International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

December 2020 | Issue 45 | Volume 12 | Number 4 | Pages 50-56

SOFT SET BASED APPROACH FOR MINING FREQUENT ITEMSETS

S. Kottam [1] V. Paul [2]

1. Research Scholar, R&D Centre, Bharathiyar University, Coimbatore, Tamilnadu, India, sankottam@rediffmail.com
2. Department of Information Technology, CUSAT, Kochi, India, vp.itcusat@gmail.com

Abstract- Currently, data miners use various algorithms to retrieve frequent itemsets. One of the popular methods is frequent pattern growth that retrieves the whole set of frequent items by avoiding candidate set production. The generation of FP-growth algorithm requires a high volume of memory space. Scanning the dataset twice reduces the full efficiency of frequent pattern growth technique and produces a huge number of frequent pattern trees. To eliminate these drawbacks, we put forward a novel algorithm called FP-soft set. Our research work modified FP-growth algorithm and discusses applications of soft set theory for extracting frequent itemset from a large data set. The proposed work has extensive value in the field of data mining. It has enormous use in multiple fields as market basket analysis, agriculture, tourism, and disease detection.

Keywords: Soft Computing, Soft Set, FP-Growth, FP-Soft Set, Frequent Itemset.

1. INTRODUCTION

Data mining or knowledge discovery from data is the process of retrieving useful patterns representing knowledge, which are stored in huge databases and other gigantic data depositories. Interesting patterns, we describe in different formats, known as data mining functionalities. Data mining processes include a sequence of functionalities. Among them, association analysis is very popular and is getting more attention from industry, academia, and research scholars.

The mining of rules helps us to bring out interesting relationships and to make more accurate decisions in different areas such as industry, health, and crime investigation. Generally, we discover frequent itemsets from huge data repositories. A good method for extracting frequent itemsets from a huge data set is Apriori algorithm. Prior knowledge of frequent itemsets properties is essential for implementing this algorithm. Even though Apriori is a classical algorithm, it has two drawbacks. Firstly, it has to generate a big collection of candidate sets. Secondly, it has to inspect the dataset continually and verify a large set of candidate sets by pattern matching.

Researchers have added active modifications to Apriori algorithm for improving its performance and flexibility, and finally generated a divide and conquer method, which is known as FP growth.

The FP algorithm brings out frequent itemsets without candidate itemsets generation. This method directly compresses the databases into a particular tree, which is known as frequent pattern tree and generates the associative rule from it. Frequent pattern growth technique scans the dataset twice and produces huge number of frequent pattern trees. Also, FP algorithm requires a high volume of memory space. Soft set (SS) theory devised by

D. Molodtsov, a renowned soft computing mathematician, is a new soft computing tool for handling uncertainties in different decision-making scenarios. Soft set theory provides sufficient parameterization techniques for managing with data uncertainty [3] and helps us in decision making [14-18]. We propose a new alternative to FP algorithm with the help of soft set theory. Conditional FP tree and frequent pattern creation are simplified by soft set theory [19].

The first section of this paper is the introduction; the second section explains the preliminaries that form the background for the research. In the following sections, the authors explain soft set theory and how it is useful for improving FP-growth algorithm. Finally, the paper concludes with a new algorithm that the authors developed with python programming.

2. BACKGROUND WORKS

Rough set theory is a popular soft computing tool for handling vague data. Prasanta Gogoi et al contributed a new method for retrieving crisp rules from incoherent data. For each concept, lower and upper approximations are computed. Following it, a training algorithm is developed and for each concept algorithm produce well defined classification rules. This method produced more accurate and reliable rules [9]. Jiye Li et al. proposed a rough set based method for selecting most suitable rules. The proposed algorithm ranks rules by measuring its importance. Selected rules are useful for making decision in all sectors like- business, education, tourism etc. [13].

Fuzzy Association Rule Miner (FARM) is a novel data mining technique proposed by Wai-Ho to mine interesting association rules. This technique uses fuzzy set properties for representing ambiguity and vagueness. It supports to retrieve association rules between quantitative values. A prominent advantage of FARM approach is that it can extract both negative and positive rules. According to Wai-Ho, the algorithm helps us to retrieve more reliable and accurate rules [10].

Multi Level Feed Forward Mining (MLFM) is a neural network based approach invented by Amith et al. It has the capability to mine different levels of rules from huge databases. This research work used a supervised neural network approach for retrieving frequent items. A single scan of database is employed for each concept level. For each concept, algorithm reads items, splits them into different hierarchical structure and sent it to a neural network for generating frequent item sets [11]. Transaction Reduction- Frequency Count Table Method (TR-FCTM) is one of the newly invented methods to find out frequent items occurs in different database transactions. It employs a single database scan, form a frequency count table and generate whole candidate items [8]. A genetic algorithm based frequent item set mining technique is proposed by Vijaya Prakash et al. This approach improved the performance of frequent item set mining and supported to reduce time complexity [12].

3. PRELIMINARIES

3.1. Association Rules

Association rule extraction is one of the distinguished approaches of knowledge mining. It intends to retrieve interesting patterns and associations from a large data set [4]. Let R be a set of items, $R=\{r_1,r_2....,r_n\}$ and B be a subset of R. For $P \subset R$ and $Q \subset R$, then $b \in B$ accommodate P if and only if $b \subseteq B$. An association rule has the format, $P \rightarrow Q$ where $(Q \vee 1) \wedge P \cap Q = \phi$. Then set P is termed the antecedent of the representation and set Q is termed the consequent.

Usually, a rule $P \rightarrow Q$ means that if an operation includes P it very likely includes Q as well. There are two parameters related to a rule: Support and Confidence. To describe these parameters, we use B_S and B_C to indicate the subset of B that includes both P and Q, and the subset of B that includes P, respectively. It is obvious $B_S \subseteq B_C \subseteq B$.

Definition: The support value of the rule $P \rightarrow Q$ derived from dataset B is the ratio of the cardinality of Bs to the cardinality of B. Hence, the support of the rule is

$$S = \frac{|B_S|}{|B|} \tag{1}$$

Definition: The confidence value of the rule $P \rightarrow Q$ obtained from dataset B is ratio of the cardinality of B_S to the cardinality of B_C. Therefore, the confidence of rule is

$$C = \frac{|B_S|}{|B_C|} \tag{2}$$

Definition: The *lift* of a rule is specified as

$$lift(P \Rightarrow Q) = \frac{(P \cup Q)}{(P) \times (Q)} \tag{3}$$

or the ratio of the examined support to that expected if P and Q were independent [20]

3.2. Apriori Rules

Apriori is a conventional association-rule-fetching algorithm invented by R. Agarwal and R. Srikant in 1994. This algorithm employs level-wise search to mining frequent itemsets.

It uses Jth level itemsets for determining (J+1)th level itemsets. Initially, the set of frequent items is decided by browsing the whole dataset to find the count for each item and gather those items that assure minimum support. Initially, the resulting set is represented as M_1; in the next stage M_1 is used to discover M_2, which is used to discover M_3 and so on, until no more frequent itemsets can be found.

For each stage, a whole examination of the dataset is necessary. This will reduce the performance of the algorithm execution. To increase the performance of level-wise examination, we use Apriori property to decrease inspection area. According to this principle, for all frequent itemset its nonempty subsets are also frequent. This algorithm uses the following steps for its process - join and prune [1]. Join action says that to create frequent itemset M_k, a set of Candidate K-itemsets (CK) is created by combining the frequent itemsets M_k-1 with itself. Prune action says that from the candidate K-itemsets filter the candidates to keep a count not less than minimum support. For reducing the complexity of this action, here we use Apriori property [21]

3.3. FP Algorithm

Frequent pattern growth algorithm is an enhancement of Apriori algorithm. It uses a conventional method-divide and conquer. A frequent pattern tree is a condensed portrayal of a dataset that helps in finding of frequent itemset without the identification of candidate itemset creation. The root node of the FP- tree is termed as 'NULL' value. Remaining nodes stand for Item Name, Node link and Count. Nodes correspond to items and have a count value [1]. Development of frequent pattern tree uses two stages over the data set. In the first stage, it finds count for each item and removes non-frequent items. The set of frequent items is arranged in descending order based on their support count.

3.4. Example 1

Consider an example in Table 1, where the items are p, q, r, s, t.

Table 1. Transaction data set

TID	ITEMS
1	{q,p}
2	{r,q,s}
3	{r,s,t,p}
4	{t,p,s}
5	{p,r,q}
6	{s,r,p,q}
7	{p}
8	{q,p,r}
9	{s,p,q}
10	{t,q,r}

These items are sorted on their support count. Sorted order is $L=\{p, q, r, s, t\}$. The first phase is the construction of FP tree. Select the first transaction $T_1=\{p, q\}$, arrange it into L order $\{p, q\}$. Construct first branch of the tree and set the count values of p and q to 1. Read the transactions 2, 3, …., 10, and add the links to FP tree. Continue this process until all transactions are arranged to a path in the frequent pattern tree. The final output of the process is shown in Figure 1.

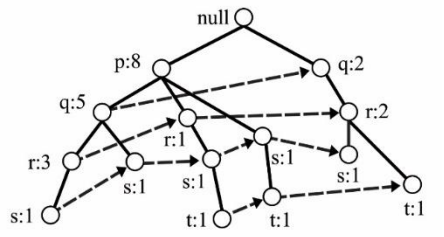

Figure 1. Final constructed FP tree

In the second stage, frequent pattern growth fetches frequent itemsets from the frequent pattern tree. First, extract the prefix path (*P*-path) sub tree finishing in an item '*t*' of Figure 2. Next processes the extracted prefix path sub tree and produce the frequent itemsets. Results are then consolidated.

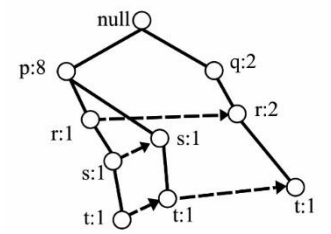

Figure 2. Prefix path sub tree ending in *t*

Using a divide and conquer method, from the above tree extract the frequent itemsets ending in '*t*'; then in *st*, *rt*, *qt* and *pt*; then in *rst*, *qst*, *pst* etc. Let Sup_{min}=2 and extract all frequent itemsets containing '*t*'. Count the number of times '*t*' repeats in the prefix path tree. If the count is greater than or equal to 2, extract {*t*} as a frequent itemset. Here it is 3 and '*t*' is a frequent itemset. Next, find the frequent itemsets ending in '*t*', i.e. *st*, *rt*, *qt* and *pt*. Decompose the problem recursively. To do this, we must first obtain the conditional FP tree for '*t*'. Revise the counts together with the *P*-paths from '*t*' to reproduce the number of transactions containing '*t*'. The *q* and *r* should be set to 1 and *p* to 2. Remove the nodes containing '*t*', since information about node '*t*' is no longer needed. Also remove infrequent items from the prefix path. Since q has a support of 1, remove it from conditional FP tree. FP tree conditional on t is given in Figure 3.

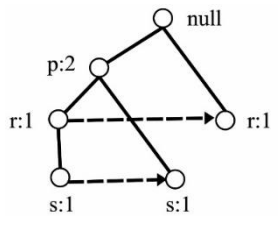

Figure 3. FP tree conditional on '*t*'

Use the conditional FP-tree for '*t*' to find frequent itemsets ending in *st*, *rt* and *pt*. For each of them (e.g., *st*), find the prefix paths from the conditional tree for t, extract frequent itemsets, generate conditional FP-tree, etc... recursively.

Example: $t \rightarrow st \rightarrow pst$ ($\{s,t\},\{p,s,t\}$) are found to be frequent. Generate conditional FP-tree for *p*, *q*, *r*, *s*, and find all of their frequent itemsets. Final output is given in Table 2.

Table 2. Frequent itemsets results

Suffix	Frequent Itemsets
T	$\{t\},\{p,s,t\},\{r,t\},\{p,t\},\{s,t\}$
S	$\{s\},\{r,s\},\{q,r,s\},\{p,r,s\},\{q,s\},\{p,q,s\},\{p,s\}$
R	$\{r\},\{p,q,r\},\{p,r\},\{q,r\}$
Q	$\{q\},\{p,q\}$
P	$\{p\}$

4. SOFT SET THEORY - A NEW SOFT COMPUTING TOOL

Now a days, data miners depends on a group of reliable and flexible techniques for handling problems having approximate reasoning and uncertainty, that group is known as soft computing [2]. To retrieve relevant data from enormous data repositories, the data mining industry uses conventional soft computing methodologies [3]. These theories have certain limitations due to insufficiency of the parameterization tools. Soft set theory has enough parameterization capability and free from the limitations mentioned.

4.1. Preliminaries of Soft Set Theory

Soft Set (Definition): Let (G,X) is a soft set defined over the set S and G is denoted as a mapping

$$G: X \rightarrow P(S) \tag{4}$$

Also, a soft set defined on S is a collection of parameterized subsets of the universal set S. Each subset $G(\varepsilon)$, $\varepsilon \in X$ is considered as the ε is the elements of the soft set (G,X). Soft set theory mainly differs from conventional mathematics on item managing. In conventional computing, we develop a concept of an item and discover the idea of precise output of that concept. SS develops an idea of the relative output and determines that output. This increases the popularity of SS theory. Soft set theory permits parameterization in different ways. We can use numbers, words, sentences, functions and many more as parameters in our real-world problems [6].

5. SOFT SET THEORY FOR IMPROVING FP GROWTH ALGORITHM

We discuss the role of soft set theory for increasing the performance of FP growth algorithm. The pre-requirement for applying soft set method for the proposed work is that data must be converted into a soft set, where each element is treated as a parameter.

5.1. Example 2

The soft set describing Table 1 is given here. Soft set (F,X) illustrates the occurrence of each element in different dealings.

Let Y denote the set of all transactions.

X is the set of elements. Each element is an item. $Y=\{Y_1,Y_2,Y_3,Y_4,Y_5,Y_6,Y_7,Y_8,Y_9,Y_{10}\}$ and $X=\{X_1,X_2,X_3,X_4,X_5\}$

where,
X_1 represents for the element 'p'
X_2 represents for the element 'q'
X_3 represents for the element 'r'
X_4 represents for the element 's'
X_5 represents for the element 't'
and
$F(X_1)=\{Y_1,Y_3,Y_4,Y_5,Y_6,Y_7,Y_8,Y_9\}$
$F(X_2)=\{Y_1,Y_2,Y_5,Y_6,Y_8,Y_9,Y_{10}\}$
$F(X_3)=\{Y_2,Y_3,Y_5,Y_6,Y_8,Y_{10}\}$
$F(X_4)=\{Y_2,Y_4,Y_6,Y_9\}$
$F(X_5)=\{Y_3,Y_4,Y_{10}\}$

The soft set (F,X) is a parameterized family $\{F(X_i), i=1, 2, 3, \ldots, 5\}$ of subsets of the set Y and gives us a collection of rough information of an item. $F(X_1)$ means "item(p)" whose functional value is the set $\{Y_1,Y_3,Y_4,Y_5,Y_6,Y_7,Y_8,Y_9\}$. The soft set (F,X) is a collection of approximation given in Figure 4. An illustration of soft set (F,X) is given in Table 3. From the soft set (F,X), representation of different transactions is given in Figure 5.

$$(F,X)=\begin{cases} p=\{Y1, Y3, Y4, Y5, Y6, Y7, Y8, Y9\}, \\ q=\{Y1, Y2, Y5, Y6, Y8, Y9, Y10\}, \\ r=\{Y2, Y3, Y5, Y6, Y8, Y10\}, \\ s=\{Y2, Y4, Y6, Y9\},\ t=\{Y3, Y4, Y10\} \end{cases}$$

Figure 4. The soft set representing Table 1

Table 3. Representation of the Soft Set in tabular format

Item	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9	Y_{10}
P	1	0	1	1	1	1	1	1	1	0
Q	1	1	0	0	1	1	0	1	1	1
R	0	1	1	0	1	1	0	1	0	1
S	0	1	0	1	0	1	0	0	1	0
T	0	0	1	1	0	0	0	0	0	1

$Y1 = \{p,q\}$
$Y2 = \{q,r,s\}$
$Y3 = \{p,r,s,t\}$
$Y4 = \{p,s,t\}$
$Y5 = \{p,q,r\}$
$Y6 = \{p,q,r,s\}$
$Y7 = \{p\}$
$Y8 = \{p,q,r\}$
$Y9 = \{p,q,s\}$
$Y10 = \{q,r,t\}$

Figure 5. The involvement of items in different transaction

Definition: Let (F,X) be a SS over the universal set Y and $E\subseteq X$. The support count of an element E is represented by $sup(E)$, i.e. sup(E) is the number of transactions Y containing the element E. The $sup(E)=|\{u: E\subseteq U\}|$, where $|E|$ support count of E [22].

From the above definition, the supported count collected for different frequent itemsets are given in Figure 6.

Next, we see how this soft set theory is useful for improving the FP growth algorithm. The whole process starts with organizing items in the downward order of support count. The result will be $L=\{p,q,r,s,t\}$. Assume minimum support count for frequent itemset as 2. FP growth algorithm extracts association rules without generating the candidate set.

sup{p}=|{Y1,Y3,Y4,Y5,Y6,Y7,Y8,
sup{q}=|{ Y1, Y2, Y5, Y6, Y8, Y9,
sup{r}=|{ Y2, Y3, Y5, Y6, Y8, Y10
sup{s}=|{ Y2, Y4, Y6, Y9}|=4
sup{t}=|{ Y3, Y4, Y10}|
sup{p, q}=|{Y1,Y5,Y6,Y8,Y9}|=5
sup{r, s}=|{Y2,Y3,Y6}|=3
sup{s, t}=|{Y3,Y4}|=2
sup{q, r}=|{Y2,Y5,Y6,Y8,Y10}|=5
sup{q, s}= |{Y2,Y6,Y9}|=3
sup{r, t}=|{Y3,Y10}|=2
sup{p, r}=|{Y3,Y5,Y6,Y8}|=4
sup{p, s}=|{Y3,Y4,Y6,Y9}|=4
sup{p, t}=|{Y3,Y4}|=2
sup{q, r, s}=|{Y2,Y6}|=2
sup{p, s, t}=|{Y3,Y4}|=2
sup{p, r, s}=|{Y3,Y6}|=2
sup{p, q, s}=|{Y6,Y9}|=2
sup{p, q, r}=|{Y5,Y6,Y8}|=3

Figure 6. The supported sets

It passes through the following steps:
1. FP-Tree construction
2. Prefix sub-path generation
3. Conditional FP Tree generation
4. Frequent Itemset generation

In the proposed approach, the first two steps remain the same. For the remaining steps, we apply the application of soft set theory. The conditional FP tree and frequent itemset mining steps are carried out by the following method. This method is performed on all items of the FP-Tree. Figure 7 shows the process diagram of new method.

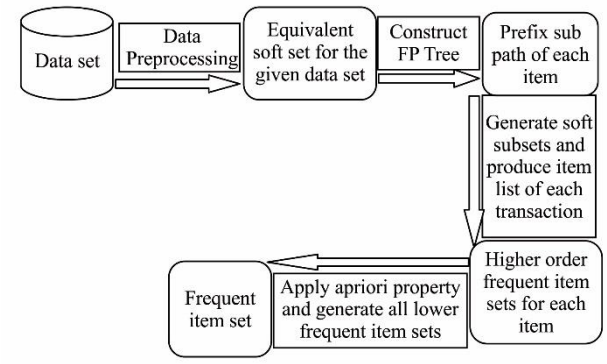

Figure 7. Process diagram of proposed method (FP-Softset)

Next, we implement example 1 using new method. We first consider prefix path sub tree for 't', which is the last item in L, rather than the first. Soft set (F,X) says, 't' occurs in three branches of the prefix path sub tree.
$F(t)= Y_3,Y_4,Y_{10}\}$
where,
$Y_3=\{p,r,s,t\}$, $Y_4=\{p,s,t\}$ and $Y_{10}=\{q,r,t\}$
If $Y_3\subseteq Y_4$ or $Y_3\supseteq Y_4$
$Y_3\cap Y_4=\{p,r,s,t\}\cap\{p,s,t\}=\{p,s,t\}$
which indicates $\{p,s,t\}$ is repeated two times in the transactions $\{Y_3,Y_4\}$, i.e., $\{p,s,t{:}2\}$ is 3-frequent itemset

Then, we use Apriori property in the new process. Use this property to find all subsets which include the item 't'. Resultant subsets are also frequent.
$\{p,t{:}2\}$, $\{s,t{:}2\}$ are 2-frequent itemsets
Next, Y_3 and Y_{10}, if $Y_3\subseteq Y_{10}$ or $Y_3\supseteq Y_{10}$

$Y_3 \cap Y_{10}=\{p,r,s,t\} \cap \{r,t\}=\{r,t\}$
$\{r,t{:}2\}$ is a 2-frequent itemset
Next, combination is Y_4 and Y_{10}
If $Y_4 \subseteq Y_{10}$ or $Y_4 \supseteq Y_{10}$
$Y_4 \cap Y_{10}=\{p,s,t\} \cap \{q,r,t\}=\{t\}$
where, $\{t{:}2\}$ is 1-frequent itemset, which is proved in Figure 6. Frequent itemsets ending with '*t*' are
$\{t\}$, $\{p,t{:}2\}$, $\{s,t{:}2\}$, $\{r,t{:}2\}$ and $\{p,s,t{:}2\}$

Apply this same method for the remaining items of *s*, *r* and *q*. Final frequent item set results are the same as Table 4. The algorithm for mining frequent itemsets using soft set theory is given in Figure 8.

Table 4. Frequent itemsets results using soft set theory

Suffix	Frequent Itemsets
T	{*t*},{*p,s,t*},{*r,t*},{*p,t*},{*s,t*}
S	{*s*},{*r,s*},{*q,r,s*},{*p,r,s*},{*q,s*},{*p,q,s*},{*p,s*}
R	{*r*},{*p,q,r*},{*p,r*},{*q,r*}
Q	{*q*},{*p,q*}
P	{*p*}

Algorithm:Fp-softset. Extract frequent itemsets using a Frequent pattern tree by soft set approach.
Input:
D, be a dataset
M_sup, the minimum support count.
Output:
Complete set of frequent items
Method:
The Frequent Pattern tree FPT is produced in the following phases:
Phase I
- Examine the dataset *D* once and find support count of each element.
- Remove item which is not keeping minimum support count.
- Use support count and arrange frequent items F, in descending order.

Phase II
- For all transactions sort frequent items according to the order of L.
- Read transactions Y1, Y2, Y3….continue until all transactions are mapped to a path in the Frequent Pattern tree.
- Call Fp_soft(FPT)

Procedure Fp_soft(FPT)
Convert Fp-tree into a soft set representation.
for each frequent item $ei \in F$, $i=1,2....n$
{
Extract prefix path sub tree PT and process it iteratively to generate the frequent itemsets.
From the prefix path sub tree PT, find F (*ei*), Result will be a set ofransactions and subset of (F, X).
For each transaction *tk* in F (*ei*), *k=1,2...sup.count(F(ei))*
For each transaction *tl* in F (*ei*), *l=k+1,...sup.count(F(ei))*
{
If $tk \subseteq tl$ *or* $tl \supseteq tk$
$C=tk \cap tl$
If $C \neq$ NULL, *C* is a frequent item set.
}
Apply apriori property on *C* and generate all subsets, which are included the item *ei*.
}

Figure 8. Fp-soft set algorithm

6. EXPERIMENT RESULTS AND DISCUSSIONS

In this section, we evaluate the new FP-Softset frequent itemset mining method with two conventional algorithms - Apriori and FP growth. Using the data sets [7] and [5], we implemented the proposed approach in Python programming language. All the algorithms are implemented consecutively on Windows 7 Professional OS running on an Intel Core i5-6200U processor CPU with 6 GB RAM.

6.1. Grocery Store Data

Grocery store data set is collected from the data repository Kaggle. It contains 11 items and twenty transactions. Column headings are tea, bournvita, maggi, cornflakes, sugar, bread, coffee, biscuit, jam, cheese, and milk [7]. Here we performed frequent itemset mining using the Apriori, FP and proposed FP-Softset algorithms. This helps us to understand the purchasing behaviours of various customers. The procedure for generating useful information is given in Table 5.

Table 5. Transaction dataset for Grocery Basket analysis

TID	ITEM	TID	ITEM
1	Milk, bread, biscuit	11	Biscuit, coffee, cock, cornflakes
2	Milk, bread, biscuit, cornflakes	12	Biscuit, coffee, cock, cornflakes
3	Bread, tea, bournvita	13	Coffee, suger, bournvita
4	Milk, bread, jam, maggi	14	Bread, coffee, cock
5	Biscuit, tea, maggi	15	Bread, biscuit, suger
6	Bread, tea, bournvita	16	Coffee, suger, cornflakes
7	Tea, magi, cornflakes	17	Bread, suger, bournvita
8	Bread, biscuit, tea, maggi	18	Bread, coffee, suger
9	Bread, tea, jam, maggi	19	Bread, coffee, suger
10	Milk, bread	20	Milk, tea, coffee, cornflakes

Assume minimum support count as two and *Y* denotes the set of all transactions. The soft set representing supported data set is given in Figure 9.

$$(F, X)=\left\{\begin{array}{l}\text{milk= \{Y1, Y2, Y4, Y10, Y20\}, bread = \{Y 1, Y2, Y3, Y4, Y6, Y8, Y9, Y10, Y14, Y15, Y17, Y18, Y19\}, biscuit= \{Y 1, Y2, Y5, Y8 ,Y11, Y12, Y15\}, Cornflakes= \{Y2, Y7, Y11, Y12, Y16,Y20\},tea=\{Y3,Y5,Y6,Y7,Y8,Y9,Y20\},}\\ \text{bournvita=\{Y3,Y6,Y13,Y17\}, jam= \{Y4, Y9\},}\\ \text{cock= \{Y11, Y12, Y14\}, maggi= \{Y4, Y5, Y7, Y8, Y9\}, Coffee= \{Y11, Y12, Y13, Y14, Y16, Y18, Y19, Y20\}, Suger= \{Y 13, Y15, Y16, Y17, Y18, Y19\}}\end{array}\right\}$$

Figure 9. The soft set representing Table 5

Construct FP tree and prefix sub path for all items present in the FP tree. Apply proposed approach on conditional FP tree and frequent itemset generation. We will get the following frequent itemsets Table 6.

Table 6. Frequent itemset results using Soft Set theory

Item	Frequent Itemsets
biscuit	{biscuit},{biscut, bread},{biscuit, coffee},……. {biscuit, maggi, tea},{biscut, cock, coffee, cornflakes}
coffee	{coffee},{coffee, bread},{coffee, cock}……. {coffee, bread, suger},{coffee, cock, cornflakes}
. .	. .
cock	{cock}, {cock, cornflakes}
jam	{jam},{jam, bread},{jam, maggi}, {jam, magi, bread}

The same sets of frequent items are derived from both Apriori and FP algorithm methods. The execution time of our proposed approach - FP-Softset - through this data set is 0.00399 s. Response time of different methods is given below Figure 10.

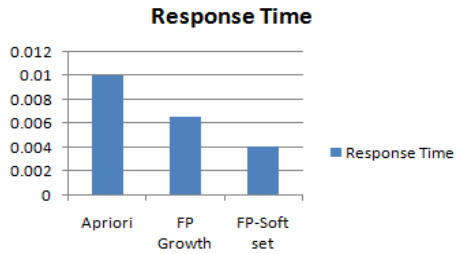

Figure 10. The comparison of executing time

Based on the above Figure 9, the improvement in response time of FP-Softset in comparison with the other two methods is given in the table below Table 7.

Table 7. Performance Progress of FP-Softset

Method	Response time improvement of FP-Softset
Apriori	56.25%
FP-Method	33.38%

6.2. Student Performance Data

This data is related to student achievement in secondary education in a Portuguese school and is collected from UCI repository. The data set contains relevant details of students for determining their performance. Dataset is contributed regarding the performance of the subject: Mathematics [5]. It consists of 33 attributes, 369 instances and used in the implementation of the above three-frequent itemset generation algorithms. The response time was found to be much faster when using FP-soft set algorithm than when using other two methods. The comparison result is given below in Figure 11.

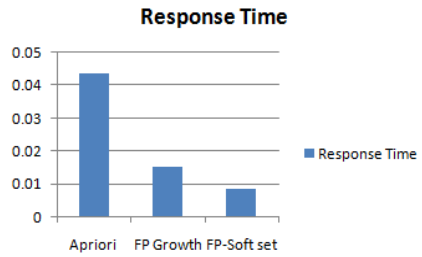

Figure 11. The comparison of executing time

In the Student Performance dataset, the response time of FP-Softset is significantly better than that of the other two traditional methods. The execution time of FP-Softset is .00875s. The improvement in response time when using FP-Softset, in comparison with the two traditional methods is given below in Table 8.

Table 8. Performance Progress of FP-Softset

Method	Response time improvement of FP-Softset
Apriori	79.96%
FP-Method	43.54%

Implementation of FP-Sofset approach on Grocery store and students performance data sets proved that its response time is better than other two conventional methods.

7. CONCLUSION AND FUTURE WORK

Soft computing methods are playing a significant role in knowledge discovery process. Different soft computing methods have emerged with reliable performance and accuracy. Among them, soft set is a powerful one and it has a wide range of applications in the data mining industry. We envisaged the potential of soft set theory in frequent itemset mining. This research work presented two well-known techniques - Apriori and FP-growth - for retrieving frequent itemset from data sets. We brought out the demerits of these methods. Frequent pattern growth is a distinctive method for finding frequent itemset. It has to go to different steps for completing its process. These steps increase memory utilization for FP- growth.

We experienced this limitation in our implementation. To overcome this limitation, we have implemented the proposed algorithm FP-Softset, which could reduce the overhead of FP-growth method. Apriori, FP-Growth, and FP-Softset algorithms implemented using Python language, were used to analyse two different data sets - Grocery and Student Performance. We compared the results of FP-Softset with the results obtained with Apriori and FP-Growth. In different experiments, FP-Softset produced a far better response time. In the future studies, we would like to extend our work to find association rules from large data sets.

REFERENCES

[1] J. Han, J. Pei, M. Kamber, "Data Mining: Concepts and Techniques", Elsevier, 3rd Edition, p. 744, 9 June 2011.

[2] S. Mitra, S.K. Pal, P. Mitra, "Data Mining in soft Computing Framework: A Survey", IEEE Transactions on Neural Networks, Vol. 13, Issue 1, pp. 3-14, 7 August 2002.

[3] D. Molodtsov, "Soft Set Theory - First Results", International Journal of Computers & Mathematics with Applications", Elsevier, Vol. 37, Issue 4-5, pp. 19-31, February-March 1999.

[4] Q. Zhao, S.S Bhowmick, "Association Rule Mining: A Survey", Technical Report, CAIS, Nanyang Technological University, Singapore, No. 2003116, January 2003.

[5] "Student Performance Data Set", https://archive.ics. uml/ datasets/studentperformance.

[6] S. Kottam, P. Varghese, "Uncertain Data Handling in Data Mining Using Soft Set Theory", International Journal of Applied Engineering Research, Vol. 10, No. 73, pp. 65-70, 2015.

[7] "Grocery Store Data Set, available at https://www.kaggle.com/ shazadudwadia/supermarket.

[8] S. Suba, T. Christopher, "An Efficient Data Mining Method to Find Frequent Itemsets in Large Database Using TR-FCTM", ICTACT Journal on Soft Computing, Vol. 6, Issue 2, 1 January 2016.

[9] P. Gogoi, R. Das, B. Borah, D.K. Bhattacharyya, "Efficient Rule Set Generation Using Rough Set Theory

for Classification of High Dimensional Data", International Journal of Smart Sensors and Ad Hoc Networks, Vol. 1, Issue 2, pp. 129-136, 2011.
[10] W.H. Au, K.C. Chan, "FARM: A Data Mining System for Discovering Fuzzy Association Rules", IEEE International Fuzzy Systems Conference Proceedings, Vol. 3, pp. 1217-1222, August 1999.
[11] A. Bhagat, S. Sharma, K.R. Pardasani, "Ontological Frequent Patterns Mining by Potential Use of Neural Network", International Journal of Computer Applications, Vol. 36, Issue 10, pp. 44-53, December 2011.
[12] R.V. Prakash, D. Govardhan, D.S. Sarma, "Mining Frequent Itemsets from Large Data Sets Using Genetic Algorithms", Artificial Intelligence Techniques-Novel Approaches & Practical Applications, No. 4, Article 7, pp. 38-43, 2011.
[13] J. Li, N. Cercone, "A Rough Set Based Model to Rank the Importance of Association Rules", International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing, pp. 109-118, Springer, Berlin, Heidelberg, August 2005.
[14] D. Chen, E.C.C. Tsang, D.S. Yeung, X. Wang, "The Parameterization Reduction of Soft Sets and its Applications", Computers and Mathematics with Applications, Vol. 49, Issues 5-6, pp. 757-763, April- May 2005.
[15] P.K. Maji, A.R. Roy, R. Biswas, "An Application of Soft Sets in a Decision Making Problem", Computers and Mathematics with Applications, Vol. 44, Issues 8-9, pp. 1077-1083, October-November 2002
[16] Z. Kong, L. Gao, L. Wang, S. Li, "The Normal Parameter Reduction of Soft Sets and its Algorithm", Computers and Mathematics with Applications, Vol. 56, Issue 12, pp. 3029-3037, December 2008.
[17] A.R. Roy, P.K. Maji, "A Fuzzy Soft Set Theoretic Approach to Decision Making Problems", Journal of Computational and Applied Mathematics, Vol. 203, Issue 2, pp. 412-418, June 2007.
[18] Y. Zou, Z. Xiao, "Data analysis Approaches of soft Sets under Incomplete Information", Knowledge Based Systems, Vol. 21, Issue 8, pp. 941-945, December 2008.
[19] J. Bilbao, E. Bravo, O. Garcia, C. Varela, C. Rebollar, "Particular Case of Big Data for Wind Power Forecasting: Random Forest", International Journal on Technical and Physical Problems of Electrical Engineering (IJTPE), Issue 42, Vol. 12, No. 1, pp. 25-30, March 2020.
[20] U. Allimuthu, K. Mahalakshmi, "Observed Survey on Efficient Route Interaction of Mobile Nodes in MANET", J. Adv. Res. Dyn. Control Syst, 10-Special, pp. 952-965, 2018.
[21] V. Yousefi, S. Kheiri, S. Rajebi, "Evaluation of K-Nearest Neighbor, Bayesian, Perceptron, RBF and SVM Neural Networks in Diagnosis of Dermatology Disease", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 42, Vol. 12, No. 1, pp. 114-120, March 2020.
[22] T. Herawan, M.M. Deris, "A Soft Set Approach for Association Rules Mining", Knowledge-Based Systems, Vol. 24, Issue 1, pp. 186-195, February 2011.

BIOGRAPHIES

Santhosh Kottam completed his B.Sc. degree in Mathematics from Mahatma Gandhi University (Kerala, India) and the Master of Computer Applications (MCA) from Madras University (Chennai, India). He is currently pursuing Ph.D. in the area of Data Mining at Bharatiyar University (Coimbatore, India). He has more than 19 years of teaching experience, which includes UG and PG. He has been serving FISAT as Assistant Professor (Senior Grade) in Department of Computer Applications since May 2008. He has published research papers in the international journals, national and international conferences.

Varghese Paul received B.Sc. degree in Electrical Engineering from Kerala University (Thiruvananthapuram, India), M. Tech degree in Electronics and Ph.D. degree in Computer Science from Cochin University of Science and Technology (Kochi, India). He is a Research Supervisor of Cochin University of Science and Technology, M G University Kottayam, Anna Technical University Chennai, Bharathiar University Coimbatore, Bharathidasan University Trichy and Karpagam University Coimbatore. Under his guidance, 29 research scholars had already completed research studies and degree awarded. His research areas are data security using cryptography, data compression, data mining, image processing and governance. He developed TDMRC coding system for character representation in computer systems and encryption system using this unique coding system. He has published many research papers in international as well as national journals and a text book also.