate clustering results than *GAFSM*.

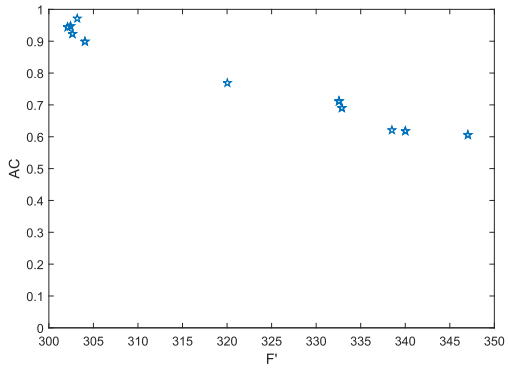


Fig. 5. Relationship between F' and ac from 20 results of Market Basket data.

D. Comparisons of SV-k-Modes With Other Clustering Algorithms

In this section, we compare the clustering results of the SV-k-modes algorithm from the five data sets in Table II with the clustering results of three clustering algorithms: a multi-instance clustering algorithm (BAMIC) [15], k-modes, and Trk-means [2]. Due to the different characteristics of the data sets and algorithms, we present the comparison results separately on different data sets. We also compare the results of the SV-k-modes algorithm with *GICCA* and random initial cluster center selection.

1) Results on Musk1 Data: On this data set, we compared the clustering results of the SV-k-modes algorithm against the results of the BAMIC algorithm [15], which is an extension to the k-medoids algorithm with Hausdorff distances to cluster unlabeled bags. In this experiment, we used three Hausdorff distances, minimal, maximal and average, in BAMIC and both random and *GICCA* initial cluster center selection methods in the SV-k-modes algorithm. A total of 50 runs were

TABLE VI
COMPARISON RESULTS OF *BAMIC* AND SV-k-MODES ALGORITHMS ON MUSK1 DATA

	AC	PE	RE	ARI	NMI
BAMIC(minimal)	0.5304 ± 0.0315	0.5312 ± 0.0332	0.5291 ± 0.0316	-0.0036 ± 0.0179	0.0055 ± 0.0134
BAMIC(maximal)	0.5109 ± 0.0000	0.5143 ± 0.0030	0.5000 ± 0.0000	-0.0100 ± 0.0009	0.0004 ± 0.0004
BAMIC(average)	0.5565 ± 0.0588	0.5605 ± 0.0607	0.5560 ± 0.0612	0.0147 ± 0.0381	0.0200 ± 0.0296
SV-k-modes +*HAFSM*	0.5522 ± 0.0428	0.5547 ± 0.0434	0.5502 ± 0.0428	0.0076 ± 0.0276	0.0131 ± 0.0204
SV-k-modes+*GICCA* +*HAFSM*	0.5870	0.5892	0.5882	0.0196	0.0229

TABLE VII
COMPARISON RESULTS OF k-MODES, Trk-MEANS, AND SV-k-MODES ALGORITHMS ON MARKET BASKET DATA

	AC	PE	RE	ARI	NMI
k-modes	0.7972 ± 0.0900	0.8068 ± 0.0805	0.7450 ± 0.1148	0.5483 ± 0.1888	0.5461 ± 0.1427
Trk-means($\gamma = 0.1$)	0.6212 ± 0.0748	0.6280 ± 0.0845	0.5856 ± 0.0740	0.2540 ± 0.1100	0.2683 ± 0.1076
Trk-means($\gamma = 0.3$)	0.8308 ± 0.1374	0.8365 ± 0.1313	0.8173 ± 0.1484	0.6513 ± 0.2599	0.6632 ± 0.2255
Trk-means($\gamma = 0.5$)	0.9035 ± 0.0801	0.9107 ± 0.0780	0.8756 ± 0.1059	0.7663 ± 0.1692	0.7341 ± 0.1338
SV-k-modes +*GICCA* +*HAFSM*	0.9144	0.9188	0.9037	0.8002	0.7801

TABLE VIII
COMPARISON RESULTS OF k-MODES, Trk-MEANS, AND SV-k-MODES ALGORITHMS ON MICROSOFT WEB DATA

	AC	PE	RE	ARI	NMI
k-modes	0.7521 ± 0.0848	0.7635 ± 0.0826	0.7487 ± 0.0857	0.2813 ± 0.1370	0.2306 ± 0.1097
Trk-means($\gamma = 0.1$)	0.7810 ± 0.1033	0.7902± 0.1011	0.7714 ± 0.1088	0.3567 ± 0.1940	0.2900± 0.1574
Trk-means($\gamma = 0.3$)	0.9172 ± 0.0662	0.9296± 0.0434	0.9151 ± 0.0620	0.7131 ± 0.1555	0.6500± 0.1167
Trk-means($\gamma = 0.5$)	0.8200 ± 0.0952	0.8523± 0.0993	0.8079 ± 0.0954	0.4441 ± 0.1812	0.4358± 0.1790
SV-k-modes +*GICCA* +*HAFSM*	0.9188	0.9339	0.9109	0.7013	0.6508

TABLE IX
COMPARISON RESULTS OF k-MODES, Trk-MEANS, AND SV-k-MODES ALGORITHMS ON ALIBABA DATA

	AC	PE	RE	ARI	NMI
k-modes	0.6620 ± 0.1138	0.8065 ± 0.0391	0.6467 ± 0.1200	0.1501 ± 0.1752	0.2092 ± 0.1557
Trk-means($\gamma = 0.1$)	0.7011 ± 0.0384	0.8065 ± 0.0153	0.7137 ± 0.0368	0.1662 ± 0.0565	0.2831 ± 0.0486
Trk-means($\gamma = 0.3$)	0.7320 ± 0.1154	0.7576 ± 0.1152	0.7379 ± 0.1158	0.2616 ± 0.1673	0.2438 ± 0.1466
SV-k-modes +*GICCA* +*HAFSM*	0.7597	0.8267	0.7696	0.2688	0.3397

conducted for each algorithm, and five evaluation measures were calculated to facilitate an evaluation of the results. Table VI shows the clustering performance mean values and standard deviations of five combinations of two algorithms with respect to five evaluation measures. Each performance value was computed from 50 clustering results.

From Table VI, we can find that SV-k-modes+*GICCA*+*HAFSM* performed significantly better than the other four algorithms. *BAMIC*(average) is slightly better than SV-k-modes+*HAFSM*, which is much better than *BAMIC*(minimal) and *BAMIC*(maximal). These results further demonstrate that *GICCA* is an effective initial cluster center selection method and much better than random selection.

2) Results on Market Basket, Microsoft Web, and Alibaba Data Sets: On these three data sets, we compared the clustering results of the SV-k-modes algorithm with the results of two clustering algorithms: k-modes and Trk-means [2]. We first converted the set-valued features into single-valued features with dummy features for the k-modes algorithm. Because the result of Trk-means is sensitive to the parameter γ, we tested three γ values: 0.1, 0.3, and 0.5. For the SV-k-modes algorithm, we used both *GICCA* and *HAFSM* because these two methods used together produced the best performance of the SV-k-modes algorithm. Each algorithm was run 50 times on each data set. The results in terms of the five evaluation measures on each data set are given in Tables VII, VIII, and IX.

From Tables VII, VIII, and IX, we can see that SV-k-modes+*GICCA*+*HAFSM* performed significantly better than the other algorithms on the three data sets. Because Trk-means could not produce meaningful results with $\gamma = 0.5$ on the Alibaba data set, the results were excluded from Table IX.

3) Results on MovieLens Data: Because the MovieLens data include both single-valued features and set-valued features, Trk-means cannot be applied to these data because the cluster representatives are meaningless on single-valued features. Therefore, we only compared the SV-k-modes algorithm with the k-modes algorithm. Table X shows the results in terms of the five evaluation measures. We can see that the k-modes algorithm performed poorly on these data. The SV-k-modes algorithm with *HAFSM* and random initial centers performed much better than the k-modes algorithm. This indicates that the SV-k-modes algorithm is a good algorithm for data with both single-valued features and set-valued features. After adding *GICCA* to the the

TABLE X
COMPARISON RESULTS OF k-MODES AND SV-k-MODES ALGORITHMS ON MOVIELENS DATA

	AC	PE	RE	ARI	NMI
k-modes	0.5856 ±0.1666	0.6318 ± 0.1591	0.5564 ± 0.1685	0.2600 ± 0.2693	0.3006 ±0.2755
SV-k-modes +*HAFSM*	0.7804± 0.0825	0.7827± 0.0987	0.7470± 0.0949	0.5561 ± 0.1256	0.5449± 0.1148
SV-k-modes+*GICCA* +*HAFSM*	0.8955	0.9126	0.8775	0.7353	0.6979

TABLE XI
SET-VALUED CLUSTER CENTERS FROM A CLUSTERING RESULT OF MOVIELENS DATA BY THE SV-k-MODES ALGORITHM

	Sex	Age	Occupation	Genres	Rating
C_1	female	35	executive/managerial	{1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17}	{3,4,5,2,1}
C_2	female	18	college/grad student	{1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 14, 15, 16, 17}	{4,3,5,2,1}
C_3	female	45	executive/managerial	{1, 2, 4, 5, 6, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18}	{4,3,5,2,1}

TABLE XII
MOVIE CATEGORIES OF THE CODES OF THE GENRES FEATURE IN TABLE XI

Id	meaning	Id	meaning
1	Action	10	Film-Noir
2	Adventure	11	Horror
3	Animation	12	Musical
4	Children's	13	Mystery
5	Comedy	14	Romance
6	Crime	15	Sci-Fi
7	Documentary	16	Thriller
8	Drama	17	War
9	Fantasy	18	Western

SV-k-modes algorithm, the performance was further improved significantly.

To further investigate the clustering results, we list the three cluster centers in Table XI from one cluster result by SV-k-modes+*GICCA*+*HAFSM*. Table XII lists the movie categories of the codes of the Genres feature. With the information shown in the cluster centers, we can give some explanations about user preferences. For example, few people like Documentary films because this category do not appear in the set-valued cluster centers of the Genres feature. Users in cluster C_3 do not like "Animation" films but love "Western" movies because of their age. Users in cluster C_1 enjoy "Film-Noir" movies, but users in the other two groups do not. Although these explanations are not profound, the results indeed reveal that the SV-k-modes algorithm can obtain more interesting information from complex set-valued data than can other clustering algorithms.

Table XIII shows the cluster centers of one clustering result produced by the k-modes algorithm. These cluster centers on the Age, Occupation, and Rating features are not interpretable. Comparatively, the information of the cluster centers by the SV-k-modes algorithm is much richer and more useful.

VI. SCALABILITY STUDIES ON SYNTHETIC DATA

In this section, we present the results of a scalability test of the SV-k-modes algorithm on synthetic data. The algorithm used to generate set-valued synthetic data is proposed, and the scalability of the SV-k-modes algorithm on synthetic data sets is demonstrated.

A. Data Generation Method

Let $\mathbf{X}$ denote a set of n set-valued objects $\{X_1, X_2, \cdots, X_n\}$ to be generated with m set-valued features $\{A_1, A_2, \cdots, A_m\}$, and let V^j be the set of categories for the set value of feature A_j, where $(j = 1, 2, \cdots, m)$. We use the following parameters to generate the synthetic data set $\mathbf{X}$ with k clusters $C = \{C_1, C_2, \cdots, C_k\}$, and we make each cluster identifiable from other clusters:

1) k: the number of clusters to be generated in $\mathbf{X}$;
2) c_i: the number of objects in C_i;
3) ρ: the overlap percentage of feature values between any two clusters.

For simplicity, we make the numbers of categories equal for all V^j, where $j = 1, 2, \cdots, m$. To generate an object X for cluster C_i, we perform the following steps.

1) Construct a set of set values of feature A_j $(j = 1, 2, \cdots, m)$ for cluster C_i with specific parameters ρ, k, and V^j.
2) Randomly select one set value as the value of object X for feature A_j, where $j = 1, 2, \cdots, m$.
3) Repeat Step 2 to generate all set-valued objects for cluster C_i and assign the cluster label to all objects in the cluster.

Repeat the above steps to generate objects for other clusters with different parameters ρ, k, and V^j. The synthetic data generation algorithm generating set-valued data algorithm (*GSDA*) is given in Algorithm 5.

B. Scalability Study

We test the scalability of the SV-k-modes algorithm against changes in the number of objects, the number of features, the number of clusters, and the number of categories of the set-valued features. A total of 10 synthetic data sets were generated with *GSDA*. The parameter ρ was set to 0.5. In each scalability experiment, the same data set was utilized 10 times, and the execution time was the average of 10 runs. The experiments were conducted on a PC with an Intel Xeon i7 CPU (3.4 GHz) and 16 GB of memory. The experimental results are reported based on four experiments below.

Experiment 1: In this experiment, we set $m = 10$, $|V^j| = 10$ $(j = 1, 2, \ldots, m)$, and $k = 2$, and the number of objects was varied from 1000 to 5000 with a step length of 1000.

TABLE XIII
CLUSTER CENTERS FROM A CLUSTERING RESULT OF MOVIELENS DATA BY THE k-MODES ALGORITHM

	Sex		Age							Occupation																						Genres																		Rating				
	0	1	1	18	25	35	45	50	56	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	1	2	3	4	5
C_1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
C_2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	1	0	0	1	0	1	1	1	1	1	1	0	1	1	1	1	1
C_3	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1

Algorithm 5 *GSDA*

Input:
- n : the number of objects in each cluster;
- m : the number of features;
- V^j: the domain values in A_j;
- ρ : the overlap percentage of domain values of each feature in different clusters;
- k : the number of clusters;

Output: A synthetic data set $\mathbf{X}$ with label;
Method:
$\mathbf{X} = \emptyset$;
for $i = 1$ to k **do**
  for $j = 1$ to m **do**
    Uniformly allocate V^j to k clusters $V_1^j, V_2^j, \cdots, V_k^j$;
  end for
end for
for $i = 1$ to k **do**
  for $p = 1$ to n **do**
    for $q = 1$ to m **do**
      Obtain the domain values of the qth features V_i^q in the ith cluster;
      for $h = 1$ to k **do**
        if i!=h **then**
          Compute the number of overlap feature values $rationum = round(|V_h^q| \times \rho)$;
          Randomly select $rationum$ values from V_h^q and add them to V_i^q;
        end if
      end for
      Randomly select r values from V_i^q as the qth component of X;
    end for
    Assign the label i to object X;
    Add X to $\mathbf{X}$;
  end for
end for
return $\mathbf{X}$;

Fig. 6 shows the scalability of the SV-k-modes algorithm against the number of objects. We can see that this algorithm was linearly scalable to the number of objects. Therefore, the SV-k-modes algorithm can efficiently cluster a large number of set-valued objects.

Experiment 2: In this experiment, we set $|\mathbf{X}| = 1000$, $|V^j| = 10$ $(j = 1, 2, \ldots, m)$, and $k = 2$, and the number of set-valued features was changed from 10 to 50 with a step length of 10.

Fig. 7 shows the scalability of the SV-k-modes algorithm against the number of features. The SV-k-modes algorithm

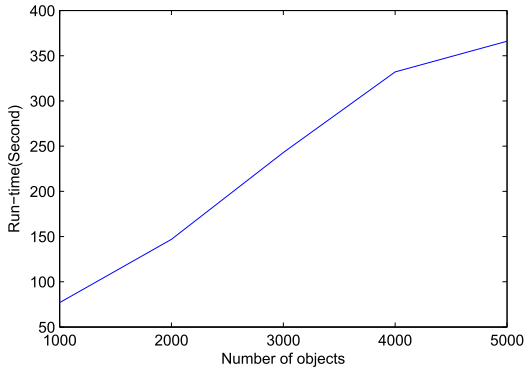


Fig. 6. Scalability of the SV-k-modes algorithm against the number of objects.

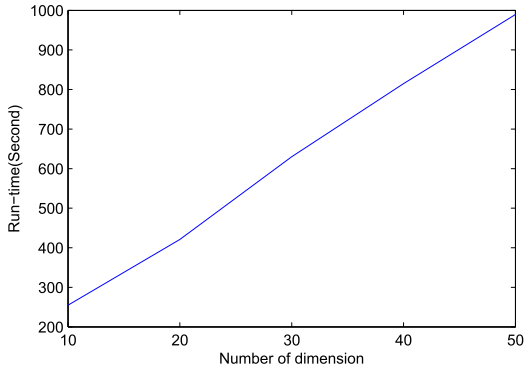


Fig. 7. Scalability of the SV-k-modes algorithm against the number of features.

was linearly scalable to the number of features. Therefore, the SV-k-modes algorithm can efficiently cluster high-dimensional data.

Experiment 3: In this experiment, we set $|\mathbf{X}| = 1000$, $|V^j| = 50$ $(j = 1, 2, \ldots, m)$, and $m = 10$, and the number of clusters was varied from 2 to 10 with a step length of 2.

Fig. 8 shows the scalability of the SV-k-modes algorithm against the number of clusters. We can see that the SV-k-modes algorithm scaled well with the number of clusters.

Experiment 4: In this experiment, we set $|\mathbf{X}| = 1000$, $m = 10$, and $k = 2$, and the number of categories in the set-valued features was varied from 10 to 50 with a step length of 10.

Fig. 9 shows the scalability of the SV-k-modes algorithm against the number of categories in the set-valued features. It can be observed that the runtime of the SV-k-modes algorithm scaled nearly linearly with the number of categories in the set-valued features.

These experiments indicate that the SV-k-modes algorithm with *HAFSM* scales linearly with the numbers of objects, features, clusters, and categories in the set-valued features.

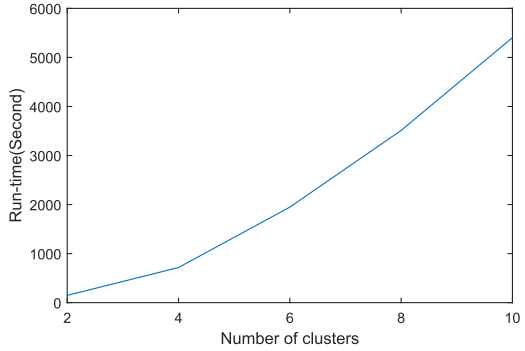


Fig. 8. Scalability of the SV-k-modes algorithm against the number of clusters.

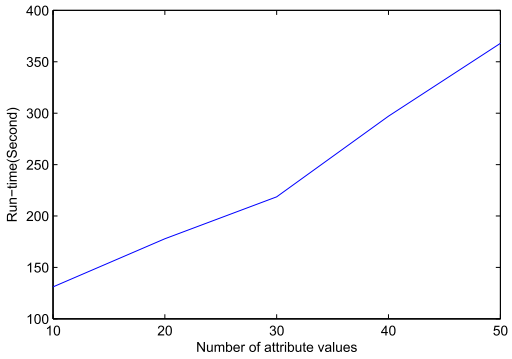


Fig. 9. Scalability of the SV-k-modes algorithm against the number of categories in the set-valued features.

VII. Related Work

In real applications, categorical data are ubiquitous. The k-modes algorithm [7] extends the k-means algorithm [4] using a simple matching dissimilarity measure for categorical objects, modes instead of means for clusters, and a frequency-based method to update modes in the clustering process to minimize the clustering objective function. These extensions have removed the numeric-only limitation of the k-means algorithm and enabled the k-means clustering process to be used to efficiently cluster large categorical data sets from real-world applications [16], [17]. There are two versions of k-modes clustering [18], [19]. Huang and Ng [20] analyzed the relationship between the two k-modes methods. So far, the k-modes algorithm and its variants [21], [22], including the fuzzy k-modes algorithm [23], the fuzzy k-modes algorithm with fuzzy centroid [24], the k-prototype algorithm [7], and w-k-means [25], [26], have been widely used in many disciplines. However, these methods, including [27]–[29], cannot be used to cluster set-valued data sets effectively. The SV-k-modes algorithm attempts to fill this gap.

VIII. Conclusion

In real applications, data with set-valued features are not uncommon, and current algorithms are not effective in clustering set-valued data. In this paper, the SV-k-modes algorithm was proposed for clustering categorical data sets with set-valued features. In the proposed algorithm, a new distance is used to compute the distance between two set-valued objects. A set-valued cluster center representation and the methods for updating the set-valued cluster centers in the iterative clustering process were developed. The convergence of the SV-k-modes clustering process was proved, and the time complexity of the SV-k-modes algorithm was analyzed. The heuristic method proposed to construct set-valued cluster centers in each iteration of the SV-k-modes algorithm speeds up the clustering process. An initialization algorithm for selecting the initial cluster centers was developed to improve the performance of the clustering algorithm. A method to generate set-valued synthetic data for scalability testing was developed. The experimental results on synthetic and real data sets have shown that the SV-k-modes algorithm outperforms other clustering algorithms in terms of clustering accuracy and that the algorithm is scalable to large and high-dimensional data.

Our future work is to accelerate the updating process of the cluster centers in *HAFSM* by building a hierarchical tree structure and applying the SV-k-modes algorithm to the behavior analysis of customers.

Acknowledgment

The authors would like to thank Prof. J. Pei at Simon Fraser University, Burnaby, BC, Canada, for his valuable suggestions. They would also like to thank the editors and reviewers for their valuable comments on this paper.

References

[1] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann, 2011.

[2] F. Giannotti, C. Gozzi, and G. Manco, "Clustering transactional data," in *Principles of Data Mining and Knowledge Discovery* (Lecture Notes in Artificial Intelligence), vol. 2431, T. Elomaa *et al.*, Eds. 2002, pp. 175–187.

[3] S. Guha, R. Rastogi, and K. Shim, "Rock: A robust clustering algorithm for categorical attributes," in *Proc. 15th Int. Conf. Data Eng.*, Sydney, NSW, Australia, Mar. 1999, pp. 512–521.

[4] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, vol. 1. Berkeley, CA, USA, Jan. 1967, pp. 281–297.

[5] P. Jaccard, "Distribution comparée de la flore alpine dans quelques régions des Alpes occidentales et orientales," *Bull. Société Vaudoise Sci. Naturelles*, vol. 37, pp. 241–272, 1901.

[6] M. Levandowsky and D. Winter, "Distance between sets," *Nature*, vol. 234, no. 5323, pp. 34–35, 1971.

[7] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining Knowl. Discovery*, vol. 2, no. 3, pp. 283–304, 1998.

[8] A. Björklund, T. Husfeldt, and M. Koivisto, "Set partitioning via inclusion-exclusion," *SIAM J. Comput.*, vol. 39, no. 2, pp. 546–563, 2009.

[9] 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.

[10] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, nos. 1–2, pp. 31–71, 1997.

[11] K. Bache and M. Lichman. (2014). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[12] S. Schiffman, L. Reynolds, and F. Young, *Introduction to Multidimensional Scaling: Theory, Methods, and Applications*. New York, NY, USA: Academic, 1981.

[13] J. Liang, L. Bai, C. Dang, and F. Cao, "The K-means-type algorithms versus imbalanced data distributions," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 4, pp. 728–745, Aug. 2012.

[14] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Dec. 2002.

[15] M.-L. Zhang and Z.-H. Zhou, "Multi-instance clustering with applications to multi-instance prediction," *Appl. Intell.*, vol. 31, no. 1, pp. 47–68, 2009.
[16] F. Cao, J. Z. Huang, and J. Liang, "Trend analysis of categorical data streams with a concept change method," *Inf. Sci.*, vol. 276, pp. 160–173, Aug. 2014.
[17] C. Wang, X. Dong, F. Zhou, L. Cao, and C.-H. Chi, "Coupled attribute similarity learning on categorical data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 4, pp. 781–797, Apr. 2015.
[18] J. D. Carroll, A. Chaturvedi, and P. E. Green, "k-means, k-medians and k-modes: Special cases of partitioning multiway data," in *Proc. Classification Soc. North Amer. Meet. Presentation*, Houston, TX, USA, 1994.
[19] A. Chaturvedi, P. E. Green, and J. D. Caroll, "k-modes clustering," *J. Classification*, vol. 18, no. 1, pp. 35–55, 2001.
[20] Z. Huang and M. K. Ng, "A note on K-modes clustering," *J. Classification*, vol. 20, no. 2, pp. 257–261, 2003.
[21] M. K. Ng, M. J. Li, J. Z. 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.
[22] 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.
[23] Z. Huang and M. K. Ng, "A fuzzy k-modes algorithm for clustering categorical data," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 4, pp. 446–452, Aug. 1999.
[24] D.-W. Kim, K. H. Lee, and D. Lee, "Fuzzy clustering of categorical data using fuzzy centroids," *Pattern Recognit. Lett.*, vol. 25, no. 11, pp. 1263–1271, 2004.
[25] J. Z. Huang, M. K. Ng, H. Rong, and Z. Li, "Automated variable weighting in k-means type clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 657–668, May 2005.
[26] L. Jing, K. Tian, and J. Z. Huang, "Stratified feature sampling method for ensemble clustering of high dimensional data," *Pattern Recognit.*, vol. 48, no. 11, pp. 3688–3702, 2015.
[27] I. Khan, J. Z. Huang, and K. Ivanov, "Incremental density-based ensemble clustering over evolving data streams," *Neurocomputing*, vol. 191, pp. 34–43, May 2016.
[28] H. Jia, Y.-M. Cheung, and J. Liu, "A new distance metric for unsupervised learning of categorical data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 5, pp. 1065–1079, May 2016.
[29] Y. Qian, F. Li, J. Liang, B. Liu, and C. Dang, "Space structure and clustering of categorical data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 10, pp. 2047–2059, Oct. 2016.

Fuyuan Cao received the M.S. and Ph.D. degrees in computer science from Shanxi University, Taiyuan, China, in 2004 and 2010, respectively.

He is currently a Professor with the School of Computer and Information Technology, Shanxi University. His current research interests include data mining and machine learning, with a focus on clustering analysis.

Joshua Zhexue Huang received the Ph.D. degree from the Royal Institute of Technology, Stockholm, Sweden.

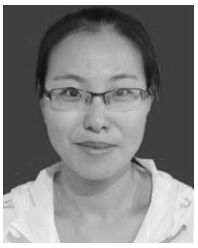
He is currently a Distinguished Professor with the College of Computer Sciences and Software Engineering, Shenzhen University, Shenzhen, China. He is known for his contributions to a series of k-means-type clustering algorithms in data mining that are widely cited and used, and some have been included in commercial software. He has led the development of the open source data mining system AlphaMiner, which is widely used in education, research, and industry. He has extensive industry expertise in business intelligence and data mining and has been involved in numerous consulting projects in Australia, Hong Kong, Taiwan, and Mainland China.

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, Shanxi University, Taiyuan, China, where he is also the Director of the Key Laboratory of Computational Intelligence and Chinese Information Processing of the Ministry of Education. He has authored more than 170 journal papers in his research fields. His current research interests include computational intelligence, granular computing, data mining, and knowledge discovery.

Xingwang Zhao received the M.S. degree in computer science from Shanxi University, Taiyuan, China, in 2012, where he is currently pursuing the Ph.D. degree with the School of Computer and Information Technology.

His current research interests include clustering analysis, data mining, and machine learning.



Yinfeng Meng received the M.S. degree from the School of Mathematical Science, Shanxi University, Taiyuan, China, in 2005, where she is currently pursuing the Ph.D. degree with the School of Computer and Information Technology.

Her current research interests include machine learning, data mining, and functional data analysis.

Kai Feng received the Ph.D. degree from Shanxi University, Taiyuan, China, in 2014.

He is currently a Lecturer with the School of Computer and Information Technology, Shanxi University. His current research interests include combinatorial optimization and interconnection network analysis.

Yuhua Qian received the M.S. and Ph.D. degrees from Shanxi University, Taiyuan, China, in 2005 and 2011, respectively.

He is currently a Professor with the Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University. He has authored more than 70 papers in his research fields.