International Journal of Computational Intelligence and Informatics, Vol. 1 : No. 1, April – June 2011

Frequent Itemset Mining of Market Basket Data using K-Apriori Algorithm

D. Ashok Kumar
Department of Computer Science,
Government Arts College,
Trichy-, 620 022,
Tamil Nadu, India
akudaiyar@yahoo.com

M. C. Loraine Charlet Annie
Department of Computer Science,
Government Arts College,
Trichy-, 620 022,
Tamil Nadu, 620 022, India
lorainecharlet@gmail.com

Abstract

The constant advance in computing power makes data collection and storage easier, so that the databases are tend to be very big. Market basket databases are very large binary databases. To identify the frequently bought items from the market basket data, a novel approach called K-Apriori algorithm is proposed here, in which binary data is clustered based on a linear data transformation technique, then frequent item sets and association rules are mined. Experimental results show that the proposal has higher performance in terms of objectivity and subjectivity.

Keywords: *Wiener, Frequent itemsets, Association rules*

1. Introduction

Mining frequent itemsets and association rules is a popular method for discovering interesting relations between variables in very large databases. The implicit information within databases, mainly the interesting association relationships among sets of objects that lead to association rules which disclose useful patterns for decision support, financial forecast, marketing policies, even medical diagnosis and many other applications. Frequent itemsets play an essential role in many data mining tasks that try to find interesting patterns from databases such as association rules[1], correlations, sequences, classifiers, clusters and many more of which the mining of association rules is one of the most popular problems.

Association rule mining finds interesting association or correlation relationships among a large set of data items [2], [4]. The original motivation for searching association rules came from the need to analyze the supermarket transaction data, that is, to examine customer behavior in terms of the purchased products. Association rules describe how often items are purchased together. Finding all such rules is valuable to segment the items based on the buying patterns of the Customers. From the Market Basket Analysis, sales people can learn more about customer behavior, can find which products perform similarly to each other, which products should be placed near each other and which products should be cross-hold.

For example, consider $X = \{X_1, X_2, \ldots, X_q\}$ be a set of literals, called items. Let X be a set of transactions, where each transaction T is a set of items such that $T \subseteq X$. Consider a transaction T contains R, a set of some items in X, if $R \subseteq T$. An association rule is an implication of the form $R \rightarrow S$, where $R \subset X$, $S \subset X$ and $R \cap S$. The rule $R \rightarrow S$ holds in the transaction set X with confidence c if $c\%$ of transactions in X that contain R also contain S. The rule $R \rightarrow S$ has support s in the transaction set X if $s\%$ of transactions in X contains $R \cup S$. Given a set of transactions X, the problem of mining association rules is to generate all association rules that have support and confidence greater than the user-specified minimum support called *min_sup* and minimum confidence called *min_conf* respectively.

The support of an itemset is defined as the proportion of transactions in the data set which contain the itemset. Confidence is defined as the measure of certainty or trustworthiness associated with each discovered pattern. Rules that have both high confidence and support are called strong rules.

For frequent itemset generation and deriving association rules from them, large number of algorithms like Apriori [2], FP-Growth Algorithm [4] and Eclat [5] are available. The first and arguably most influential algorithm for efficient frequent itemset mining and association rule discovery is Apriori. Apriori-inspired algorithms show good performance with sparse datasets such as market basket data so Apriori algorithm is considered here. The Apriori algorithm extracts a set of frequent itemsets from the data, and then pulls out the association rules with the highest information content.

2. The Apriori algorithm

Apriori is an influential algorithm for mining frequent itemsets for Boolean association rules. Algorithm uses prior knowledge of frequent itemset properties [5]. Apriori requires that input and output fields all be categorical.

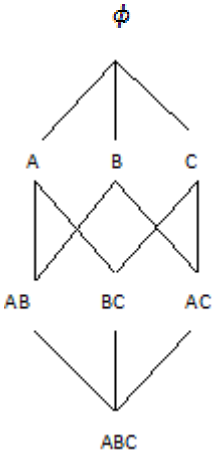
Apriori, developed in [2], is a level-wise, breadth-first algorithm which counts transactions. Apriori employs an iterative approach known as a level-wise search, where *n-itemsets* are used to explore *(n+1) - itemsets*. First, the set of frequent *1-itemsets* is found. This set is denoted L_1. L_1 is used to find L_2, the frequent *2-itemsets*, which is used to find L_3, and so on, until no more frequent *n-itemsets* can be found. Finding of each L_n 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 that is the support for an itemset never exceeds the support for its subsets.

Algorithm 1: Apriori algorithm for Frequent Itemset Mining

C_n: Candidate itemset of size n
L_n: frequent itemset of size n
L_1 = {frequent items};
For (n=1; L_n !=Φ ; n++)
Do begin
C_{n+1} = candidates generated from L_n;
For each transaction T in database do
Increment the count of all candidates in C_{n+1}
that are contained in T
L_{n+1}= candidates in C_{n+1} with min_sup
End
Return $\cup_n L_n$

2.1 Data structure

The data structure used in the Apriori algorithm is a root directed tree. The root is defined to be at depth 0, and a node at depth *d* can point to nodes at depth *d+1*. If node *u* points to node *v* then *u* is the parent of *v*, and *v* is a child node of *u*.



Every leaf *l* represents a word which is the concatenation of the letters in the path from the root to *l*. Note that if the first '*g*' letters are the same in two words, then the first *g* steps on their paths are the same as well. In finite ordered sets, a link is labeled by an element of the set, and the tree contains a set if there exists a path where the links are labeled by the elements of the set, in increasing order.

2.2 Generating association rules from frequent itemsets

Once the frequent itemsets from transactions in a database *X* have been found, it is straightforward to generate strong association rules from them, [5] where strong association rules satisfy both minimum support and minimum confidence. Based on confidence, association rules can be generated as follows.

a) For each frequent itemset, *F*, generate all non-empty subsets of *F*.
b) For every non-empty subset *E*, of *F*, output the rule $E \rightarrow F - E$ if the minimum confidence threshold (min_conf) is satisfied.

$$\frac{support\_count(F)}{support\_count(E)} \geq \min\_conf$$

Since the rules are generated from frequent itemsets, then each rule automatically satisfies minimum support.

The main issues of Apriori algorithm is the Database scanning of the whole dataset for every iteration, that is, full database scan is needed every time so the computational efficiency is very less. In situations with many frequent itemsets, long itemsets, or very low minimum support, it still suffers from scanning the entire database repeatedly to check a large set of candidate itemsets. Discovering pattern of length 100 requires at least 2^{100} candidates, it means n_r of subsets, repeated database scanning is very costly. To overcome these issues K-Apriori algorithm is proposed which is explained in section 3.

2.3. Wiener Transformation

The binary data is pre-processed in K-Apriori algorithm by transforming into real data using the Wiener Transformation, which is a statistical transformation. The approach is based on a stochastic framework. Wiener Transform is efficient on large linear spaces.

The input for wiener transformation is stationary with known autocorrelation. It is a causal transformation. It is based upon linear estimation of statistics [3]. The Wiener transformation is optimal in terms of the mean square error. The Wiener filter is a filter proposed by Norbert Wiener. The syntax for Wiener filter is *Y = wiener2 (X, [p q], noise)* for two-dimensional image which is normally used for image restoration. The same formula is used here for data mining task.

The input *X* for K-Apriori algorithm is a two-dimensional matrix and the output matrix *Y* is of the same size. Wiener2 uses an element-wise adaptive Wiener method based on statistics, estimates from a local neighborhood for each element. Input is a 2-dimensional matrix hence wiener2 function is used here. Wiener estimates the local mean μ and variance σ^2 around each element using the equations given below.

$$\mu = \frac{1}{pq} \sum_{n_1 n_2 \varepsilon \eta} X(n_1, n_2) \quad (1)$$

$$\sigma^2 = \frac{1}{pq} \sum_{n_1 n_2 \varepsilon \eta} (X^2(n_1, n_2) - \mu) \quad (2)$$

Where η is the *p-by-q* local neighborhood of each element in the input matrix *X*. Wiener then creates a element-wise Wiener filter using these estimates,

$$Y(n_1, n_2) = \mu + \frac{\sigma^2 - v^2}{\sigma^2} (X(n_1, n_2) - \mu) \quad (3)$$

where v^2 is the average of all the local estimated variances.

2.4 K-means algorithm

The wiener transformed data is clustered with the Standard K-means algorithm [6] which uses a multi-pass technique. Its main advantage is the short computational time it takes to find a solution so that the clustering process is very efficient. The algorithm iterates between the following steps till convergence:

a) Initialize *K* centroids at random for *K* clusters and assign each vector/ transaction to the closest cluster centroid.
b) Compute the centroids of all current clusters.
c) Generate a new partition by assigning each item to the closest cluster centroid.
d) If cluster membership changes comparing to the last iteration, go to step 2,
else stop.

3. K-Apriori Algorithm

A novel method, K-Apriori algorithm for mining Frequent itemsets and deriving Association rules from binary data are proposed here. K-Apriori is an enhanced version of Apriori algorithm based on the Apriori property and the Association rule generation procedure.

In K-Apriori algorithm, the binary data is transformed into real dom-ain using linear Wiener transformation on vector basis because Market basket transactions are vectors of the database. The Wiener transformed data is partitioned using the multi-pass K-means algorithm into *K* clusters. Apriori procedure is used for frequent itemset and association rule generation for the clusters. The items in the clusters are very similar, so that multiple and very efficient frequent itemsets are generated in the K-Apriori algorithm for normal and high support and confidence values. The Apriori algorithm should be executed multiple *(K)* times for the clusters, so K-Apriori is a multi pass algorithm. K-Apriori algorithm is described in Algorithm 2.

Algorithm 2 : K-Apriori Algorithm for Frequent Itemset Mining

Input *: Binary data matrix X of size p x q,*
K (number of clusters)

Output *: Frequent Itemsets and Association rules*

//Binary data is transformed to real data in a vector basis using Wiener transformation.
V = Call function wiener2 (X_i)
// X_i is a vector i of X

//Calculate K clusters (C_1, C_2, … C_k) for V using K-means algorithm
Call function kmeans (V, K)

//Apriori procedure
For each cluster C_i
Cd_n : Candidate itemset of size n
L_n: frequent itemset of size n
L_1 = {frequent items};
For (n=1; L_n != ϕ ; n++)
Do begin
C_{n+1} = candidates generated from L_n;
For each transaction T in database do
Increment the count of all candidates in C_{n+1}
that are contained in T
L_{n+1}= candidates in C_{n+1} with min_support
End
Return $\cup_n L_n$
End

Function wiener2 (X_i)

Input *: Binary data vector X_i of size 1 X q*
Output *: Transformed data vector Y_i of size 1 X q*

Step 1: *Calculate the mean μ for the vector n in the input vector b around each element*

$$\mu = \frac{1}{pq} \sum_{n_1 n_2 \varepsilon \eta} X(n_1, n_2)$$

where η is the local neighbourhood of each element

Step 2: *Calculate the variance σ^2 around each element*

$$\sigma^2 = \frac{1}{pq} \sum_{n_1 n_2 \varepsilon \eta} (X^2(n_1, n_2) - \mu)$$

where η is the local neighbourhood of each element

Step 3: *Perform wiener transformation for each element in every vector using this equation*

$$Y(n_1, n_2) = \mu + \frac{\sigma^2 - v^2}{\sigma^2} (X(n_1, n_2) - \mu)$$

Function kmeans (V, K)

Input *: Transformed data matrix V and number of clusters K.*
Output *: K clusters*

Step 1:

Choose initial cluster centers Z_1 , Z_2,…, Z_K *randomly from the N points*

$X_1, X_2, \ldots X_p$, $X_i \in R^q$

where q is the number of features/attributes

Step 2:

Assign point X_i, $i = 1, 2, \ldots, p$ *to cluster* C_j, $j = 1, 2, \ldots, K$

if and only if $|| X_i - Z_j|| < || X_i - Z_t||$, $t = 1, 2,\ldots,K$. *and* $j \neq t$.

Ties are resolved arbitrarily.

Step 3:

Compute the new cluster centers Z_1^*, Z_2^*, …, Z_K^* *as follows:*

$$Z_i^* = \frac{1}{l_j} \sum_{X_j \in C_j} X_i$$

where $i = 1, 2,\ldots, K$ *and* l_j *=Number of points in* C_j.

Step 4:

If $Z_i^* = Z_i$, $i = 1, 2,\ldots, K$ *then terminate. Otherwise* $Z_i \leftarrow Z_i^*$ *and go to step 2.*

4. Empirical Study and Results

K-Apriori algorithm is an enhanced version of Apriori algorithm. In K-Apriori algorithm, the binary data is transformed into real domain using linear Wiener transformation. The Wiener transformed data is partitioned using the multi-pass K-means algorithm. For the clusters, Apriori procedure is used from which frequent itemsets and Association rules are generated. Large datasets are partitioned so that the candidate itemsets generated will be very less and Database scanning will also be done for adequate data which increases the efficiency.

A market basket data sample is taken here with 988 transactions of a supermarket, Anantha Stores in Tirunelveli, Tamil Nadu. For sample, 988 transactions is considered here for experiments which is the one day transactions (04-April-2011) of the store.

Apriori algorithm generates two frequent *4-itemsets*, *ADIN* and *AINV* for 25% support. Twenty one *1-itemsets*, twenty eight *2-itemsets* and twelve *3-itemsets* also generated. Strong rule generated is *NW→A* for 100% confidence which means if *N* and *W* are purchased then *A* will also be purchased.

Table 1: Apriori & K-Apriori Result analysis for Market Basket dataset with support =25%

Confidence (%)	Maximum Number of Frequent Itemsets		Total Number of Frequent Itemsets		Total Number of Association Rules	
	Apriori	K-Apriori	Apriori	K-Apriori	Apriori	K-Apriori
50	4	5	63	311	133	1218
60	4	5	63	311	121	934
70	4	5	63	311	85	598
80	4	5	63	311	47	338
90	4	5	63	311	17	110
100	4	5	63	311	1	24

But for 25% support, *ADINV, ADNVW* and *AINVW* are the frequent *5-itemsets* generated by K-Apriori algorithm. K-Apriori generates twenty three *1-itemsets*, one hundred and thirty nine *2-itemsets*, ninety six *3-itemsets*, twenty five *4-itemsets* and three *5-itemsets* are generated. The ARs generated are given below.

AIVW→N *AIW→N*

CDN→A *CN→A*

DKN→A *DMN→A*

DNVW→A *DNW→A*

HMN→A *HN→A*

IMN→A *INVW→A*

INW→A *KN→A*

LN→A *MN→A*

MNV->A *NQW→A*

NU→A *NVW→A*

NW→A *BFK→A*

IN→A *N→A*

From the ARs generated, it is observed that *A* is the main item of the Market. If *N, W, V, I and D* is bought then *A* will also be purchased. If *A, I, V* and *W* is purchased then *N* will also be purchased. To improve the profit, if *A* has

some promotional offers, sales will be improved. Other frequent itemsets and ARs are given in the Table 1 as a summary.

Table 2: Apriori & K-Apriori Result analysis for Market Basket dataset with support =50%

Confidence (%)	Maximum Number of Frequent Itemsets		Total Number of Frequent Itemsets		Total Number of Association Rules	
	Apriori	K-Apriori	Apriori	K-Apriori	Apriori	K-Apriori
50	2	3	4	13	2	32
60	2	3	4	13	2	32
70	2	3	4	13	1	27
80	2	3	4	13	1	15
90	2	3	4	13	1	5
100	2	3	4	13	0	0

For 50% support, *A, D and N* are *1-itemsets* and *AN* is the *2-itemset* generated by Apriori algorithm. $N \rightarrow A$ is the AR generated for 90% confidence level.

For 50% support, *A, C, D, I, M, N and Q* are *1-itemsets*; *DQ, DN, DM, AN and AD* are *2-itemsets* generated by K-Apriori.

AR generated is $N \rightarrow A$ for 90%confidence. The sample set ARs generated for 90% confidence are presented below.

$AI \rightarrow N$

$IN \rightarrow A$

$I \rightarrow A$

$N \rightarrow A$

$W \rightarrow A$

It means *A* is the main item of the Market. If *N, W and I* are bought then *A* will also be purchased. If *A and I* is purchased then *N* will also be purchased. Other frequent itemsets and ARs are given in the Table 2 as a summary. Total number of frequent itemsets generated for Apriori algorithm are 4 but 13 for K-Apriori algorithm for 50% support. Apriori generates 1 AR for 90% confidence but K-Apriori generates 5 ARs for 90% confidence, it means these generated strong rules can be used to improve the sales of the Market. Due to similarity in groups of items, more number of efficient ARs are generated.

5. Conclusion

Market basket data analysis is an important data mining issue to be handled in very large databases. Market databases are homogenous databases in binary format. In the very large binary databases to find the correlation among the purchased items, frequent itemsets are used from which Association rules can be derived for decision making. Apriori is the simple and efficient algorithm for frequent itemset mining; the antimonocity property makes it simple for binary databases. A new K-Apriori Algorithm is proposed here to perform frequent itemset mining in an efficient manner. Initially the binary data is clustered using the multi-pass K-means algorithm based on the linear wiener transformation. The similar groups of data (clusters) are used in Apriori algorithm for frequent itemset generation. Apriori algorithmic procedure is used for *K* clusters and the frequent itemsets are generated from which Association rules are derived. Experiments are performed using real and synthetic data and found K-Apriori is more efficient compared to Apriori algorithm.

References

[1]. Aaron Ceglar and John Roddick. " Association Mining", ACM Computing Surveys, Vol. 38, No. 2, Article 5, July 2006.

[2]. R.Agrawal and R.Srikant. " Fast algorithms for mining association rules". In Proceedings of the 20th VLDB conference, pp 487–499, 1984.

[3]. Frederic Garcia Becerro. "Report on Wiener filtering", Image Analysis, Vibot, 2007. [Online] http://eia.udg.edu/~fgarciab/.

[4]. J.Han, H.Pei, and Y.Yin . "Mining Frequent Patterns without Candidate Generation", In Proc. Conf. on the Management of Data SIGMOD, Dallas, TX. ACM Press, New York, USA, 2000.

[5]. J.Han and M.Kamber. "Data Mining: Concepts and Techniques". Morgan Kaufmann Publishers, San Francisco, CA, 2001.

[6]. J.McQueen. "Some methods for classification and analysis of multivariate observations", In Proc. of 5th Berkeley Symp Mathematics, statistics and probability, S281-296, 1967.