Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Systems 31 (2006) 170–186
www.elsevier.com/locate/infosys

Adherence clustering: an efficient method for mining market-basket clusters

Ching-Huang Yun, Kun-Ta Chuang, Ming-Syan Chen*

Department of Electrical Engineering, National Taiwan University, No. 1, Sec. 14. Roosevelt Rd., Taipei, Taiwan, ROC

Received 7 January 2004; received in revised form 26 September 2004; accepted 3 November 2004

Abstract

We explore in this paper the efficient clustering of market-basket data. Different from those of the traditional data, the features of market-basket data are known to be of high dimensionality and sparsity. Without explicitly considering the presence of the taxonomy, most prior efforts on clustering market-basket data can be viewed as dealing with items in the leaf level of the taxonomy tree. Clustering transactions across different levels of the taxonomy is of great importance for marketing strategies as well as for the result representation of the clustering techniques for market-basket data. In view of the features of market-basket data, we devise in this paper a novel measurement, called the *category-based adherence*, and utilize this measurement to perform the clustering. With this category-based adherence measurement, we develop an efficient clustering algorithm, called algorithm *k-todes*, for market-basket data with the objective to minimize the category-based adherence. The distance of an item to a given cluster is defined as the number of links between this item and its nearest tode. The category-based adherence of a transaction to a cluster is then defined as the average distance of the items in this transaction to that cluster. A validation model based on *information gain* is also devised to assess the quality of clustering for market-basket data. As validated by both real and synthetic datasets, it is shown by our experimental results, with the taxonomy information, algorithm k-todes devised in this paper significantly outperforms the prior works in both the execution efficiency and the clustering quality as measured by information gain, indicating the usefulness of category-based adherence in market-basket data clustering.

Keywords: Data mining; Clustering market-basket data; Category-based adherence; k-todes

1. Introduction

Data clustering is an important technique for exploratory data analysis [1,2]. Explicitly, data clustering is a well-known capability studied in information retrieval [3], data mining [4], machine learning [5], and statistical pattern recognition [6].

*Corresponding author. Tel.: +886 2 2363 5251; fax: +886 2 2367 1597.
E-mail addresses: chyun@arbor.ee.ntu.edu.tw (C.-H. Yun), doug@arbor.ee.ntu.edu.tw (K.-T. Chuang), mschen@cc.ee.ntu.edu.tw (M.-S. Chen).

0306-4379/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.is.2004.11.008

In essence, clustering is meant to divide a set of transactions into some proper groups in such a way that transactions in the same group have similar features while transactions in different group are dissimilar. Many data clustering algorithms have been proposed in the literature. These algorithms can be categorized into nearest neighbor clustering [7], fuzzy clustering [8], partitional clustering [9,10], hierarchical clustering [11,12], artificial neural networks for clustering [13], and statistical clustering algorithms [14]. However, finding optimal clustering result is known to be an NP-hard problem [15] and thus clustering algorithms usually employ some heuristic processes to find local optimal results.

In market-basket data (also called transaction data), each transaction contains a set of items purchased by a customer. Market-basket data has been well studied in mining association rules for discovering the set of frequently purchased items [16–19]. However, mining association rules is generally useful in the cross-selling of items. For marketing strategies, the clusters with representative subjects (consisting of items or categories) are informative for planning a product promotion. Clustering market-basket data techniques can be used to identify the subjects with similar buying patterns in the same cluster. For promotion of a cluster, the items are identified as the products to be sold and the transactions could be used to identify the target customers. In this paper, we focus on clustering market-basket data for identifying representative subjects. One of the important features of market-basket data sets is that they are generated at rapid pace (million transactions per day) and thus requires the data mining algorithms to be scalable and capable of dealing with the large data set.

It is important to note that since customers purchase desired items with the corresponding categorical meanings, the implications from purchasing supports of items and the taxonomy of items are in fact entangled, and both of them are of great importance in reflecting customer behaviors. Explicitly, the support of item i is defined as the percentage of transactions which contain i. Note that in mining association rules, a *large item* is basically an item with frequent occurrences in transactions [16]. Thus, item i is called a large item if the support of item i is larger than the pre-given minimum support count. The taxonomy of items defines the categorical relationships of items and it can be represented as a taxonomy tree. In the taxonomy tree, the leaf nodes are called the item nodes and the internal nodes are called the category nodes. For the example shown in Fig. 1, "War and Peace" is an item node and "Novel" is a category node. As formally defined in Section 2, a *large item/category* (i.e., item or category) is basically an item/category with its occurrence count in transactions exceeding a given threshold. If an item/category is large, its corresponding node in the taxonomy tree is called a *tode* (standing for taxonomy node). The todes in each

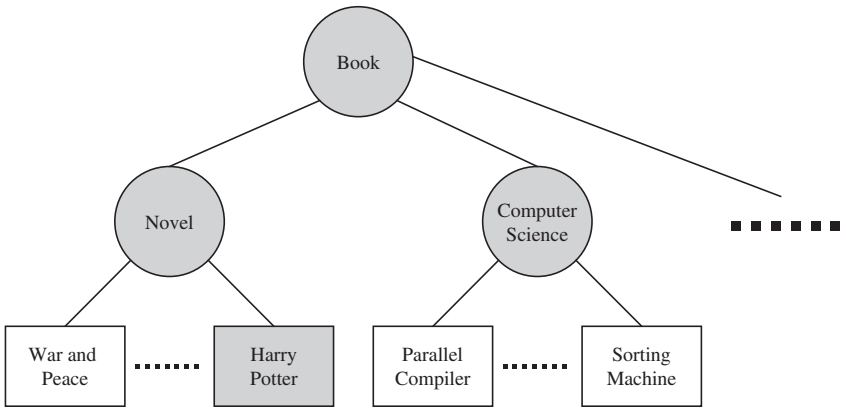

Fig. 1. An example taxonomy tree for books.

cluster can be viewed as the representatives of the cluster. For the example shown in Fig. 1, nodes marked gray are todes. Based on the definition of a large item, if item i is large, the categories containing i are also large. For example, because item "Harry Potter" is large, its ancestors, category "Novel" and category "Book", are also large. In other words, in the taxonomy tree, if a node is a tode, its ancestor nodes are also todes. The characteristic of todes is helpful in efficiently discovering todes of clusters. As formally defined in Section 3.1, the todes and the taxonomy tree both are used to identify the *nearest todes*.

In view of the features of market-basket data, we devise in this paper a novel measurement, called the *category-based adherence*, and utilize this measurement to perform the clustering. The *distance* of an item to a given cluster is defined as the number of links between this item and its nearest tode in the taxonomy tree. The *adherence* of a transaction to a cluster is then defined as the average distance of the items in this transaction to that cluster.[1] With this category-based adherence measurement, we develop an efficient clustering algorithm, called algorithm *k-todes*, for market-basket data. Explicitly, algorithm k-todes employs the category-based adherence as the similarity measurement between transactions and clusters, and allocates each transaction to the cluster with the minimum adherence. To the best of our knowledge, without explicitly considering the presence of the taxonomy, previous efforts on clustering market-basket data unavoidably restricted themselves to deal with the items in the leaf level (also called item level) of the taxonomy tree. However, clustering transactions across different levels of the taxonomy is of great importance for the efficiency and the quality of the clustering techniques for market-basket data. Note that in the real market-basket data, there are many transactions containing only single items, and many items are purchased infrequently. Hence, without considering the taxonomy tree, one may inappropriately treat a transaction (such as the one containing "parallel compiler" in Fig. 1) as an outlier. However, as indicated in Fig. 1, purchasing "parallel compiler" is in fact instrumental for the category node "computer science" to become a tode (i.e., a representative). In contrast, by employing category-based adherence measurement for clustering, many transactions will not be mistakenly treated as outliers if we take taxonomy relationships of items into consideration, thus leading to a better clustering quality. The details of k-todes will be described in Section 7. A validation model based on *Information Gain* (*IG*) is also devised in this paper for evaluating the clustering results. As validated by both real and synthetic datasets, it is shown by our experimental results, with the taxonomy information, that algorithm k-todes devised in this paper significantly outperforms the prior works [20,21] in both the execution efficiency and the clustering quality evaluated by IG, indicating the usefulness of category-based adherence in market-basket data clustering.

[1]The formal definitions of these terms will be given in Section 2.1.

1.1. Related works

Many data clustering algorithms have been proposed in the literature. Numerical attributes are those with finite or infinite number of ordered values, such as the height of a customer. On the other hand, categorical attributes are those with finite unordered values, such as the sex of a customer. In market-basket data, the purchase record is unordered and thus non-numeric. In addition, a transaction can be represented as a vector with boolean attributes where each attribute corresponds to a single item [22]. Boolean attributes themselves form a special case of categorical attributes because they are unordered [2].

The k-means algorithm is efficient in the clustering of numerical data [23]. There are other fast algorithms designed for clustering large numerical data sets, such as CLARANS [24], BIRCH [25], DBSCAN [26], CURE [27], and CSM [28]. In addition, several approaches in [29–31] are proposed to solve the high dimensionality and data sparsity problems of numerical data. The k-modes algorithm extends the k-means algorithm for clustering categorical data by replacing means of clusters with modes and using a

frequency-based method to update modes. For each attribute, the mode is the highest frequency values. However, in clustering market-basket data sets, k-modes will view each item as one boolean attribute. For each item, k-modes chooses True or False as the highest value to perform the clustering and suffers unstable clustering quality in market-basket data. The approach in [32] is an extension of k-means algorithm to cluster categorical data by converting multiple category attributes into binary attributes which are computed as numerical data. However, it is very time-consuming in matrix computing and needs a large memory to store the matrices for clustering market-basket data. ROCK is an agglomerative hierarchical clustering algorithm by treating market-basket data as categorical data and using the links between the data points to cluster categorical data [22]. ROCK utilizes the concept of *links* for clustering, where a link is defined as the number of common "*neighbors*" between two transactions. Here two transactions are said to be the *neighbor* if their Jaccard-coefficient [1] is larger than or equal to the user defined threshold θ. The time complexity of ROCK could be prohibitive because the number of transactions is very large in the market-basket data. Only by properly choosing value of θ, ROCK could generate the clustering results with good qualities. However, in practice, the threshold θ is difficult to be determined by users [33]. CORE [34] is a gravity based clustering algorithm by using the ensemble of correlated-forces between two clusters as the similarity measurement to perform subspace categorical clustering. Conceptual-based clustering in machine learning is developed for clustering categorical data [5,35,36]. In general, the clustering techniques proposed in [5,22,34–36] have high time complexity and thus are not suitable for market-basket data. The concept of nodes in [37] is a set of distinct categorical values where the emphasis is in constructing the categorical clusters by both STIRR [37] and CACTUS [38]. However, how to cluster transactions was not addressed. Explicitly, STIRR is an iterative algorithm according to nonlinear dynamic systems. In addition, CACTUS is devised by using a summarization procedure based on the assumption that all attributes are independent. COOLCAT in [33] utilizing the entropy analysis for clustering categorical data sets is also under the attribute independence assumption. However, the items (each of which represents a boolean attribute) in market-basket data sets usually have high associations [16], meaning that the assumption of having independent attributes needs further justification.

The authors in [39] proposed a hypergraph partitioning algorithm to find the clusters of items and transactions based on the large item sets. The work in [40] devised a top-down hierarchical algorithm by using association rules with high confidences to discover the clusters of customers. BitOp is a greedy grid-based clustering algorithm for clustering association rules where the cause attributes are quantitative and the consequence attribute is categorical [41]. The authors in [42] proposed an EM-based algorithm by using the maximum likelihood estimation method for clustering transaction data. OPOSSUM is a graph-partitioning approach based on a similarity matrix to cluster transaction data [43]. The work in [21] proposed a k-means based algorithm by using large items as the similarity measurement to divide the transactions into clusters with a cost function to minimize the overlap of large items (corresponding to inter-cluster cost) and minimize the union summation of small items (corresponding to intra-cluster cost). In this approach, an item which is not large is called a small item which is used to measure the intra-cluster cost. However, with this disposition, the support difference between a large item and a small one could be as few as one, which could make the clustering quality be very data dependent. OAK in [44] combined hierarchical and partitional clustering techniques for transaction data. CLOPE in [45] proposed a heuristic approach by increasing the height-to-width ratio for clustering transaction data. CLOPE did not explicitly address the inter-cluster dissimilarity issue. In addition, there is no explicitly statement for describing the statistical relationship between the repulsion parameter and the intra-cluster similarity. In market-basket data, the taxonomy of items defines the generalization relationships for the concepts in different abstraction levels [46]. Item taxonomy (i.e., *is-a* hierarchy) is well

addressed with respect to its impact to mining association rules of market-basket data [17,19] and can be represented as a tree, called *taxonomy tree*. Similar techniques for extracting synonyms, hypernyms (i.e., a *kind* of) and holonyms (i.e., a *part* of) from the lexical database are derived in [47,48].

This paper is organized as follows. Preliminaries are given in Section 2. In Section 3, algorithm k-todes is devised for clustering market-basket data. Experimental studies are conducted in Section 4. This paper concludes with Section 5.

2. Preliminary

The problem description will be presented in Section 2.1. In Section 2.2, we describe a new validation model, *IG* validation model, for the assessment to the quality of different clustering algorithms.

2.1. Problem description

In this paper, the market-basket data is represented by a set of transactions. A database of transactions is denoted by $D = \{t_1, t_2, \ldots, t_h\}$, where each transaction t_j is represented by a set of items $\{i_1, i_2, \ldots, i_r\}$. An example database for clustering market-basket data is described in Table 1 where there are twelve transactions, each of which has a transaction identification (abbreviated as TID) and a set of purchased items. For example, transaction ID 40 has items h and item z. A clustering $U = <C_1, C_2, \ldots, C_k>$ is a partition of transactions into k clusters, where C_j is a cluster consisting of a set of transactions.

Items in the transactions can be generalized to multiple concept levels of the taxonomy. An example taxonomy tree is shown in Fig. 2. In the taxonomy tree, the leaf nodes are called the *item nodes* and the internal nodes are called the *category nodes*. The root node in the highest level is a virtual concept of the generalization of all categories. In this taxonomy tree, item g *is-a* category B, category B *is-a* category A, and item h *is-a* category B, etc. In this paper, we use the measurement of the occurrence count to determine which items or categories are the representatives of each cluster.

Definition 1. The support of an item i_k in a cluster C_j, denoted by $Sup(i_k, C_j)$, is defined as the number of transactions containing this item i_k in cluster C_j. An item i_k in a cluster C_j is called a *large item* if $Sup(i_k, C_j)$ exceeds the minimum support count.

Definition 2. The support of a category c_k in a cluster C_j, denoted by $Sup(c_k, C_j)$, is defined as the number of transactions containing items under this category c_k in cluster C_j. A category c_k in a cluster C_j is called a *large category* if $Sup(c_k, C_j)$ exceeds the minimum support count.

Note that one transaction may include more than one item from the same category, in which case the support contributed by this transaction to that category is still one. In this paper, the minimum support percentage S_p is a given parameter for determining the large items/categories of the taxonomy tree in the cluster. For a cluster C_j, the minimum support count $S_c(C_j)$ is defined as follows.

Definition 3. For cluster C_j, the minimum support count $S_c(C_j)$ is defined as

$$S_c(C_j) = S_p * |C_j|.$$

where $|C_j|$ denotes the number of transactions in cluster C_j.

Consider the example database in Table 1 as an initial cluster C_0 with the corresponding taxonomy tree recording the supports of the items/categories shown in Fig. 2. Then, $Sup(g, C_0) = 5$ and $Sup(E, C_0) = 7$. With $S_p = 50\%$, we have $S_c(C_0) = 6$.

Table 1
An example database D

TID	10	20	30	40	50	60	70	80	90	100	110	120
Items	g, x	m, y	y, z	h, z	g, x, y	g, n	k, m, n	y	g, k, n	m, n	y, z	g, h, n

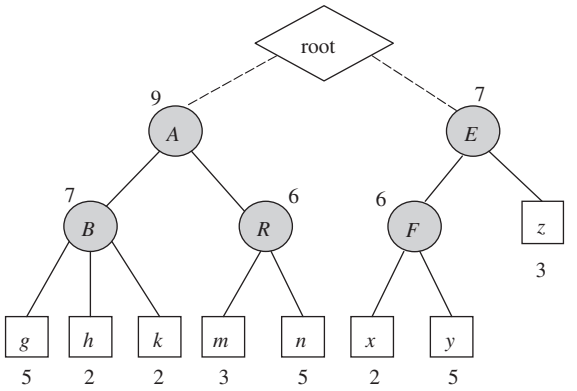

Fig. 2. An illustrative taxonomy example whose transactions are shown in Table 1 ($S_p = 0.5$).

In this example, all categories are large but all items are not.

2.2. Information gain validation model

To evaluate the quality of clustering results, some experimental models were proposed [49,50]. In general, *square error criterion* is widely employed in evaluating the efficiency of numerical data clustering algorithms [49]. In addition, authors in [51] proposed a novel clustering validation scheme which uses the variance and the density of each cluster to measure the inter-cluster dissimilarity and the intra-cluster similarity. Note that the nature feature of numeric data is quantitative (e.g., weight or length), whereas that of categorical data is qualitative (e.g., color or gender) [2]. Thus, validation schemes using the concept of variance are thus not applicable to assessing the clustering result of categorical data. To remedy this problem, some real data with good classified labels, e.g., mushroom data, congressional votes data, soybean disease [52] and Reuters news collection [53], were taken as the experimental data for categorical clustering algorithms [22,54,21,44]. In view of the feature of market-basket data, we propose in this paper a validation model based on Information Gain (IG) to assess the qualities of the clustering results. It is noted that information gain is widely used in the classification problem [55,56]. Explicitly, ID3 [55] and C4.5 [56] used information gain measurement to select the test attribute with the highest information gain for splitting when constructing the decision tree.

The definitions required for deriving the information gain of a clustering result are given below.

Definition 4. The entropy of an attribute J_a in a database D is defined as

$$I(J_a, D) = -\sum_{i=1}^{n} \frac{|J_a^i|}{|D|} * \log_2 \frac{|J_a^i|}{|D|},$$

where $|D|$ is the number of transactions in D and $|J_a^i|$ denotes the number of the transactions whose attribute J_a is classified as the value J_a^i in D.

Definition 5. The entropy of an attribute J_a in a cluster C_j is defined as

$$I(J_a, C_j) = -\sum_{i=1}^{n} \frac{|J_{a,c_j}^i|}{|C_j|} * \log_2 \frac{|J_{a,c_j}^i|}{|C_j|},$$

where $|C_j|$ is the number of transactions in cluster C_j, and $|J_{a,c_j}^i|$ denotes the number of the transactions whose attribute J_a is classified as the value J_a^i in C_j.

Definition 6. Let a clustering U contain $C_1, C_2, \ldots, C_m$ clusters. Thus, the entropy of an attribute J_a in the clustering U is defined as

$$E(J_a, U) = \sum_{C_j \in U} \frac{|C_j|}{|D|} I(J_a, C_j).$$

Definition 7. The information gain obtained by separating J_a into the clusters of the clustering U is defined as

$$Gain(J_a, U) = I(J_a, D) - E(J_a, U).$$

Definition 8. The information gain of the clustering U is defined as

$$IG(U) = \sum_{J_a \in I} Gain(J_a, U),$$

where I is the data set of the total items purchased in the whole market-basket data records.

A completely numerical example on the use of these definitions will be given in Section 3.3. For

Table 2
The meanings of various parameters

Notation	Meaning
D	The database
$Sup(i, C_j)$	The support of i in cluster C_j
$IG(U)$	The information gain of clustering U
$IG_{item}(U)$	The information gain obtained on items in clustering U
$IG_{cat}(U)$	The information gain obtained on categories in clustering U
$IG_{total}(U)$	The total information gain in clustering U
$d(i_k, C_j)$	The distance of item i_k to cluster C_j
$H(t, C_j)$	The adherence of transaction t to cluster C_j

clustering market-basket data, the larger an IG value, the better the clustering quality is. In market-basket data, with the taxonomy tree, there are three kinds of IG values, i.e., $IG_{item}(U)$, $IG_{cat}(U)$, and $IG_{total}(U)$, for representing the quality of a clustering result. Specifically, $IG_{item}(U)$ is the information gain obtained on items and $IG_{cat}(U)$ is the information gain obtained on categories. $IG_{total}(U)$ is the total information gain, i.e., $IG_{total}(U) = IG_{item}(U) + IG_{cat}(U)$. In general, market-basket data set is typically represented by a 2-dimensional table, in which each entry is either 1 or 0 to denote purchased or non-purchased items, respectively. In IG validation model, we treat each item in market-basket data as an attribute J_a with two classified labels, 1 or 0. Explicitly, for an item i_k, $I_{i_k}^{Yes}$ and $I_{i_k}^{No}$ are the two classified labels of item i_k to represent purchased and non-purchased values. The meanings of various parameters are shown in Table 2. It will be shown in Section 4 that with the category-based adherence measurement, algorithm k-todes outperforms the prior works [20,21] in the clustering quality based on the IG validation model.

3. Design of algorithm k-todes

In this section, we describe the details of k-todes algorithm. The similarity measurement of k-todes, called category-based adherence, will be described in Section 3.1. The procedure of k-todes is devised in Section 3.2 and an illustrative example is given in Section 3.3. The complexity of k-todes is analyzed in Section 3.4.

3.1. Similarity measurement: category-based adherence

Some terminologies for the similarity measurement employed by algorithm k-todes are defined as follows.

Definition 9 (*Tode*). If an item/category is large, its corresponding node in the taxonomy tree is called a *tode* (standing for taxonomy node). In this paper, the todes in each cluster are the representatives of the cluster. For the example shown in Fig. 3, nodes marked gray are todes.

Definition 10 (*Nearest tode of an item to a cluster*). In the taxonomy tree, the *nearest tode* of an item i_k is itself if i_k is a tode. Otherwise, the nearest tode is the category node which is the lowest generalized concept level node among all ancestor todes of item i_k. Note that if an item/category node is identified as tode, all its high level category nodes will also be todes. For the example shown in Fig. 3, the nearest tode of item k to cluster C_1 is category B and the nearest tode of item k to cluster C_2 is category A.

Definition 11 (*Distance of an item to a cluster*). - For an item i_k of a transaction, the *distance* of i_k to a cluster C_j, denoted by $d(i_k, C_j)$, is defined as the number of links between i_k and the nearest tode of i_k to cluster C_j. If i_k is a tode in cluster C_j, then $d(i_k, C_j) = 0$. For the example shown in Fig. 3, the distance of item k to cluster C_1 is $d(k, C_1) = 1$ and the distance of item k to cluster C_2 is $d(k, C_2) = 2$.

Definition 12 (*Adherence of a transaction to a cluster*). For a transaction $t = \{i_1, i_2, \ldots, i_p\}$,the adherence of t to a cluster C_j, denoted by $H(t, C_j)$, is defined as the average distance of distances of the items in t to C_j and shown below.

$$H(t, C_j) = \frac{1}{p}\sum_{k=1}^{p} d(i_k, C_j),$$

where $d(i_k, C_j)$ is the distance of i_k to cluster C_j. For the example shown in Fig. 3, the adherence of TID

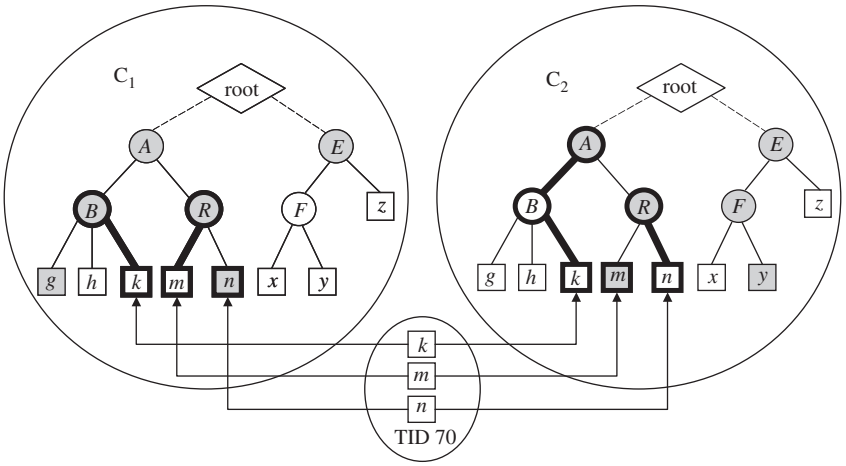

Fig. 3. The adherence represents the similarity measurement.

70 to cluster C_1 is $H(70, C_1) = \frac{1}{3}(d(k, C_1) + d(m, C_1) + d(n, C_1)) = \frac{1}{3}(1 + 1 + 0) = \frac{2}{3}$ and the adherence of TID 70 to cluster C_2 is $H(70, C_2) = \frac{1}{3}(d(k, C_2) + d(m, C_2) + d(n, C_2)) = \frac{1}{3}(2 + 0 + 1) = 1$.

Note that the todes are the representatives of a cluster. The adherence of a transaction to a cluster is a measurement of the distance between the transaction and the representatives of the cluster. Thus, the adherence is smaller, the similarity is higher. In this example shown in Fig. 3, because $H(70, C_1) = \frac{2}{3} < H(70, C_2) = 1$, TID 70 is more similar to C_1 than C_2.

3.2. Procedure of algorithm k-todes

The overall procedure of algorithm k-todes is shown in Fig. 4. In Step 1, algorithm k-todes randomly selects k transactions as the seed transactions of the k clusters from the database D. For each cluster, the items and categories of the corresponding seed transaction are counted once in the taxonomy tree. In each cluster, the items and their ancestors are all large in the very beginning because their support percentages are all 100% in the only seed transaction, larger than the minimum support percentage. For each initial cluster, they are the todes. In Step 2, algorithm k-todes reads each transaction sequentially and allocates it to the cluster with the minimum category-based adherence. After one transaction is allocated to a cluster C_j, the supports of the items and their ancestors are increased by one in the corresponding nodes in the taxonomy tree of C_j. After all transactions are allocated, the minimum support counts of clusters are updated. In Step 3, algorithm k-todes updates the todes of each cluster based on the supports of nodes in the taxonomy tree. In Step 4, algorithm k-todes repeats Steps 2 and 3 until no transaction is moved between clusters. In Step 5, algorithm k-todes outputs the taxonomy tree of the final clustering result for each cluster, where the items, categories, and their corresponding counts are presented.

Procedure of Algorithm k-todes

Step 1. Randomly select k transactions as the seed transactions of the k clusters from the database D.

Step 2. Read each transaction sequentially and allocates it to the cluster with the minimum category-based adherence. For each moved transaction, the supports of items and their ancestors are increased by one.

Step 3. Update the todes of each cluster.

Step 4. Repeat Step 2 and Step 3 until no transaction is moved between clusters.

Step 5. Output the taxonomy tree for each cluster as the visual representation of the clustering result.

Fig. 4. The overall procedure of algorithm k-todes.

3.3. An illustrative example

An illustrative example is given to describe the execution of k-todes in Section 3.3.1 and an example for describing the measurement of information gain is given in Section 3.3.2.

3.3.1. Execution of k-todes

For the example database D shown in Table 1, we set $k = 2$ and $S_p = 50\%$. In Step 1, algorithm k-todes randomly chooses TID 10 and TID 20 as the seed transaction of the cluster C_1 and C_2, respectively. Then, for cluster C_1 shown in Fig. 5a, nodes marked gray are the purchased items of TID 10 and the corresponding categories in the taxonomy tree. The gray nodes are identified as todes. Similarly, for cluster C_2, shown in Fig. 5b, nodes marked gray are todes. In Fig. 5, the support of each node is illustrated nearby. For example, $Sup(g, C_1)$ is 1 and $Sup(g, C_2)$ is 0. In Step 2, algorithm k-todes first allocates TID 30 to cluster C_2 because $H(30, C_2) = \frac{1}{2}(1 + 0) = \frac{1}{2}$ is smaller than $H(30, C_1) = \frac{1}{2}(1 + 1) = 1$. Similarly, TIDs 40, 50, 60, 90, and 120 are allocated to cluster C_1 which is shown in Fig. 6a. TIDs 30, 70, 80, 100, and 110 are allocated to cluster C_2 which is shown in Fig. 6b. In Step 3, algorithm k-todes updates the todes in cluster C_1 to be $\{A, E, B, R, g, n\}$ and the todes in cluster C_2 to be $\{A, E, R, F, m, y\}$. Explicitly, algorithm k-todes derives $S_c(C_1) = 3$ and $S_c(C_2) = 3$ by $S_p * |C_1| = 0.5 * 6 = 3$ and $S_p * |C_2| = 0.5 * 6 = 3$, respectively. Because $Sup(g, C_1) > S_c(C_1)$, item g is identified as a large node in cluster C_1 and marked gray. In Step 4, algorithm k-todes proceeds to iteration 2 by repeating Steps 2 and 3. In iteration 2, two transactions, TID 50 and TID 70 are moved. TID 50 is moved from cluster C_1 to cluster C_2 because $H(50, C_1) = \frac{1}{3}(0 + 2 + 2) = \frac{4}{3} > H(50, C_2) = \frac{1}{3}(2 + 1 + 0) = 1$, and TID 70 is moved from cluster C_2 to cluster C_1 due the $H(70, C_1) = \frac{1}{3}(1 + 1 + 0) = \frac{2}{3} < H(70, C_2) = \frac{1}{3}(2 + 0 + 1) = 1$. Then, algorithm k-todes updates the todes again. In iteration 3, only one transaction TID 100 is moved from cluster C_2 to cluster C_1. In iteration 4, there is no movement and thus algorithm k-todes proceeds to Step 5. The final taxonomy trees of clustering U_1 are shown in Fig. 7.

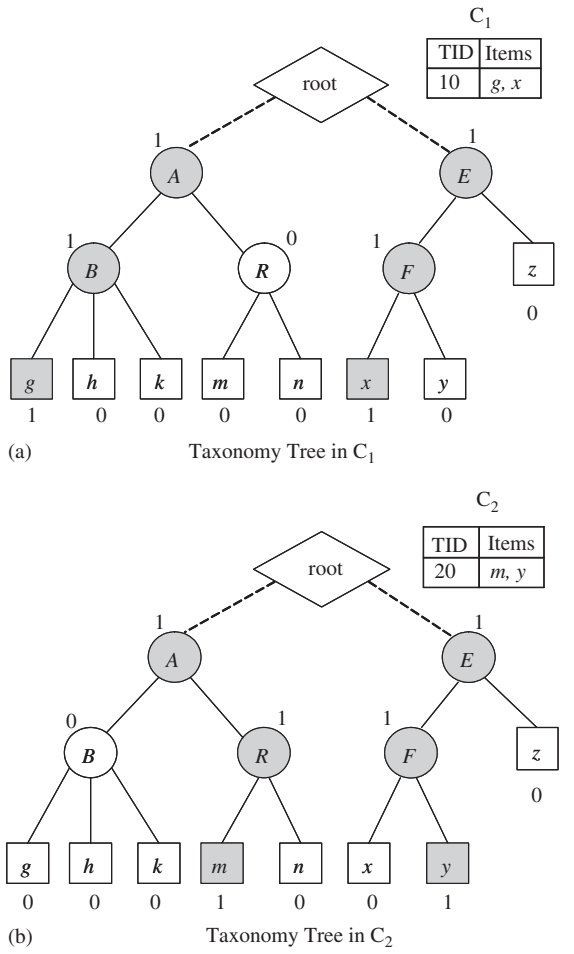

Fig. 5. In Step 1, algorithm CBA randomly chooses the seed transaction for each cluster.

Note that a transaction at item level may not be similar to any cluster. For example, TID 10 $\{g, x\}$ and TID 40 $\{h, z\}$ have no common items, but item g and item h have common category B and item x and item z have common category E. Thus, TID 10 is similar to TID 40 in the high level concept. By taking category-based adherence measurement, many transactions may not be taken as outliers if we take categorical relationships of items into consideration. In addition, transactions at the item level may have the same similarities in different clusters. However, by summarizing the similarities of all items across their category levels, algorithm k-todes allocates each transaction to a proper

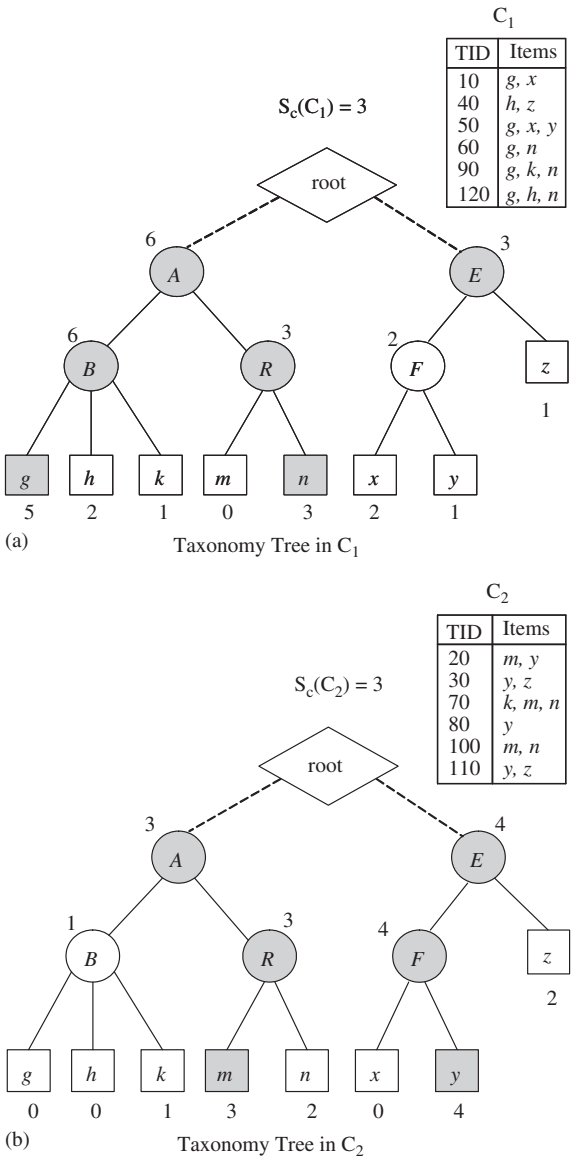

Fig. 6. In Step 2, algorithm CBA reads each transaction sequentially and allocates it to the cluster with the minimum category-based adherence.

cluster. For example, TID 50 has three items: g, x, and y. Item g is large in cluster C_1 and item y is large in cluster C_2. Thus, TID 50 has the same similarities in both C_1 and C_2. However, item x is a category F which is a tode in C_2. Thus, TID 50 is allocated to C_2.

3.3.2. Measurement by information gains

To provide more insight into the quality of k-todes, we calculate the *IG* values of the clustering U_1 shown in Fig. 7. Note that for an item i_k, $I_{i_k}^{Yes}$ and $I_{i_k}^{No}$ are the two classified labels of item i_k for representing purchased and non-purchased values. For item g, the information gain $Gain(g, U_1) = I(g, D) - E(g, U_1) = (-\frac{5}{12} log_2 \frac{5}{12} - \frac{7}{12} log_2 \frac{7}{12}) - [\frac{7}{12}(-\frac{4}{7} log_2 \frac{4}{7} - \frac{3}{7} log_2 \frac{3}{7}) + \frac{5}{12}(-\frac{1}{5} log_2 \frac{1}{5} - \frac{4}{5} log_2 \frac{4}{5})] = 0.10$. Similarly, $Gain(h, U_1) = 0.15$, $Gain(k, U_1) = 0.48$, $Gain(m, U_1) = 0.31$, $Gain(n, U_1) = 0.48$, $Gain(x, U_1) = 0$, $Gain(y, U_1) = 0.98$, and $Gain(z, U_1) = 0.39$. Hence, $IG_{item}(U_1) = \sum_{J_a \subset I} Gain(J_a, U_1) = 2.89$, where I is the set of items $\{g, h, k, m, n, x, y, z\}$. Similarly, $Gain(B, U_1) = 0.33$, $Gain(R, U_1) = 0.2$, $Gain(A, U_1) = 0.41$, $Gain(F, U_1) = 0.65$, $Gain(E, U_1) = 0.48$, and thus $IG_{cat}(U_1) = sum_{J_a \subset C} Gain(J_a, U_1) = 2.07$, where C is the set of categories $\{A, B, E, F, R\}$. Then, $IG_{total}(U_1) = IG_{item}(U_1) + IG_{cat}(U_1) = 4.96$.

3.4. Complexity analysis of algorithm k-todes

The time complexity and the space complexity of algorithm k-todes are analyzed by the following two theorems.

Theorem 1. *The time complexity of k-todes is* $O(rk(|D|vN^L + N^N))$, *where* r *is the number of iterations*, k *is the given cluster number*, $|D|$ *is the database size*, v *is the average transaction length*, N^L *is the number of taxonomy levels, and* N^N *is the number of nodes in the taxonomy tree.*

Proof. We first define following symbols to analyze the complexity of each iteration in detail: I_t is the item set in transaction t, $1 \leqslant t \leqslant |D|$, I_t^m is the mth item in transaction t, $cost(I_t^m, C_k)$ is the cost for I_t^m to find the nearest tode in cluster C_k, and $x(I_t^m)$ is the number of levels from item I_t^m to its highest ancestor, $1 \leqslant x(I_t^m) \leqslant N^L$. Note that an iteration consists of Steps 2 and 3. For each transaction t, the adherence of t to every cluster is obtained in Step 2. Thus, the time complexity of this sub-step is $\sum_k \sum_{I_t \in D} \sum_{I_t^m} cost(I_t^m, C_k)$. After obtaining the cluster C_a in which t has the minimum adherence, t is allocated to C_a and the supports of items and related categories in the taxonomy tree of C_a will be increased by one.

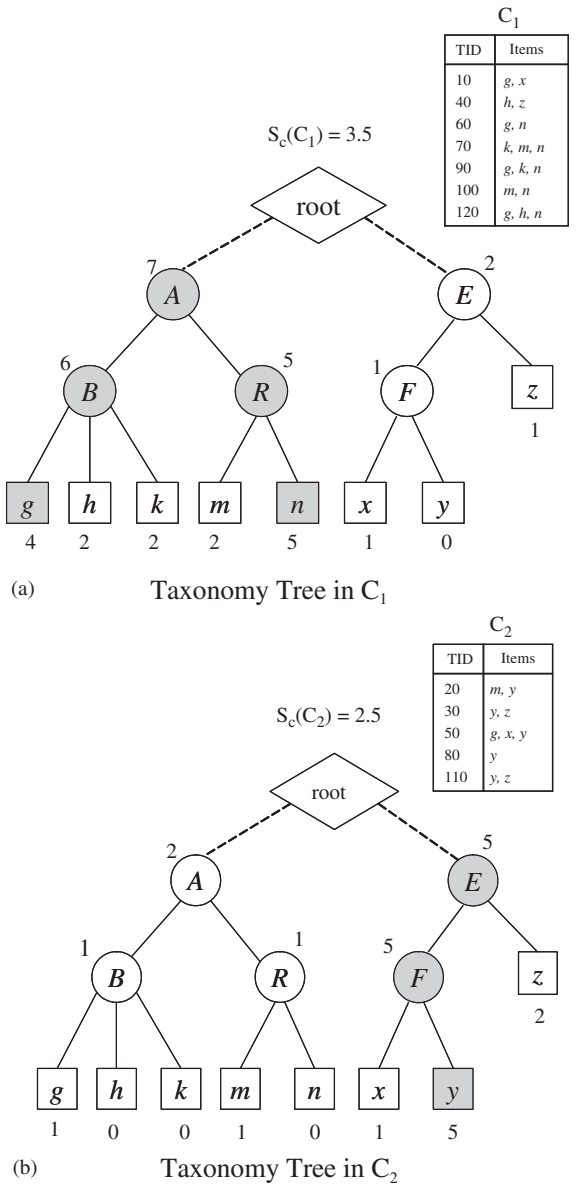

Fig. 7. In Step 5, algorithm k-todes generates the clustering U_1.

Thus, the time complexity of this sub-step is $\sum_{I_t^m} x(I_t^m)$. In Step 3, algorithm k-todes updates the todes of each cluster. Thus, the time complexity of Step 3 iskN^N where N^N is the number of nodes in the taxonomy tree. With r iterations for running Steps 2 and 3, the total time complexity is therefore

$$\begin{aligned}
&\sum_r \left[\left\{ \sum_k \sum_{I_t \in D} \sum_{I_t^m} cost(I_t^m, C_k) \right. \right. \\
&\qquad \left. \left. + \sum_{I_t^m} x(I_t^m) \right\} + \{kN^N\} \right] \\
&\leqslant rk|D|vN^L + r|D|vN^L + rkN^N \\
&= r(k+1)|D|vN^L + rkN^N \\
&= O(rk(|D|vN^L + N^N)),
\end{aligned}$$

where r is the number of iterations, k is the given cluster number, $|D|$ is the database size, v is the average transaction length, N^L is the number of taxonomy levels, and N^N is the number of nodes in the taxonomy tree. □

Theorem 2. *The space complexity of k-todes is* $O(|D| + kA)$, *where* $|D|$ *is the database size,* k *is the given cluster number, and* A *is the number of nodes, including category nodes and item nodes, in the taxonomy tree.*

Proof. First, before k-todes is executed, all data must be loaded and the space requirement is $O(|D|)$. In each cluster, there is only an array structure needed to store the supports of all nodes, whose space requirement is $O(A)$. Because the number of clusters is k, the space requirement is $O(kA)$ for all clusters. Thus, the overall space complexity of k-todes is $O(|D| + kA)$. □

4. Experimental results

To assess the performance of algorithm k-todes, we have conducted a series of experiments. These experiments are performed on a computer with a 1 Ghz Intel CPU and 512M of memory. We compare k-todes with k-modes algorithm [20] and the algorithm proposed in [21] (for the convenience, the algorithm is named as *Basic* in this paper). By extending both previous approaches with taxonomy consideration in market-basket data, we also implement algorithm *k-modesT* (standing for *k-modes with Taxonomy*) and algorithm *BasicT* (standing for *Basic with*

Taxonomy) for comparison purposes. The details of data generation are described in Section 4.1. The experimental results are shown in Section 4.2.

4.1. Data generation

The meanings of various parameters used in our experiments are shown in Table 3. We take the real market-basket data from a large bookstore company for performance study. In this real data set whose item distribution is shown in Fig. 8, there are $|D| = 100K$ transactions, $N^I = 58909$ items, and $N^L = 3$ levels. Note that in this real data, there are many transactions containing only single items, and many items are purchased infrequently. In this real data, there are $58909 - 31846 = 27063$ items which are purchased only once among the $100K$ transactions. In addition, the number of the taxonomy level in this real data set is 3. To provide more insight into this study, we use a well-known market-basket synthetic data generated by the IBM Quest Synthetic Data Generation Code [16], as the synthetic data for performance evaluation. This code will generate volumes of transactions over a large range of data characteristics. These transactions mimic the transactions in the real world retailing environment. This generation code also assumes that people will tend to buy sets of items together, and each such set is potentially a maximal large itemset. An example of such a set might be sheets, pillow case, comforter, and ruffles. However, not all items purchased by customers are large itemsets. The average size of the transactions, denoted by $|T|$, is set to 5 as default. The average size of the maximal potentially large itemsets, denoted by $|I|$, is set to 2 as default. The number of maximal potential large itemsets, denoted by $|L|$, is set to $2K$. The number of items in database, denoted by N^I, is set to $60K$ as default. The number of roots, denoted by N^R, is set to 100 and the number of the taxonomy level, denoted by N^L, is set to 3.

Table 3
The meanings of various parameters used in experimental results

Notation	Meaning
$\|D\|$	The database size
$\|T\|$	Average size of the transactions
$\|I\|$	Average size of the maximal potential large itemsets
$\|L\|$	Number of large itemsets within database
N^I	Number of items in database
N^R	Number of the roots
N^L	Number of the taxonomy levels

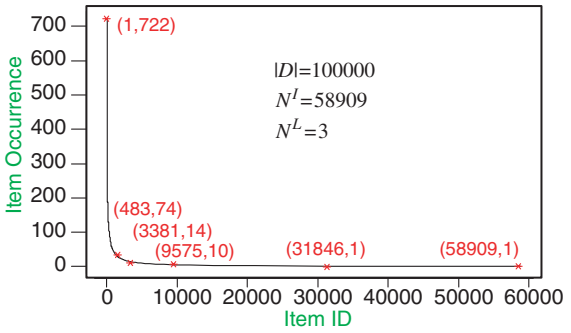

Fig. 8. The data distribution of real market-basket data obtained from the large bookstore.

4.2. Performance study

We conduct experiments in this section for performance study and the clustering quality is evaluated by the *IG* values. For algorithms k-todes, Basic, and BasicT, the minimum support percentage S_p is set to 0.5%. Recall that there are three kinds of *IG* values, i.e., IG_{item}, IG_{cat}, and IG_{total}, for evaluating the quality of the clustering result. IG_{item} is the information gain obtained on items and IG_{cat} is the information gain obtained on categories. $IG_{total} = IG_{item} + IG_{cat}$.

4.2.1. Experiment one: Comparison on the clustering results

Fig. 9a shows the relative qualities of clustering results of k-todes, k-modes, k-modesT, Basic, and BasicT in real data set where $|D| = 100K$, $N^L = 3$, and $N^I = 58909$. In addition, the number of clusters k is 50. As described in [57], a term with a higher discrimination value will be associated with a longer distance between data points in the database. Because different items may belong to the same categories, the discrimination values of categories are lower than those of items for the

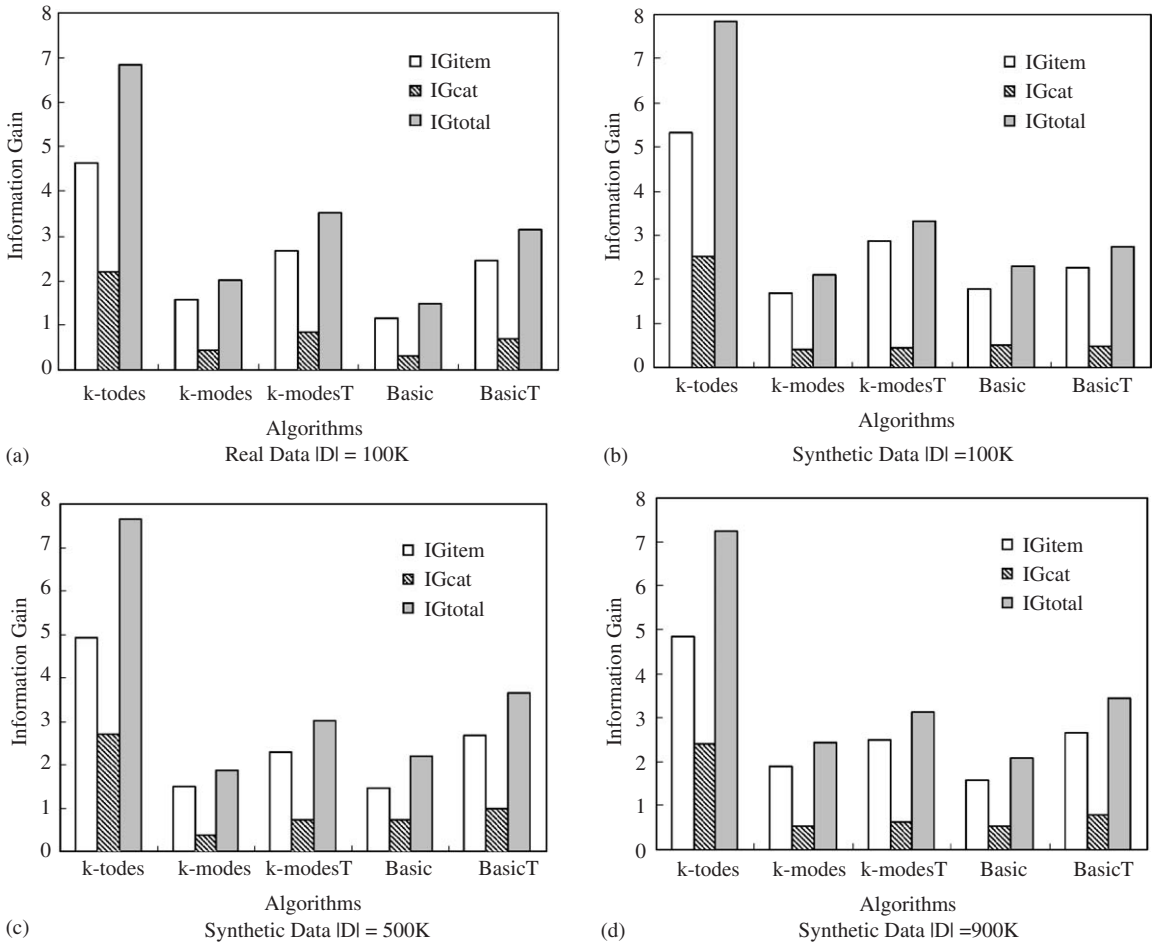

Fig. 9. The comparison of information gain values.

transactions in the database. For identifying the large and small terms in Basic and BasicT, the discrimination values of the items and the categories are aggregated in the similarity measurements for clustering market-basket data. Thus, BasicT obtains higher IG values than Basic. Similarly, k-modesT obtains higher IG values than k-modes. By considering the item similarities across their category levels, algorithm k-todes utilizes the category-based adherence measurement to allocate each transaction to a proper cluster so that k-todes in general outperforms other algorithms in the three IG values. To provide more insight into the performance comparisons of algorithms, we also conduct experiments on synthetic data set. In the experiments shown from Fig. 9b–d, we set $|D| = 100K$, $|T| = 5$, $|L| = 2K$, $|I| = 2$, $N^I = 60K$, $N^R = 100$, $N^L = 3$ with three different synthetic database sizes ($|D| = 100K$, $|D| = 500K$, and $|D| = 900K$).

4.2.2. Experiment two: when the database size $|D|$ varies

It is shown in Fig. 10, the scalability of k-todes is evaluated by both the real data and the synthetic data. By varying the real database size $|D|$ from 20 to $100K$, it is shown in Fig. 10a that k-todes significantly outperforms other algorithms in

execution efficiency. The execution time of k-todes increases linearly as the database size increases, indicating the good scale-up feature of algorithm k-todes. In the experiment shown in Fig. 10b, we set $|D| = 100K$, $|T| = 5$, $|L| = 2K$, $|I| = 2$, $N^I = 60K$, $N^R = 100$, $N^L = 3$, and $|D|$ varies from 100 to $900K$.

4.2.3. Experiment three: when the number of items N^I varies

In the synthetic data experiment shown in Fig. 11a, we set $|D| = 100K$, $|T| = 5$, $|L| = 2K$, $|I| = 2$, $N^R = 100$, $N^L = 3$, and N^I varies from 20 to $100K$. Similarly, in the synthetic data experiment shown in Fig. 11b, we set $|D| = 500K$, $|T| = 5$, $|L| = 2K$, $|I| = 2$, $N^R = 100$, $N^L = 3$, and N^I varies from 50 to $250K$. Note that each item could be viewed as a boolean attribute and N^I is thus viewed as the number of dimensions in the boolean space. With todes as the representatives, algorithm k-todes increases approximately linearly as the number of items increases.

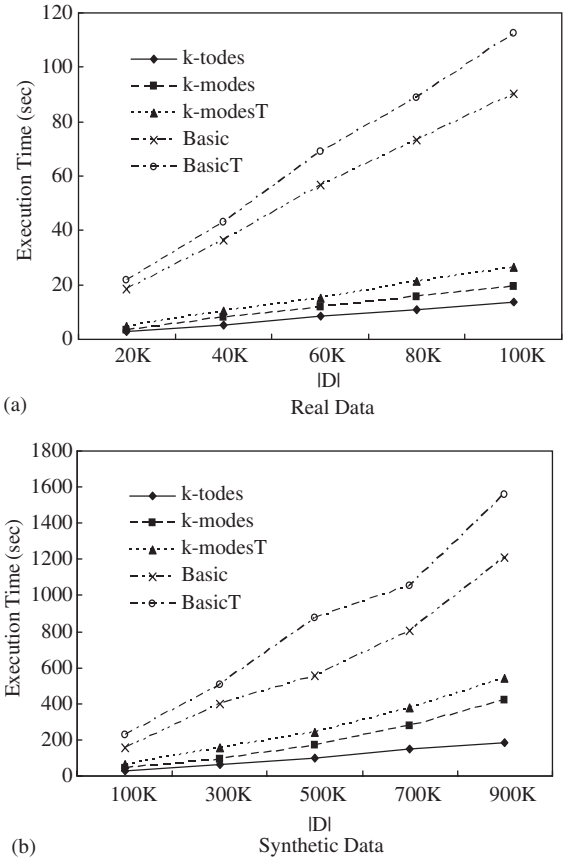

Fig. 10. Execution time for algorithms when the database size $|D|$ varies.

4.2.4. Experiment four: when the average size of maximal potential large itemsets $|I|$ varies

In the synthetic data experiments shown in Fig. 12, we set $|T| = 5$, $|L| = 2K$, $N^I = 60K$, $N^R = 100$, $N^L = 3$, and $|I|$ varies from 1 to 4. It is shown in Fig. 12a that when $|I|$ increases, the IG_{total} value also increases. This can be explained by the reason that when $|I|$ increases, the number of transactions containing co-occurrence itemsets increases and thus most transactions are allocated to the corresponding clusters with smaller adherences to the todes. Explicitly, many members of the transactions containing an item i_k are allocated to a cluster C_j because these transactions also

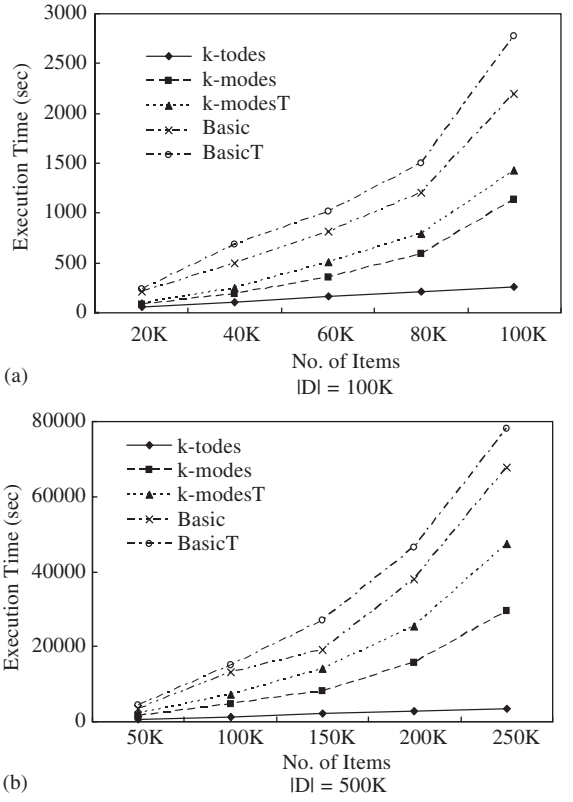

Fig. 11. Execution time when the number of items varies.

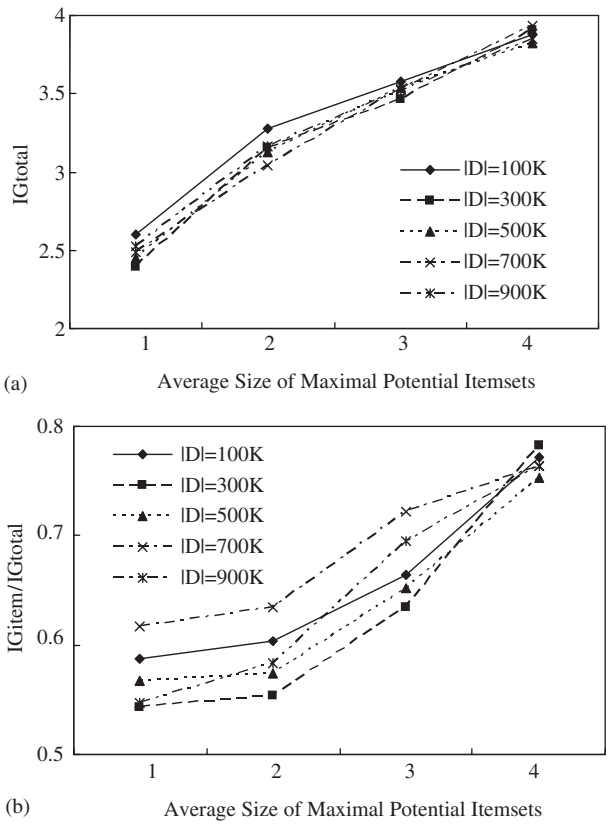

Fig. 12. When the average size of maximal potential large itemset $|I|$ varies.

contain other frequently-purchased items which are purchased together with i_k. When $|I|$ increases, the number of such items as i_k also increases so that more transactions containing i_k are allocated to one cluster instead of being allocated to several clusters separately. Therefore, the value of IG_{total} increases. In addition, it is shown in Fig. 12b that the percentage of IG_{item} in IG_{total} also increases when $|I|$ increases.

4.2.5. Experiment five: when the number of taxonomy levels N^L varies

In the synthetic data experiment shown in Fig. 13, we set $|T| = 5$, $|I| = 2$, $|L| = 2K$, $N^I = 60K$, $N^R = 100$, and N^L varies from 3 to 6. When N^L increases, k-todes has more category levels to distinguish the items by calculating their adherences. Thus, the percentage of IG_{cat} in IG_{total} increases, indicating the good feature of k-todes.

5. Conclusion

In this paper, we devised an efficient method to cluster market-basket data by identifying representative subjects. One of the important features of market-basket data sets is that they are generated at rapid pace and thus requires the data mining algorithms to be scalable and capable of dealing with the large data set. In view of the features of market-basket data, we devised in this paper a novel measurement, called the category-based adherence, and utilized this measurement to perform the clustering. With this category-based adherence measurement, we developed an efficient

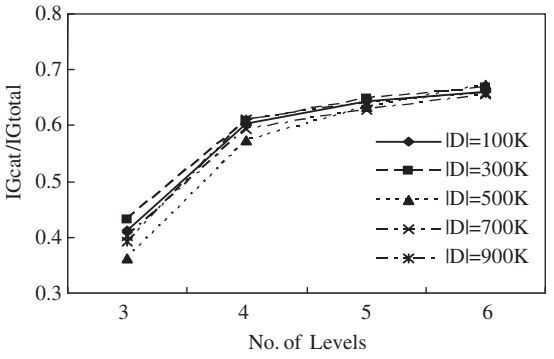

Fig. 13. When the number of taxonomy levels N^L varies.

clustering algorithm, called algorithm k-todes, for market-basket data with the objective to minimize the category-based adherence. A validation model based on Information Gain (IG) was also devised in this paper to assess the quality of clustering for market-basket data. As validated by both real and synthetic datasets, it was shown by our experimental results, with the taxonomy information, algorithm k-todes devised in this paper significantly outperforms the prior works in both the execution efficiency and the clustering quality for market-basket data.

Acknowledgements

The work was supported in part by the National Science Council of Taiwan, R.O.C., under Contracts NSC93-2752-E-002-006-PAE.

References

[1] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood cliffs, NJ, 1988.
[2] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surveys 31 (3) (1999).
[3] M. Charikar, C. Chekuri, T. Feder, R. Motwani, Incremental clustering and dynamic information retrieval, Proceedings of the 29th ACM Symposium on Theory of Computing, 1997.
[4] M.-S. Chen, J. Han, P.S. Yu, Data mining: an overview from a database perspective, IEEE Trans. on Knowledge and Data Eng. 8 (6) (1996) 833–866.
[5] D.H. Fisher, Knowledge acquisition via incremental conceptual clustering, Machine Learning 2 (2) (1987) 139–172.
[6] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Trans. on Pattern Anal. and Machine Intelligence, (2000) pp. 4–37.
[7] S.Y. Lu, K.S. Fu, A sentence-to-sentence clustering procedure for pattern analysis, IEEE Trans. Syst. Man Cybern. 8 (1978) 381–389.
[8] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, NY, 1981.
[9] R.C. Dubes, How many clusters are best?—an experiment, Pattern Recognition 20 (6) (1987) 645–663.
[10] C.-R. Lin, M.-S. Chen, On the Optimal Clustering of Sequential Data, in: Proceedings of the second SIAM International Conference on Data Mining, April 2002.
[11] B. King, Step-wise clustering procedures, J. Am. Stat. Assoc. 69 (1967) 86–101.
[12] P.H.A. Sneath, R.R. Sokal, Numerical Taxonomy, Freeman, London, UK, 1973.
[13] J. Hertz, A. Krogh, R.G. Palmer, Introduction to the Theory of Neural Computation, Westview Press, 1991.
[14] J. Tantrum, A. Murua, W. Stuetzle, Hierarchical Model-Based Clustering of Large Datasets Through Fractionation and Refractionation, in: Procreedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 2002.
[15] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman and Company, New York, 1979.
[16] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, Proceedings of the 20th VLDB Conference, September 1994, pp. 478–499.
[17] J. Han, Y. Fu, Discovery of multiple-level association rules from large databases, Proceedings of the 21st VLDB Conference, September 1995, pp. 420–431.
[18] J.-S. Park, M.-S. Chen, P.S. Yu, An effective hash based algorithm for mining association rules, Proceedings of the ACM SIGMOD Conference, May 1995, pp. 175–186.
[19] R. Srikant, R. Agrawal, Mining generalized association rules, Proceedings of the 21st VLDB Conference, September 1995, pp. 407–419.
[20] Z. Huang, Extensions to the k-means algorithm for clustering large data sets with categorical values, Data Mining and Knowledge Discovery 2 (3) (1998) 283–304.
[21] K. Wang, C. Xu, B. Liu, Clustering transactions using large items, Proceedings of the ACM CIKM International Conference on Information and Knowledge Management, 1999.
[22] S. Guha, R. Rastogi, K. Shim, ROCK: A robust clustering algorithm for categorical attributes, Journal of Information Systems 25 (5) (2000) 345–366.
[23] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.

[24] R.T. Ng, J. Han, Efficient and effective clustering methods for spatial data mining, Proceedings of the 20th Annual International Conference on Very Large Data Bases, 1994, pp. 144–155.
[25] T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: an efficient data clustering method for very large databases, ACM SIGMOD International Conference on Management of Data, vol. 25(2) June 1996, pp. 103–114.
[26] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD 96), August 1996, pp. 226–231.
[27] S. Guha, R. Rastogi, K. Shim, CURE: an efficient clustering algorithm for large databases, ACM SIGMOD International Conference on Management of Data, vol. 27(2), June 1998, pp. 73–84.
[28] C.-R. Lin, M.-S. Chen, A Robust and Efficient Clustering Algorithm based on Cohesion Self-Merging, Proceedings of the 8th ACM SIGKDD Intern'l Conference on Knowledge Discovery and Data Mining (KDD-2002), July 2002.
[29] C.C. Aggarwal, C.M. Procopiuc, J.L. Wolf, P.S. Yu, J.-S. Park, Fast algorithms for projected clustering, ACM SIGMOD International Conference on Management of Data, June 1999, pp. 61–72.
[30] C.C. Aggarwal, P.S. Yu, Finding generalized projected clusters in dimensional spaces, ACM SIGMOD International Conference on Management of Data, May 2000, pp. 70–81.
[31] A. Hinneburg, D.A. Keim, Optimal grid-clustering: towards breaking the curse of dimensionality in high-dimensional clustering, Proceedings of the 25th VLDB Conference, September 1999, pp. 506–517.
[32] H. Ralambondrainy, A conceptual version of the k-means algorithm, Pattern Recognition Lett. 16 (1995) 1147–1157.
[33] D. Barbara, Y. Li, J. Couto, COOLCAT: an entropy-based algorithm for categorical clustering, Proceedings of the 2002 ACM CIKM International Conference on Information and Knowledge Management, November 2002, pp. 590–599.
[34] K.-T. Chuang, M.-S. Chen, Clustering categorical data by utilizing the correlated-force ensemble, Proceedings of the 4th SIAM Conference on Data Mining, April 2004.
[35] M. Lebowitz, Experiments with incremental concept formation, Machine Learning 2 (2) (1987) 103–138.
[36] R.S. Michalski, R.E. Stepp, Automated construction of classifications: conceptual clustering versus numerical taxonomy, IEEE Trans. Pattern Anal. and Machine Intelligence 5 (4) (1983) 396–410.
[37] D. Gibson, J. Kleinberg, P. Raghavan, Clustering categorical data: an approach based on dynamical systems, Proceedings of the 24th Annual International Conference on Very Large Data Bases, 1998, pp. 311–322.
[38] V. Ganti, J. Gehrke, R. Ramakrishnan, CACTUS-clustering categorical data using summaries, Proceedings of ACM SIGKDD International Conference on Knowledge discovery and data mining, 1999.
[39] E.-H. Han, G. Karypis, V. Kumar, B. Mobasher, Clustering based on association rule hypergraphs, ACM SIGMOD'97 Workshop on Research Issues on Data Mining and Knowledge Discovery, 1997.
[40] W.A. Kosters, E. Marchiori, A.A.J. Oerlemans, Mining Clusters with Association Rules, Lecture Notes in Computer Science, 1642, 1999.
[41] B. Lent, A.N. Swami, J. Widom, Clustering Association Rules, Proceedings of the 13th International Conference on Data Engineering, April 1997, pp. 220–231.
[42] C. Ordonez, E. Omiecinski, FREM: fast and robust EM clustering for large data sets, Proceedings of the 2002 ACM CIKM International Conference on Information and Knowledge Management, November 2002, pp. 590–599.
[43] A. Strehl, J. Ghosh, A Scalable approach to balanced, high-dimensional clustering of market-baskets, Proceedings of the 7th International Conference on High Performance Computing, December 2000.
[44] Y. Xiao, M.H. Dunham, Interactive clustering for transaction data, Proceedings of the 3rd International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2001), September 2001.
[45] Y. Yang, X. Guan, J. You, CLOPE: a fast and effective clustering algorithm for transactional data, The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Poster), July 2002.
[46] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, Los Altas, CA, 2000.
[47] C. Fellbaum, WordNet: An Electronic Lexical Database, MIT Press, 1998.
[48] S. Scott, S. Matwin, Text classification using wordNet hypernyms, Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems, 1998, pp. 38–44.
[49] R. Duda, P. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[50] M. Halkidi, Y. Batistakis, M. Vazirgiannis, On clustering validation techniques, J. Intelligent Information Systems, 2001.
[51] O.R. Zaiane, A. Foss, C.-H. Lee, W. Wang, On Data Clustering Analysis: Scalability, Constraints and Validation, PAKDD02, 2002.
[52] UCI Machine Learning Repository. http:://www.ics.uci.edu/~mlearn/MLRepository.html.
[53] Reuters-21578 news collection, http://www.research.att.com/lewis/reuters21578.html.
[54] F.-X. Jollois, M. Nadif, Clustering Large Categorical Data, PAKDD02, 2002.
[55] J.R. Quinlan, Induction of decision trees, Machine Learning, 1986.
[56] J.R. Quinlan, C.45: programs for machine learning, Morgan Kaufmann, Los Atlas, CA, 1993.
[57] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley, Reading, MA, 1999.