VIETNAM GENERAL CONFEDERATION OF LABOUR

TON DUC THANG UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY

REPORT OF THESIS RESEARCH 1

PROBABILISTIC WEIGHTED FREQUENT ITEMSET MINING OVER UNCERTAIN DATA STREAMS

Advised by: **Dr Nguyễn Chí Thiện**

Authors: **Nguyễn Đình Quý – 520H0675**

Nguyễn Nhất Thống – 52000808

HO CHI MINH CITY, 2024

VIETNAM GENERAL CONFEDERATION OF LABOUR

TON DUC THANG UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY

REPORT OF THESIS RESEARCH 1

PROBABILISTIC WEIGHTED FREQUENT ITEMSET MINING OVER UNCERTAIN DATA STREAMS

Advised by: **Dr Nguyễn Chí Thiện**

Authors: **Nguyễn Đình Quý – 520H0675**

Nguyễn Nhất Thống – 52000808

HO CHI MINH CITY, 2024

ACKNOWLEDGMENT

We sincerely thank the Faculty of Information Technology for providing us with the opportunity to access and complete the report. We would like to express our heartfelt gratitude to Dr Nguyen Chi Thien for guiding us in completing the report.

During the process of preparing the report, due to limited knowledge and experience, there may be some shortcomings. We greatly appreciate any feedback from you so that we can learn more skills and experiences, and improve further.

We sincerely thank you!

Ho Chi Minh City, day month year 2024
Author
(Signature and full name)

THE REPORT WAS COMPLETED
AT TON DUC THANG UNIVERSITY

I hereby declare that this is our own research work, conducted under the scientific supervision of Dr. Nguyen Chi Thien. The research contents and results in this topic are truthful and have not been previously published in any form. The data presented in tables and figures, serving for analysis, comments, and evaluations, are collected by the author from various sources, clearly referenced in the reference section.

Furthermore, the project also incorporates some comments, evaluations, as well as data from other authors, and different organizations, all of which are appropriately cited and referenced.

If any misconduct is discovered, I take full responsibility for the content of my project. Ton Duc Thang University is not liable for any copyright infringements or violations caused by me during the implementation process (if any).

Ho Chi Minh City, day month year 2024
Author
(Signature and full name)

ABSTRACT

Mining frequent itemsets from uncertain data streams is a critical task in data analysis, yet it poses unique challenges due to the inherent uncertainty and dynamic nature of streaming data. This report presents a novel approach, focusing on probabilistic weighted frequent itemset mining over uncertain data streams.

Our method, PFIT, leverages an efficient in-memory index to manage data synopsis, facilitating real-time output of probabilistic frequent itemsets within sliding windows. We introduce PFIMoS, a dynamic depth-first algorithm, to construct and update PFIT, optimizing runtime and memory usage by estimating probabilistic support ranges. Additionally, we tackle the computational overhead associated with low minimum support thresholds and dense data through PFIMoS+, an error-parameter-guided heuristic algorithm.

Our methods are developed based on the algorithmic proposals found in the study by *Li, H., Zhang, N., Zhu, J., Wang, Y., & Cao, H. (2018) on Probabilistic Frequent Itemset Mining over Uncertain Data Streams*. Furthermore, to enhance flexibility in usage, improve the quality of outcomes, and increase efficiency in processing large datasets, we have incorporated a probability weighting approach based on the research by *Li, Z., Chen, F., Wu, J., Liu, Z., & Liu, W. (2020) in Efficient Weighted Probabilistic Frequent Itemset Mining in Uncertain Databases*. This integration aims to refine our methodology by considering the importance and uncertainty associated with each data item, thereby enriching our analysis and enabling more nuanced interpretations of complex data environments.

Through empirical evaluation, we demonstrate the effectiveness and efficiency of our approach, offering insights into scalable probabilistic frequent itemset mining in uncertain data stream environments.

CONTENTS

LIST OF FIGURES

LIST OF TABLE, DIAGRAM

ABBREVIATIONS

λ	Minimum support
τ	Minimum probability
i	Distinct item
I	Uncertain item
I	Uncertain itemset
UT	Uncertain Transaction
UD	Uncertain Database
η	Sliding window
p(X)	Occurrence probability of itemset
S(X)	Support of itemset
$\Lambda^{E}(\mathrm{X})$	Expected Support of itemset
w(X)	Weight of itemset
Λ_{τ}^{P}	Probability support
$P_{\Lambda(\mathrm{X})\geq i} > \tau$	Probability support in itemset but greater than minimum probability
$lb(\Lambda_{\tau}^{P}(\mathrm{X}))$	Lower bound
$ub'(\Lambda_{\tau}^{P}(\mathrm{X}))$	Upper bound
$\mathrm{w}\Lambda_{\tau}^{P}$	Probability support with weight

CHAPTER 1. INTRODUCTION AND OVERVIEW TOPIC

1.1 Introduction

In the field of data mining, uncovering frequent itemsets within streams of data is crucial for extracting meaningful insights. The exploration of probabilistic frequent itemsets in uncertain data streams, as presented by ***Li et al. (2018)*** in "Probabilistic frequent itemset mining over uncertain data streams" in Expert Systems With Applications, marks a significant advancement in this domain. Their work introduces the PFIMoS and PFIMoS+ algorithms, which efficiently mine such itemsets by employing an innovative in-memory index, PFIT, and by estimating the range of probabilistic support to reduce computational demands.

Building upon the foundation laid by ***Li et al. (2018)***, this project seeks to extend the existing methodology by introducing weights to the frequent probability itemsets. This modification aims to enhance the granularity of the analysis by accounting for the varying significance of item occurrences within itemsets, thereby offering a more refined understanding of data stream patterns. The weighted approach promises to improve the algorithms' applicability across different contexts, enabling a prioritized analysis of itemsets based on their relevance or importance to specific applications.

This report aims to detail the theoretical underpinnings of this modification, its implementation, and the potential impacts it holds for the field of data mining. Through a meticulous examination of the proposed approach and its comparison with the foundational work of ***Li et al. (2018)***, this report endeavors to highlight the innovation and added value brought about by incorporating weights into the mining of probabilistic frequent itemsets.m mining research.

1.2 Challenges and our contributions

- **Sliding Window Model and PFIT**:
  - **Approach**: Using a sliding window model to analyze data streams.

- **Data Structure**: Introducing PFIT (Probabilistic Frequent Item-set Tree), an in-memory data structure designed to store data synopsis of probabilistic frequent item-sets in a bottom-up manner.
- **Objective**: Efficiently reducing the search space during the analysis of uncertain data streams.

- **PFIMoS Algorithm**:
  - **Algorithm Name**: PFIMoS (Probabilistic Frequent Itemset Mining over Streams).
  - **Objective**: Incrementally discovering and maintaining probabilistic frequent itemsets.
  - **Innovation**: Utilizing a probabilistic support estimating method that computes upper and lower bounds of probabilistic support at a significantly lower cost.
  - **Performance Improvement**: The method, when used in conjunction with PFIT, is claimed to yield better overall performance.
- **PFIMoS+ Algorithm**:
  - **Algorithm Name**: PFIMoS+ (an improved version of PFIMoS).
  - **Enhancement**: Introducing a heuristic rule to reduce the count of probabilistic support computing that is not pruned by the bounds.
  - **Effectiveness**: The heuristic rule is claimed to be particularly effective when mining over streams, significantly reducing runtime costs, especially in scenarios with low minimum support or dense data.
- **Weighted PFIT, Weighted PFIMoS, Weighted PFIMoS+:**
  - **Algorithm Name:** Weighted PFIT, Weighted PFIMoS, Weighted PFIMoS+
  - **Objective**: Enhance both the performance and accuracy compared to the algorithm's original version before the integration of weights
- **Evaluation of Algorithms**:
  - **Datasets**: Evaluation conducted on 2 synthetic datasets and 4 real-life datasets.

- **Outcome**: The proposed algorithms are reported to be effective and efficient

1.3 Report organization

Chapter 1 Introduction and Overview: This initial chapter sets the stage by introducing the topic of probabilistic weighted frequent itemset mining. It outlines the challenges associated with uncertain data streams and enumerates our contributions to addressing these challenges. The chapter concludes with an outline of related work, establishing the context and relevance of our research.

Chapter 2 Theoretical Basis: The second chapter delves into the theoretical underpinnings of our study. It begins with the preliminaries and problem definition, laying the groundwork for understanding the algorithms and methods employed. Subsequent sections detail the various components of our approach, including the computation of probabilistic support bounds, the construction of a probabilistic frequent itemset tree, and the description of the PFIMoS algorithm. We also discuss algorithms for adding and deleting uncertain transactions and introduce an enhanced version of PFIMoS.

Chapter 3 Experiments: In this chapter, we present the empirical evaluation of our proposed methods. We describe the experimental setup, including the computing environment and datasets used. The results of our experiments are then thoroughly discussed

Chapter 4 Conclusion: The final chapter concludes the report, summarizing the key findings and contributions of our research.

References: At the end of the report, a comprehensive list of references is provided, documenting the scholarly works and sources cited throughout the report.

1.4 Related work:

To explore deeply into this topic. Two new definitions must be clearly. Firstly, ***Expected Frequent Item-set (Chui,Kao & Hung, 2007)***. For expected frequent itemset,this definition focuses on calculating the expected support of itemsets. The expected support can be computed with O(n) time complexity and

O(1) space complexity. Several algorithms have been devised to discover expected frequent itemsets, mainly based on a priori rules and traditional data structures like ***FP-tree (Cuzzocrea & Leung, 2016; Leung & MacKinnon, 2014)*** and ***H-struct (Aggarwal, Li, Wang & Wang, 2009)***.

However, expected support cannot show the total of probabilistic characteristic of data. Therefore, ***Probabilistic Frequent item-set (Bernecker, Kriegel, Renz, Verhein, & Zuefle, 2009)*** was invented.Mining probabilistic frequent itemsets can better discover probabilistic features. Algorithms like ***ProApriori (Bernecker et al., 2009)*** and ***ProFPGrowth (Bernecker, Kriegel, Renz, Verhein, & Züfle, 2012)*** have been proposed to discover probabilistic frequent itemsets, utilizing a priori rules and the ***ProFP-tree (Sun et al., 2010)*** data structure to maintain itemsets.

In the realm of uncertain data mining over data streams, algorithms such as ***FEMP (Akbarinia & Masseglia, 2013)*** and ***PFIMUDS (Akbarinia & Masseglia, 2013)*** have been proposed to discover probabilistic frequent itemsets in data streams. These methods focus on computing probabilistic support and employ techniques like a priori rules to reduce computational costs.

Further advancing this domain, **Li et al. (2018)** developed **PFIMoS** and **PFIMoS+** algorithms to efficiently mine probabilistic frequent itemsets in uncertain data streams, using an in-memory index (**PFIT**) and probabilistic support ranges to significantly cut down on computational time and memory use, surpassing previous methods like **TODIS-Stream** and **FEMP**. This breakthrough enhances uncertain data stream mining for practical use. In a subsequent study by **Li, Chen, Wu, Liu, and Liu (2020)**, a new algorithm for mining weighted probabilistic frequent itemsets **(w-PFIs)** in uncertain databases was introduced, employing a novel probability model and three pruning techniques to efficiently reduce the search space. This research advances weighted uncertain data mining by combining weights and probabilistic assessments, providing a thorough analysis of itemset mining's significance and uncertainty.

CHAPTER 2. THEORETICAL BASIS

2.1 Preliminaries

- **Definition 1**: Distinct Item (i)
  - **Concept:** The list of different items in uncertain database
  - **Formula:** Where n is the number of distinct items. Available on all data

$$i = \{\, i_1, i_2, \dots, i_\mathrm{n} \,\} \tag{2.1}$$

  - **Explanation and Example:** This is 5 different items in uncertain database

$$i = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$$

- **Definition 2:** Uncertain Item
  - **Concept:** It is a random variable. Corresponding to each item, there will be a different occurrence probability
  - **Formula:**

$$\mathrm{I} \ = \{\, i_1, \mathrm{p}_1 \,\} \tag{2.2}$$

  - **Explanation and Example:** This presents the item 'A' and the occurrence probability '0.3'

$$\mathrm{I} \ = \{\mathrm{A}, 0.3\}$$

- **Definition 3:** Uncertain itemset
  - **Concept:** It is Mutitional random variables
  - **Formula:** Where n is the number of random variables.

$$\mathbf{I} \ = \{(\, \mathrm{i}_1, \mathrm{p}_1), (\, \mathrm{i}_2, \mathrm{p}_2), \dots, (\, \mathrm{i}_\mathrm{n}, \mathrm{p}_\mathrm{n})\} \tag{2.3}$$

  - **Explanation and Example:** This example shows the Uncertain itemset with 3 pairs of uncertain items and the probabilities are (A, 0.3), (B, 0.6), (D, 0.8)

$$\mathbf{I} \ = \{(\mathrm{A}, 0.3), (\mathrm{B}, 0.6), (\mathrm{D}, 0.8)\}$$

- **Definition 4:** Uncertain Transaction
  - **Concept:** Uncertain Transaction is a list of multidimensional random variables. Corresponding to a multi-dimensional random variable, there will be a symbolic ID.
  - **Formula:** Where n is the number of uncertain transactions

$$\mathbf{UT} = \{\mathrm{ID}, \{(\mathrm{i}_1, \mathrm{p}_1), (\mathrm{i}_2, \mathrm{p}_2), \ldots, (\mathrm{i}_\mathrm{n}, \mathrm{p}_\mathrm{n})\}\} \tag{2.4}$$

  - **Explanation and Example:** This example shows 1 Uncertain transaction consists of a list of multidimensional random variables including (A, 0.3), (B, 0.6), (D, 0.8) and symbolic ID

$$\mathbf{UT} = \{\mathrm{ID}, \{(\mathrm{A}, 0.3), (\mathrm{B}, 0.6), (\mathrm{D}, 0.8)\}\}$$

- **Definition 5:** Uncertain Database
  - **Concept:** Uncertain Database is a collection of uncertain transactions.
  - **Formula:** where n is the number of uncertain transactions

$$\mathrm{UT} = \{\mathrm{UT}_1, \mathrm{UT}_2, \ldots, \mathrm{UT}_\mathrm{n}\} \tag{2.5}$$

  - **Explanation and Example:** This example shows an uncertain database as a table where each of its rows is an uncertain transaction

ID1	{(A,0.3), (B, 0.6), (D, 0.8)}
ID2	{(C, 0.4), (E, 0.7)}
ID3	{(B, 0.9)}
ID4	{(D, 0.6), (E, 0.7)}
ID5	{(A,0.5), (B, 0.4),(C, 0.6),(D, 0.8), (E, 0.8)}

Table 2.1 Example of a Uncertain Database

- **Definition 6**: Support
  - **Concept:** Support is the frequency of an itemset appearing in the entire uncertain database.
  - **Formula:**

$$\mathrm{S(I)} = \sum_{t\epsilon T} 1_{\{i\epsilon t\}} \tag{2.6}$$

- **Explanation and Example:** It compute the sum of items occur in uncertain database. Similarly, according to **Table 2.1,** itemset A occurs in the first Uncertain transaction 1 time so increase to 1, Otherwise, in Uncertain transaction 2, 3, 4 don't have so increase to 0. Finally, in the last uncertain transaction also own 'A' so we continue rising to 1. At the end of computing support, collecting support is 2

- **Definition 7:** Expected Frequent Itemset:
  - **Concept:** An itemset is considered a frequent itemset when expected support > minimum support. Below is the formula to calculate expected support. (Expected support is a form of support but is often used in uncertain data streams)
  - **Formula:**

$$\Lambda^{E}(\mathrm{X}) = \sum_{i\epsilon \mathrm{UT}} \Lambda(\mathrm{X})\mathrm{p}(\mathrm{X}) \tag{2.7}$$

  - **Explanation and Example**: Calculate expected frequent itemset based on the sum of each item's probability on each transaction

- **Definition 8:** Probabilistic Frequent Item-set
  - **Concept:** An itemset is considered frequent when probabilistic support > minimum support. Below is the calculation formula
  - **Formula:**

$$\Lambda_{\tau}^{P} = Max\{i|P_{\Lambda(\mathrm{X})\geq i} > \tau\} \tag{2.8}$$

Where:

$$P_{\Lambda(\mathrm{X})\geq i}(X) = \sum_{PW\epsilon\psi,\Lambda_{PW}(X)\geq i} p(PW) \tag{2.9}$$

  - **Explanation and Example:** For itemset 'A' we see that with minimum probability is 0.2 so first transaction and five transaction have probability greater than minimum probability so we have probability support equals to 2.

- **Definition 9:** Probability Support weight
  - **Concept**: Is a variation of probability support. But we will calculate more with weighted to block the conditions in ADDTRANS and DELTRANS functions
  - **Formula**:

$$\mathrm{w}\Lambda_{\tau}^{P} = \mathrm{w(X)}Max\{i|P_{\Lambda(\mathrm{X})\geq i} > \tau\} \quad (2.10)$$

Where: $$P_{\Lambda(\mathrm{X})\geq \mathrm{i}}(X) = \sum_{i\epsilon \mathrm{UT},\Lambda(X)\geq i} p(\mathrm{X}) \quad (2.11)$$

  - **Explanation and Example**:Similar to probability support but we will multiply each Probability support with weight of itemset

- **Definition 10**: Lower bound
  - **Concept**: The smallest value that a random variable, a set of numbers, or a quantity can take or not exceed towards the bottom. In the context of an algorithm or a probability distribution, the lower bound has can indicate the minimum value below which the probability of finding a value of the variable is very low or non-existent.
  - **Formula**:

$$lb(\Lambda_{\tau}^{P}(\mathrm{X})) = Max(lb'(\Lambda_{\tau}^{P}(\mathrm{X})), 0) \quad (2.12)$$

Where: $$lb'(\Lambda_{\tau}^{P}(\mathrm{X})) = \Lambda^{E}(\mathrm{X}) - \sqrt{-2\Lambda^{E}(\mathrm{X})ln(1-\tau)} \quad (2.13)$$

  - **Explanation and Example**: Base on Expected support and minimum probability we compute the lower bound and check the condition if it smaller than 0 we mark it 0

- **Definition 11**: Upper bound
  - **Concept**: The maximum value that a random variable, a set of numbers, or a quantity can reach or not exceed. In an algorithm or model, an upper limit can be used to indicates the value above which the probability of finding the value of the variable is very low or non-existent.

- **Formula**:

$$ub(\Lambda_{\tau}^{P}(\mathrm{X})) = Min(ub'(\Lambda_{\tau}^{P}(\mathrm{X})), \Lambda(\mathrm{X})) \tag{2.14}$$

Where:
$$ub'(\Lambda_{\tau}^{P}(\mathrm{X})) = \frac{2\Lambda^{E}(\mathrm{X}) - ln\tau + \sqrt{ln^{2}\tau - 8\Lambda^{E}(\mathrm{X})ln\tau}}{2} \tag{2.15}$$

- **Explanation and Example**: Base on Expected support and Support we compute the lower bound and check the condition if it smaller than 0 we mark it support

- **Definition 12:** Weighted
  - **Concept:** Is in the number of each itemset
  - **Formula:**

$$\mathrm{w}(X) = \frac{1}{|\mathrm{X}|}\sum_{x \epsilon X} \mathrm{w(x)} \tag{2.16}$$

  - **Explanation and Example**: We will compute the weight in each itemset base on the average weight of each item.

A	B	C	D	E
0.2	0.4	0.1	0.9	0.7

Table 2.2 Table describe the result of calculating weight of each item

2.2 UML Class Diagram

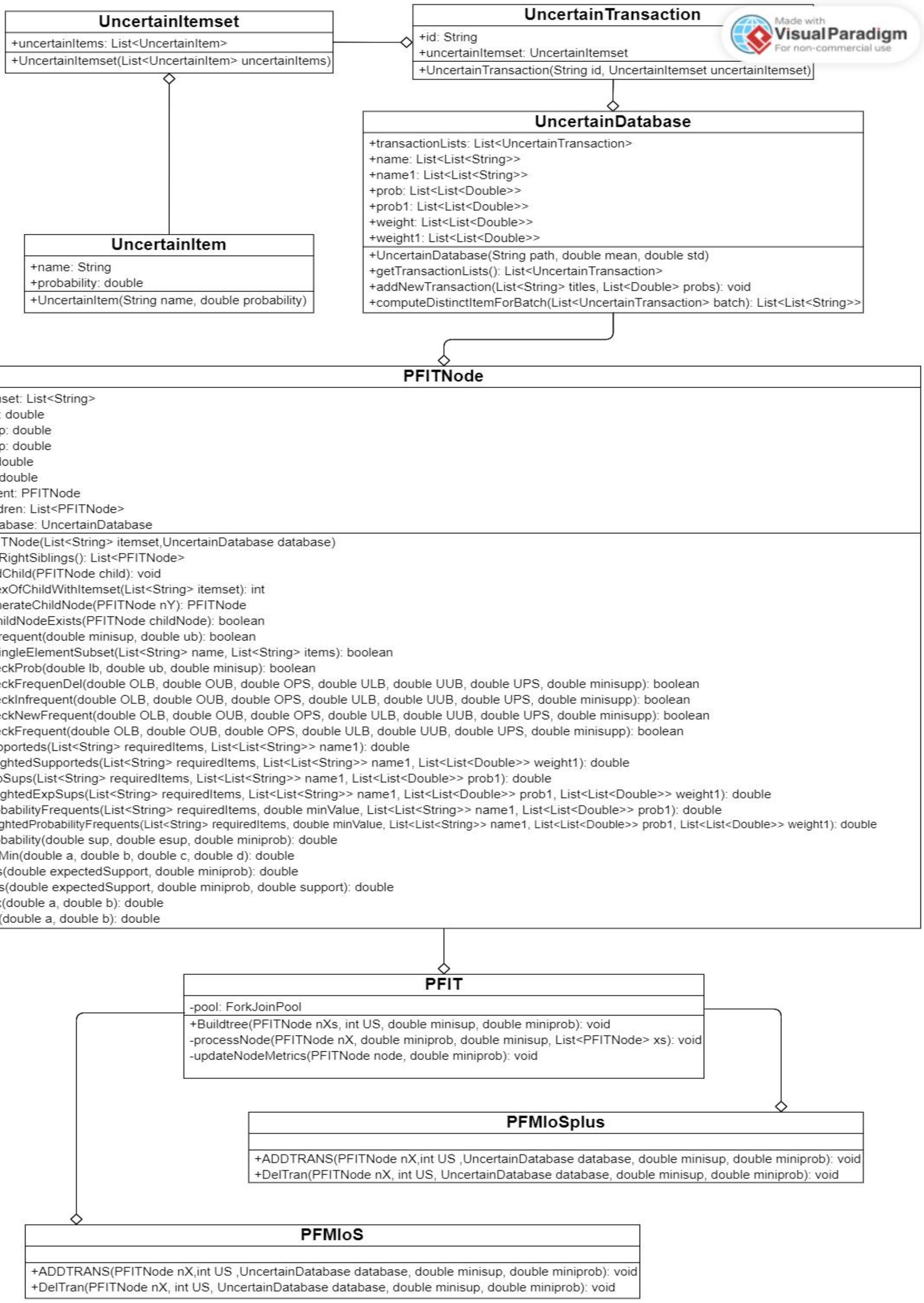

UncertainItemset
+uncertainItems: List<UncertainItem>
+UncertainItemset(List<UncertainItem> uncertainItems)
UncertainTransaction
+id: String
+uncertainItemset: UncertainItemset
+UncertainTransaction(String id, UncertainItemset uncertainItemset)
UncertainDatabase
+transactionLists: List<UncertainTransaction>
+name: List<List<String>>
+name1: List<List<String>>
+prob: List<List<Double>>
+prob1: List<List<Double>>
+weight: List<List<Double>>
+weight1: List<List<Double>>
+UncertainDatabase(String path, double mean, double std)
+getTransactionLists(): List<UncertainTransaction>
+addNewTransaction(List<String> titles, List<Double> probs): void
+computeDistinctItemForBatch(List<UncertainTransaction> batch): List<List<String>>
UncertainItem
+name: String
+probability: double
+UncertainItem(String name, double probability)
PFITNode
-itemset: List<String>
-sup: double
-esup: double
-psup: double
-lb: double
-ub: double
-parent: PFITNode
-children: List<PFITNode>
+database: UncertainDatabase
+PFITNode(List<String> itemset,UncertainDatabase database)
+getRightSiblings(): List<PFITNode>
+addChild(PFITNode child): void
-indexOfChildWithItemset(List<String> itemset): int
+generateChildNode(PFITNode nY): PFITNode
-isChildNodeExists(PFITNode childNode): boolean
+isFrequent(double minisup, double ub): boolean
+isSingleElementSubset(List<String> name, List<String> items): boolean
+checkProb(double lb, double ub, double minisup): boolean
+checkFrequenDel(double OLB, double OUB, double OPS, double ULB, double UUB, double UPS, double minisupp): boolean
+checkInfrequent(double OLB, double OUB, double OPS, double ULB, double UUB, double UPS, double minisupp): boolean
+checkNewFrequent(double OLB, double OUB, double OPS, double ULB, double UUB, double UPS, double minisupp): boolean
+checkFrequent(double OLB, double OUB, double OPS, double ULB, double UUB, double UPS, double minisupp): boolean
+Supporteds(List<String> requiredItems, List<List<String>> name1): double
+weightedSupporteds(List<String> requiredItems, List<List<String>> name1, List<List<Double>> weight1): double
+ExpSups(List<String> requiredItems, List<List<String>> name1, List<List<Double>> prob1): double
+weightedExpSups(List<String> requiredItems, List<List<String>> name1, List<List<Double>> prob1, List<List<Double>> weight1): double
+ProbabilityFrequents(List<String> requiredItems, double minValue, List<List<String>> name1, List<List<Double>> prob1): double
+weightedProbabilityFrequents(List<String> requiredItems, double minValue, List<List<String>> name1, List<List<Double>> prob1, List<List<Double>> weight1): double
+Probability(double sup, double esup, double miniprob): double
-findMin(double a, double b, double c, double d): double
+LBs(double expectedSupport, double miniprob): double
+UBs(double expectedSupport, double miniprob, double support): double
-Max(double a, double b): double
-Min(double a, double b): double
PFIT
-pool: ForkJoinPool
+Buildtree(PFITNode nXs, int US, double minisup, double miniprob): void
-processNode(PFITNode nX, double miniprob, double minisup, List<PFITNode> xs): void
-updateNodeMetrics(PFITNode node, double miniprob): void
PFMIoSplus
+ADDTRANS(PFITNode nX,int US ,UncertainDatabase database, double minisup, double miniprob): void
+DelTran(PFITNode nX, int US, UncertainDatabase database, double minisup, double miniprob): void
PFMIoS
+ADDTRANS(PFITNode nX,int US ,UncertainDatabase database, double minisup, double miniprob): void
+DelTran(PFITNode nX, int US, UncertainDatabase database, double minisup, double miniprob): void

Figure 2.1 UML Class Diagram

2.3 PFIT with weight:

Function: Buildtree

Require: nXs: PFITNode, US: int, minisup: double, miniprob: double

```
Initialize a new RecursiveAction
    Define compute() method
    Initialize an empty list xs of PFITNode
    For each child nX of nXs in parallel do
        Set nX's support, expected support, LB, UB
        If nX is not frequent then
            Continue to the next child
        Set the probability of nX
        For each right sibling node of nX in parallel do
            If the sibling node is frequent then
                Generate a child node
                Set child's support, expected support, LB, UB
                If the child node's LB <= minisup and UB >= minisup then
                    Set the probability of the child
                Synchronize access to xs
                    Add the child node to xs
    Add all nodes in xs to the children of nXs
```

Table 2.3 Pseudcode Buildtree function in PFIT with weight

Explanation and Example: List itemset includes ‘A’, ‘B’, ‘C’, ‘D’,’E’ in **Table 2.1** can see that, with minimum support (lamda) is 2, minimum probability (t) is 0.6. itemset ‘A’ has support 2 and it is greater than or equal minimum support so it define frequent. If itemset is not frequent, it will continue to check the right parents. The righr parent is frequent so continuing generate children with the itemset not frequent. The condition lowerbound and upper bound keep computing probability support less consume running time cost. In this probability we multiply probability support with weight of itemset (dựa trên bài báo gì đó). Finally, we add all childnode to root.

Output: This output is the calculation result between ***Table 2.1 Example of a Uncertain Database***

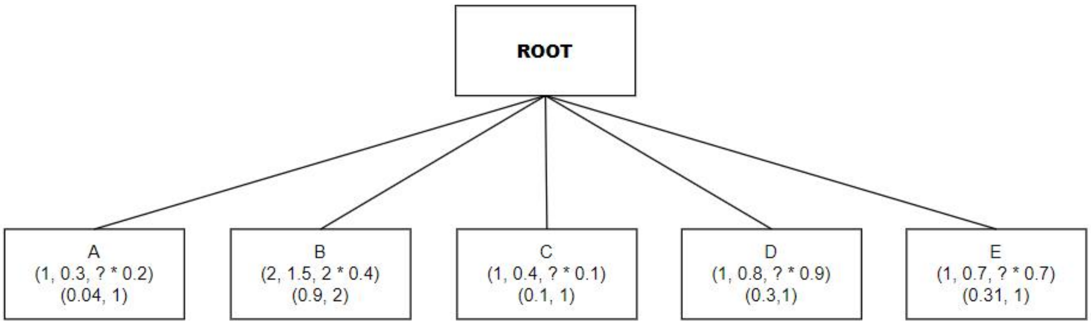


Figure 2.2 Diagram describe the example's result of Buildtree Function

2.4 PFMIoS with weight

2.4.1 ADDTRANS function

```
Function: ADDTRANS
Require: nX: PFITNode, US: int, database: UncertainDatabase, minisup: double, miniprob: double
Retrieve the last value, probability, and weight from the database
Initialize three lists: childrenCopy, newfre, and frequent
If nX has no children then
    Return
For each child nY of nX do
```

```
    Store nY's original lower bound, upper bound, and probability
    If nY is a subset of the transaction (value) then
        Update nY's support, expected support, lower bound, upper
        bound, and reset probability
        If nY's updated bounds indicate it's potentially frequent then
            Update nY's probability
    If nY transitions to a frequent state then
        Add nY to newfre list
        Generate new child nodes for all right siblings of nY
        Update support and expected support for each child
        Add each child to childrenCopy and as a child of nY
    If nY remains frequent and is a subset of the transaction then
        Add nY to frequent list
    For each node in frequent then
        For each right sibling that is newly frequent then
            Generate and update a new child node
            Add this new child to childrenCopy and as a child of the
            current node
    Add all nodes in childrenCopy to the children of nX
```

Table 2.4 Pseudocode ADDTRANS function in PFMIoS with weight

Explanation and Example: When owning a tree include a list of itemset frequent or not frequent, we continue to addtran to check conditions in uncertain data stream. When new uncertain transaction add, store a original upper bound, lower bound and probability support (with weight). Iteration between each childnode and check if itemsets are on list or not. The condition become true so moving to update again support, expected support, lower bound, upper bound and probability support. When done, check itemset are new frequent or not (i - f). if new frequent, Build again Tree, If itemsets keep frequent (f - f) after updating number, we loop all right parents and check if parents are new frequent, integrating itemset with parents. At the end, add all childNode to root children.

Output: This output is the calculation result between ***Table 2.1 Example of a Uncertain Database*** and the output of the **Buildtree function** in ***Figure 2.2***

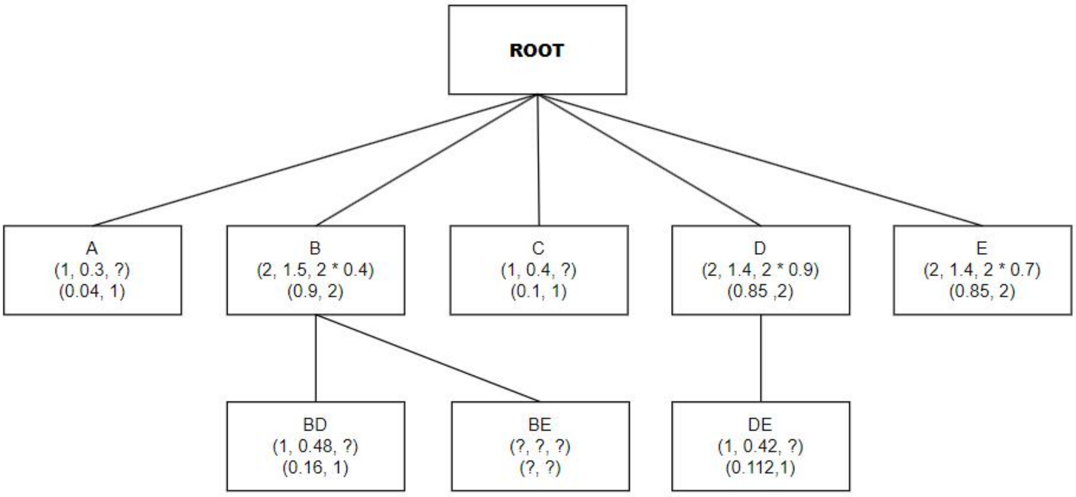

ROOT
A (1, 0.3, ?) (0.04, 1)
B (2, 1.5, 2 * 0.4) (0.9, 2)
C (1, 0.4, ?) (0.1, 1)
D (2, 1.4, 2 * 0.9) (0.85 ,2)
E (2, 1.4, 2 * 0.7) (0.85, 2)
BD (1, 0.48, ?) (0.16, 1)
BE (?, ?, ?) (?, ?)
DE (1, 0.42, ?) (0.112,1)

Figure 2.3 Diagram describe the example's result of ADDTRANS Function

2.4.2 DELTRANS function

```
Function: DELTRANS
Require: nX: PFITNode, US: int, database: UncertainDatabase, minisup: double,
miniprob: double
Retrieve the first set of names, probabilities, and weights from the database
Initialize an infre list to track infrequent nodes
If nX has no children then
        Return
Copy nX's children for iteration
Remove the first set of names, probabilities, and weights from the database
```

```
For each node nY in the copied list of children do
    Store nY's original lower bound, upper bound, and probability
    If nY's items are a subset of the transaction (list) then
        Update nY's support, expected support, lower bound, and upper bound without database access
        Update nY's probability based on the new bounds
        Set nY's probability to 0 if it is not in the frequent bounds anymore
    For each child nZ of nY do
        If nZ becomes infrequent after the transaction removal then
            Remove nZ from nX's children
            Add nZ to the infre list
        If nZ remains frequent but needs updates due to the transaction removal then
            Remove all children of nZ from nX's children if they are related to the transaction
```

Table 2.5 Pseudocode DELTRANS function in PFMIoS with weight

Explanation and Example: When updating complete we need delete the old uncertain transaction, especially, first row. We take first uncertain transaction and check item sets include in first transaction. If it occur, we remove all unnecessary value. When done, check after removing value we move to identify itemset become infrequent (f - i). if it infrequent, remove all childnode of this node. And if itemset keep frequent, we will remove childnode of this.

Output: This output is the calculation result between ***Table 2.1 Example of a Uncertain Database,*** the output of the **Buildtree function** in ***Figure 2.2*** and the output of the **ADDTRANS function** in ***Figure 2.3***

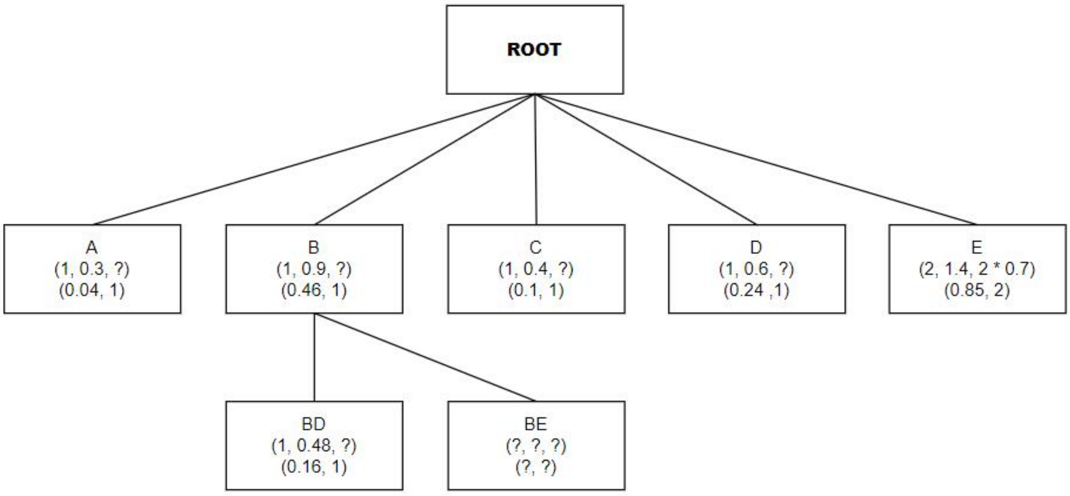


Figure 2.4 Diagram describe the example's result of DELTRANS Function

2.5 PFIMoS+ with weight

Explanation and Example: Similar to PFMIoS but we compute again Probability support which base on ***Herutical rules*** in document ***Li et al. (2018)***. The new thing we change in this is about add itemset's weight in each probability support to make time consuming in the condition ADDTRANS and DELTRANS more efficient than the older. However, it reduce quite small compared to the older without weight. Here is the formula, we use for this function:

$$\xi = min\left(\begin{matrix}\frac{2\sqrt{-2\varepsilon ln(1-\tau)}-ln\tau+\sqrt{ln^2\tau-8\varepsilon ln\tau}}{2\eta}, \\ \frac{2\varepsilon-ln\tau+\sqrt{ln^2\tau-8\varepsilon ln\tau}}{2\eta}, \\ \frac{\Lambda(\mathrm{X})-\varepsilon+\sqrt{-2\varepsilon ln(1-\tau)}}{\eta}, \\ \frac{\Lambda(\mathrm{X})}{\eta}\end{matrix}\right) \tag{2.17}$$

CHAPTER 3. EXPERIMENTS

3.1 Running environment and datasets

- **Programming language**: Java
- **Compile environment**: java 21.0.2 2024-01-16 LTS, Java(TM) SE Runtime Environment (build 21.0.2+13-LTS-58), Java HotSpot(TM) 64-Bit Server VM (build 21.0.2+13-LTS-58, mixed mode, sharing)
- **IDE used for compilation**: Visual Studio Code
- **Device**: Laptop LENOVO AMD Ryzen 5 4600H with Radeon Graphics 3.00 GHz and 8.00 GB of main memory
- **Datasets**:

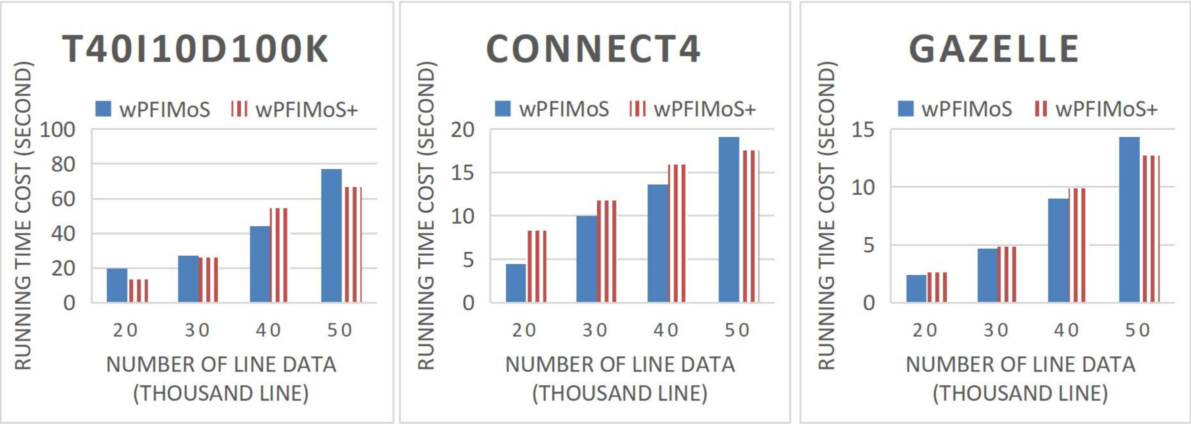
Datasets	Transactions count	Mean	Variance
T40I10D100K	100 000	0.79	0.61
CONNECT4	67 557	0.78	0.65
GAZELLE	59 602	0.94	0.08

Table 3.1 Details about the datasets used for experiment

- **Processing data**: To address the lack of item probabilities and weights in the dataset, we used a **Gaussian distribution** to aggregate the probabilities, applying a **sigmoid function** to prevent overflow, following ***Li et al. (2018)*** and assign random weights between 0 and 1 according to ***Li et al. (2020)***.

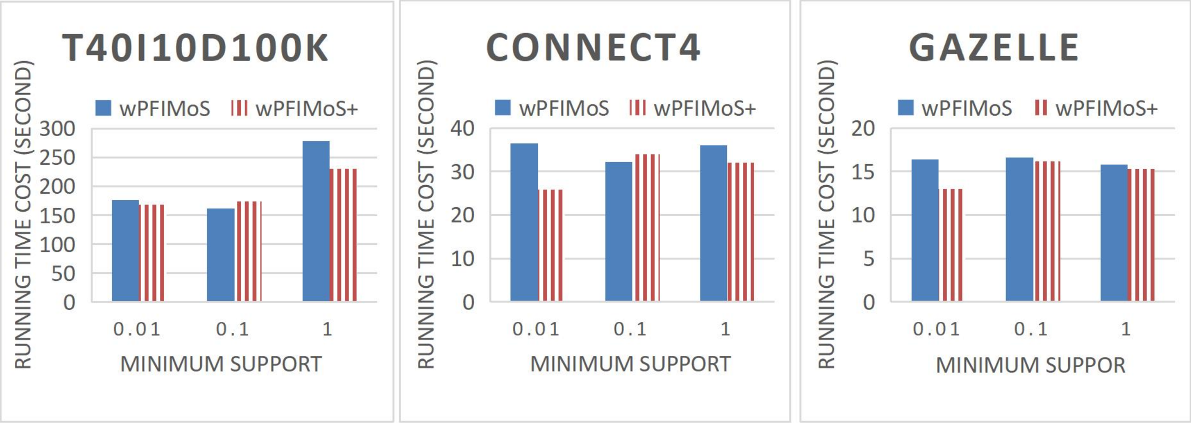
3.2 Experiment Results and Discussion

3.2.1 Experiment Results



a. $\lambda_r = 0.9, \eta = 10\text{K}$ b. $\lambda_r = 0.9, \eta = 10\text{K}$ c. $\lambda_r = 0.9, \eta = 10\text{K}$

Figure 3.1 Effect of Number of Line Data(runtime cost)



d. $\tau = 0.9, \eta = 10\text{K}$ e. $\tau = 0.9, \eta = 10\text{K}$ f. $\tau = 0.9, \eta = 10\text{K}$

Figure 3.2 Effect of Number of Minimum Support(runtime cost)

3.2.2 Discussion

- **According to *Figure 3.1:***
  - There is a clear upward trend in running time cost as the number of line data increases for both algorithms, which is a typical behavior as more data generally requires more processing time.
  - wPFIMoS+ shows an advantage over wPFIMoS in handling larger datasets, particularly evident in the T40I10D100K dataset at 50 thousand lines of data.
  - The incremental running time cost with increasing data volume appears to be more linear for wPFIMoS+, while wPFIMoS exhibits a steeper increase,

especially in the T40I10D100K dataset. This indicates better scalability of wPFIMoS+ under heavier data loads.

- **According to *Figure 3.2***:
  - Across all datasets, wPFIMoS+ consistently performs better (i.e., lower running time) than wPFIMoS. This suggests that the enhancements made in the wPFIMoS+ algorithm effectively reduce computation time compared to its predecessor.
  - The running time cost tends to decrease for both algorithms as the minimum support increases from 0.01 to 1. This is an expected behavior because a higher minimum support threshold usually results in fewer itemsets being considered frequent, thus reducing the computational workload.
  - The T40I10D100K dataset appears to be more computationally demanding than the CONNECT4 and GAZELLE datasets, as indicated by the overall higher running times for both algorithms on this dataset.
  - The relative difference in performance between wPFIMoS and wPFIMoS+ is most pronounced at the lower minimum support level (0.01), particularly in the T40I10D100K dataset. This suggests that the improvements in wPFIMoS+ are especially beneficial when dealing with denser datasets and more stringent (lower) minimum support thresholds.
  - While all datasets show improved running times at higher minimum support levels, the T40I10D100K dataset demonstrates a less pronounced improvement compared to the other datasets. This could indicate scalability issues with larger or more complex datasets and might be an area for further optimization.

Overall, these diagrams suggest that the wPFIMoS+ algorithm is more efficient than wPFIMoS, particularly when dealing with larger datasets and when the minimum support threshold is high. Additionally, the diagrams also indicate that algorithm performance can vary significantly depending on the characteristics of the dataset being processed.

CHAPTER 4. CONCLUSION

Our research into Probabilistic Weighted Frequent Itemset Mining over Uncertain Data Streams has presented a approach that integrates the concepts of weight and probability in the analysis of uncertain data streams. This integration is pivotal in addressing the dynamic and inherently uncertain nature of streaming data, which poses significant challenges in the field of data mining.

The introduction of the PFIT, PFIMoS, and PFIMoS+ algorithms represents a significant leap forward in our ability to efficiently and effectively mine weighted frequent itemsets from such streams. The 'weighted' aspect of these algorithms is particularly noteworthy, as it allows for a more nuanced understanding of the data by considering not only the occurrence frequency of itemsets but also their respective weights. This approach acknowledges that not all itemsets contribute equally to the insights derived from the data, hence offering a more sophisticated analysis tool that aligns more closely with real-world complexities.

Through our experimental evaluations, we have demonstrated that the incorporation of weights into the probabilistic mining process significantly enhances the performance and accuracy of the mining task. The weighted approach enables the algorithms to prioritize the analysis of itemsets based on their relevance or importance, which is crucial for applications where the significance of the data varies.

Moreover, our experiments using synthetic and real-life datasets have unequivocally shown the superior efficacy of our proposed methods in terms of runtime and memory usage. This not only validates our approach but also emphasizes the practical utility of weighted itemset mining in uncertain data streams, offering potential applications in a wide array of domains such as market basket analysis, real-time monitoring systems, and beyond.

In conclusion, the weighted dimension introduced in our probabilistic frequent itemset mining methodology has opened up new avenues for more refined and relevant data analysis in the face of uncertainty. The success of this approach underscores the importance of continuing to explore and optimize weighted mining techniques, with future work aimed at further enhancing the efficiency and applicability of these algorithms in other complex data environments. Our research thus provides a substantial contribution to the ongoing development of data mining technologies, offering a robust, efficient, and adaptable solution for uncovering the intricate patterns hidden within uncertain data streams.

REFERENCES

1. Aggarwal, C. C., Li, Y., Wang, J., & Wang, J. (2009). Frequent pattern mining with uncertain data. In Proceedings of the 15th acm sigkdd international conference on knowledge discovery and data mining (pp. 29–38). ACM
2. Akbarinia, R., & Masseglia, F. (2013). Fast and exact mining of probabilistic data streams. In Machine learning and knowledge discovery in databases (pp. 493–508).Springer.
3. Bernecker, T., Kriegel, H.-P., Renz, M., Verhein, F., & Zuefle, A. (2009). Probabilistic frequent itemset mining in uncertain databases. In Proceedings of the 15th acm sigkdd international conference on knowledge discovery and data mining (pp. 119–128). ACM.
4. Bernecker, T., Kriegel, H.-P., Renz, M., Verhein, F., & Züfle, A. (2012). Probabilistic frequent pattern growth for itemset mining in uncertain databases. In Scientific and statistical database management (pp. 38–55). Springer
5. Chui, C.-K., Kao, B., & Hung, E. (2007). Mining frequent itemsets from uncertain data. In Advances in knowledge discovery and data mining (pp. 47–58). Springer.
6. Cuzzocrea, A., & Leung, C. K. (2016). Computing theoretically-sound upper bounds to expected support for frequent pattern mining problems over uncertain big data. In International conference on information processing and management of uncertainty in knowledge-based systems (pp. 379–392). Springer.
7. Leung, C. K.-S., & MacKinnon, R. K. (2014). Blimp: A compact tree structure for uncertain frequent pattern mining. In Data warehousing and knowledge discovery (pp. 115–123). Springer.
8. Li, H., Zhang, N., Zhu, J., Wang, Y., & Cao, H. (2018). Probabilistic Frequent Itemset Mining over Uncertain Data Streams. Proceedings of the 2018

International Conference on Database Systems for Advanced Applications, 11382, 646–661.

9. Li, Z., Chen, F., Wu, J., Liu, Z., & Liu, W. (2020). Efficient weighted probabilistic frequent itemset mining in uncertain databases. Expert Systems, e12551. https://doi.org/10.1111/exsy.12551
10. Sun, L., Cheng, R., Cheung, D. W., & Cheng, J. (2010). Mining uncertain data withprobabilistic guarantees. In Proceedings of the 16th acm sigkdd international conference on knowledge discovery and data mining (pp. 273–282). ACM.
11. Mike Novey, Tülay Adali, Anindya Roy.(2010). A Complex Generalized Gaussian Distribution— Characterization, Generation, and Estimation. In IEEE Transactions on Signal Processing (Volume: 58, Issue: 3). 10.1109/TSP.2009.2036049
12. Jun Han, Claudio Moraga. (2005). The influence of the sigmoid function parameters on the speed of backpropagation learning. In Computational Models of Neurons and Neural Nets. https://doi.org/10.1007/3-540-59497-3_175
13. Seok-Ho Chang, Pamela C. Cosman, Laurence B. Milstein. (2011) . Chernoff-Type Bounds for the Gaussian Error Function. In IEEE Transactions on Communications. 10.1109/TCOMM.2011.072011.100049
14. Lin, J.CW., Gan, W., Fournier-Viger, P. et al. Weighted frequent itemset mining over uncertain databases. Appl Intell 44, 232–250 (2016). https://doi.org/10.1007/s10489-015-0703-9
15. Toon Calders, Calin Garboni, Bart Goethals. (2010). Approximation of Frequentness Probability of Itemsets in Uncertain Data. In 2010 IEEE International Conference on Data Mining. 10.1109/ICDM.2010.42.
16. Chengjie Luo, Clement Yu, and Jorge Lobo, Gaoming Wang, Tracy Pham, Clement Yu. (1996). Computation of Best Bounds of Probabilities from Uncertain Data. https://doi.org/10.1111/j.1467-8640.1996.tb00276.x

17. Razieh Davashi. (2021). UP-tree & UP-Mine: A fast method based on upper bound for frequent pattern mining from uncertain data. https://doi.org/10.1016/j.engappai.2021.104477
18. Carson Kai-Sang Leung, Boyu Hao. (2009). Mining of Frequent Itemsets from Streams of Uncertain Data. In 2009 IEEE 25th International Conference on Data Engineering. 10.1109/ICDE.2009.157
19. Islam, M.S., Kar, P.C., Samiullah, M. et al. Discovering probabilistically weighted sequential patterns in uncertain databases. Appl Intell 53, 6525–6553 (2023). https://doi.org/10.1007/s10489-022-03699-7
20. Gan, W., Lin, J.CW., Fournier-Viger, P., Chao, HC. (2016). Mining Recent High Expected Weighted Itemsets from Uncertain Databases. In: Li, F., Shim, K., Zheng, K., Liu, G. (eds) Web Technologies and Applications. APWeb 2016. Lecture Notes in Computer Science(), vol 9931. Springer, Cham. https://doi.org/10.1007/978-3-319-45814-4_47