



Jerry Chun-Wei Lin

Tree-based Algorithms for Incremental, Utility, and Fuzzy Data Mining

 **LAMBERT**
Academic Publishing

Jerry Chun-Wei Lin

**Tree-based Algorithms for
Incremental, Utility, and Fuzzy Data
Mining**



LAP LAMBERT Academic Publishing

Impressum / Imprint

Bibliografische Information der Deutschen Nationalbibliothek: Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Alle in diesem Buch genannten Marken und Produktnamen unterliegen warenzeichen-, marken- oder patentrechtlichem Schutz bzw. sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Inhaber. Die Wiedergabe von Marken, Produktnamen, Gebrauchsnamen, Handelsnamen, Warenbezeichnungen u.s.w. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutzgesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Bibliographic information published by the Deutsche Nationalbibliothek: The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <http://dnb.d-nb.de>.

Any brand names and product names mentioned in this book are subject to trademark, brand or patent protection and are trademarks or registered trademarks of their respective holders. The use of brand names, product names, common names, trade names, product descriptions etc. even without a particular marking in this work is in no way to be construed to mean that such names may be regarded as unrestricted in respect of trademark and brand protection legislation and could thus be used by anyone.

Coverbild / Cover image: www.ingimage.com

Verlag / Publisher:

LAP LAMBERT Academic Publishing

ist ein Imprint der / is a trademark of

OmniScriptum GmbH & Co. KG

Heinrich-Böcking-Str. 6-8, 66121 Saarbrücken, Deutschland / Germany

Email: info@lap-publishing.com

Herstellung: siehe letzte Seite /

Printed at: see last page

ISBN: 978-3-659-67722-9

Zugl. / Approved by: Tainan, Taiwan, National Cheng Kung University, Diss., 2010

Copyright © 2015 OmniScriptum GmbH & Co. KG

Alle Rechte vorbehalten. / All rights reserved. Saarbrücken 2015

Jerry Chun-Wei Lin

**Tree-based Algorithms for Incremental, Utility, and Fuzzy Data
Mining**

ABSTRACT

Data mining, also referred to as knowledge discovery, has recently emerged as an important research topic, and association rules mining is considered as one of the most referenced sub-topics in data mining. In the past, traditional algorithms process all records in a batch way for mining association rules. In real-world applications, records are constantly being inserted, deleted or modified in dynamic databases. Designing an algorithm that can efficiently maintain association rules in dynamic databases is critically important. In the first part of this dissertation, three Pre-FUFP maintenance algorithms are thus proposed to efficiently maintain and update the FUFP-tree structures regardless of whether records are inserted, deleted or modified in dynamic databases. Based on two support thresholds of pre-large concepts, it helps avoid the need to re-build the tree structure until after a number of records have been processed. The FP-growth-like algorithm is then implemented to mine the desired information for the updated FUFP trees.

In the association rules mining, it treats items as binary variables in databases, which considers whether an item is bought in a record or not. Utility mining was thus proposed to reflect any other implicit factors, such as prices or profits. In the second part of this dissertation, a novel HUP-tree algorithm is proposed to efficiently mine the high utility itemsets based on the downward closure property. A HUP tree is first designed to keep the related information for later mining process. A HUP-growth mining algorithm is then presented to efficiently mine high utility itemsets from it.

In the past, most association rules mining focused on processing binary variables in databases. In recent years, many fuzzy data mining algorithms have been proposed for managing quantitative data, and most of them are processed in the level-wise approaches. In the third part of this dissertation, we attempt to extend the FP-tree algorithm for handling quantitative data from the global values of fuzzy regions. Thus, the fuzzy FP-tree algorithm,

the compressed fuzzy frequent pattern tree (CFFP-tree) algorithm, and the upper-bound fuzzy frequent pattern tree (UBFFP-tree) algorithm are then proposed to efficiently mine the fuzzy frequent itemsets. The maximum cardinality is used to make the number of fuzzy regions processed equivalent to the number of the original items for reducing the processing time. Three mining algorithm are then proposed to mine the fuzzy frequent itemsets based on the designed tree structures, respectively.

Experimental results showed that the performance of the proposed algorithms in three parts of this dissertation for handling association rules mining, high utility mining and fuzzy data mining, respectively.

Keywords: data mining, fuzzy data mining, utility mining, tree structure, maintenance algorithm.

List of Figures

| | |
|---|----|
| FIGURE 2.1: FOUR CASES WHEN NEW RECORDS ARE INSERTED INTO EXISTING DATABASES..... | 9 |
| FIGURE 2.2: THE FP TREE AFTER THE FIRST RECORD IS PROCESSED | 12 |
| FIGURE 2.3: THE FP TREE AFTER THE SECOND RECORD IS PROCESSED..... | 13 |
| FIGURE 2.4: THE RESULTING HEADER_TABLE AND FP TREE IN THE EXAMPLE | 13 |
| FIGURE 2.5: THE CONDITIONAL FP TREE FOR $\{P\}$ | 15 |
| FIGURE 2.6: THE CONDITIONAL FP TREE FOR $\{M\}$ | 15 |
| FIGURE 2.7: THE CONDITIONAL FP-TREE FOR $\{AM\}$ | 16 |
| FIGURE 2.8: NINE CASES WHEN NEW RECORDS ARE INSERTED INTO EXISTING DATABASES | 17 |
| FIGURE 2.9: THE CONCEPT OF FUZZY DATA MINING | 21 |
| FIGURE 3.1: THE PROPOSED FRAMEWORK OF THE MAINTENANCE ALGORITHMS | 26 |
| FIGURE 3.2: THE HEADER_TABLE AND THE FUIFP TREE CONSTRUCTED | 29 |
| FIGURE 3.3: THE HEADER_TABLE AND THE FUIFP TREE AFTER STEP 4..... | 35 |
| FIGURE 3.4: THE HEADER_TABLE AND THE FUIFP TREE AFTER STEP 9..... | 37 |
| FIGURE 3.5: THE FINAL FUIFP TREE AFTER ALL THE NEW RECORDS ARE PROCESSED..... | 38 |
| FIGURE 3.6: THE COMPARISON OF THE EXECUTION TIMES FOR DIFFERENT THRESHOLD VALUES | 39 |
| FIGURE 3.7: THE COMPARISON OF THE NUMBERS OF NODES FOR DIFFERENT THRESHOLD VALUES | 40 |
| FIGURE 3.8: THE COMPARISON OF THE EXECUTION TIME FOR SEQUENTIALLY INSERTED NEW RECORDS..... | 41 |
| FIGURE 3.9: THE COMPARISON OF THE NUMBERS OF NODES FOR SEQUENTIALLY INSERTED NEW RECORDS..... | 42 |
| FIGURE 3.10: THE HEADER_TABLE AND THE FUIFP TREE CONSTRUCTED | 45 |
| FIGURE 3.11: THE HEADER_TABLE AND THE FUIFP TREE AFTER STEP 4..... | 51 |
| FIGURE 3.12: THE HEADER_TABLE AND THE FUIFP TREE AFTER STEP 10..... | 53 |
| FIGURE 3.13: THE COMPARISON OF THE EXECUTION TIMES FOR DIFFERENT THRESHOLD VALUES | 55 |
| FIGURE 3.14: THE COMPARISON OF THE NUMBERS OF TREE NODES FOR DIFFERENT THRESHOLD VALUES | 56 |
| FIGURE 3.15: THE COMPARISON OF THE EXECUTION TIMES FOR SEQUENTIALLY DELETED RECORDS..... | 56 |
| FIGURE 3.16: THE COMPARISON OF THE NUMBERS OF TREE NODES FOR SEQUENTIALLY DELETED RECORDS..... | 57 |
| FIGURE 3.17: THE HEADER_TABLE AND THE FUIFP TREE CONSTRUCTED | 60 |
| FIGURE 3.18: THE UPDATED FUIFP TREE AFTER STEP 5 | 67 |
| FIGURE 3.19: THE UPDATED FUIFP TREE AFTER STEP 9 | 68 |
| FIGURE 3.20: THE FINAL UPDATED FUIFP TREE AFTER ALL MODIFIED RECORDS ARE PROCESSED..... | 70 |
| FIGURE 3.21: THE COMPARISON OF THE THREE MODIFICATION ALGORITHMS IN EXECUTION TIME..... | 71 |
| FIGURE 3.22: THE COMPARISON OF THE THREE MODIFICATION ALGORITHMS FOR NUMBERS OF TREE NODES..... | 72 |
| FIGURE 3.23: THE COMPARISON OF THE EXECUTION TIME FOR MODIFIED RECORDS AT A 6% THRESHOLD..... | 73 |
| FIGURE 3.24: THE COMPARISON OF THE NUMBERS OF NODES FOR MODIFIED RECORDS AT A 6% THRESHOLD..... | 73 |
| FIGURE 4.1: THE HUP TREE AFTER THE FIRST UPDATED RECORD IS PROCESSED | 84 |

| | |
|---|-----|
| FIGURE 4.2: THE HUP TREE AFTER THE SECOND UPDATED RECORD IS PROCESSED | 85 |
| FIGURE 4.3: THE FINAL CONSTRUCTED HUP TREE..... | 85 |
| FIGURE 4.4: THE COMPARISON OF THE EXECUTION TIME | 89 |
| FIGURE 4.5: THE NUMBERS OF TREE NODES GENERATED BY THE THREE DIFFERENT ORDERING METHODS | 90 |
| FIGURE 5.1: THE PROPOSED FUZZY DATA MINING FRAMEWORK | 94 |
| FIGURE 5.2: THE MEMBERSHIP FUNCTIONS USED IN THIS EXAMPLE | 99 |
| FIGURE 5.3: THE HEADER_TABLE FORMED AFTER STEP 5..... | 101 |
| FIGURE 5.4: THE FUZZY FP TREE AFTER THE FIRST RECORD IS PROCESSED | 103 |
| FIGURE 5.5: THE FUZZY FP-TREE AFTER THE SECOND RECORD IS PROCESSED | 104 |
| FIGURE 5.6: THE FINAL FUZZY FP TREE CONSTRUCTED IN THE EXAMPLE | 104 |
| FIGURE 5.7: THE CONDITIONAL FUZZY PATTERN TREE FOR THE FUZZY REGION $\{D.Low\}$ | 105 |
| FIGURE 5.8: THE NUMBERS OF NODES IN THE FUZZY FP TREE ALONG WITH DIFFERENT MINIMUM SUPPORT VALUES IN THE FIRST PART | 107 |
| FIGURE 5.9: THE NUMBERS OF NODES IN THE FUZZY FP TREE ALONG WITH DIFFERENT MINIMUM SUPPORT VALUES IN THE SECOND PART | 108 |
| FIGURE 5.10: THE EXECUTION TIME ALONG WITH DIFFERENT MINIMUM SUPPORT THRESHOLDS IN THE FIRST PART | 109 |
| FIGURE 5.11: THE EXECUTION TIME ALONG WITH DIFFERENT MINIMUM SUPPORT THRESHOLDS IN THE SECOND PART | 109 |
| FIGURE 5.12: THE MEMBERSHIP FUNCTIONS USED IN THIS EXAMPLE | 114 |
| FIGURE 5.13: THE HEADER_TABLE FORMED AFTER STEP 6..... | 117 |
| FIGURE 5.14: THE CFFP TREE AFTER THE FIRST RECORD IS PROCESSED | 119 |
| FIGURE 5.15: THE FINALLY CONSTRUCTED CFFP TREE | 120 |
| FIGURE 5.16: THE COMPARISONS OF THE EXECUTION TIME AT FUZZY 2-REGIONS | 123 |
| FIGURE 5.17: THE COMPARISONS OF THE NUMBERS OF TREE NODES AT FUZZY 2-REGIONS..... | 123 |
| FIGURE 5.18: THE COMPARISONS OF THE EXECUTION TIME AT FUZZY 3-REGIONS | 124 |
| FIGURE 5.19: THE COMPARISONS OF THE NUMBERS OF TREE NODES AT FUZZY 3-REGIONS..... | 124 |
| FIGURE 5.20: THE MEMBERSHIP FUNCTIONS USED IN THIS EXAMPLE | 129 |
| FIGURE 5.21: THE HEADER_TABLE FORMED..... | 132 |
| FIGURE 5.22: THE UBFFP TREE AFTER THE FIRST RECORD IS PROCESSED | 133 |
| FIGURE 5.23: THE UBFFP TREE AFTER THE SECOND RECORD IS PROCESSED | 134 |
| FIGURE 5.24: THE FINALLY CONSTRUCTED UBFFP TREE..... | 135 |
| FIGURE 5.25: THE TWO PATHS WITH THE FUZZY REGION $\{D.Low\}$ | 139 |
| FIGURE 5.26: THE EXECUTION TIME OF THE THREE APPROACHES FOR TWO FUZZY REGIONS..... | 142 |
| FIGURE 5.27: THE NUMBERS OF TREE NODES GENERATED FROM THE THREE APPROACHES FOR TWO FUZZY REGIONS | 142 |
| FIGURE 5.28: THE EXECUTION TIME OF THE THREE APPROACHES FOR THREE FUZZY REGIONS..... | 143 |
| FIGURE 5.29: THE NUMBERS OF TREE NODES GENERATED FROM THE THREE APPROACHES FOR THREE FUZZY | |

| | |
|--------------|-----|
| REGIONS..... | 144 |
|--------------|-----|

List of Tables

| | |
|--|----|
| TABLE 2.1: FOUR CASES AND THEIR FUP RESULTS..... | 9 |
| TABLE 2.2: THE DATABASES WITH FIVE RECORDS | 11 |
| TABLE 2.3: ALL THE ITEMS WITH THEIR COUNTS..... | 11 |
| TABLE 2.4: THE RECORDS WITH ONLY SORTED LARGE ITEMS | 11 |
| TABLE 2.5: NINE CASES AND THEIR PRE-LARGE RESULTS..... | 18 |
| TABLE 3.1: A SUMMARY OF THE RELATED ASSOCIATION RULES MINING..... | 25 |
| TABLE 3.2: THE ORIGINAL DATABASES IN THE EXAMPLE..... | 28 |
| TABLE 3.3: THE PRE-LARGE ITEMS IN THE ORIGINAL DATABASES..... | 29 |
| TABLE 3.4: THE THREE NEW RECORDS | 33 |
| TABLE 3.5: THE COUNTS OF ALL ITEMS IN THREE NEW RECORDS | 33 |
| TABLE 3.6: THREE PARTS OF ITEMS FROM THE NEW RECORDS..... | 34 |
| TABLE 3.7: THE CORRESPONDING BRANCHES FOR THE ORIGINAL RECORDS WITH ITEM $\{D\}$ | 36 |
| TABLE 3.8: THE CORRESPONDING BRANCHES FOR THE NEW RECORDS WITH ITEMS $\{B, A, F, D\}$ | 37 |
| TABLE 3.9: THE ORIGINAL DATABASES IN THE EXAMPLE..... | 44 |
| TABLE 3.10: THE PRE-LARGE ITEMS IN THE ORIGINAL DATABASES..... | 45 |
| TABLE 3.11: THE COUNTS OF ALL ITEMS IN THE DELETED RECORDS..... | 49 |
| TABLE 3.12: THREE PARTS OF ITEMS FROM THE DELETED RECORDS..... | 50 |
| TABLE 3.13: THE CORRESPONDING BRANCHES FOR THE DELETED RECORDS WITH ITEMS $\{C, D\}$ | 51 |
| TABLE 3.14: THE CORRESPONDING BRANCHES FOR THE UPDATED DATABASES WITH ITEM $\{H\}$ | 52 |
| TABLE 3.15: THE ORIGINAL DATABASES IN THE EXAMPLE..... | 59 |
| TABLE 3.16: THE PRE-LARGE ITEMS IN THE ORIGINAL DATABASES..... | 60 |
| TABLE 3.17: THE THREE RECORDS AFTER MODIFICATION..... | 60 |
| TABLE 3.18: THE COUNT DIFFERENCE OF EACH ITEM IN M | 65 |
| TABLE 3.19: THREE PARTS OF ITEMS FROM THE DELETED RECORDS..... | 66 |
| TABLE 3.20: THE CORRESPONDING BRANCHES FOR THE RECORDS BEFORE MODIFICATION | 68 |
| TABLE 3.21: THE CORRESPONDING BRANCHES FOR THE UNMODIFIED RECORDS IN $BRANCH_ITEMS$ | 69 |
| TABLE 3.22: THE CORRESPONDING BRANCHES WITH ITEMS IN $INCREASED_ITEMS$ FOR THE MODIFIED RECORDS.... | 69 |
| TABLE 4.1: A SUMMARY OF THE RELATED HIGH UTILITY MINING | 77 |
| TABLE 4.2: THE QUANTTTATIVE DATABASES IN THE EXAMPLE | 80 |
| TABLE 4.3: THE UTILITY TABLE | 81 |
| TABLE 4.4: THE UTILITIES AND THE OCCURRENCE FREQUENCIES OF ALL ITEMS IN EACH RECORD..... | 81 |
| TABLE 4.5: THE TRANSACTION-WEIGHTED UTILITY(twu) OF EACH ITEM..... | 82 |
| TABLE 4.6: THE CONSTRUCTED HEADER_TABLE IN THE EXAMPLE | 83 |
| TABLE 4.7: THE UPDATED RECORDS IN THE QUANTITATIVE DATABASES..... | 83 |
| TABLE 4.8: THE ITEMSETS CONTAINING ITEM $\{B\}$ WITH THEIR SUMMED QUANTITIES..... | 87 |

TABLE 4.9: THE ACTUAL UTILITY VALUES OF THE ITEMSETS ASSOCIATED TO ITEM $\{B\}$ 87

TABLE 4.10: THE FINAL HIGH UTILITY ITEMSETS 88

TABLE 5.1: A SUMMARY OF THE RELATED FUZZY DATA MINING ALGORITHMS 93

TABLE 5.2: SIX QUANTITATIVE RECORDS IN THE EXAMPLE 98

TABLE 5.3: THE FUZZY SETS TRANSFORMED FROM THE DATA IN TABLE 5.2 99

TABLE 5.4: THE COUNTS OF FUZZY REGIONS 100

TABLE 5.5: THE SET OF FUZZY FREQUENT REGIONS IN THE EXAMPLE 101

TABLE 5.6: THE TRANSFORMED RECORDS WITH FREQUENT FUZZY ITEMS 101

TABLE 5.7: THE RECORDS WITH ONLY THE SORTED FREQUENT FUZZY ITEMS 102

TABLE 5.8: ALL THE FUZZY FREQUENT ITEMSETS OBTAINED IN THE EXAMPLE 106

TABLE 5.9: SIX QUANTITATIVE RECORDS IN THE EXAMPLE 114

TABLE 5.10: THE FUZZY SETS TRANSFORMED FROM THE DATA IN TABLE 5.9 115

TABLE 5.11: THE COUNTS OF FUZZY REGIONS 116

TABLE 5.12: THE COUNTS AND THE OCCURRENCE NUMBERS OF THE FUZZY REGIONS 117

TABLE 5.13: THE UPDATED RECORDS AFTER STEP 7 117

TABLE 5.14: THE FINALLY DERIVED FUZZY FREQUENT ITEMSETS 122

TABLE 5.15: THE QUANTITATIVE DATABASES IN THE EXAMPLE 129

TABLE 5.16: THE FUZZY SETS TRANSFORMED FROM THE DATA IN TABLE 5.15 130

TABLE 5.17: THE COUNTS OF FUZZY REGIONS 131

TABLE 5.18: THE COUNTS AND THE OCCURRENCE NUMBERS OF THE FUZZY REGIONS 132

TABLE 5.19: THE UPDATED RECORDS AFTER STEP 7 132

TABLE 5.20: THE FINAL SET OF CANDIDATE FUZZY ITEMSETS 140

TABLE 5.21: THE FINALLY DERIVED FUZZY FREQUENT ITEMSETS 141

Contents

| | |
|---|----|
| ABSTRACT | I |
| LIST OF FIGURES | VI |
| LIST OF TABLES | IX |
| CHAPTER 1 INTRODUCTION..... | 1 |
| 1.1 MOTIVATION..... | 1 |
| 1.2 OVERVIEW OF THE DISSERTATION..... | 4 |
| 1.2.1 <i>The Maintenance Algorithms in Dynamic Databases</i> | 4 |
| 1.2.2 <i>Mining High Utility Itemsets</i> | 4 |
| 1.2.3 <i>The Algorithms for Fuzzy Data Mining</i> | 5 |
| 1.3 ORGANIZATION OF THE DISSERTATION | 6 |
| CHAPTER 2 REVIEW OF RELATED WORKS..... | 7 |
| 2.1 THE DATA MINING PROCESS FOR ASSOCIATION RULES | 7 |
| 2.2 THE FREQUENT PATTERN TREE..... | 10 |
| 2.2.1 <i>Construction of the FP tree</i> | 10 |
| 2.2.2 <i>Mining of Large Itemsets</i> | 13 |
| 2.3 THE PRE-LARGE CONCEPTS..... | 16 |
| 2.4 THE UTILITY MINING | 18 |
| 2.5 FUZZY DATA MINING..... | 19 |
| CHAPTER 3 THE MAINTENANCE ALGORITHMS FOR MINING ASSOCIATION RULES IN DYNAMIC DATABASES | 23 |
| 3.1 INTRODUCTION..... | 23 |
| 3.2 THE FRAMEWORK OF THE PRE-FUFP MAINTENANCE ALGORITHMS | 26 |
| 3.3 THE MAINTENANCE ALGORITHM FOR RECORD INSERTION | 27 |
| 3.3.1 <i>Notation</i> | 27 |
| 3.3.2 <i>The Proposed Maintenance Algorithm</i> | 28 |
| 3.3.3 <i>An Example</i> | 33 |
| 3.3.4 <i>Experimental Results</i> | 38 |
| 3.3.5 <i>Summary</i> | 42 |
| 3.4 THE MAINTENANCE ALGORITHM FOR RECORD DELETION | 43 |
| 3.4.1 <i>Notation</i> | 43 |
| 3.4.2 <i>The Proposed Maintenance Algorithm</i> | 44 |
| 3.4.3 <i>An Example</i> | 49 |
| 3.4.4 <i>Experimental Results</i> | 53 |

| | | |
|--|---|-----------|
| 3.4.5 | Summary | 57 |
| 3.5 | THE MAINTENANCE ALGORITHM FOR RECORD MODIFICATION | 58 |
| 3.5.1 | Notation | 58 |
| 3.5.2 | The Proposed Modification Algorithm | 59 |
| 3.5.3 | An Example | 64 |
| 3.5.4 | Experimental Results | 70 |
| 3.5.5 | Summary | 74 |
| 3.6 | ANALYSIS AND DISCUSSION | 74 |
| CHAPTER 4 MINING HIGH UTILITY ITEMSETS | | 76 |
| 4.1 | INTRODUCTION | 76 |
| 4.2 | THE PROPOSED HUP-TREE CONSTRUCTION ALGORITHM | 77 |
| 4.3 | THE HUP-GROWTH MINING ALGORITHM | 85 |
| 4.4 | EXPERIMENTAL RESULTS | 88 |
| 4.5 | SUMMARY | 90 |
| 4.6 | ANALYSIS AND DISCUSSION | 91 |
| CHAPTER 5 MINING FUZZY FREQUENT ITEMSETS..... | | 92 |
| 5.1 | INTRODUCTION | 92 |
| 5.2 | THE FRAMEWORK OF FUZZY DATA MINING ALGORITHMS | 94 |
| 5.3 | THE PROPOSED FUZZY FP TREE ALGORITHM | 95 |
| 5.3.1 | Notation | 95 |
| 5.3.2 | The Proposed Fuzzy FP-tree Algorithm | 95 |
| 5.3.3 | An Example | 98 |
| 5.3.4 | Experimental Results | 107 |
| 5.3.5 | Summary | 110 |
| 5.4 | THE PROPOSED CFFP-TREE ALGORITHM | 111 |
| 5.4.1 | Notation | 111 |
| 5.4.2 | The Proposed CFFP- tree Construction Algorithm | 111 |
| 5.4.3 | An Example of CFFP-tree Construction Algorithm | 114 |
| 5.4.4 | The Proposed CFFP-growth Mining Algorithm | 120 |
| 5.4.5 | An Example of the CFFP-growth Mining Algorithm | 121 |
| 5.4.6 | Experimental Results | 122 |
| 5.4.7 | Summary | 125 |
| 5.5 | THE PROPOSED UBFFP-TREE ALGORITHM | 125 |
| 5.5.1 | Notation | 126 |
| 5.5.2 | The Proposed UBFFP- tree Construction Algorithm | 126 |
| 5.5.3 | An Example | 128 |
| 5.5.4 | The Proposed UBFFP-growth Mining Algorithm | 135 |

| | | |
|--|---|-----|
| 5.5.5 | <i>An Example of the UBFP-growth Mining Algorithm</i> | 138 |
| 5.5.6 | <i>Experimental Results</i> | 141 |
| 5.5.7 | <i>Summary</i> | 144 |
| 5.6 | ANALYSIS AND DISCUSSION | 145 |
| CHAPTER 6 CONCLUSION AND FUTURE WORKS..... | | 147 |
| BIBLIOGRAPHY | | 150 |

CHAPTER 1

INTRODUCTION

1.1 Motivation

Years of effort in data mining have been produced a variety of efficient techniques. Depending on the type of databases processed, the mining approaches may be classified as finding association rules [2, 4-7, 10-11, 46-48, 59], classification rules [29, 52], clustering rules [34, 39], sequential patterns [3, 50, 53], among others. Among them, association rules mining is the most commonly seen in data mining. It consists of two main steps to derive the association rules from the transaction databases. The first step is to discover the frequent itemsets from databases based on the minimum support threshold; and the second one is to create the association rules from the found frequent itemsets during the first step based on the minimum confidence threshold. That is, mining frequent itemsets from databases is a fundamental task of finding association rules.

Numerous methods were proposed in the past to discover frequent itemsets, such as level-wise approaches and pattern-growth ones. In the level-wise approaches, most of which were based on the Apriori algorithm [2, 4-5], which generated and tested candidate itemsets level-by-level. In the pattern-growth approaches [1, 16, 19, 35, 56], most of which were based on the Frequent-Pattern-tree (FP-tree) structure [20] for efficiently mining association rules without generation of candidate itemsets. Both of the Apriori and the FP-tree mining approaches, however, are processed in the batch way. Cheung *et al.* then proposed the noticeable Fast UPdate (FUP) algorithm and FUP2 algorithm to maintain the discovered rules for record insertion [12] and record deletion [13], respectively. Hong *et al.* then attempted to modify the batch procedure of the FP-tree algorithm based on the FUP concept and proposed

a Fast Updated FP-tree (FUFPP-tree) structure for easily updating the tree. Three maintenance algorithms were also proposed to maintain the FUFPP tree whether the records are inserted [12, 27], deleted [13, 28] or modified [13, 26] in dynamic databases.

Although the FUP and FUP2 algorithms could indeed improve mining performance for record insertion and record deletion, the original databases still needed to be re-scanned whenever necessary. Hong *et al.* thus proposed three pre-large algorithms for record insertion [24], record deletion [21] and record modification [22], respectively, to further reduce the need for rescanning the original databases based on two support thresholds of pre-large algorithms. Based on the pre-large concepts, the original databases are unnecessary to rescan until a number of records have been processed. Since rescanning the databases spent much computation time, the maintenance cost could thus be reduced in the pre-large algorithms.

In the first part of this dissertation, a maintenance framework is proposed for effectively updating the constructed FUFPP-tree structures and then deriving the desired frequent itemsets from it. It consisted of three Pre-FUFPP maintenance algorithms for record insertion, record deletion, and record modification in dynamic databases, respectively. The proposed three maintenance algorithms do not require rescanning the original databases to re-construct the FUFPP tree until a number of records have been processed. The number is determined from the two support thresholds and the size of the original databases. In the experimental results, the proposed three Pre-FUFPP maintenance algorithms ran faster than the batch FP tree and FUFPP tree but generated nearly the same number of tree nodes. That is, the proposed algorithms can thus achieve a good trade-off between execution time and tree complexity.

In the association rules mining, it treats all items in the databases as binary variables. That is, they only consider whether an item is bought in a record or not. In this case, frequent itemsets just reveal the occurrence importance of the itemsets in the records, but do not reflect any other implicit factors, such as prices or profits. Utility mining was thus proposed to partially solve the above problem [9, 41, 55]. Liu *et al.* then presented the two-phase

algorithm for fast discovering all high utility itemsets based on the downward-closure property to generate and test candidate high utility itemsets in a level-wise way [42]. The databases-scanning time is, however, a bottleneck of the approach. In second part of this dissertation, a new high utility pattern tree (HUP-tree) algorithm with the aid of the HUP-tree structure is first designed for mining high utility itemsets. An array is then attached to each node for keeping the quantities of its super-items in the path for later mining process. The HUP-growth mining algorithm based on the proposed HUP-tree structure is then presented to efficiently mine the high utility itemsets. In the experimental results, the proposed algorithm for mining high utility itemsets can thus be efficiently than the two-phase algorithm in a level-wise way.

In addition to binary variables in databases of association rules mining, transaction data in real-world applications, however, usually consisted of quantitative values. In recent years, the fuzzy set theory [32, 58] has been used more and more frequently in intelligent systems because of its simplicity and similarity to human reasoning. Several fuzzy learning algorithms for inducing rules from given sets of data have been designed and used to good effect with specific domains [8, 17, 33, 49, 51, 57]. Hong *et al.* proposed the fuzzy mining algorithms [23, 25] for managing quantitative data in a level-wise approach of Apriori algorithm. Papadimitriou *et al.* then proposed the fuzzy frequent pattern tree (FFPT) algorithm to find fuzzy association rules [45] in the pattern-growth approach. In the third part of this dissertation, we attempt to extend the FP-tree mining process for handling quantitative data from the global values of fuzzy regions. A fuzzy data mining framework is proposed to efficiently mine the fuzzy frequent itemsets from quantitative databases. It consists of three fuzzy data mining algorithms called fuzzy FP-tree algorithm, the compressed fuzzy frequent pattern tree (CFFP-tree) algorithm, and the upper-bound fuzzy frequent pattern tree (UBFFP-tree) algorithm for constructing the tree structures and mining the fuzzy frequent itemsets from it, respectively. Experimental results also show that the performance of the