Application of Data Mining Techniques For Diabetic DataSet

1Runumi Devi and 2Vineeta Khemchandani
Dept of Computer Application, JSS Academy of Technical Education, NOIDA, Uttar Pradesh
E-Mails: 1mailto:runumi@jssaten.ac.in, 2vkhemchandani@jssaten.ac.in

ABSTRACT
Medical data mining has great potential for exploring the hidden patterns in the data sets of the medical domain. These patterns can be utilized for fast and better clinical decision making, and also to curb the occurrence of particular disease by physicians. However, the available raw medical data are widely distributed, heterogeneous in nature and voluminous. Data mining and Statistics both strive towards discovering hidden patterns and structures in data. Statistics deals with heterogeneous numbers only, whereas data mining deals with heterogeneous fields. We have identified one area of healthcare where data mining techniques can be applied for knowledge discovery. In this paper, the impact of two Data Mining techniques (FP-Growth and Apriori) on a known diabetic dataset has been examined. Also rules generated by the FP-Growth approach are being matched and co-related with those being generated by Apriori algorithm.

KEYWORDS
Data mining, Knowledge discovery in database(KDD), Association Rules(AR), FPtree, Ttree.

1.0 INTRODUCTION
The healthcare environment is generally perceived as being ‘information rich’ yet ‘knowledge poor’. [6,13] There is a wealth of data available within the healthcare systems. However, there is a lack of effective analysis tools to discover hidden relationships and trends in data. Knowledge discovery and data mining have found numerous applications in business and scientific domain. Valuable knowledge can be discovered from application of data mining techniques in healthcare system[6]. The characteristics of clinical data as it originates during the process of clinical documentation, including issues of data availability and complex representation models, can make data mining applications challenging. Data preprocessing and transformation are required before one can apply data mining to clinical data. In today’s Information Technology(IT) driven society, knowledge is the most significant asset of any organization. Knowledge discovery in databases is a well-defined process consisting of several distinct steps. Data mining is the core step, which results in discovery of hidden but useful knowledge from massive databases[6].

A formal definition of data mining in databases is as follows:-

- **valid**: hold on new data with some certainty.
- **novel**: non-obvious to the system
- **useful**: should be possible to act on the item.
- **understandable**: humans should be able to interpret the pattern.

The role of (IT) in health care is well established. Knowledge management in Health care offers many challenges in creation, dissemination and preservation of health care knowledge using advanced technologies. The healthcare environment is usually information rich but knowledge poor. However data mining techniques can be applied to create a knowledge rich healthcare environment [6]. Application of data mining techniques to Diabetes, Acquired Immune Deficiency Syndrome(AIDS) datasets can be a highly important area. In a thickly populated country with scarce resources such as India, information dissemination and knowledge discovery from large databases seem to be the only solution to reduce the number of diabetes patients and to check the spread of AIDS.[13]

The article is organized as follows: Section2 introduces description about the diabetic dataset. Section3 explains how discretization can be performed on a dataset having continuous data values for different attributes. Section4 provides definition of association rule mining as well as Apriori and FP-Growth Approach. Section5 discusses about Ttree data structure that is being used in the association rule mining. Section6 presents the number of frequent itemsets and rules generated by the different approaches. Section7 presents a discussion and conclusion on the rules generated and Section8 presents some directions for future work.

2.0 DATASET DESCRIPTION
The Dataset which is being used is Pima Indian Diabetes Database (PIDD).

2.1THE PIMA INDIAN DIABETIC DATABASE
The Pima Indians may be genetically predisposed to diabetes (Hanson, Ehm et al. 1998), and it was noted that their diabetic rate was 19 times that of a typical town in Minnesota (Knowler, Bennett et al. 1978). The National Institute of Diabetes and Digestive and Kidney Diseases of the NIH originally owned the Pima Indian Diabetes Database (PIDD).[12] In 1990 it was received by the UC-Irvine Machine Learning Repository and can be downloaded at www.ics.uci.edu/~mlearn/MLRepository.html. The database has n=768 patients each with 9 numeric variables. Ages range from 21 to 81 and all are female. Out of the nine condition attributes, six attributes describe the result of physical examination, rest of the attributes of chemical examinations.
There have been many studies applying data mining techniques to the PIDD. The independent or target variable is diabetes status within 5 years, represented by the 9th variable (class=1). The attributes are:
1. number of times pregnant.
2. 2-hour OGGT plasma glucose.
3. diastolic blood pressure
4. triceps skin fold thickness
5. 2-hour serum insulin
6. BMI
7. diabetes pedigree function
8. age
9. diabetes onset within 5 years (0, 1).

The goal is to use the first 8 variables to predict 9th attribute values.

### 3.0 DATA DISCRETIZATION:

Many real-world data mining tasks involve continuous attributes. Data discretization is defined as a process of converting continuous data attribute values into a finite set of intervals and associating with each such specific data value. There are no restrictions on discrete values associated with a given data interval except that these values must induce some ordering on the discretized attribute domain. Discretization significantly improves the quality of discovered knowledge and also reduces the running time of various data mining tasks such as association rule discovery, classification, and prediction.[4]

Good discretization may lead to new and more accurate knowledge. On the other hand, bad discretization leads to unnecessary loss of information or in some cases to false information with disastrous consequences. Any discretization process generally leads to a loss of information. The goal of the good discretization algorithm is to minimize such information loss. If discretization leads to an unreasonably small number of data intervals, then it may result in significant information loss. If a discretization method generates too many data intervals, it may lead to false information. Discretization of continuous attributes has been extensively studied. There are a wide variety of discretization methods starting with the naive methods often referred to as unsupervised methods such as equal-width and equal-frequency to much more sophisticated methods often referred to as supervised methods such as Minimum Description Length (MDL) and Pearson’s $X^2$ or Wilks’ $G^2$ statistics based discretization algorithms.[3, 4]

Both unsupervised and supervised discretization methods can be further subdivided into top-down or bottom-up methods.[3]

A top-down method starts with a single interval that includes all data attribute values and then generates a set of intervals by splitting the initial interval into two or more intervals. A bottom-up method initially considers each data point as a separate interval. It then selects one or more adjacent data points merging them into a new interval.

**Equal-depth(frequency) partitioning**

It divides the range into N intervals (bins), each containing approximately the same number of samples. Smoothing is then performed by finding bin means.

### 4.0 ASSOCIATION RULE MINING:

Association rule mining techniques are used to identify relationships among a set of items in database.[1] These relationships are not based on inherent properties of the data themselves (as with functional dependencies), but rather based on cooccurrence of the data items. Association rules are more appropriate when we search for completely new rules.[1, 7]

In this context, the association rule mining technique may generate the probable causes of the particular disease (Diabetes) in the form of association rules which can be used for fast and better clinical decision-making.

Let D be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$, I=\{i1, i2, ..., in\} be a set of literals, called items. Given the set of transactions D the problem is to find association rules that have support and confidence greater than the user specified minimum support and minimum confidence[1].

An association rule is an implication of the form $X \Rightarrow Y$, where $X \subseteq I$, $Y \subseteq I$, $X \cap Y = \emptyset$. The rule $X \Rightarrow Y$ holds in the transaction set D with confidence c, if c% of transactions in D that contain X also contain Y. The rule $X \Rightarrow Y$ has support s in the transaction set D, if s% of transactions in D contain $X \cup Y$[7, 8].

Given the set of transactions T, one may be interested in generating all rules that satisfy certain fixed constraints for support and confidence. Support and confidence are measures of the interestingness of the rule. A high level of support indicates that the rule is frequent enough for the organization to be interested in it. A high level of confidence shows that the rule is true often enough to justify a decision based on it.[14]

Thus for a rule $X \Rightarrow Y$,

$$Support(XY) = \frac{\text{(Number of times X and Y appear together)}}{D}$$

$$Confidence(XY) = \frac{Support(XY)}{Support(X)}$$

### 4.1 APRIORI:

This algorithm may be considered to consist of two parts.[10, 14]

- **First part**- finding frequent itemsets
- **Second part**- finding the rules

**For finding frequent itemsets following steps are followed:**

1. **Step 1:** Scan all transactions and find all frequent items that have support above 5%. Let these frequent items be L.
2. **Step 2:** Build potential sets of k items from $L_{k-1}$ by using pairs of itemsets in $L_{k-1}$ such that each pair has the first k-2 items in common. Now the k-2 common items and the one remaining
item from each of the two itemsets are combined to form a k-itemset. The set of such potentially frequent k itemsets is the candidate set $C_k$. (For $k=2$, we build the potential frequent pairs by using the frequent itemset $L_1$ appears with every other item in $L_1$. The set so generated is the candidate set $C_2$).

Step 3: Scan all transactions and find all k-itemsets in $C_k$ that are frequent. The frequent set so obtained is $L_2$.

The first pass of the Apriori algorithm simply counts item occurrences to determine the large 1-itemsets. A subsequent pass, say pass $k$, consists of two phases. First, the large itemsets $L_{k-1}$ found in the $(k-1)$th pass are used to generate the candidate itemsets $C_k$, using the apriori-gen function. Next, the database is scanned and the support of candidates in $C_k$ is counted. For fast counting, we need to efficiently determine the candidates in $C_k$ that are contained in a given transaction $t$. [11,14]

For finding rules we follow the following straightforward algorithm:

We take a large frequent itemset, say $l$, and find each non-empty subset $a$. For every such subset $a$, output a rule of the form $a \Rightarrow (l-a)$ if support($l$)/support($a$) satisfies minimum confidence.

4.2 FREQUENT PATTERN GROWTH

This algorithm comprises two major steps:

First: compress a large database into a compact, Frequent Pattern tree (FP-tree) structure.

Secondly: develop an efficient, FP-tree based frequent pattern mining.

The major difference between FP-growth and the Apriori algorithm discussed above is that FP-growth does not generate the candidate itemsets and then tests $[5,9]$.

5.0 DATA STRUCTURE FOR ASSOCIATION RULE MINING

Association Rule Mining (ARM) obtains a set of rules which indicate that the consequent of a rule is likely to apply if the antecedent applies. [1] To generate such rules, the first step is to determine the support for sets of items (I) that may be present in the data set, i.e., the frequency with which each combination of items occurs. After eliminating those I for which the support fails to meet a given minimum support threshold, the remaining large I can be used to produce ARs of the form $A \Rightarrow B$, where $A$ and $B$ are disjoint subsets of a large I. The ARs generated are usually pruned according to some notion of confidence in each AR. However to achieve this pruning, it is always necessary to first identify the “large” I contained in the input data. This in turn requires an effective storage structure. One of the efficient data storage mechanism for itemset storage is the T-tree. [2]

5.1 TOTAL SUPPORT TREE (T-TREE):

A T-tree is a set enumeration tree structure which is used to store frequent itemset information. The difference between the T-tree and other set enumeration tree structure is:

1. Array is used to define the levels in each sub-branch of the tree which permits “indexing in” at all levels which in turn offers computational advantages.

2. To make the indexing at all levels the tree is built in “reverse”. Here, each branch is founded on the last element of the frequent sets to be stored.

The most significant overhead when considering ARM data structures is that the number of possible combinations represented by the items (columns) in the input data scales exponentially with the size of the record. A partial solution is to store only those combinations that actually appear in the data set. A further mechanism is to make use of the downward closure property of itemsets—“if any given itemset I is not large, any superset of I will also not be large.” This can be used effectively to avoid the need to generate and compute support for all combinations in the input data. However, the approach requires: 1) a number of passes of the data set and 2) the construction of candidate sets to be counted in the next pass.

The implementation of this structure can be optimized by storing levels in the tree in the form of arrays, thus reducing the number of links needed and providing direct indexing. For the latter purpose, it is more convenient to build a “reverse” version of the tree, referred to as a T-tree (Total support tree).

6.0 FIRST RESULT

In the following table 1 we can see how many frequent itemsets and rules were produced using different approaches and different parameter settings.

<table>
<thead>
<tr>
<th>Parameter Value</th>
<th>FP-Growth</th>
<th>Apriori</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support(4%)</td>
<td>121</td>
<td>121</td>
</tr>
<tr>
<td>Confidence(80%)</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Support(4%)</td>
<td>121</td>
<td>121</td>
</tr>
<tr>
<td>Confidence(70%)</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>Support(4%)</td>
<td>121</td>
<td>121</td>
</tr>
<tr>
<td>Confidence(60%)</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>Support(4%)</td>
<td>121</td>
<td>121</td>
</tr>
<tr>
<td>Confidence(50%)</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>Support(5%)</td>
<td>68</td>
<td>No Rules Generated</td>
</tr>
<tr>
<td>Confidence(80%)</td>
<td>68</td>
<td>No Rules Generated</td>
</tr>
<tr>
<td>Support(5%)</td>
<td>68</td>
<td>6</td>
</tr>
<tr>
<td>Confidence(70%)</td>
<td>68</td>
<td>6</td>
</tr>
<tr>
<td>Support(5%)</td>
<td>121</td>
<td>121</td>
</tr>
<tr>
<td>Confidence(60%)</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>Support(5%)</td>
<td>68</td>
<td>68</td>
</tr>
<tr>
<td>Confidence(50%)</td>
<td>49</td>
<td>No Rules Generated</td>
</tr>
<tr>
<td>Support(6%)</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td>Confidence(80%)</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td>Support(6%)</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td>Confidence(70%)</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td>Support(6%)</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td>Confidence(60%)</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td>Support(6%)</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td>Confidence(50%)</td>
<td>195</td>
<td>195</td>
</tr>
<tr>
<td>Support(3%)</td>
<td>195</td>
<td>195</td>
</tr>
<tr>
<td>Confidence(80%)</td>
<td>74</td>
<td>74</td>
</tr>
<tr>
<td>Support(3%)</td>
<td>195</td>
<td>195</td>
</tr>
<tr>
<td>Confidence(50%)</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>
7.0 DISCUSSION AND CONCLUSION

Our goal was to observe the impact of data mining technique on diabetic dataset. In this study, we have implemented two algorithms of association rule mining technique, viz Apriori and FP-Growth techniques. In FP-Growth a novel data structure, frequent pattern tree (FP-tree), is being implemented for storing compressed, crucial information about frequent pattern. A method, FP-growth, is also followed for efficient mining of frequent patterns in large database.

There are several advantages of FP-growth over other Apriori approach: 1) It constructs a highly compact FP-tree, which is usually substantially smaller than the original database and thus saves the costly database scans in the subsequent mining processes. 2) It applies a pattern growth method which avoids costly candidate generation and test by successively concatenating frequent 1-itemset found in the (conditional) FP-trees. 3) It applies a partitioning-based divide-and-conquer method which dramatically reduces the size of the subsequent conditional pattern bases and conditional FP-trees.

We have observed that both the techniques generates the same number of frequent sets as a consequence same number of rules for the same known dataset under the same constraints. There were two exceptions in the rules generated by the two techniques. Both the techniques (FP-Growth, Apriori) generate no association rules within the limitations (supp= 5% conf=80% and supp=6%, conf=80%) . Here the limitations seem to be too hard as we are considering 80% reliability as well as 5% or 6% occurrence. Moreover, rule generation time graph proved the efficiency of the FP-Growth approach in terms of generation time.

Although most of these rules provided valuable knowledge, we only describe some of the most beneficial rules under 100.0% confidence, according to secondary data source which is given below:

1. IF (OGTT=127) THEN (number of times pregnant=3)
2. IF (Diastolic Blood Pressure =75) AND (BMI=30) THEN (Diabetis pedigree function=0.5)
   Unit of BMI—weight in Kg/(height in m)^2
3. IF (number of times pregnant=3) AND (Age=35) THEN (Diabetis pedigree function=0.33)
4. IF (Diastolic Blood Pressure =50) THEN Not Diabetic
5. IF (number of times pregnant=6) AND (BMI=34) THEN Not Diabetic
6. IF(Triceps Skinfold Thickness=22) AND (Diabetis pedigree function=0.5) THEN Not Diabetic
7. IF (number of times pregnant=5) AND (Diabetis pedigree function=0.66) THEN Not Diabetic
8. IF (number of times pregnant=7) AND (Diabetis pedigree function=0.66) THEN Not Diabetic
9. IF (BMI=35) AND (Diabetis pedigree function=0.66) THEN Not Diabetic
10. IF (OGTT=103) THEN Diabetic
11. IF (OGTT=105) THEN Diabetic
12. IF (OGTT=119) THEN Diabetic
13. IF (OGTT=120) THEN Diabetic
14. IF (Diastolic Blood Pressure =63) THEN Diabetic
15. IF (number of times pregnant=2) AND (Diastolic Blood Pressure =75) THEN Diabetic
16. IF (number of times pregnant=3) AND (Triceps Skinfold Thickness=17) THEN Diabetic
17. IF (number of times pregnant=2) AND (BMI=30) THEN Diabetic
18. IF (Diastolic Blood Pressure =75) AND (BMI=30) THEN Diabetic
19. IF (number of times pregnant=3) AND (BMI=33) THEN Diabetic
20. IF (OGTT=113) AND (Diabetis pedigree function=0.33) THEN Diabetic
21. IF (OGTT=120) AND (Diabetis pedigree function=0.33) THEN Diabetic
22. IF (BMI=32) AND (Diabetis pedigree function=0.33) THEN Diabetic
23. IF (OGTT=119) AND (Diabetis pedigree function=0.5) THEN Diabetic
24. IF (Diastolic Blood Pressure =75) AND (Diabetis pedigree function=0.5) THEN Diabetic
25. IF (Diastolic Blood Pressure =75) AND (BMI=30) THEN (Diabetis pedigree function=0.5) AND Diabetic
26. IF (Diastolic Blood Pressure =75) AND (BMI=30) AND (Diabetis pedigree function=0.5) THEN Diabetic
27. IF (Diastolic Blood Pressure =75) AND (BMI=30) AND Diabetic THEN (Diabetis pedigree function=0.5)
28. IF (BMI=31) AND (Diabetis pedigree function=0.5) THEN Diabetic

Continued on Page No. 368
29. IF(Age=23) THEN Diabetic.
30. IF (number of times pregnant=1) AND (Age=25) THEN Diabetic
31. IF (BMI=31) AND (Age=25) THEN Diabetic
32. IF (number of times pregnant=3) AND (Age=26) THEN Diabetic
33. IF (Diabetes pedigree function=0.33) AND (Age=28) THEN Diabetic
34. IF (number of times pregnant=3) AND (Age=29) THEN Diabetic
35. IF (Diabetes pedigree function=0.33) AND (Age=31) THEN Diabetic
36. IF (number of times pregnant=2) AND (Age=33) THEN Diabetic
37. IF (BMI=30) AND AND (Age=33) THEN Diabetic

The group with confidence 100% shows if the value of 2-hour OGTT plasma glucose is between 103 to 120 then the person is likely to become diabetic. If Diastolic Blood Pressure=75 and Body Mass Index =30 and has diabetes history in family then chances of becoming diabetic is more. Triceps Skinfold Thickness which is an estimated measure of subcutaneous fat in a body also plays a major role for diabetes. Moreover, if a lady is 1 or 2 or 3 times pregnant and age is also either 25,26,28,29,31,33 ,there may be more number of diabetic cases. Summarizing, these results show that apart from the values of chemical examinations, results of physical examination is necessary for better clinical decision making by the medical expert. Some risk factor have less importance because they appear less frequently in the dataset.

These rules have the potential to improve the expert system and to make better clinical decision making. In a thickly populated country with scarce resources such as India, public awareness can also be achieved through the dissemination of the above knowledge.

8. FUTURE WORK

The above application software is designed for generating association rules using two different association rule mining techniques which provide same set of rules. It is observed that association rule mining techniques generate large number of rules that are very difficult to be analyzed by users. Also, it is found that most of the rules are redundant. Therefore different filtering techniques as well as rule pruning can also be performed on these two techniques. Moreover, Decision Tree techniques can be an alternative solution where knowledge is represented in more compact and optimized way. Also Decision Trees may be more convenient in ‘goal oriented’ cases, where we look for rules with fixed consequence or outcome, like some specific diagnosis whereas association rule mining techniques are good, when we search for completely new rules.

9. REFERENCES

[9] Jiawei Han, Jian Pei and Yiwen Yin. Mining Frequent Patterns without Candidate Generation. International Conference on Management of Data Proceedings of the 2000 ACM SIGMOD international conference on Management of data.