A Review on Frequent Itemset Mining based on Partitioning Approach

[1]Ashok Kumar D, [2]Loraine Charlet Annie M. C.
[1,2]Government Arts College, Trichy, TamilNadu, India

Abstract
Data Mining is the process of extracting useful and nontrivial information from databases which helps the management to make proper decision to improve the sales. Frequent Itemset Mining helps the "supermarkets' Managementto" understand the customer needs better and they can react to customer needs faster. Fast and scalable frequent itemset mining techniques are increasingly becoming more important because of its complexity. In this paper, a survey of existing Frequent Itemset Mining algorithms is discussed. To improve the efficiency of these algorithms partitioning techniques can be used such that repeated database scanning can be avoided. Thus the existing partitioning approaches are also discussed in this paper.

Keywords
Frequent Itemset Mining, Association Rule Mining, Clustering, Market Basket Analysis

I. Introduction

Data mining deals with the automatic discovery of implicit information or knowledge within the databases. Big databases make the mining process more interesting but also more difficult. Studies of Frequent Itemset Mining is acknowledged in the data mining field because of its broad applications in mining association rules, correlations and graph pattern constraints. For example, frequently sold itemsets may be an interesting piece of information for supermarkets / hypermarkets and also for online stores. Knowing the frequently sold items can be used as a powerful advertising tool. Discounted prices on these items bundled with not very high selling items may boost sales. Finding frequent item sets is a NP-hard problem that can be solved in exponential time.

Supermarkets are very common nowadays and it contains large number of items so that one time purchase can be done easier which attracts the customers. Supermarkets contain different items based upon quality, price and different brands. The supermarkets can establish better customer relationship by giving them exactly what they need. The supermarkets can find, attract and retain customers; by ordering the items utilizing the acquired insight of customer requirements so that customers' valuable purchase time is reduced and more number of transactions done in lesser time. They can increase profitability by arranging/placing the items based on the frequent itemsets mined. They can even find the customer who might default to a competitor the company will try to retain the customer by providing promotional offers to the specific group of customer and thus reducing the risk of losing a customer or customers.

Market Basket Analysis is a modelling technique based upon the theory that if you buy a certain group of items, you are more or less likely to buy another group of items. A famous example about association rule mining is the "beer and diaper" purchase. A purported survey of behavior of supermarket shoppers discovered that customers presumably young men who buy diapers tend also to buy beer. This anecdote became popular as an example of how unexpected association rules might be found from everyday data. Daniel Powers says in 1992, Thomas Blischok, manager of a retail consulting group at Teradata, and his staff prepared an analysis of 1.2 million market baskets from about 25 Osco Drug stores. Database queries were developed to identify affinities. The analysis "did discover that between 5:00 and 7:00 p.m. those consumers bought beer and diapers". To improve the sales frequent items need to be identified which is very tedious due to various customer behaviours, trends and climatic conditions.

Databases involved in frequent itemset mining are very large, thus imperative to have fast algorithms for this task. The size of the collected data is substantial and even the detection of Strong rules requires sophisticated algorithms.

II. Frequent Itemset Mining Algorithms

The algorithms for performing market basket analysis are fairly straightforward and deal with the large amounts of transaction data that may be available. The efficiency of the algorithms will depend on the particular characteristics of the data sets. The size of the itemsets is a key factor in determining the performance of frequent itemset mining and association rule discovery algorithms.

Association mining is user-centric as the objective is the elicitation of useful or interesting rules from which new knowledge can be derived. The key characteristics of usefulness suggested in the literature are that the rules are novel, externally significant, unexpected, nontrivial, and actionable [16]. The majority of current research involves the specialization of fundamental association mining algorithms to address specific issues, such as the development of incremental algorithms to facilitate dynamic dataset mining or the inclusion of additional semantics such as time, space, ontologies, etc. to discover, for example, temporal or spatial association rules [17]. Relevant research can be organized into four groups:

- constraining the exploration, through the development and incorporation of measuresof interest (MOI) and efficient pruning strategies;
- reducing I/O, through hardware advances, enabling large datasets to become memory resident, and techniques such as intelligent sampling;
- creating useful data structures to make analysis more tractable.
- producing condensed inference sets allowing the entire result set to be inferred froma reduced set of inferences, reducing storage and facilitating user interpretation.

Over the past decade a variety of algorithms have been developed that address these issues through the refinement of search strategies, pruning techniques, data structures, and the use of alternative dataset organizations. While most algorithms focus on the explicit discovery of all inferences for a given dataset by incorporating domain knowledge. Here the existing algorithms are given a brief overview.

A. FP-Growth Algorithm

A compact data structure, called frequent pattern tree, or FP-tree is introduced for frequent itemset mining [18]. Based on this structure an FP-tree-based pattern fragment growth method was developed. FP-growth uses a combination of the vertical and

horizontal database layout to store the database in main memory. Instead of storing the cover for every item in the database, it stores the actual transactions from the database in a tree structure and every item has a linked list going through all transactions that contain that item.

Essentially, all transactions are stored in a tree data structure. A frequent pattern tree is a tree structure which consists of one root labeled as "root", a set of item prefix subtrees as the children of the root, and a frequent-item header table, Each node in the item prefix sub-tree consists of three fields: item-name, count and node-link, where item-name registers which item this node represents, count registers the number of transactions represented by the portion of the path reaching this node, and node-link links to the next node in the FP-tree carrying the same item-name, or null if there is none. Each entry in the frequent-item header table consists of item-name and. head of node-link, which points to the first node in the FP-tree carrying the item-name.

B. Broglet's FP-Growth

Broglet implemented an efficient FP-Growth [19] algorithm. The FP-growth in his implementation preprocesses the transaction database according to is as follows:

1. In an initial scan the frequencies of the items (support of single element item sets) are determined.
2. All infrequent items, that is, all items that appear in fewer transactions than a user-specified minimum number are discarded from the transactions, since, obviously, they can never be part of a frequent item set.
3. The items in each transaction are sorted, so that they are in descending order with respect to their frequency in the database.

The main issue is to maintain Complex data structures and and need to perform a lot of dereferencing.

C. Eclat

Eclat [20-21] algorithm is basically a depth-first search algorithm using set intersection. It uses a vertical database layout i.e. instead of explicitly listing all transactions; each item is stored together with its cover (also called tidlist) and uses the intersection based approach to compute the support of an itemset. In this way, the support of an itemset X can be easily computed by simply intersecting the covers of any two subsets Y, $Z \in X$, such that $Y \cup Z = X$. It states that, when the database is stored in the vertical layout, the support of a set can be counted much easier by simply intersecting the covers of two of its subsets that together give the set itself.

In this algorithm each frequent item is added in the output set. After that, for every such frequent item i, the i-projected database Diis created. This is done by first finding every item j that frequently occurs together with i. The support of this set {i, j} is computed by intersecting the covers of both items. If {i, j} is frequent, then j is inserted into D_i together with its cover. The reordering is performed at every recursion step. Then the algorithm is used recursively to find all frequent itemsets in the new database Di. It essentially generates the candidate itemsets using only the join step from Apriori. Again all the items in the database is reordered in ascending order of support to reduce the number of candidate itemsets that is generated, and hence, reduce the number of intersections that need to be computed and the total size of the covers of all generated itemsets. Since the algorithm doesn't fully exploit the monotonicity property, but generates a candidate itemset based on only two of its subsets, the number of candidate itemsets that are generated is much larger as compared to a breadth-first approach such as Apriori. Eclat essentially generates candidate itemsets using only the join step. A technique that is regularly used is to reorder the items in support ascending order to reduce the number of candidate itemsets that is generated. In Eclat, such reordering can be performed at every recursion step in the algorithm. Also that at a certain depth d, the covers of at most all n-itemsets with the same n-1 prefix are stored in main memory, with $n < d$. Because of the item reordering, this number is kept small.

The main issues are the algorithm requires the original database to be stored in main memory and it need to store covers of n-Itemsets and it doesn't fully exploit the monotonicity property, so that the number of candidate itemsets that are generated is much larger as compared to a breadth-first approach such as Apriori.

D. SaM Algorithm

The SaM (Split and Merge) algorithm [22] has a single transaction list in horizontal representation is stored as an array. This array is processed with a simple split and merge scheme, which computes a conditional database, processes this conditional database recursively, and finally eliminates the split item from the original database. SaMpreprocesses a given transaction using the steps given below:

1. The transaction database is taken in its original form.
2. The frequencies of individual items are determined from this input in order to be able to discard infrequent items immediately.
3. The (frequent) items in each transaction are sorted according to their frequency in the transaction database, since it is well known that processing the items in the order of increasing frequency usually leads to the shortest execution times.
4. The transactions are sorted lexicographically into descending order, with item comparisons again being decided by the item frequencies; here the item with the higher frequency precedes the item with the lower frequency.
5. The data structure on which SaM operates is built by combining equal transactions and setting up an array, in which each element consists of two fields: An occurrence counter and a pointer to the sorted transaction array of contained items. This data structure is then processed recursively to find the frequent item sets. The basic operations of the recursive processing are based on depth-first/divide-and conquer scheme.

In the split step the given array is split with respect to the leading item of the first transaction. All array elements referring to transactions starting with this item are transferred to a new array. The new array created in the split step and the rest of the original arrays are combined with a procedure that is almost identical to one phase of the well-known merge sort algorithm. The main reason for the merge operation in SaM is to keep the list sorted, so that all transactions with the same leading item are grouped together and equal transactions or transaction suffixes can be combined, thus reducing the number of objects to process.

Each transaction is represented as a simple array of item identifiers (which are integer numbers). The transaction list is prepared which are stored in a simple array, each element ofwhich contains a support counter and a pointer to the head of the list. The list elements themselves consist only of a successor pointer and a pointer to the transaction. The transactions are inserted one by one into this structure by simply using their leading item as an index.

E. Apriori Algorithm

Apriori algorithm is an influential algorithm for mining frequent itemsetsfor Boolean association rules [24]. Uses a Level-wise search, where n-itemsetsare used to explore (n+1)-itemsets, to mine frequent itemsets from transactional database for Boolean association rules. An itemset that contains n items is an-itemset. Reducing the search space to avoid finding of each Ln requires one full scan of the database. If an itemset I does not satisfy the minimum support threshold, min_sup, the Iis not frequent, that is, P (I) <min_sup. If an item A is added to the itemset I, then the resulting itemset (i.e., I∪A) cannot occur more frequently than I. Therefore, I∪A is not frequent either, that is, P (I∪A) <min_sup.

First, the set of frequent 1-itemsets is found. This set is denoted L_1. L_1 is used to find L_2, the frequent 2-itemsets, which is used to find L_3, and so on, until no more frequent k-itemsets can be found. Finding of each Ln requires one full scan of the database. To improve the efficiency of the level-wise generation of frequent itemsets, an important property called the Apriori property is used. Apriori property means "All non-empty subsets of a frequent itemset must also be frequent". This is made possible because of the anti-monotone property of support measure - the support for an itemset never exceeds the support for its subsets.

A two-step process is followed, consisting of join and prune actions.

1. The Join Step

n-itemsets candidate set L_n is generated by joining L_{n-1} with itself and this set of candidates is represented here as Cd_n. Consider l_1 and l_2 are the itemsets in L_{n-1}. The notation $l_i[j]$ refers to the jth item in l_i. Apriori assumes that itemsets are sorted in increasing lexicographic order. The join, is performed, where members of L_{n-1} are joinable if their first (n - 2) items are in common. The condition $l_1[n - 1] < l_2[n - 1]$ simply ensures that no duplicates are generated.

2. The Prune Step

Cd_n is a superset of L_n, it means that its members may or may not be frequent, but all of the frequent n-itemsets are included in Cd_n. To determine the count of each candidate in Cd_n a database scan is done which results in the determination of L_n i.e., all candidates having a count no less than the minimum support count are frequent by definition, and therefore belong to L_n. Cd_n will be huge, and so this involves heavy computations. To reduce the size of Cd_n, the Apriori property is used here as that any (n-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset. Hence, if any (n-1)-subset of a candidate n-itemset is not in L_{n-1}, then the candidate cannot be frequent either and so can be removed from Cd_n. This subset testing can be done quickly by maintaining a hash tree of all frequent itemsets.

The first and arguably most influential algorithm for efficient association rule discovery is Apriori. Apriori-inspired algorithms show good performance with sparse datasets such as market basket data. As the minimum threshold becomes lower; an exponentially large number of itemsets are generated.

III. Binary Representation

In supermarkets/hypermarkets, the databases are very large and the item's purchases can be represented in binary format it means 0 and 1 easily. Boolean variables are used to represent the presence or absence of the item or attribute in the purchase / transaction / basket.

In this paper, Homogenous Binary databases are considered for market basket databases. These are databases which can be represented in binary format. In this binary format, each entry is either 0 or 1, and the columns or the rows can be stored as bit vectors. A transaction data set has a natural tabular representation obtained by using bit vectors. Consider a 0 /1-table twhose columns correspond to the items i_1; : : : ; i_m; the rows of tcorrespond to the transactions of T. If t_{pq} is the entry of tlocated in row p and column q, then

$$t_{pq} = \begin{cases} 1, & if\ i_q \epsilon t_p \\ 0, & otherwise \end{cases}$$

The column of tthat corresponds to an item i is a bit vector b_i. so that $b^i \epsilon \{0,1\}^n$. The notion of bit vector has been frequently used for discovery of association rules. The notion of bit vector of an item is extended her to bit vectors for item sets.

Binary data are used in information systems because data are categorized for attributes value by 0 and 1 to represent either a category absence or presenceof the attribute. Binary data sets are interesting and useful for a variety reasons. It can be used to represent categorical data. Close ended questions (yes/no questions) can be used easily for different categories. For instance, surveys can be easily conducted with yes/no questions for all the categories we want to analyse so that people can easily and effectively participate in surveys; which can be directly used as binary data. From a clustering point of view they offer several advantages. There is no concept of noise like that of quantitative data, they can be used to represent categorical data and they can be efficiently stored, indexed and retrieved. They reflect the qualitative nature of data.

Each basket is represented by a row of the table. If $item_j$ is present in basket number i, then the entry located in row i and column j of the table is set to 1; otherwise, that entry is set to 0. A market basket database is stored using bit vectors $b^{item}{}_1 : : : b^{item}{}_n$ as shown in Table 1. For example, basket no. 1 contains the items $item_2$, and $item_n$. And basket no. 2 contains $item_2$, additionally $item_1$ but not $item_n$.

Table 1: Market Basket Data Representation

Basket Number	$Item_1$	$Item_2$	……	$Item_n$
1	0	1	……	1
2	1	1	……	0
3	1	0	……	0
4	0	0	……	1
……				
m	0	1	……	0

IV. Clustering

Clustering algorithms partition a data set into several dis-joint groups such that points in the same group are similar to each other according to some similarity metric. Binary Data Clustering means that all the objects in the dataset are labelled as either zero or one and clustered based on the similarity on the attributes.

Typical applications for binary data clustering include market basket data clustering and document clustering. For market-basket data, each data transaction can be represented as a binary vector where each element indicates whether or not any of the corresponding item/product exists. For a document in web mining, each element indicates whether a given word/term was present or not.

Market basket analysis' ultimate goal is to satisfy the customer effectively. If the customers are segmented means then the frequent itemset mining process will be easier and efficient. To solve this, frequent itemset mining technique is combined here with the clustering technique. Initially, the binary data is clustered and then frequent itemset mining is done on the binary data. The existing algorithms for binary data clustering are explained in the following section.

In literature, the clustering techniques forbinary data are analysed in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. Existing clusteringapproaches for binary data are given a brief overview.

A. Partitional Clustering

Partitional Clusteringdecomposes the dataset into a set of disjoint clusters. It determines an integer number of partitions that optimises a certain criterion function. It selects a criterion and evaluates it for all possible partitions containing K clusters. Symmetric similarity measure like Euclidean distance on binary points is acceptable, since a 0/0 match is as important as a 1/1 match on some dimension [5]. Optimized approach for partitional clustering is to start with an initial partition and move objects so that the value of the criterion function improves. The criterion function may emphasize the local or global structure of the data and its optimisation is an iterative procedure. It is further subdivided into medoid based and centroid based clustering methods. In medoid based method, the cluster is represented by one of its points. Whereas, in centroid based the mean or average is used to represent the cluster. General procedure for partitioning methods is given below.

1. Initial Partition

- Select k seed points at random or by taking the centroid as the first seed point and the rest at a certain minimum distance from this seed point.
- Cluster the remaining points to the closest seed point.

2. Algorithm

- Select an initial partition with K clusters. Repeat steps 2 through 5 until the cluster membership stabilizes.
- Generate a new partition by assigning each pattern to its closest cluster center.
- Compute new cluster centers as the centroids of the clusters.
- Repeat steps 2 and 3 until an optimum value of the criterion is found.
- Adjust the number of clusters by merging and splitting existing clusters or by removing small or outlier clusters.

To obtain a partition which, for a fixed number of clusters, minimizes the square-error where square-error is the sum of the Euclidean distances between each pattern and its cluster center. Data partitioning algorithms divides data into subsets. Since checking all possible subset systems is computationally not feasible, certain greedy heuristics are used in the form of iterative optimization. Specifically, this means different relocation schemes that iteratively reassign points between the K clusters. Unlike traditional hierarchical methods, in which clusters are not revisited after being constructed, partitional algorithms gradually improve clusters. With appropriate input parameters this results in high quality clusters. One approach to data partitioning is to take a conceptual point of view that identifies the cluster with a certain model whose unknown parameters have to be found.

3. Problem

- Combinatorial blow-up of the number of partitions to be evaluated.
- Might converge to local minima.

Another approach for partitioning starts with the definition of objective function depending on a partition. The pairwise distances or similarities can be used to compute measures of inter and intra cluster relations. Depending on how representatives are constructed, iterative optimization partitioning algorithms are subdivided into medoid and centroid based methods. Medoid based methods have the most appropriate data point within a cluster that represents it. In K-means algorithm, a cluster is represented by its centroid, which is a mean of points within a cluster. Centroids have the advantage of clear geometric and statistical meaning.

B. K-means Algorithm

The K-means algorithm [14, 25] is the most popular clustering tool used in scientific and industrial applications. The name comes from representing each of K clusters by the mean or weighted average Z of its points, the so called centroid. While this obviously does not work well with categorical attributes, it has the good geometric and statistical sense for numerical attributes. The sum of discrepancies between a point and its centroid expressed through appropriate distance is used as the objective function.

Two versions of K-means iterative optimization are known. The first version is similar to Expectation Maximization (EM) [23] algorithm called Forgy's clustering algorithm and consists of two-step major iterations

- Reassign all the points to their nearest centroids.
- Recompute centroids of newly assembled groups.

Iterations continue until a stopping criterion is achieved with no reassignments. The second, classic in iterative optimization version of K-means iterative optimization reassigns points based on more detailed analysis of effects on the objective function caused by moving a point from its current cluster to a potentially new one. If a move has a positive effect, the point is relocated and the two centroids are recomputed.

Reassigning points and immediately re-computing centroids works much better. The wide popularity of K-means algorithm is well deserved. It is simple, straightforward, and is based on the firm foundation of analysis of variances. When the data are continuous and normally distributed K-means algorithm performs well.

Partitional clustering approaches are in high need since it works for all data types like continuous (factor scores for example), binary indicators (0/1 flags) and ranks (1=low, 2=middle, 3=high).

Binary data streams are clustered by [5] On-line K-means, Scalable K-means and Incremental K-means algorithms. Here clustering is based on mean-based initialization and incremental learning. These algorithms use extra data structures like summary tables and have some extra matrix operations which will increase the processing time and memory usage. From the study it is observed that the standard K-means is best for normal binary data clustering.

C. Tao Li's Model

Tao Li [12]developed an optimization procedure for binary data clustering. Clustering is based upon two matrices, data matrix and feature matrix. The data matrix specifies the cluster memberships for the corresponding data (respectively, features). The feature matrix specifies the cluster memberships for the corresponding feature. For clustering, it is customary to assume that each data point is assigned to one and only one cluster. Approximation of

the original data is found by the dot product of data and feature matrices. Using Frobenius norm for approximation data matrix and the original data matrix, optimization is done to minimize the mean squared error. In this way binary data clustering is transformed into the computation of dataand feature matrix which minimizes the mean squared error.

The objective function can be minimized using an alternating least-squares procedure by alternatively optimizing one of the matriceswhile fixing the other. Given an estimate of feature matrix, new least-squares estimates of the entries of data matrixcan be determined by assigning each data point to the closest cluster. If a feature has similar association to all clusters, then it is viewed as an outlier at the current stage. The optimization procedure for minimizing squared error alternates between updating data matrixand feature matrices with one of them fixed. After each iteration, the objective criterion i.e squared error is computed. If the value is decreased, then repeat the above process; otherwise, the process has arrived at a local minimum.

The main issues for this binary data clustering procedure are approximations are used for clustering which converges to local minimum.

D. Luke's Model

Brian T.Luke [13] suggested that the bit strings that characterize two objects can also be used to calculate a "distance." This effective distance can then be used with a clustering algorithm to place the objects into groups. One of the most popular methods for calculating the distance between two objects is to first determine their similarity. If a general similarity metric is denoted S, it usually has a value between 0.0 (the bit strings have no ON bits in common) and 1.0 (the two bit strings are identical). A distance can then be given by

A(1-S)M , where M is a non-negative exponent and A is a constant. Conversely, each bit string can be thought of as components of an L-dimensional vector and the similarity as just the cosine of the angle between them (which is actually the case for the Ochini similarity). Therefore, a distance constant times the value of this angle.

If the bit string has a length of L, it is possible to go down this string and count the number of times a bit is ON in both strings, ON in one and OFF in the other, or OFF in both strings. The four sums are presented in the table below.

Table 2: Chemical Molecule Representation

Object	0	1
0	B_{00}	B_{01}
1	B_{10}	B_{11}

Some other symbols that will be used in this section are

B_i (= B_{10} + B_{11}) is the total number of ON bits in object i.

B_j (= B_{01} + B_{11}) is the total number of ON bits in object j.

B_c (= B_{11}) is the total number of times a bit is ON in both bit strings.

B_l (= B_{00} + B_{11}) is the total number of times the two bit strings agree.

L (= B_{00} + B_{01} + B_{10} + B_{11}) is the length of the bit string.

A dissimilarity metric, D, is a measure of how different two bit strings are, and can therefore be scaled by any constant to yield a distance. This metric has a value of 0.0 for identical bit strings and 1.0 for strings with no bits in common. These can easily be formed by D=(1-S), where S is a similarity value from the table above. Dissimilarities measures are proportional to either the sum or the product if the disagreement between the bit strings, B_{10} and B_{01}, that is, $B_{10}+B_{01}=L-B_l$.

A Squared Euclidean Substructure Distance, DSES. If compound i is smaller than compound j (less bits turned ON), DSESi-j may yield a more accurate distance. Similariy, if object j has less bits turned ON than i (Bj < Bi), $DSES_{j-i}$ can be used.

With any of these distance metrics, a clustering algorithm can be used to place these binary objects into groups. Since it is not possible to calculate an average position between two binary objects (unless they are identical), one cannot determine a centroid for a group of binary objects. To circumvent this, a medoid can be used instead. A medoid is simply a bit string that minimizes the sum of the distance to all objects in the group, and can be used with Wards Method in Agglomerative Linkagesand K-means clustering.

To cluster nucleotide sequences due to the large variation in the number of 1's for different bit strings, Brian Luke T. suggests the following distance metric produced better results.

$D'_{SE} = B_j(B_i-B_c)/Bi^2$

One final point is that it is possible to weight each bit in the string. If bim is the bit in position m(m=1,2,...L) for object i, it can be weighted by w_k.

The main issues of Luke's model areweight is assigned for each bit which can be related to the "inverse frequency" of being turned on. In other words, if this bit is on for all N objects to be clustered, it cannot be used as a discriminator. Conversely, if it is on in only a few of the objects, it may be an important discriminator and should be given a relatively large weight. One such inverse frequency weight is given by the expression

$w_k = \log(N/f_k)$

where f_k is the number of objects that have this bit turned on.Here weights are provided for the chemical molecules but for items which are used in market databases, weights can't be provided. Our goal is to find the frequency of items from the very large market database with the unknown parameters.

E. Latent Class (LC) Model

Latent Class (LC) Model is a method that uses categorical variables todiscover hidden or latent groups. LC Model [26-29] the subjectsinto classes where the manifest variables are unrelated. Latent ClassModel are widely used for analyzing correlated binary data. It canbe used in market segmentation and medical research. LC Model is amodel based approach [27, 45-46]. This means that a statistical modelis postulated for the population from which the data sample is obtained.LC Model clusters the objects based on their response patterns. Moreprecisely, it is assumed that a mixture of underlying probability distributions generates the data. In LC Model, a K-class latent variableis used to explain the associations among a set of observed variables. Each latent class, like each cluster, groups together similar cases. Thefundamental assumption underlying LC models is that of local independence which states that objects (persons, cases) in the same latentclass share a common joint probability distribution among the observedvariables. Since persons in the same latent class or cluster cannot bedistinguished from each other based on their observed responses, theyare similar to each other, homogeneous with respect to these observedvariables. Persons are classified into that class having the highest posterior membership probability of belonging given the set of responsesfor that case.

LC Model are used to explain the associations in situations whereit is plausible to assume existence of some latent categorical

variable.For instance, in the case of diagnosing illness with multiple symptomsit can be hypothesized that there are two latent classes representing theabsence or presence of the disease [42]. Those who are really sick havea higher chance of showing the disease symptoms than those who arenot. LC Model postulate the existence of a latent categorical variableand each individual in the sample can be classified as a member of oneof the latent classes. In the diagnostic test situation,one could considerthere being two latent classes, diseased and non-diseased.When using the maximum likelihood method for parameter estimation, the clustering problem involves maximizing a log-likelihoodfunction. This is similar to standard non-hierarchical cluster techniquessuch as K-means clustering, in which the allocation of objects to clustersshould be optimal according to some criteria. These criteria typicallyinvolve minimizing the within-cluster variation or equivalently maximizing the between-cluster variation. LC Model assumes that the relationship among the binary observed variables is accounted completelyby the latent categorical variable. The observed variables are independent conditional on the latent class membership. This is refered asconditional independence or local independence. LC Model is usedwhen the data is best collected in frequencies of observations in categories.

The main issue of latent class model is it uses a mixture modelwith binary responses on each subject that are independent conditionalon cluster membership. However, in many practical applications, theresponses are correlated because they are observed on the same subject;this is known as local dependence leads to local optimum. LC Modelassumes that the observed variables are independent conditional on thelatent membership, but this is rarely valid in practice.

IV. FISM and Clustering

A partitioning technique can be used to find candidate itemsets [21] in frequent itemset mining so that high informativeFrequent itemsets and Association rules can be generated effectively. It consists of two phases. In Phase I, the algorithm subdivides the transactions of database into clusters using binary data clustering techniques. In Phase II, for each partition (cluster), all frequent itemsets within the partition are found and corresponding Association rules are derived. As the market database is partitioned, main memory can hold the partition fully so that repeated database scanning is reduced.For K partitions K database scanning is done hence the overall frequent itemset mining efficiency is improved. As the clustering technique is used to partition the database, its similarity property helps to find proper association rules.

V. Conclusion

Efficient algorithms for mining frequent itemsets are crucial for mining association rules. The main aim is to optimize the process of finding itemsets which should be efficient, scalable and can detect the important itemsets by partitioning the database. Reviews of various itemset mining algorithms with its pros and cons are reviewed here. The clustering should be efficient so that the clusters generated are highly similar and scalable which means time complexity should be lesser. The clustering algorithms used to cluster categorical/binary data were discussed with its merits and demerits in this paper. Also it gives better directions to use simple, effective and fast Partitional clustering approaches for handling binary data.

References

[1] Top 10 algorithms in data mining, XindongWu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou Michael Steinbach, David J. Hand, Dan Steinberg, 4 KnowlInfSyst 14, pp. 1–37, Springer-Verlag London Limited, 2008.

[2] Johannes Grabmeier, Andreas Rudolph,"Techniques of Cluster Algorithms in Data Mining", Data Mining and Knowledge Discovery, 6, pp. 303–360, 2002.

[3] Jain, A.K., Dubes, R.C.(1988),"Algorithms for Clustering Data", Prentice-Hall Inc, Everitt. B., Cluster Analysis (3rd Ed.), Edward Arnold,London, UK, 1993.

[4] Jain.A.K, Murty.M.N. Flynn.P.J.,"Data Clustering: A Review, ACM Computing Surveys", 31, 3, pp. 264-323, 1999.

[5] Kaufman.L., Rousseeuw.P.J.,"Finding Groups in Data -An Introduction to Cluster Analysis", Wiley, 1990.

[6] Carlos Ordonez,"Clustering Binary Data Streams with K-means", DMKD03: 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2003, San Diego, USA.

[7] Kaufman, L., Rousseeuw, P.J., 1990,"Finding Groups in Data: An Introduction to Cluster Analysis", John Wiley & Sons] and the algorithm CLARA (Clustering LARge Applications) [Kaufman, L., Rousseeuw, P.J., 1990,"Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons.

[8] Kaufman, L., Rousseeuw, P.J., 1990,"Finding Groups in Data: An Introduction to Cluster Analysis", John Wiley & Sons] and the algorithm CLARA (Clustering LARge Applications) [Kaufman, L., Rousseeuw, P.J.,"Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, 1990.

[9] Z. Huang,"Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values", Data Mining and Knowledge Discovery 2(3), pp. 283-304, 1998.

[10] Z. Huang,"A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining", In Proceedings of the ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, 1997.

[11] Z. Huang,"Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values", Data Mining and Knowledge Discovery 2(3), pp. 283-304, 1998.

[12] A General Model for Clustering Binary Data, Tao Li, KDD'05, August, 2005, ACM.

[13] Brian T.Luke, Luke, B. T.,"Clustering Binary Objects", Lin, D., An Information-Theoretic Definition of Similarity. Toit, du S.H.C.; Steyn, A.G.W.; Stumpf, R.H.; Graphical Exploratory Data Analysis; Chapter 3, p. 77, 1986; Springer-Verlag.

[14] Hartigan.J., Wong.M., A K-Means Clustering Algorithm, Applied Statistics, 28, pp. 100-108, 1979, Highleyman.W.H., The Design and Analysis of Pattern Recognition Experiments, Bell Syst. Tech,J., vol. 41, pp. 723-727, 1962.

[15] Selim.S.Z. and Ismail.M.A., K-Means Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality, IEEE Transactions On Pattern Analysis and Machine Intelligence, vol. 6, No.1, pp. 81-87, 1984.

[16] Abraham Silberschatz, Alexander Tuzhilin: What Makes Patterns Interesting in Knowledge Discovery Systems. IEEE Transactions on Knowledge Data Engineering 8(6), pp. 970-974, 1996.

[17] Hilderman, R.J., and Hamilton, H.J. Measuring the Interestingness of Discovered Knowledge: A Principled Approach,Intelligent Data Analysis, 7(4): 347-382, 2003.
[18] J. Han, H. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. In: Proc. Conf. on the Management of Data (SIGMOD'00, Dallas, TX). ACM Press, New York, NY, USA 2000.
[19] J. Han, H. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. In: Proc. Conf. on the Management of Data (SIGMOD'00, Dallas, TX). ACM Press, New York, NY, USA 2000.
[20] M.J. Zaki. Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engineering, 12(3):372–390, May/June 2000.
[21] J. Han, and M. Kamber, 2000. Data Mining Concepts and Techniques. Morgan Kanufmann.
[22] C. Borgelt. SaM: Simple Algorithms for Frequent Item Set Mining. IFSA/EUSFLAT 2009 conference- 2009.
[23] Mclachlan.G. and Krishnan.T., The EM Algorithm and Extensions. John Wiley & Sons, New York, 1997.
[24] Agrawal R, Srikant R, Fast algorithms for mining association rules, In Proceedings of the 20th VLDB conference,pp.487 - 499, 1994.
[25] McQueen J, "Some methods for classi_cation and analysis of multivariate observations", Proc. 5th Berkeley SympMathematics,statistics and probability, 281-296, 1967.
[26] Flaherty Brian P., Ki_ Cara J., Cooper Harris, Camic, Paul M., Long Debra L., Panter A. T., Rindskopf David, Sher Kenneth J., Latent class and latent pro_le models., APA handbook of research methods in psychology, Vol 3: Data analysis and research publication., pp. 391-404, Washington, DC, US: American Psychological Association, 2012.
[27] HuipingXu, "Latent Variable Modeling of Multivariate Binary Data Using General Correlation Structures", Purdue University, Indiana, 2007.
[28] Jay Magidson and Jeroen K. Vermunt, "Latent class models for clustering:A comparison with K-means", Canadian Journal of Marketing Research", Volume 20, pp.37-44, 2002.
[29] Jay Magidson and Jeroen K. Vermunt, "Latent class models", Statistical Innovations Inc.,Belmont, 2003.

D. Ashok Kumar did his Master degree in Mathematics and Computer Applications in 1995 and completed Ph.D., on Intelligent Partitional Clustering Algorithm's in 2008, from Gandhigram Rural Institute–Deemed University, Gandhigram, Tamilnadu, INDIA. He is currently working as Senior Grade Assistant Professor and Head in the Department of Computer Science, Government Arts College, Trichy- 620 022, Tamilnadu, INDIA. His research interest includes Pattern Recognition and Data Mining by various soft computing approaches viz., Neural Networks, Genetic Algorithms, Fuzzy Logic, Rough set, etc.,.

Loraine Charlet Annie M.C., completed her M.E. in Computer Science and Engineering in Noorul Islam College of Engineering affiliated to Anna University, India. Currently doing research in data mining techniques at ManonmaniamSundaranar University, Tirunelveli, India. Her research interests include Pattern Recognition, Data Mining and Association Rule mining.