An Efficient Approach of Discovery of Frequent Data Set from Big Operational Database

Priyanka V. Gajare.
Computer Engineering
Alard College of Engineering
Pune, India.
gajare.priyanka2@gmail.com

Mrs. Sonali Patil.
Computer Engineering
Alard College of Engineering
Pune, India

Abstract - Currently in real world scenario data uncertainty is the most major issue in the real time applications where these data are generated from various devices daily from various users. So, the important part is to find the important data from them. In this paper, we propose to measure pattern frequentness based on the various possible world semantics. We are looking to establish two uncertain sequence data models abstracted from many real-life applications involving uncertain sequence data, and based on that formulate the problem of mining probabilistically frequent sequential patterns (or p-FSPs) from data that conform to our models. By using the projection strategy of famous prefixspan algorithm, we are looking to develop an algorithm called U-PrefixSpan for probabilistically frequent sequential pattern mining. UPrefixSpan avoids the problem of "possible world explosion" and when combined with pruning techniques and one validating technique achieves good performance. Theoretically study and analysis shows that our work proposed do the better with compare to existing system.

Keywords: *– Uncertain datasets, frequent sequential patterns, Prefix Span algorithm*

____*****____

I. INTRODUCTION

The problem of Sequential Pattern Mining in which involves discovery of frequent sequences of events in data with a temporal component. Frequent sequential pattern mining has become a classical and well-studied problem in data mining techniques. In previous frequent Sequential pattern mining techniques presented, the database to be mined consists of tuples present in table. A tuple may record a retail transaction (event) by a customer (source) or we can say an observation of an object/person (event) by a sensor/camera (source). The components of the tuple are certain, or completely determined.

It is recognized that data obtained from a wide range of data sources is inherently uncertain such as those data arising from sensor readings and GPS trajectories. This paper is concerned with frequent sequential pattern mining in probabilistic databases or uncertain databases, a popular framework for modeling uncertainty.

In our proposed work we consider the problem of mining frequent sequential patterns in the context of uncertain datasets or probabilistic dataset. In contrast to previous work that adopts *expected support* to measure pattern frequentness here we are looking to define pattern frequentness based on the various possible world semantics. This proposed framework gives us effective mining of high quality patterns with respect to a formal probabilistic datasets or uncertain datasets. There are two uncertain sequence data models (sequence-level and element-level models) abstracted from many real-life applications involving uncertain sequences.

To our knowledge this is the first work that attempts to solve the problem of p-FSP mining. We consider two general uncertain sequence data models that are abstracted from many real-life applications involving uncertain sequence data first the sequence-level uncertain model and the second element-level uncertain model. Based on the prefix-projection method of PrefixSpan algorithm, we present two new U-PrefixSpan algorithms that mine p-FSPs from uncertain data conforming to our models. Various Pruning techniques and a fast validating method are developed to further improve the efficiency of U-PrefixSpan algorithm.

II. LITERATURE SURVEY

A comprehensive survey of traditional data mining problems such as frequent pattern mining in the context of uncertain data can be found in [6]. Some of the concepts and issues arising from traditional sequential pattern mining and the mining of uncertain data are presented below

A. UApriory Algorithm

Frequent item set mining algorithm based on the expected support was proposed by Chui et al. [1].The frequent item set mining algorithm Apriori algorithm for uncertain environment is an extension of the well-known and expected to find support based frequent item sets to generate and test framework uses. But it does not scale well to large datasets is in limit. Due to the uncertain nature of the data associated with each item as a potential value item sets is required to act with these values. Possibilities exist for low value especially when the efficiency degrades and the problem becomes more serious and uncertain datasets.

B. UApriory with data trimming

To improve the efficiency of the U-Apriori algorithm, the data trimming technique was proposed [2].The main idea of the original dataset to trim off the items with low survival probabilities and instead is to mine trimmed dataset. So that insignificant increment candidate can reduce the computational cost. Sorting dataset size is much smaller than the original one, because besides, I / O cost can be reduced. Apriori trimming process for the application of the framework needs to be changed.

Trimming of the mining process modules in a precarious dataset D project starts by passing. The first row of data items once received by D scanning. A trimmed dataset D is smaller than a threshold trimming possibilities exist is constructed by removing all items.

C. Tree based Approaches

Using the tree structure of the tree-based approach [3] .From the tree structure to store persistent data, rather than making frequent item set to find the candidate generation and candidate does not include sorting steps are based on the different item sets Apriori. There are also modified F for uncertain data growth as these algorithms mining algorithms can be used.

D. *Sequential pattern mining*

Frequent item set mining, graph pattern mining and sequential pattern mining has been studied in the context of uncertain datasets that are very important pattern mining problems. For the problem of frequent pattern mining, the general pattern of work is expected to measure and support frequentness uses [4]. Where, some experimental results are expected to support the use of [5] found that may render the missing pattern is important. As a result, recent research has focused on the possible use of support.

III. IMPLEMENTATION DETAILS

Here we introduce a new pattern-growth method for mining frequent sequential patterns which is called as Sequential UPrefixSpan. The main idea behind is that instead of projecting sequence databases by considering all the possible occurrences of frequent subsequences here the projection is based only on frequent prefixes because any frequent subsequence can always be found by growing a frequent prefix. SO here we have presented

1. Sequence-level U-PrefixSpan
2. Element-level U-PrefixSpan

Each of them have their own issues to handle and deal with the handling the sequence pattern generation and mining those from the datasets. Along with this mentioned strategies to deal with we are going to implement the one of it i.e. Sequence-level U-PrefixSpan which is the core part of the proposed work.

A. Mathematical Model

Presence Probability: The probability of the presence of a patterm α in a probabilistic sequence *s* is given by

$$\Pr\{\alpha \dot{\subseteq} s\} = \sum_{\alpha \dot{\subseteq} s_i} \Pr\,(Pwi)$$

Where,
s_i - deterministic instance of probabilistic sequence S in the possible word *pwi* .
Pr(pwi) - existence probability of possible world *pwi*.

Expected Support: The sum of the expected probabilities of the presence of *a* in each of the sequences in databases.
Pattern *a* is expectably frequent if

$$\text{expsup}(a) > \tau_{\text{sup}}$$

where,
expsup(a) – expected support of a pattern *a*
τ_{sup} - support threshold

Probabilistic frequentness: Pattern *a* is probabilistically frequent iff

$$\Pr\{\text{sup}\ (a) \geq \tau_{sup}\} \geq \tau_{prob}$$

Where,
τ_{sup} - support threshold
τ_{prob} – probability threshold
Given a sequence-level probabilistic sequence *si* and a pattern α we now discuss how to obtain the α-projected probabilistic sequence *Si*|*α.*

Conceptually, the α-projected database D|α is constructed by projecting each probabilistic sequence *si* ∈ *D* onto *Si*|*α.*

we grow α by appending to it one element *e* to obtain a new pattern *αe* and then recursively checking the frequentness of αe. To keep the number of such new patterns small in each growing step, we maintain an element table T|α that stores only those elements *e* that still have a chance of making αe *frequent.*

We construct *T*|αe from *T*|α that is the element table during the pattern growth using Algorithm presented below.

B. Algorithms

Algorithm 1: PMFCheck(VECα)

Input: probability vector: VECα
Output: mark of frequentness: tag; pmf: fa

```
If |VECa|=1 then
    $f_a(0) \leftarrow 1$- VECa[1], $f_a(1) \leftarrow$VECa[1]
    return (1-$F_a(\tau_{sup} - 1) \geq \tau_{prob}$),$f_a$)
Partition VECa into $VEC^1_a$ and $VEC^2_a$ , where |$VEC^1_a$ |=$[\frac{n}{2}]$
($tag_1$,$f^1_a$) ←PMFCheck($VEC^1_a$)
If $tag_1$= TRUE then
    return(True , Φ)
($tag_2$,$f^2_a$) ←PMFCheck($VEC^2_a$)
If $tag_2$= TRUE then
    return(True , Φ)
$f_a$ ← convolution($f^1_a$,$f^2_a$)
return (1-$F_a(\tau_{sup} - 1) \geq \tau_{prob}$,$f_a$)
```

Algorithm 2: Prune(T|α,D|αe)

Input: Elemtn table T|*α, projected probabilistic databse D*|*αe*
Output: Elemet table T|*αe*

```
$T|\alpha e \leftarrow \Phi$
For each element $l$ € $T|\alpha$ do
Check CntPrune with pattern $l$ on $D|\alpha e$
If $l$ is not pruned then
Check markovPrune with pattern $l$ on $D|\alpha e$
If $l$ is not pruned then check ExpPrune with pattern $l$ on $D|\alpha e$
If $l$ is not pruned then $T|\alpha e \leftarrow T|\alpha e$ U $\{l\}$
```

Algorithm 3: SeqU-PrefixSpan (αe, D|α),T|α)

Input : Current path αe, projected probabilistic databse D|α, element table T|α

```
VECαe $\leftarrow \Phi$
For each projected sequence $S_i|\alpha$ € $D|\alpha$ do
$Pr(S_i \mid \alpha e) \leftarrow 0$
For each instance $S_{ij} \mid \alpha = <pos, pr(S_{ij}) >$ € $S_i|\alpha$ do
Find its corresponding sequence $S_{ij}$ € $D$
If $e$ € $S_{ij}$ [pos +1],.....,len($S_{ij}$)] then
$Pr(S_i| \alpha e) \leftarrow Pr(S_i| \alpha e) + Pr(S_{ij})$
$C' \leftarrow \min_{C \geq pos+1} \{S_{ij}[c]=e\}$
Append $(C', \Pr(S_{ij}))$ to $S_i|\alpha e$
If $pr(S_i|\alpha e) > 0$ then
Append $Si|ae$ to $D|ae$
Append $pr(S_i|\alpha e)$ to VECαe
(tag,fαe) $\leftarrow$ PMFcheck(VECαe)
If tag=true then
Output αe
$T\alpha e \leftarrow$ Prune $(T|\alpha , D|\alpha e)$
For each element $l$ € $T|\alpha e$ do
Seq U-Prefix span(αel, $D|\alpha e$, $T|\alpha e$)
```

In this section for giving details of this method, we direct the problem of p-FSP mining on datasets that conform to the sequence-level uncertain model. In our proposed work a pattern-growth algorithm for this which called *SeqU-PrefixSpan* to overcome this problem. Compared with *PrefixSpan* the *SeqU-PrefixSpan* algorithm needs to addresses the following additional issues coming from the sequence - level uncertain model which are as follow:

1. Frequentness validating
2. Pattern Frequentness Checking
3. Candidate Elements for Pattern Growth

These are the main core issues associated with this technique of probabilistic sequence patterns mining with sequence-level U-PrefixSpan which we are going to concern in our proposed work along with the implementation of the algorithm for the same. We will see the algorithm details in the sub-section below:

SeqU-PrefixSpan algorithm in our work recursively performs pattern growth from the previous pattern say *a* to the current B= *αe*, by appending an element *e* ∈ T|a. where T|a is set of elements which are nothing but generated from the local datasets. We also construct the current projected probabilistic database for the generation of local datasets D|B using the previous projected probabilistic database in the sequential pattern. For the execution and testing of the above algorithm work we are going to use one application scenario where we are going to generate the datasets locally in that application from the local user of the proposed architecture workflow which as follows:

So it goes in the following way where the performance of *SeqU-PrefixSpan* is checked by the, implementation of data generated which datasets that relate to the sequence-level uncertain model. Given the configuration *(n, m, l, d)*, our generator generates *n* probabilistic sequences. For each probabilistic sequence, the number of sequence instances is randomly chosen from the range [1,*m*] which is decided from the local datasets. The length of a sequence instance is randomly chosen from the range [1,*l*], and each element in the sequence instance is randomly picked from an element table with *d* elements these are the important parameters in the proposed architecture for the finding the sequence patterns based on the probability of the patterns

C. Fast Validating Method

In this part, we present this method of fast validation for speeding up the *U-PrefixSpan* algorithm and to increase the efficiency of the same. Fast validating method involves two approximation techniques which checks the probabilistic frequentness of patterns and reducing the time complexity from *O(n* log2 *n)* to *O(n)* which is achive with the help of this method means its work as the complimentary for the our proposed algorithm to enhance the efficiency of the algorithm. So here we are going to apply the two models in the proposed architecture of the system design which are namely (for e.g. a Poisson or Normal model) by which we can verify our p-FSPs very fastly and the efficiently.

D. Software and hardware requirements

a. Hardware Configuration
- Processor - PentiumIV 2.6 ghz
- RAM - 512 mbdd ram
- Monitor - 15" color
- Hard Disk - 20 GB
- Key Board - Standard Windows Keyboard

b. Software Configuration
- Operating System - Windows XP/7
- Programming Language - Java
- Database - MySQL
- Tool – Netbeans

IV. RESULTS

We report on an experimental evaluation of our proposed work. Our implementations are Java, executed on a machine with a 3.2GHz Intel CPU and 3GB RAM running windows 7. We begin by describing the dataset used for experiment. After that we demonstrate the scalability of our algorithms (reported running times are averages from multiple runs).
The dataset we are using here is a synthetic dataset.

Following figure shows the output screen for finding frequency of data items to mine data items according to their frequency.

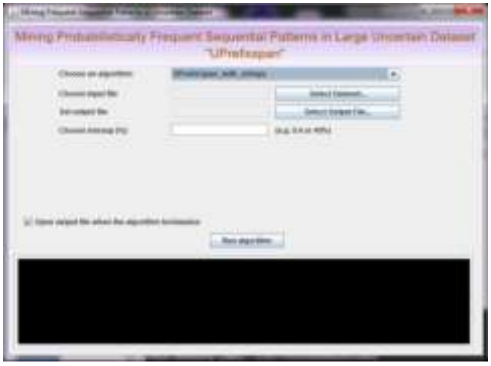


Fig: Selecting Algorithm

Mining Probabilistically Frequent Sequential Patterns in Large Uncertain Dataset
"UPrefixspan"
Save
Save In:
Desktop
File Name:
Files of Type:
All Files
Save
Cancel

Fig: Loading Dataset

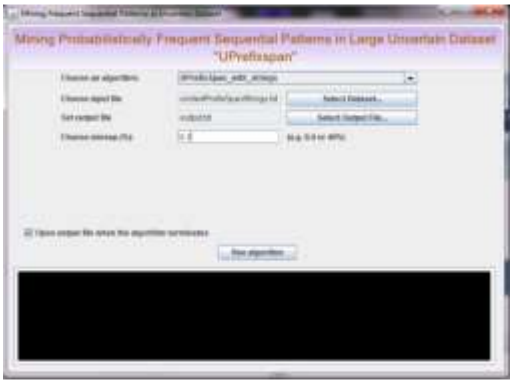
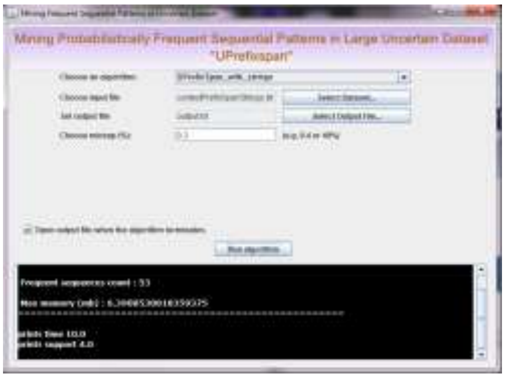


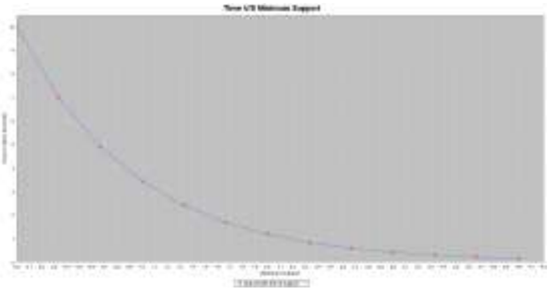
Fig: Setting Output folder and Minimum Support



```
Frequent sequences count : 33
Max memory (mb) : 6.3[illegible]
=============================================================
[illegible] time 10.0
[illegible] support 4.0
```

Fig: Output after Running Project
`

The results of our project can be analyzed based on minimum support value against the execution time. Then the attribute analysis graph is shown in following figure and the time measurement is taken in milliseconds.



V. CONCLUSION AND FUTURE SCOPE

Here in our proposed work we have designed and probably continuous sequential pattern mining problem in uncertain database study. Our study involving uncertain sequence data are fundamental to many real-life applications that are based on two different algorithms U-apriory algorithm and prefix span algorithm. We also generate the local datasets as well as various applications in the context of uncertain sequence pattern mining efficiency improvement frequentness check pattern, to accelerate the development of novel sorting rules and a quick valid method.

To increase the performance of algorithm effectively expand and enhance the work of the future can be mentioned in Section C at the local level to implement the model for the generation of datasets.

VI. REFERENCES

[1] Chiu, C.K. Chui, B. Kao, "Mining Frequent Item sets from Uncertain Data," Proc. 11th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining, 2007.

[2] L. Wang, R. Cheng, S.D. Lee, "Accelerating Probabilistic Frequent Item set Mining: A Model-Based Approach," Proc. 19th ACM Int'l Conf. Information and Knowledge Management (CIKM), 2010.

[3] Q. Zhang, F. Li "Finding Frequent Items in Probabilistic Data," Proc. ACM SIGMOD Int'l Conf. Management of Data, 2008

[4] C. C. Aggarwal, J. Wang. "Frequent Pattern Mining with Uncertain Data". In *SIGKDD*, 2009.

[5] Q. Zhang, K. Yi "Finding Frequent Items in Probabilistic Data". In *SIGMOD*, 2008

[6] C. C. Aggarwal, and P. S. Yu, "A survey of uncertain data algorithms and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 21,no. 5, pp. 609–623, May 2008.

[7] sequential patterns in uncertain databases," in *Proc. 15th Int. Conf. EDBT*, New York, NY, USA, 2012. [10] C. Gao and J. Wang, "Direct mining of discriminative patterns for classifying uncertain data," in *Proc. 16th ACM SIGKDD*,Washington, DC, USA, 2010.

[8] N. Pelekis, I. Kopanakis, E. E. Kotsifakos, E. Frentzos, and Y. Theodoridis, "Clustering uncertain trajectories," *Knowl. Inform. Syst.*, vol. 28, no. 1, pp. 117–147, 2010.

[9] H. Chen, W. S. Ku, H. Wang, and M. T. Sun, "Leveraging spatio temporal reundancy for RFID data cleansing," in *Proc. ACM SIGMOD*, Indianapolis, IN, USA, 2010.

[10] Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proc. 13th Int. Conf. VLDB*, Toronto, ON, Canada, 2004.