การพัฒนาการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอจากฐานข้อมูลรายการ

ปรีชา สิทธิชัยทวีกุล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา
ธันวาคม 2558

ENHANCING OF MINING TOP-K FREQUENT-REGULAR PATTERNS FROM TRANSACTIONAL DATABASES

PREECHA SITTICHAITAWEEKUL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE MASTER DEGREE OF SCIENCE IN INFORMATION TECHNOLOGY
FACULTY OF INFORMATICS BURAPHA UNIVERSITY
DECEMBER 2015

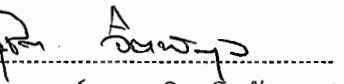
คณะกรรมการควบคุมวิทยานิพนธ์และคณะกรรมการสอบวิทยานิพนธ์ได้พิจารณาวิทยานิพนธ์ของ ปรีชา สิทธิชัยทวีกุล ฉบับนี้แล้ว เห็นสมควรรับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ ของมหาวิทยาลัยบูรพาได้

คณะกรรมการควบคุมวิทยานิพนธ์

อาจารย์ที่ปรึกษา

(ดร.โกเมศ อัมพวัน)

คณะกรรมการสอบวิทยานิพนธ์

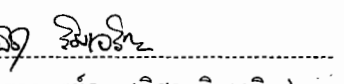


ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.อนุชิต จิตพัฒนกุล)

กรรมการ

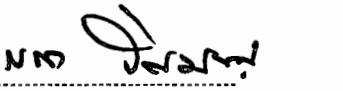
(ดร.โกเมศ อัมพวัน)



กรรมการ

(ผู้ช่วยศาสตราจารย์ดร.สุนิสา ริมเจริญ)

คณะวิทยาการสารสนเทศ อนุมัติให้รับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ ของมหาวิทยาลัยบูรพา



คณบดีคณะวิทยาการสารสนเทศ

(ผู้ช่วยศาสตราจารย์ ดร.สุวรรณา รัศมีขวัญ)

วันที่ 19 เดือน ธันวาคม พ.ศ. 2558

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยความกรุณาจากอาจารย์ ดร.โกเมศ อัมพวัน อาจารย์ปรึกษาหลักที่กรุณาอบรมสั่งสอน ให้ความรู้ ให้คำปรึกษา แนะแนวทางในการทำวิทยานิพนธ์และให้กำลังใจ ผู้ทำวิจัยรู้สึกซาบซึ้งในการช่วยเหลือในการทำวิทยานิพนธ์เป็นอย่างยิ่ง

ขอขอบคุณคณาจารย์คณะวิทยาการสารสนเทศที่คอยอบรมสั่งสอนและให้ความรู้ความสามารถในการจัดทำวิทยานิพนธ์

ขอขอบคุณผู้ช่วยศาสตราจารย์ ดร.อนุชิต จิตพัฒนกุล ที่กรุณาให้เกียรติเป็นประธานโดยมีผู้ช่วยศาสตราจารย์ ดร.สุนิสา ริมเจริญ เป็นกรรมการในการสอบวิทยานิพนธ์

ขอขอบพระคุณคุณพ่อ คุณแม่ ญาติพี่น้อง ที่เป็นกำลังใจและให้การสนับสนุนในการเรียนของผู้ทำวิจัยเสมอมา

ขอขอบคุณรุ่นพี่และรุ่นน้องสาขาเทคโนโลยีสารสนเทศ และเจ้าหน้าที่คณะวิทยาการสารสนเทศที่ให้การสนับสนุน ข้อแนะนำและให้ความช่วยเหลือในการทำวิทยานิพนธ์

คุณค่าและประโยชน์ของวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบเป็นกตัญญูกตเวทิตาแด่บุพการีและบูรพาจารย์ และผู้ที่มีพระคุณทุกท่านทั้งในอดีตและปัจจุบัน ที่ทำให้ข้าพเจ้าเป็นผู้มีความรู้และประสบความสำเร็จในด้านต่าง ๆ จนถึงทุกวันนี้

ปรีชา สิทธิชัยทวีกุล

54910165: สาขาเทคโนโลยีสารสนเทศ; วท.ม. (เทคโนโลยีสารสนเทศ)
คำสำคัญ: ดาต้าไมนิง, รูปแบบปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ, รูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอ

ปรีชา สิทธิชัยทวีกุล: การพัฒนาการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอจากฐานข้อมูลรายการ (Enhancing of Mining Top-k Frequent-regular Patterns from Transactional Databases) คณะกรรมการควบคุมวิทยานิพนธ์: โกเมศ อัมพวัน, ปร.ด., 71 หน้า. ปี พ.ศ. 2558.

การค้นหารูปแบบปรากฏบ่อยและปรากฏสม่ำเสมอถูกนำมาใช้ในงานหลาย ๆ ด้าน เช่น ธุรกิจค้าปลีก การวิเคราะห์ดีเอ็นเอ การติดตามพฤติกรรมผู้สูงอายุ และอื่น ๆ ในการค้นหารูปแบบปรากฏบ่อยและปรากฏอย่างสม่ำเสมอจะทำการวัดความสำคัญหรือความน่าสนใจของรูปแบบด้วยการประยุกต์ใช้ค่าขีดแบ่งสนับสนุนและค่าขีดแบ่งความสม่ำเสมอในการปรากฏซ้ำ ซึ่งเป็นที่ทราบกันดีว่าการกำหนดค่าสนับสนุนขั้นต่ำให้เหมาะสมกับข้อมูลนั้นสามารถทำได้ยาก ดังนั้นการอนุญาตให้ผู้ที่ต้องการค้นหารูปแบบดังกล่าวทำการกำหนดจำนวนรูปแบบที่ต้องการจะช่วยลดความยุ่งยากในการกำหนดค่าขีดแบ่งที่เหมาะสมได้ ด้วยเหตุนี้ จึงได้มีการคิดค้นที่จะทำการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและและปรากฏอย่างสม่ำเสมอขึ้นมาเพื่อแก้ไขปัญหาความยุ่งยากดังกล่าว ภายใต้แนวคิดนี้ ผู้ที่ต้องการวิเคราะห์ข้อมูลจะสามารถกำหนดจำนวนผลลัพธ์ที่ต้องการ (k) และกำหนดค่าขีดแบ่งความสม่ำเสมอในการปรากฏซ้ำ เพื่อกำหนดระยะห่างในการปรากฏซ้ำของเซตรูปแบบ ซึ่งเซตรูปแบบที่ได้จะมีความถี่สูงสุดและปรากฏอย่างสม่ำเสมอ อย่างไรก็ตามวิธีการนี้มักจะให้ผลลัพธ์เป็นเซตรูปแบบที่มีขนาดเล็ก ซึ่งไม่สามารถนำมาวิเคราะห์ความสัมพันธ์ที่น่าสนใจได้ จากเหตุการณ์ดังกล่าวงานวิจัยนี้จึงมุ่งเน้นที่จะทำการพัฒนาการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอ โดยทำการกำหนดขนาดขั้นต่ำของผลลัพธ์ให้เป็นเซตรูปแบบที่ประกอบไปด้วยสองรายการขึ้นไปและได้เสนอขั้นตอนวิธีที่มีประสิทธิภาพที่มีชื่อว่า ETFRP (Enhancing of Mining Top-k Frequent-regular Patterns) เพื่อทำการค้นหารูปแบบในลักษณะดังกล่าว ขั้นตอนวิธีที่นำเสนอจะทำการอ่านข้อมูลเพียงครั้งเดียว ประยุกต์ใช้การค้นหาแบบดีสุดก่อน (Best-first search) และใช้โครงสร้างลิงค์ลิสต์ในการจัดเก็บเซตรูปแบบเพื่อประหยัดพื้นที่หน่วยความจำ โดยจากการทดลองเพื่อทดสอบประสิทธิภาพของขั้นตอนวิธีที่นำเสนอปรากฏว่าขั้นตอน ETFRP สามารถทำงานได้อย่างมีประสิทธิภาพและสามารถค้นหาเซตรูปแบบตามจำนวนที่ผู้ใช้ต้องการได้

54910165: MAJOR: INFORMATION TECHNOLOGY; M.Sc. (INFORMATION TECHNOLOGY)
KEYWORDS: DATA MINING, FREQUENT-REGULAR PATTERN, TOP-K FREQUENT-REGULAR PATTERN

PREECHA SITTICHAITAWEEKUL: ENHANCING OF MINING TOP-K FREQUENT-REGULAR PATTERNS FROM TRANSACTIONAL DATABASES. ADVISORY COMMITTEE: KOMATE AMPHAWAN, Ph.D., 71 P. 2015.

Frequent-regular pattern mining have played an important role in a wide range of applications such as retail business, DNA analysis, Elderly habits monitoring, etc. To find these patterns, two parameters, support and regularity thresholds, have to be specified in order to measure/evaluate significant or interestingness of patterns. However, it is well-known that setting appropriate threshold is very difficult and causes suffering to the users and it is more reasonable to specify the number of required patterns. Thus, the task of top-k frequent-regular pattern mining is introduced to alleviate this difficulty. Based on this framework, the users have to assign only the value of k (the number of patterns to be mined) and a regularity threshold to control the interval of occurrence. Then, the patterns with regular occurrence and have highest support values would be returned. Unfortunately, this approach usually generates small size patterns which is insufficient to extract interesting relationships. From above issue, this research aims to enhance ability of top-k frequent-regular pattern mining by setting the minimum length of patterns to be mined to be two. Thus, the relationship of the discovered patterns can be easily extracted. To mine such patterns, we propose an efficient single-pass algorithm, called *ETFRP*, applying best-first search strategy to quickly discover the results and employing a linked-list structure to maintain patterns during mining process. Experimental studies show that *ETFRP* can effectively and efficiently discover patterns that meet the users' interest.

สารบัญ

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญตาราง (ต่อ)

สารบัญภาพ

สารบัญภาพ (ต่อ)

บทที่ 1
บทนำ

ความเป็นมาและความสำคัญของปัญหา

ในยุคปัจจุบันการดำเนินธุรกิจมีการแข่งขันกันอย่างมากมาย ทำให้ข้อมูลสารสนเทศเข้ามามีส่วนร่วมในการช่วยตัดสินใจในการดำเนินธุรกิจมากขึ้น เทคนิคการทำเหมืองข้อมูล (Data mining) เป็นวิธีการหนึ่งที่สามารถค้นหาสารสนเทศหรือองค์ความรู้จากข้อมูลดิบที่ถูกจัดเก็บในองค์กรต่าง ๆ ได้ และได้ถูกประยุกต์ใช้กันอย่างแพร่หลายในหลาย ๆ ธุรกิจ อาทิเช่น งานด้านวิทยาศาสตร์ ด้านการแพทย์ ด้านพันธุวิศวกรรมศาสตร์ ด้านกฎหมาย และธุรกิจการค้าต่าง ๆ การทำเหมืองข้อมูลจะสามารถดำเนินการได้หลายรูปแบบและหลายวิธี อาทิเช่น การหากฎความสัมพันธ์ของข้อมูล (Association rule mining (Rakesh Agrawal, Tomasz Imielinski & Arun Swami, 1993; Rakesh Agrawal & Ramakrishnan Srikant, 1994) การจำแนกประเภทข้อมูล (Classification) การจัดกลุ่มข้อมูล (Clustering) การวิเคราะห์สิ่งผิดปกติ (Outlier analysis) และอื่น ๆ ในงานวิจัยนี้จะสนใจในส่วนของการหากฎความสัมพันธ์ของข้อมูล ซึ่งจะประกอบไปด้วยสองขั้นตอนการทำงานหลัก คือ 1) การค้นหารูปแบบปรากฏบ่อย (Frequent pattern) ที่มีค่าความถี่ของการปรากฏมากกว่าหรือเท่ากับค่าขีดแบ่งสนับสนุน (Support threshold) และ 2) การสร้างกฎความสัมพันธ์จากผลลัพธ์ที่ได้ในข้อ 1) โดยกฎความสัมพันธ์ที่น่าสนใจจะต้องมีค่าความเชื่อมั่น (Confidence value) มากกว่าหรือเท่ากับค่าขีดแบ่งความเชื่อมั่น (Confidence threshold)

ในการค้นหาเซตรูปแบบที่ปรากฏบ่อยและการกำหนดค่าความความเชื่อมั่นที่ใช้ในการวัดความน่าสนใจของกฎความสัมพันธ์ ได้มีงานวิจัยที่เกี่ยวข้อง ได้แก่ Mining Frequent Patterns without Candidate Generation (Jiawei Han, Jian Pei & Yiwen Yin, 2000), MAFIA: A Maximal Frequent Itemset Algorithm for Transactional Databases (Doug Burdick Manuel Calimlim & Johaannes Gehrke, 2001), Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach (Jiawei Han, Jian Pei, Yiwen Yin, Runying Mao, 2004), Fast Algorithms for Frequent Itemset Mining using Fp-trees (G¨osta Grahne & Jianfei Zhu, 2005) ฯลฯ ถึงอย่างไรก็ตามในการวัดความสำคัญของเซตรูปแบบที่ปรากฏบ่อยโดยใช้ความถี่เพียงอย่างเดียวไม่เพียงพอต่อความต้องการในการใช้งานบางอย่าง ตัวอย่างเช่น การจัดซื้อสินค้าเข้าสต็อกสินค้า ผู้จัดหาสินค้าต้องทราบว่าสินค้าแต่ละรายการจะหมดลงเมื่อใดโดยประมาณ ซึ่งเซตรูปแบบที่ปรากฏบ่อยไม่สามารถตอบคำถามในส่วนนี้ได้ จึงเกิดงานวิจัยในการค้นหาเซตรูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ (Frequent-regular pattern) การปรากฏอย่างสม่ำเสมอ คือ ข้อมูลรายการจะปรากฏในทรานแซกชันห่างกันเป็นช่วง ๆ ในระยะห่างที่ผู้ใช้ให้ความสำคัญ เพื่อตอบสนองความต้องการในงานด้านต่าง ๆ ได้มีงานวิจัยที่เกี่ยวข้อง ได้แก่ Mining Weighted Patterns in a Sequence Database with a Time-interval Weight (Joong Hyuk Chang, 2011), Discovering Periodic-Frequent Patterns in Transactional Databases (Syed Khairuzzaman Tanbeer, Chowdhury Farhan Ahmed, Byeong-Soo Jeong, Young-Koo Lee, 2009) ฯลฯ ในการค้นหาเซตรูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ ผู้ใช้ต้องกำหนดพารามิเตอร์ 2 ตัว คือ ค่าขีดแบ่ง

สนับสนุนและค่าขีดแบ่งความสม่ำเสมอ (Regularity threshold (σ_r)) ซึ่งเป็นเกณฑ์ในการวัดความสำคัญของเซตรายการที่ใช้ในการค้นหาเซตรูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ ซึ่งเป็นที่ทราบกันดีว่าการกำหนดค่าขีดแบ่งสนับสนุนของรายการเป็นเรื่องที่ทำได้ยาก เนื่องจากผู้ใช้ส่วนใหญ่ขาดประสบการณ์ในการกำหนดค่าขีดแบ่งสนับสนุน ถ้ากำหนดค่าขีดแบ่งสนับสนุนจำนวนมากเกินไปก็ทำให้รายการที่นำมาค้นหาเซตรูปแบบที่ปรากฏบ่อยและปรากฏแบบสม่ำเสมอมีปริมาณน้อยหรือไม่มีรายการที่นำมาใช้ในการค้นหาเซตรูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอเลย หากมีการกำหนดค่าขีดแบ่งสนับสนุนน้อยเกินไปก็จะทำให้มีจำนวนรายการที่นำมาค้นหาเซตรูปแบบที่ปรากฏบ่อยและปรากฏแบบสม่ำเสมอจำนวนมาก ทำให้ใช้เวลาในการประมวลผลเป็นเวลานาน และผลลัพธ์ที่ได้อาจเป็นข้อมูลที่ไม่น่าสนใจ ด้วยเหตุนี้จึงได้เกิดกระบวนการค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอเคอันดับแรกโดยไม่กำหนดค่าขีดแบ่งสนับสนุน เพื่อลดความยุ่งยากของการกำหนดค่าขีดแบ่งสนับสนุนของผู้ใช้ วิธีการนี้ผู้ใช้จะต้องระบุจำนวนผลลัพธ์ที่ต้องการเคอันดับแทนการกำหนดค่าขีดแบ่งสนับสนุนของรายการและกำหนดค่าขีดแบ่งความสม่ำเสมอ ซึ่งการกำหนดจำนวนผลลัพธ์สามารถทำได้ง่ายกว่าการกำหนดค่าขีดแบ่งสนับสนุน และการกำหนดค่าขีดแบ่งความสม่ำเสมอก็เป็นระยะห่างที่ผู้ใช้ให้ความสำคัญ โดยมีงานวิจัยที่เกี่ยวข้อง ได้แก่ Mining Top-k Frequent Closed Patterns without Minimum Support (Jiawei Han, Jianyong Wang, Ying Lu & Petre Tzvetkov, 2002), Mining Top-k Closed Sequential Patterns (Jianyong Wang, Jiawei Han, Ying Lu & Petre Tzvetkov, 2005), TFP: an Efficient Algorithm for Mining Top-k Frequent Closed Itemsets (Jianyong Wang, Jiawei Han, Ying Lu & Petre Tzvetkov, 2005), Mining N-most Interesting Itemsets (Ada Wai-chee Fu, Renfrew Wang-wai Kwong & Jian Tang, 2004), Mining N-most Interesting Itemsets without Support Threshold by the COFI-tree (Sze-Chung Ngan, Tsang Lam, Raymond Chi-Wing Wong & Ada Wai-Chee Fu, 2005), Efficient Mining Top-k Regular-frequent Itemset using Compressed Tidsets (Komate Amphawan, Philippe Lenca & Athasit Surarerks, 2012), Mining Top-*k* Regular-frequent Itemsets using Database Partitioning and Support Extimation (Komate Amphawan, Philippe Lenca & Athasit Surarerks, 2012), Mining Top-k Periodic-Frequent Pattern from Transactional Databases without Support Threshold (Komate Amphawan, Philippe Lenca & Athasit Surarerks, 2009), Mining Top-k Frequent-regular Patterns Based on User-given Trade-off Between Frequency and Regularity (Komate Amphawan & Philippe Lenca, 2013) ฯลฯ ซึ่งผลที่ได้จากงานวิจัยต่าง ๆ สามารถลดปัญหาการกำหนดค่าขีดแบ่งสนับสนุน และเพิ่มประสิทธิภาพในการค้นหาให้ตรงกับความต้องการของผู้ใช้ได้ดีขึ้น ถึงอย่างไรก็ตามการค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอเคอันดับแรกโดยไม่กำหนดค่าขีดแบ่งสนับสนุนจะมีผลลัพธ์เซตรูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอที่มีขนาดหนึ่งรายการ (1-patterns) จำนวนมาก ซึ่งเซตรูปแบบที่มีขนาดหนึ่งรายการไม่สามารถที่จะนำมาวิเคราะห์ความสัมพันธ์ระหว่างรายการได้ และทำให้เซตรูปแบบที่มีค่าสนับสนุนน้อยกว่าถูกตัดทิ้งเป็นจำนวนไม่น้อย ทำให้ผู้ทำวิจัยได้เกิดความคิดในการพัฒนาคุณภาพของผลลัพธ์ โดยนำเสนอวิธีการในการค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอเคอันดับแรก โดยไม่มีผลลัพธ์ของรูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอที่มีขนาดเซตรายการเป็นหนึ่ง ซึ่งการเพิ่มความยาวผลลัพธ์ของ

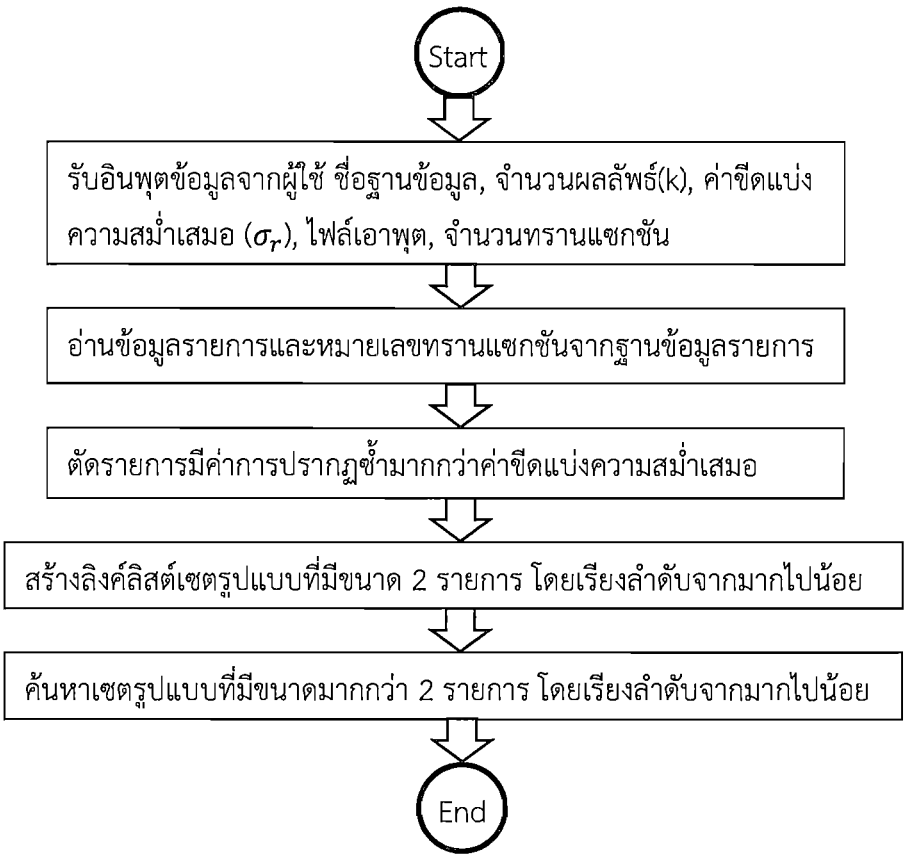
เซตรูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ จะช่วยให้การวิเคราะห์ความสัมพันธ์ของรายการที่ปรากฏขึ้นพร้อมกันมีประสิทธิภาพ และตรงกับความต้องการของผู้ใช้ โดยการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอ ผู้ทำวิจัยได้นำเสนออัลกอริทึมที่มีประสิทธิภาพที่เรียกว่า ETFRP โดยใช้การค้นหาที่ดีที่สุด (Best first search) ในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอ และใช้โครงสร้างลิงค์ลิสต์ในการจัดเก็บเซตรูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอทั้งหมดในระหว่างขั้นตอนการค้นหา เนื่องจากโครงสร้างลิงค์ลิสต์มีความยืดหยุ่นของการประหยัดพื้นที่ในการเก็บข้อมูล ซึ่งผลการทดลองแสดงให้เห็นว่าอัลกอริทึม ETFRP มีประสิทธิภาพสามารถค้นหารูปแบบปรากฏบ่อยและปรากฏอย่างสม่ำเสมอที่ตรงกับความสนใจของผู้ใช้ได้ดีขึ้น

แนวทางในการแก้ปัญหา

จากปัญหาในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอที่แสดงผลลัพธ์ของเซตรูปแบบที่มีขนาดเป็นหนึ่งรายการจำนวนมาก ทำให้ไม่สามารถวิเคราะห์ความสัมพันธ์ของรายการได้ ผู้วิจัยจึงมีแนวคิดในการแก้ไขปัญหา โดยจะปรับปรุงคุณภาพผลลัพธ์ของเซตรูปแบบให้มีขนาดตั้งแต่ 2 รายการขึ้นไป และมีคุณสมบัติดังนี้

1. มีค่าสนับสนุนการปรากฏสูงสุดเคอันดับแรก
2. มีค่าการปรากฏอย่างสม่ำเสมอไม่เกินค่าขีดแบ่งความสม่ำเสมอ (σ_r)
3. มีขนาดของเซตรูปแบบตั้งแต่สองขึ้นไป

โดยใช้โครงสร้างข้อมูลลิงค์ลิสต์ในการจัดเก็บเซตรูปแบบทั้งหมดเนื่องจากลิงค์ลิสต์มีความยืดหยุ่นในเรื่องการประหยัดพื้นที่ในการจัดเก็บข้อมูล และเหมาะกับการเก็บข้อมูลที่ไม่สามารถระบุจำนวนได้ชัดเจน โดยมีขั้นตอนดังภาพที่ 1-1



ภาพที่ 1-1 ขั้นตอนการค้นรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอที่มีขนาดรายการมากกว่าหนึ่ง

วัตถุประสงค์ของวิทยานิพนธ์

1. เพื่อเพิ่มคุณภาพของผลลัพธ์ของการค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ โดยพยายามที่จะค้นหาผลลัพธ์ที่สามารถบ่งบอกถึงความสัมพันธ์ของวัตถุหรือเหตุการณ์ที่ปรากฏพร้อม ๆ กันได้ดียิ่งขึ้น

2. เพื่อพัฒนาขั้นตอนวิธีในการค้นหารูปแบบข้างต้นให้สามารถค้นหาผลลัพธ์ได้อย่างมีประสิทธิภาพ

ขอบเขตของวิทยานิพนธ์

1. ในการค้นหารูปแบบปรากฏบ่อยเคอันดับแรกและปรากฏอย่างสม่ำเสมอ ผู้ใช้จะต้องทำการกำหนดจำนวนผลลัพธ์ที่ต้องการ (k) และค่าขีดแบ่งความสม่ำเสมอ (σ_r) ที่จะใช้เป็นเครื่องมือในการชี้วัดการปรากฏขึ้นอย่างสม่ำเสมอของรูปแบบที่ทำการพิจารณา

2. ในการทดสอบประสิทธิภาพของขั้นตอนวิธีที่คิดค้นขึ้นจะทดสอบในเชิงเวลาและหน่วยความจำที่ใช้ในการคำนวณ และยังพิจารณาถึงความยาวของรูปแบบที่สามารถค้นหาได้

3. ชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของขั้นตอนวิธีที่คิดค้นขึ้นจะเป็นข้อมูลแบบฐานข้อมูลรายการ (Transactional database) ที่ประกอบไปด้วยชุดข้อมูลสังเคราะห์ (Synthetic dataset) และ ชุดข้อมูลจริง (Real dataset) ที่สามารถดาวน์โหลดได้จากเว็บไซต์ http://fimi.ua.ac.be/data/

ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถค้นหารูปแบบที่แสดงถึงความสัมพันธ์ของรูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอได้ดียิ่งขึ้น

2. ได้ขั้นตอนวิธีสำหรับการค้นหารูปแบบข้างต้นได้อย่างมีประสิทธิภาพ ที่สามารถนำไปประยุกต์ใช้ในงานหลาย ๆ ด้านได้

3. สามารถพัฒนาโมเดลต้นแบบในการค้นหารูปแบบดังกล่าว เพื่อนำไปใช้ในการศึกษาหรือพัฒนาต่อยอดต่อไปได้

แผนการดำเนินงาน

ในการดำเนินการวิจัยนี้ดำเนินการภายใต้ระยะที่ระบุไว้ในตารางที่ 1-1

ตารางที่ 1-1 ระยะเวลาในการดำเนินการวิจัย

แผนการดำเนินงานวิจัย	ปี 2556				ปี 2557			ปี 2558		
	เดือน				เดือน			เดือน		
	1-4	5-7	8	9-12	1-5	4-8	9-12	1-4	5-8	9-12
1. ศึกษางานวิจัยที่เกี่ยวข้องและจัดเตรียมอุปกรณ์ในการทำวิจัย	● →									
2. กำหนดขอบเขตความของงานวิจัย และนำเสนอวิธีการในการดำเนินงานวิจัย		● →	→							
3. ศึกษาอัลกอริทึมและเขียนโปรแกรมทดสอบและจัดทำเอกสารตีพิมพ์			● →	→						
4. จัดทำเอกสารและทำการสอบ 3 บท					● →					
5. ทดสอบการทำงานและปรับแก้โปรแกรมและจัดทำเอกสารตีพิมพ์					●	→	→			
6. จัดทำเอกสารฉบับจบและสอบวิทยานิพนธ์						●	→	→	→	→

บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

ในช่วงหลายปีที่ผ่านมา ปัญหาการทำเหมืองข้อมูลเพื่อค้นหากฎความสัมพันธ์ (Association rules mining) ได้รับการศึกษาอย่างกว้างขวางดังที่กล่าวมาแล้วในบทที่หนึ่ง ในบทนี้จะกล่าวถึงขั้นตอนและนิยามต่าง ๆ ในการค้นหากฎความสัมพันธ์ของฐานข้อมูลรายการและงานวิจัยที่เกี่ยวข้องในการค้นหากฎความสัมพันธ์ ดังนี้

1. การค้นหากฎความสัมพันธ์ของฐานข้อมูลรายการ (Association rules mining)
2. การค้นหารูปแบบที่ปรากฏบ่อย (Frequent patterns mining)
3. การค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ (Frequent-regular patterns mining)
4. การค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรก (Top-k frequent patterns mining)
5. การค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอ (Top-k frequent-regular patterns mining)

การค้นหากฎความสัมพันธ์ของฐานข้อมูลรายการ (Association rules mining)

ในการค้นหากฎความสัมพันธ์ของฐานข้อมูลรายการถูกนำเสนอเมื่อ ปี ค.ศ. 1993 โดย Rakesh Agrawal, et al. ได้ค้นหากฎความสัมพันธ์ของฐานข้อมูลรายการจากสินค้าในตระกร้าของผู้ที่มาซื้อสินค้า (Market basket) ซึ่งการค้นหากฎความสัมพันธ์ของฐานข้อมูลรายการประกอบไปด้วย 2 ขั้นตอน คือ การค้นหารูปแบบที่ปรากฏบ่อย (Frequent patterns) และการสร้างกฎความสัมพันธ์ (Association rule) การค้นหารูปแบบที่ปรากฏบ่อยเป็นการค้นหาเซตรูปแบบที่ปรากฏขึ้นร่วมกันบ่อย ๆ ในฐานข้อมูลโดยมีความถี่ไม่น้อยกว่าค่าขีดแบ่งสนับสนุน (Support threshold) ที่ผู้ใช้เป็นผู้กำหนด เมื่อได้รูปแบบที่ปรากฏบ่อยแล้ว จะนำเซตรูปแบบที่ได้มาสร้างกฎความสัมพันธ์ ซึ่งในกฎความสัมพันธ์แต่ละตัวจะมีระดับความเชื่อมั่น (Confidence threshold) ของตัวเอง โดยนำจำนวนครั้งในการปรากฏขึ้นร่วมกันของรายการหารด้วยจำนวนครั้งในการปรากฏของเซตรายการตัวที่อยู่ด้านหน้าคูณด้วยหนึ่งร้อย ซึ่งกฎความสัมพันธ์ที่ได้จะเป็นกฎความสัมพันธ์ที่หน้าสนใจก็ต่อเมื่อมีความถี่ในการปรากฏร่วมกันไม่น้อยกว่าค่าขีดแบ่งสนับสนุน และมีค่าความเชื่อมั่นไม่น้อยกว่าค่าขีดแบ่งความเชื่อมั่นที่ผู้ใช้กำหนด โดยมีนิยามที่เกี่ยวข้องดังนี้

นิยามที่ 2.1 กำหนดให้ I แทนเซตของสินค้าในฐานข้อมูล มีสมาชิก $I = \{i_1, i_2, i_3,..., i_n\}$ โดยที่ i_j เป็นสินค้าในฐานข้อมูล และ $1 \le j \le n$

นิยามที่ 2.2 กำหนดให้ TDB แทนค่าฐานข้อมูลรายการ ซึ่งประกอบไปด้วยเซตของทรานแซกชันแต่ละตัว $TDB=\{t_1, t_2, t_3,..., t_n\}$ โดยที่ t_j เป็นทรานแซกชันแต่ละตัวในฐานข้อมูลรายการ โดยที่ $1 \le j \le n$ แต่ละทรานแซกชันจะประกอบไปด้วยเซตของรายการสินค้า โดยที่ j เป็นหมายเลขของทรานแซกชันที่ไม่ซ้ำกัน $t_j = \{i_1, i_2, i_3,..., i_k\}$ โดยที่ $t_j \subseteq I$

ถ้าเซต *A* หรือเซต *B* เป็นเซตรายการสินค้าใด ๆ โดยที่ *A, B* $\subseteq$ *I* แล้ว จะสามารถให้นิยามกฎความสัมพันธ์โดยการอุปนัยของข้อมูลในรูปแบบ *A*→*B* เมื่อ *A, B* $\subseteq$ I และ*A* $\cap$ *B* = $\emptyset$ กล่าวคือ เมื่อมีการซื้อสินค้า *A* แล้วจะมีการซื้อสินค้า *B* ไปด้วย

เกณฑ์การวัดที่ใช้ในการค้นหากฎความสัมพันธ์ของฐานข้อมูลรายการมีด้วยกันสองตัวแปร โดยผู้ใช้เป็นผู้กำหนดได้แก่ ค่าขีดแบ่งสนับสนุนและค่าความเชื่อมั่น เพราะฉะนั้นเราต้องหาค่าตัวแปรทั้งสองของเซตรายการที่ปรากฏบ่อยเพื่อนำมาตรวจสอบกับเกณฑ์ดังกล่าว เพื่อพิจารณาว่าเป็นกฎความสัมพันธ์ที่หน้าสนใจหรือไม่ โดยหาค่าตัวแปรทั้งสองได้จาก

1. ค่าสนับสนุน (Support value) ของกฎความสัมพันธ์ *A*→*B* มีค่าเท่ากับจำนวนที่ปรากฏขึ้นร่วมกันของ *A* และ *B* หารด้วยจำนวนรายการทั้งหมดในฐานข้อมูล โดยคิดเป็นร้อยละ ดังสมการ

$$sup(A \rightarrow B) = \frac{Frequent(AUB) * 100}{|TDB|}$$

2. ค่าความเชื่อมั่น (Confidence value) ของกฎความสัมพันธ์ *A*→*B* มีค่าเท่ากับจำนวนที่ปรากฏขึ้นร่วมกันของ *A* และ *B* หารด้วยจำนวนปรากฏขึ้นของ *A* โดยคิดเป็นร้อยละ ดังสมการ

$$conf(A \rightarrow B) = \frac{Frequent(AUB) * 100}{Frequent(A)}$$

ตัวอย่างการค้นหากฎความสัมพันธ์ของฐานข้อมูลรายการ โดยมีข้อมูลฐานข้อมูลตัวอย่างดังตารางที่ 2-1

ตารางที่ 2-1 ฐานข้อมูลตัวอย่างการค้นหากฎความสัมพันธ์

Database	
Transaction	Purchase item
T1	Bread, Milk, Yam
T2	Bread, Yam
T3	Bread, Jelly, Yam
T4	Beer, Bread
T5	Beer, Milk

หากผู้ใช้กำหนดค่าขีดแบ่งสนับสนุนไว้ 60% และค่าขีดแบ่งความเชื่อมั่น 50% จะได้กฎความสัมพันธ์ดังนี้

{Bread} → {Yam} จะมีการปรากฏขึ้นร่วมกัน 3 ครั้ง จากทั้งหมด 5 ทรานแซกชัน {Bread} → {Yam} จะมีค่าสนับสนุนเท่ากับ $\frac{3}{5}$ x 100 = 60 เปอร์เซ็นต์

{Bread} → {Yam} จะมีการปรากฏขึ้นร่วมกัน 3 ครั้ง และ {Bread} มีการปรากฏขึ้นทั้งหมด 4 ครั้ง {Bread} → {Yam} จะมีค่าความเชื่อมั่นเท่ากับ $\frac{3}{4}$ x 100 = 75 เปอร์เซ็นต์

{Yam} → {Bread} จะมีการปรากฏขึ้นร่วมกัน 3 ครั้ง จากทั้งหมด 5 ทรานแซกชัน {Yam} → {Bread} จะมีค่าสนับสนุนเท่ากับ $\frac{3}{5}$ x 100 = 60 เปอร์เซ็นต์

{Yam} → {Bread} จะมีการปรากฏขึ้นร่วมกัน 3 ครั้ง และ {Yam} มีการปรากฏขึ้นทั้งหมด 3 ครั้ง {Yam} → {Bread} จะมีค่าความเชื่อมั่นเท่ากับ $\frac{3}{3}$ x 100 = 100 เปอร์เซ็นต์

เพราะฉะนั้นจากเงื่อนไขที่ผู้ใช้กำหนดให้จะได้กฎความสัมพันธ์ที่หน้าสนใจที่ผ่านเกณฑ์ค่าขีดแบ่งสนับสนุน 60% และค่าขีดแบ่งความเชื่อมั่น 50% คือ {Bread → Yam} และ {Yam → Bread}

กระบวนการค้นหากฎความสัมพันธ์ในขั้นตอนแรก หรือการค้นหารูปแบบที่ปรากฏบ่อยใช้ระยะเวลาในการค้นหาเป็นเวลานาน เนื่องจากเมื่อจำนวนรายการสินค้ามีจำนวนเพิ่มมากขึ้น รูปแบบที่ปรากฏบ่อยมีความเป็นไปได้ทั้งหมดจะมีจำนวน 2^I เซต โดยที่ I แทนจำนวนสมาชิกในเซตรายการสินค้าในฐานข้อมูล ซึ่งการค้นหารูปแบบที่ปรากฏบ่อยจะเพิ่มขึ้นแบบเอ็กโปรเนเชียล ทำให้การค้นหารูปแบบที่ปรากฏบ่อยใช้เวลานาน ใช้หน่วยความจำและพื้นที่ในการจัดเก็บข้อมูลจำนวนมาก

การค้นหารูปแบบที่ปรากฏบ่อย (Frequent patterns mining)

การค้นหารูปแบบที่ปรากฏบ่อยเป็นกระบวนการแรกในการค้นหากฎความสัมพันธ์ ซึ่งเป็นขั้นตอนที่ใช้เวลานานที่สุดในการค้นหากฎความสัมพันธ์ จึงมีการวิจัยเพื่อลดปริภูมิการค้นหาโดยสร้างวิธีการใหม่ ๆ ในการค้นหาและใช้โครงสร้างข้อมูลเข้ามาช่วยปรับปรุงการค้นหาเพื่อลดเวลาและพื้นที่ในการจัดเก็บข้อมูลให้น้อยลง ซึ่งการวัดความสำคัญของรูปแบบที่ปรากฏบ่อยจากค่าความถี่ของการปรากฏของเซตรูปแบบหรือค่าสนับสนุนเป็นค่าการปรากฏขึ้นทั้งหมดในฐานข้อมูลรายการของเซตรูปแบบ การที่จะเป็นรูปแบบที่ปรากฏบ่อยได้ต้องมีค่าสนับสนุนมากกว่าหรือเท่ากับค่าขีดแบ่งสนับสนุน ซึ่งเป็นสิ่งที่ผู้ใช้เป็นผู้กำหนด โดยมีองค์ประกอบข้อมูลดังนี้

กำหนดให้ I แทนเซตของสินค้าในฐานข้อมูล มีสมาชิก $I = \{i_1, i_2, i_3,..., i_n\}$ โดยที่ i_j เป็นสินค้าในฐานข้อมูลรายการ โดยที่ $1 \le j \le n$ ให้ TDB แทนค่าของฐานข้อมูล ซึ่งประกอบไปด้วยเซตของทรานแซกชันแต่ละตัว $TDB = \{t_1, t_2, t_3,..., t_n\}$ โดยที่ t_j เป็นทรานแซกชันแต่ละตัวในฐานข้อมูลรายการ โดยที่ $1 \le j \le n$ แต่ละทรานแซกชันจะประกอบไปด้วย สมาชิกของสินค้าแต่ละรายการ ซึ่งแต่ละทรานแซกชันจะมีหมายเลขของทรานแซกชันที่ไม่ซ้ำกัน กำหนดให้ $X = \{i_1, i_2, i_3,..., i_l\} \subseteq I$ จะเรียก X ว่าเป็นเซตรูปแบบ (Pattern) หรือ เซตรูปแบบที่มีขนาด l (l-Pattern) ถ้า $X \subseteq Y$ และ T_Z มีเซตของ X ปรากฏอยู่ใน T_Z กำหนดให้ S_Z เป็นค่าสนับสนุนของเซตรูปแบบ X ที่ปรากฏในฐานข้อมูล $S_Z = \{ T_Z \mid 1 \le Z \le |TDB|, T_Z \in TDB$ และ $X \in TDB\}$ ซึ่ง S_Z เป็นจำนวนทรานแซกชันทั้งหมดที่มีเซตรูปแบบ X ปรากฏอยู่ในทรานแซกชันนั้น ๆ ของฐานข้อมูล ผู้ใช้ต้องกำหนดค่าขีดแบ่งสนับสนุน และจะเรียกเซตรูปแบบของ X ว่าเป็นเซตรูปแบบที่ปรากฏบ่อยได้ ก็ต่อเมื่อมีค่าสนับสนุนของการปรากฏไม่น้อยกว่าค่าขีดแบ่งสนับสนุน กฎความสัมพันธ์นั้นจะมีสมการที่อยู่รูปถ้าแล้ว เมื่อมีการซื้อสินค้า X แล้วจะมีการซื้อสินค้า Y ด้วยเขียนแทนด้วย $X \rightarrow Y$ โดยที่ $X \subseteq I$ และ $Y \subseteq I$ และ $X \cap Y = \emptyset$ ซึ่ง X และ Y เป็นเซตรูปแบบที่ปรากฏบ่อย แต่เซตรูปแบบที่ปรากฏบ่อยอาจมีรายการมากกว่าหรือเท่ากับหนึ่ง

รูปแบบฐานข้อมูลที่นำมาใช้ในการค้นหารูปแบบที่ปรากฏบ่อยจะประกอบด้วยฐานข้อมูลสองรูปแบบ คือ ฐานข้อมูลแนวตั้ง (Vertical Database) และฐานข้อมูลแนวนอน (Horizontal Database) โดยฐานข้อมูลแนวตั้งจะแสดงรายการสินค้าแต่ละรายการว่ามีปรากฏในทรานแซกชันใดบ้าง ดังตารางที่ 2-2 ส่วนฐานข้อมูลแนวนอนจะแสดงทรานแซกชันไอดีที่ประกอบไปด้วยรายการใดบ้าง ดังตารางที่ 2-3 แต่โดยปกติฐานข้อมูลทั่วไปจะอยู่ในรูปแบบฐานข้อมูลแนวนอน ซึ่งจะต้องนำมาแปลงเป็นฐานข้อมูลแนวตั้งเพื่อความสะดวกในการค้นหารูปแบบที่ปรากฏบ่อย

ตารางที่ 2-2 ฐานข้อมูลแนวตั้ง (Vertical database)

Item	Tids
a	1, 4, 6, 7, 8, 10, 11, 12
b	1, 3, 4, 6, 7, 8, 11, 12
c	2, 3, 5, 6, 7, 8, 9, 10
d	1, 2, 3, 4, 6, 7, 9, 11, 12
e	1, 2, 5, 8, 10
f	3, 4, 11
g	1, 3, 4, 5, 6, 10, 12

ตารางที่ 2-3 ฐานข้อมูลแนวนอน (Horizontal database)

Tid	Items
1	a, b, d, e, g
2	c, d, e
3	b, c, d, f, g
4	a, b, d, f, g
5	c, e, g
6	a, b, c, d, g
7	a, b, c, d
8	a, b, c, e
9	b, c, d
10	a, c, e, g
11	a, b, d, f
12	a, b, d, g

ใน ปี ค.ศ. 1994 Rakesh Agrawal, et al. ได้เสนออัลกอรึทึมเอไพโอริ (Apriori algorithm) เพื่อลดการเปรียบเทียบรายการที่จะเป็นรูปแบบที่ปรากฏบ่อย โดยสับเซตของเซตรูปแบบใดที่เป็นเซตเซตรูปแบบไม่ปรากฏบ่อยเซตรูปแบบของข้อมูลชุดนั้นที่มีขนาดใหญ่กว่าก็ย่อมไม่ปรากฏบ่อยด้วย ซึ่งอัลกอริทึมนี้ใช้การค้นหาแนวกว้าง (Breath first search) ซึ่งเริ่มจากอ่านฐานข้อมูล เพื่อนับค่าสนับสนุนเซตรูปแบบที่มีขนาดหนึ่งรายการ (1 pattern) หากเซตรูปแบบใดมีค่าสนับสนุนไม่น้อยกว่าค่าขีดแบ่งสนับสนุน จะทำการเก็บเซตรูปแบบนั้นไว้ในรูปแบบที่ปรากฏบ่อย หากเซตรูปแบบใดที่มีค่าสนับสนุนน้อยกว่าค่าขีดแบ่งสนับสนุนก็จะทำการตัดเซตรูปแบบนั้นทิ้งไป ขั้นตอนต่อมานำรูปแบบที่ปรากฏบ่อยมาสร้างเซตรูปแบบที่น่าจะเป็นกฎความสัมพันธ์ขนาดสองรายการ (Candidate patterns set) โดยนำเซตรูปแบบมารวมกันเป็นคู่ ๆ แล้วทำการนับค่าสนับสนุนของเซตรูปแบบทั้งสองที่ปรากฏร่วมกันจากฐานข้อมูลรายการ และทำการเปรียบเทียบค่าสนับสนุนของเซตรูปแบบขนาดสองรายการกับค่าขีดแบ่งสนับสนุน หากเซตรูปแบบมีค่าสนับสนุนไม่น้อยกว่าค่าขีดแบ่งสนับสนุนก็ทำการเก็บในรูปแบบที่ปรากฏบ่อยขนาดสองรายการนั้นไว้

ขั้นตอนต่อมาทำการสร้างเซตรูปแบบที่เป็นไปได้ขนาดสามรายการ โดยนำสมาชิกในเซตรูปแบบสองรายการสองเซตนำมารวมกัน โดยค่า prefix ในตำแหน่งที่หนึ่งจนถึงตำแหน่งรองสุดท้ายต้องมีสมาชิกที่เหมือนกัน ซึ่งค่า prefix ที่เหมือนกันต้องเป็นรูปแบบที่ปรากฏบ่อยด้วย หากสับเซตใดไม่เป็นรูปแบบที่ปรากฏบ่อยก็จะทำการตัดเซตรูปแบบที่น่าจะเป็นกฎความสัมพันธ์นั้นทิ้งไป จากนั้นทำการนับค่าสนับสนุนของเซตรูปแบบที่ได้จากฐานข้อมูล แล้วทำการเทียบกับค่าขีดแบ่งสนับสนุน โดยที่เซตรูปแบบใดที่มีค่าไม่น้อยกว่าค่าขีดแบ่งสนับสนุน ก็จะนำเซตรูปแบบนั้นไปสร้างเซตรูปแบบที่น่าจะเป็นกฎความสัมพันธ์ที่มีขนาดใหญ่กว่า และทำการหารูปแบบที่ปรากฏบ่อยขนาดต่าง ๆ จนกระทั่งไม่สามารถหารูปแบบที่ปรากฏบ่อยได้ อัลกอริทึมนี้ไม่เหมาะกับการทำงานกับข้อมูลจำนวนมาก ๆ หรือฐานข้อมูลขนาดใหญ่ เนื่องจากต้องสร้างเซตรูปแบบที่น่าจะเป็นกฎความสัมพันธ์ (Candidate patterns set) จำนวนมากและมีการอ่านข้อมูลจากฐานข้อมูลหลายครั้ง เนื่องจากทุกครั้งที่ต้องการตรวจสอบว่าเซตรูปแบบที่เป็นไปได้เป็นรูปแบบที่ปรากฏบ่อยหรือไม่และจะต้องอ่านข้อมูลจากฐานข้อมูลเพื่อหาค่าสนับสนุนในการปรากฏทุกครั้ง

ใน ปี ค.ศ. 2000 Jiawei Han, et al. ได้นำเสนออัลกอริทึมเอฟพีโกร (Frequent pattern growth: FP-growth) ในการค้นหารูปแบบที่ปรากฏบ่อย โดยไม่ต้องสร้างเซตรูปแบบที่น่าจะเป็นกฎความสัมพันธ์และทำการอ่านฐานข้อมูลรายการเพียงสองครั้ง อัลกอริทึมนี้ใช้โครงสร้างต้นไม้เอฟพีทรี (Frequent pattern tree: FP-tree) ในการสร้างเอฟพีทรีประกอบด้วย 2 ขั้นตอน ขั้นตอนแรกเริ่มต้นจากอ่านข้อมูลจากฐานข้อมูลรายการ เพื่อนับค่าสนับสนุนของเซตรูปแบบที่มีขนาดหนึ่งรายการ หากข้อมูลใดมีค่าสนับสนุนน้อยกว่าค่าขีดแบ่งสนับสนุนก็ทำการตัดรายการนั้นทิ้ง จากนั้นทำการเรียงค่าสนับสนุนของเซตรูปแบบจากมากไปน้อย และนำมาเซตรูปเก็บไว้ที่ตารางแจกแจงความถี่ (Header table) ขั้นตอนที่สองในแต่ละทรานแซกชันให้เรียงลำดับรายการตามความถี่ของรายการในตารางแจกแจงความถี่จากมากไปน้อย โดยรายการที่ไม่อยู่ในตารางแจกแจงความถี่จะถูกตัดทิ้งด้วย จากนั้นให้อ่านข้อมูลรายการอีกครั้ง เพื่อสร้างเอฟพีทรีโดยนำค่ารายการที่เรียงลำดับความถี่ในแต่ละทรานแซกชันไปใส่ในเอฟพีทรีเพื่อแสดงรูปแบบการเกิดขึ้นของข้อมูลรายการหากข้อมูลใดที่ปรากฏซ้ำ ๆ จะนับค่าสนับสนุนเพิ่มขึ้นทีละหนึ่ง ขั้นตอนในการค้นหารูปแบบที่ปรากฏบ่อยจากเอฟพีทรีมี 3 ขั้นตอน คือ (i) สร้าง

พรีฟิกพาทสับทรี (Prefix Path Subtree) ของรายการที่ต้องการ (ii) สร้างคอนดิชันนอลเบสแพทเทิน (Conditional Base Pattern) และ (iii) สร้างรูปแบบที่ปรากฏบ่อย ขั้นตอนแรกเริ่มจากการสร้างพรีฟิกพาทสับทรีของรายการที่ต้องการ โดยพิจารณาจากข้อมูลลำดับสุดท้ายของตารางแจกแจงความถี่เป็นลำดับแรก จากนั้นทำการค้นหากิ่งที่มีรายการที่ต้องการพิจารณาจากต้นไม้ แล้วนำมาสร้างพรีฟิกพาทสับทรีข้อมูลที่ได้จากการสร้างพรีฟิกพาทสับทรี จะเป็นข้อมูลรายการที่ปรากฏขึ้นร่วมกันโดยจะสร้างให้ครบทุกรายการ ขั้นตอนที่สองนำพรีฟิกพาทสับทรีมาสร้างคอนดิชันนอลเบสแพทเทิน จากนั้นสร้างต้นไม้จากพรีฟิกพาทสับทรีกับข้อมูลที่พิจารณา (Conditional FP-tree) แล้วทำการนับข้อมูลที่ปรากฏขึ้นร่วมกันทุก ๆ รายการ หากเซตรูปแบบที่ปรากฏขึ้นร่วมกันเซตใดมีค่าสนับสนุนไม่น้อยกว่าค่าขีดแบ่งสนับสนุนที่กำหนดไว้ เซตรูปแบบดังกล่าวก็จะเป็นรูปแบบที่ปรากฏบ่อย ทำการพิจารณาข้อมูลทุกเซตรูปแบบในตารางแจกแจงความถี่จนถึงข้อมูลลำดับแรกสุด (Root node) ก็จะได้รูปแบบที่ปรากฏบ่อยทั้งหมด

การค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ (Frequent-regular patterns mining)

การทำเหมืองข้อมูลรูปแบบที่ปรากฏบ่อย (Frequent patterns) โดยใช้ความถี่ของการปรากฏเป็นเกณฑ์วัดความสำคัญของเซตรูปแบบเพียงอย่างเดียวไม่สามารถตอบปัญหาในการทำงานบางอย่างได้ เช่น ปัญหาการจัดการสต็อกสินค้า เป็นต้น จึงได้มีการนำเสนอเกณฑ์ในการวัดความสำคัญของรูปแบบที่ปรากฏบ่อยเพิ่มขึ้นเพื่อตอบปัญหาในงานเหล่านั้น นั่นคือ รูปแบบที่ปรากฏอย่างสม่ำเสมอ (Regular patterns) รูปแบบที่ปรากฏอย่างสม่ำเสมอมีชื่อเรียกอีกอย่างว่า “Periodic patterns” ปัญหาการหารูปแบบที่ปรากฏอย่างสม่ำเสมอเป็นเรื่องที่เกี่ยวข้องกับช่วงเวลาจะมุ่งเน้นไปที่พฤติกรรมการปรากฏซ้ำอีกครั้งของข้อมูลรายการในฐานข้อมูลทั้งหมดหรือบางส่วนของฐานข้อมูลรายการโดยกำหนดเป็นช่วงเวลา การทำเหมืองข้อมูลรูปแบบที่ปรากฏอย่างสม่ำเสมอได้รับการศึกษาควบคู่กับการทำเหมืองข้อมูลรูปแบบลำดับ ถึงแม้ว่าการค้นหารูปแบบที่ปรากฏอย่างสม่ำเสมอจะเกี่ยวข้องอย่างใกล้ชิดกับการค้นหารูปแบบที่ปรากฏบ่อย แต่บ่อยครั้งที่รูปแบบที่ปรากฏอย่างสม่ำเสมอไม่สามารถนำไปใช้โดยตรงกับการค้นหารูปแบบที่ปรากฏบ่อยจากฐานข้อมูลรายการได้ เพราะมีเหตุผลหลักสองประการ คือ ประการที่หนึ่งรูปแบบที่เกิดแบบสม่ำเสมอจะพิจารณาอย่างใดอย่างหนึ่งระหว่างช่วงเวลาใดเวลาหนึ่ง ประการที่สองรูปแบบปรากฏแบบสม่ำเสมอไม่ได้พิจารณาจากค่าขีดแบ่งสนับสนุนซึ่งเป็นเงื่อนไขที่ใช้ในการวัดรูปแบบที่ปรากฏบ่อยทั้งหมดเท่านั้น ในปี ค.ศ. 2009 Syed Khairuzzaman Tanbeer, et al. ได้เสนอเทคนิคการค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ โดยนำการปรากฏอย่างสม่ำเสมอและค่าขีดแบ่งสนับสนุนมาใช้เป็นเกณฑ์วัดความสำคัญของเซตรูปแบบในฐานข้อมูลรายการ ปัญหาของการค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอมีการกำหนดข้อมูลดังนี้

กำหนดให้ I แทนเซตของรายการในฐานข้อมูลรายการ $I = \{i_1, i_2, i_3,..., i_n\}$ โดยที่ $n \geq 1$ กำหนดให้เซต $X = \{ i_1, i_2, i_3,..., i_l \} \subseteq I$ จะเรียก X ว่าเป็นเซตรูปแบบ (Pattern) หรือเซตรูปแบบที่มีขนาด l ให้ฐานข้อมูลรายการ $TDB=\{ t_1, t_2, t_3,..., t_n \}$ โดยฐานข้อมูลรายการมีขนาด n ทรานแซกชัน $n = |TDB|$ (ขนาดของทรานแซกชัน) แต่ละทรานแซกชัน $t_q = (tid, Y)$ เป็นคู่ลำดับ โดยที่ q เป็นค่า

หมายเลขทรานแซกชันที่ไม่ซ้ำกัน และ $Y \subseteq I$ ซึ่ง Y คือเซตรูปแบบ ถ้า $X \subseteq Y$ กล่าวได้ว่า มีเซตรูปแบบ X อยู่ภายใน t_q เขียนสัญลักษณ์แทนได้ t_q^X จะได้เซตทรานแซกชันทั้งหมดที่มีเซตรูปแบบ X ปรากฏอยู่เขียนแทนด้วย $T^X = \{ t_p^X, t_{p+1}^X, ..., t_q^X \}$ โดยที่ $p,q \in TDB$ ค่าสนับสนุนของเซตรูปแบบ X ในฐานข้อมูลรายการ $Sup_X = |T^X|$ เป็นจำนวนทรานแซกชันที่มีเซตตรายการ X ปรากฏในฐานข้อมูล ให้ t_p^X และ t_{p+1}^X เป็นทรานแซกชันที่ติดกันในเซตของ T^X และไม่มีทรานแซกชันใดที่อยู่ระหว่างกลาง t_p^X และ t_{p+1}^X จะได้ $rtt_{p+1}^X = t_{p+1}^X - t_p^X$ เป็นค่าระยะห่างการปรากฏซ้ำของทรานแซกชันที่มีเซตรูปแบบ X ปรากฏอยู่ของทรานแซกชัน ซึ่งค่าระยะห่างของทรานแซกชันที่มีเซตรูปแบบ X ปรากฏตัวแรกและตัวสุดท้ายเขียนแทนด้วย $fr^X = t_1^X$ และ $lr^X = |TDB| - t_{|T^X|}^X$ ตามลำดับ เซตของค่าระยะห่างทรานแซกชันทั้งหมดที่มีเซตรูปแบบ X ปรากฏอยู่เขียนแทนด้วย $RTT^X = \{ fr^X, rtt_2^X, rtt_3^X, ..., rtt_{|T^X|}^X, lr^X \}$ โดยค่าระยะการปรากฏขึ้นของเซตรูปแบบ X ที่นำมาใช้ ได้จากสมการ $r^X = max(fr^X, rtt_2^X, rtt_3^X, ..., rtt_{|T^X|}^X, lr^X)$ โดยเซตรูปแบบที่มีเซต X ปรากฏอยู่จะเรียกว่า รูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอได้ต้องผ่านเงื่อนไขสองข้อ ดังนี้

1. ความสม่ำเสมอในการปรากฏซ้ำของเซตรูปแบบต้องไม่เกินค่าขีดแบ่งความสม่ำเสมอ (σ_r) ที่ผู้ใช้กำหนด

2. ค่าสนับสนุนของเซตรูปแบบต้องไม่น้อยกว่าค่าขีดแบ่งสนับสนุนที่ผู้ใช้กำหนด

ดังนั้น ปัญหาการค้นหาของรูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ คือ การค้นหาเซตรูปแบบที่ผ่านเงื่อนไขทั้งสองจากฐานข้อมูลรายการ โดยที่ผู้ใช้ต้องเป็นผู้กำหนดค่าขีดแบ่งสนับสนุนและค่าขีดแบ่งความสม่ำเสมอที่เป็นเกณฑ์ในการทดสอบ โดยผู้ใช้อาจจะกำหนดโดยคิดเป็นอัตราร้อยละของขนาดฐานข้อมูลรายการ

อย่างไรก็ตามผู้ใช้ต้องกำหนดเกณฑ์ค่าขีดแบ่งสนับสนุน ซึ่งการกำหนดค่าขีดแบ่งสนับสนุนให้ขีดแบ่งสนับสนุนให้เหมาะสมกับข้อมูลรายการนั้นทำได้ยาก เพราะเมื่อกำหนดค่าน้อยเกินไปก็จะมีการสร้างรูปแบบที่ปรากฏบ่อยจำนวนมาก ซึ่งใช้เวลาและพื้นที่ในการจัดเก็บข้อมูลจำนวนมาก อีกทั้งผลลัพธ์กฎความสัมพันธ์ที่ได้อาจไม่น่าสนใจ หากกำหนดค่ามากเกินไปก็ทำให้การสร้างรูปแบบที่ปรากฏบ่อยจำนวนน้อยหรือไม่มีการสร้างเลย จากเหตุการณ์ดังกล่าวจึงทำให้เกิดงานวิจัยในการทำเหมืองข้อมูลเพื่อค้นหารูปแบบปรากฏบ่อยสูงสุดเคอันดับแรกบนฐานข้อมูลรายการโดยไม่ต้องกำหนดค่าขีดแบ่งสนับสนุน

การค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรก (Top-k frequent patterns mining)

การค้นหารูปแบบที่ปรากฏบ่อย (Frequent Patterns Mining) จะมีปัญหาในการที่ผู้ใช้ต้องทำการกำหนดค่าขีดแบ่งสนับสนุน (Support Threshold) ซึ่งเป็นเรื่องยากสำหรับผู้ใช้เพราะผู้ใช้ส่วนใหญ่มักจะไม่มีประสบการณ์ในการพิจารณาค่าขีดแบ่งสนับสนุนที่เหมาะสมกับข้อมูล เนื่องจากหากกำหนดค่าขีดแบ่งสนับสนุนน้อยเกินไปก็จะทำให้เกิดการสร้างรูปแบบที่ปรากฏบ่อยจำนวนมาก ทำให้ใช้เวลาและพื้นที่ในจัดเก็บข้อมูลจำนวนมาก รูปแบบที่ปรากฏบ่อยที่ได้อาจไม่น่าสนใจ หากผู้ใช้มีการกำหนดค่าขีดแบ่งสนับสนุนมากเกินไปก็จะทำให้รูปแบบที่ปรากฏบ่อยมีจำนวนน้อย ซึ่งทำให้การหาผลลัพธ์ของกฎความสัมพันธ์ไม่มีประสิทธิภาพ และผลลัพธ์ที่ต้องการอาจถูกซ่อนอยู่ในฐานข้อมูลรายการ จากเหตุการณ์นี้ทำให้เกิดกระบวนการที่ใช้การกำหนดจำนวนผลลัพธ์ที่ต้องการแทนการกำหนด

ค่าขีดแบ่งสนับสนุน ซึ่งวิธีการดังกล่าวเรียกว่า “Top-k” ซึ่งจะกำหนดจำนวนผลลัพธ์ที่มีค่าความถี่ในการปรากฏสูงสุดเคอันดับแรกในฐานข้อมูลรายการ ซึ่งวิธีการนี้ช่วยให้ผู้ใช้ไม่ต้องกำหนดค่าขีดแบ่งสนับสนุน ทำให้สะดวกต่อการใช้งาน ลดปัญหาความยุ่งยากในการกำหนดค่าขีดแบ่งสนับสนุนและผลลัพธ์ที่ได้ตรงกับความต้องการของผู้ใช้มากยิ่งขึ้น

นิยามที่ 2.3 เซตรูปแบบใด ๆ ที่จะเป็นรูปแบบที่ปรากฏบ่อยเคอันดับแรกได้ จะต้องมีรูปแบบที่ปรากฏบ่อยที่มีค่าสนุนมากกว่าไม่เกิน *k-1* เซต

ต่อมาได้มีการนำเสนอวิธีการค้นหารูปแบบที่ปรากฏบ่อยสูงสุด N เซตแรก (Yin-Ling Cheung, et al., 2002) โดยมีการค้นหารูปแบบที่ปรากฏบ่อยจำนวน N ที่มีค่าสนันสนุนสูงสุด รูปแบบที่ปรากฏบ่อยมีความยาวตั้งแต่1 จนถึงความยาวสูงสุดและ N คือจำนวนที่ผลลัพธ์ที่ต้องการ (k patterns set) โดยผู้ใช้เป็นผู้กำหนด โดยมีสามอัลกอริทึมที่ถูกนำเสนอในการค้นหารูปแบบที่ปรากฏบ่อย คือ LOOPBACK, BOLB และ BOMO ทั้งสามอัลกอริทึมได้ดัดแปลงมาจากวิธีเอฟพีทรี อัลกอลิทึม BOMO ประกอบไปด้วยสองขั้นตอน โดยขั้นตอนแรกจะสร้างเอฟพีทรี ที่มีความสมบูรณ์ของรายการทั้งหมดในฐานข้อมูลเพื่อหาเกณฑ์ค่าขีดแบ่งสนับสนุนของรูปแบบที่ปรากฏบ่อย N ลำดับแรก จากนั้นก็จะทำการค้นหา รูปแบบที่ปรากฏบ่อย ในระหว่างขั้นตอนการค้นหาค่าสนับสนุนของเซตรูปแบบทั้งหมดจะเพิ่มขึ้นโดยการพิจารณาจากค่าขีดแบ่งสนับสนุนของรูปแบบที่ปรากฏบ่อยที่สุด N ลำดับแรก ซึ่งถูกใช้เป็นค่าสำหรับตัดรูปแบบที่ปรากฏบ่อยที่มีค่าสนับสนุนน้อยกว่าค่าขีดแบ่งสนับสนุนใน N ลำดับแรก อัลกอลิทึม LOOPBACK จะสร้าง FP-tree และกำหนดค่าขีดแบ่งสนับสนุนได้จากค่าสนับสนุนจากรูปแบบที่ปรากฏบ่อย N ลำดับแรก จำนวนของรูปแบบที่ปรากฏบ่อยมีค่าน้อยกว่า N ลำดับ จะทำการสร้างเอฟพีทรีเพิ่มจากการหาค่าสนับสนุนที่มีขนาดน้อยลงมาตามลำดับ เพื่อให้มีจำนวนรูปแบบที่ปรากฏบ่อยมากขึ้นเพื่อใช้ในขั้นตอนการค้นหาถัดไป อัลกอลิทึม BOLB เป็นวิธีการนำ BOMO และ LOOPBACK มาใช้รวมกัน โดยจะเหมือนอัลกอลิทึม BOMO โดยสร้างเอฟพีทรีสมบูรณ์เพียงครั้งเดียว และขั้นตอนการทำเหมืองข้อมูลได้ประยุกต์จากเทคนิคของ LOOPBACK โดย Jianyong Wang, et al. ได้เสนอวิธีการค้นหารูปแบบที่ปรากฏบ่อยแบบปิดเคอันดับแรก (TFP) โดยมีความยาวของเซตรูปแบบไม่น้อยกว่า min_l (Jiawei Han et al., 2002; Jianyong Wang et al., 2005) ค่าเคเป็นจำนวนของรูปแบบที่ปรากฏบ่อยแบบปิดที่ต้องการค้นหาและ min_l เป็นความยาวขั้นต่ำของรูปแบบที่ปรากฏบ่อยแบบปิด อัลกอริทึม TFP จะเริ่มค่าขีดแบ่งสนับสนุนที่ 0 โดยจะทำการตัดข้อมูลรายการที่มีการปรากฏน้อยกว่า min_l ออก จากนั้นนำข้อมูลรายการที่ตัดแล้วมาสร้างเอฟพีทรีเพื่อเพิ่มค่าขีดแบ่งสนับสนุนและใช้เป็นเกณฑ์ในการตัดเอฟพีทรี ซึ่งขั้นตอนในการสร้างเอฟพีทรีเพื่อหาค่าขีดแบ่งสนับสนุนตัวสุดท้ายใช้เวลานานและการใช้ค่าขีดแบ่งสนับสนุนตัวนี้ในการตัดเอฟพีทรีถ้าฐานข้อมูลมีทรานแซกชันจำนวนมากและมีรูปแบบยาว ๆ นอกจากนี้ TFP อัลกอรึทึม (Petre Tzvetkov et al., 2005) ยังมีการจัดเก็บเซตรูปแบบที่น่าจะเป็นไปได้ (Candidate patterns set) เพื่อใช้ตรวจสอบว่ารูปแบบที่ปรากฏบ่อยที่ได้เป็นรูปแบบที่ปรากฏบ่อยแบบปิดจริง

การค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอ (Top-k frequent-regular patterns mining)

เพื่อเพิ่มประสิทธิภาพในการวัดความสำคัญของรูปแบบปรากฏบ่อยสุดเคอันดับแรกให้สามารถตอบปัญหาในงานบางประเภทที่รูปแบบที่ปรากฏบ่อยไม่สามารถตอบปัญหาได้ จึงมีการเพิ่มเกณฑ์ค่าขีดแบ่งความสม่ำเสมอในการปรากฏซ้ำของรูปแบบที่ปรากฏบ่อยเคอันดับแรก

นิยามที่ 2.4 เซตรูปแบบใด ๆ ที่จะเป็นรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏสม่ำเสมอได้ จะต้องมีคุณสมบัติ 2 ข้อดังนี้ (i) มีค่าการปรากฏอย่างสม่ำเสมอไม่เกินค่าขีดแบ่งความสม่ำเสมอ (σ_r) (ii) มีรูปแบบที่ปรากฏบ่อยที่มีค่าสนุนมากกว่าไม่เกิน *k-1* เซต

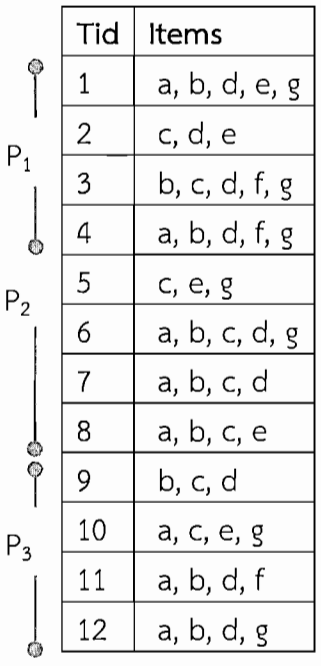
ในปีค.ศ. 2009 Komate Amphawan et al. ได้นำเสนออัลกอรึทึม MTKPP ในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอจากฐานข้อมูลรายการโดยผู้ใช้ไม่ต้องกำหนดค่าขีดแบ่งสนับสนุนโดยใช้การอ่านข้อมูลจากฐานข้อมูลเพียงครั้งเดียวทำให้ประหยัดเวลา โดยค้นหาเซตรูปแบบที่มีค่าสนับสนุนสูงสุดเคอันดับแรก โดยเรียงค่าสนับสนุนเซตรูปแบบจากมากไปน้อย และใช้การค้นหาที่ดีที่สุด (Best first search) ในการค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ และใช้โครงสร้างลิงค์ลิสต์ในการจัดเก็บข้อมูล ทำให้ใช้เวลาในการประมวลผลและใช้พื้นที่ในการจัดเก็บข้อมูลน้อยลง แต่ในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอที่มีความยาวตั้งแต่ 2 รายการขึ้นไป ยังใช้เวลาในการเปรียบเทียบการปรากฏร่วมกันของรายการในแต่ละทรานแซกชันยังใช้เวลาค่อนข้างมาก

ใน ปี ค.ศ. 2011 Komate Amphawan et al. จึงได้นำเสนอการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอโดยใช้การแบ่งฐานข้อมูลรายการและการประมาณค่าสนับสนุน (TKRIMPE) เพื่อปรับปรุงเวลาในการค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอโดยการแบ่งฐานข้อมูลรายการออกเป็นส่วน ๆ ด้วยค่าขีดแบ่งความสม่ำเสมอ โดยมีสมการการแบ่งฐานข้อมูลรายการดังนี้

สมการการแบ่งฐานข้อมูลรายการ $P_n = |TDB|/\sigma_r$ โดยฐานข้อมูลจะถูกแบ่งออกเป็นพาทิชันและจัดเก็บข้อมูลรายการที่ปรากฏในทรานแซกชันตามจำนวน P_n โดยกำหนดให้เซตรูปแบบ X ปรากฏใน P_m เขียนแทนด้วย $T^X_m=\{t_q \mid X \subseteq t_q; t_q \subseteq P_m\}$ และ T^X เป็นทรานแซกชันที่มีเซตรูปแบบ X ปรากฏ $T^X = \{ T^X_1,..., T^X_{pn} \}$ ให้ S^X_m เป็นค่าสนับสนุนของเซตรูปแบบที่มีเซตรูปแบบ X ปรากฏในฐานข้อมูลรายการพาทิชัน m เขียนแทนด้วย $S^X_m= |T^X_m|$ จำนวนค่าสนับสนุนของเซตรูปแบบที่มีเซต X ปรากฏทั้งหมดในฐานข้อมูลเขียนแทนด้วย $S^X = \sum_{m=1}^{pn} S^x_m$

ตัวอย่าง โดยฐานข้อมูลรายการมีจำนวนทรานแซกชันทั้งหมด 12 รายการและกำหนดค่าขีดแบ่งความสม่ำเสมอไว้ที่ 4 พิจารณารายการ a จะปรากฏในทรานแซกชันหมายเลข {1, 4, 6, 7, 8, 10,11,12} ($T^a = \{t_1, t_4, t_6, t_7, t_8, t_{10}, t_{11}, t_{12}\}$)

เมื่อทำการแบ่งช่วงของฐานข้อมูลรายการด้วยค่าขีดแบ่งความสม่ำเสมอ จะแบ่งฐานข้อมูลรายการออกเป็น 3 พาทิชันดังภาพที่ 2-1 โดยทรานแซกชัน {1, 4} จะถูกเก็บใน T_{a1} ซึ่งเป็นทรานแซกชันที่มีรายการ a ปรากฏ เซตทรานแซกชัน {6, 7,8} และ {10,11,12} จะถูกเก็บใน T_{a2} และ T_{a3} ตามลำดับ และแสดงทรานแซกชันที่รายการ a ปรากฏทั้งหมด Ta = {{1, 4}, {6, 7, 8},{10,11,12}} ค่าสนับสนุนทั้งหมดของรายการ a เท่ากับ $S^a = 2 + 3 + 3 = 8$



Tid	Items
1	a, b, d, e, g
2	c, d, e
3	b, c, d, f, g
4	a, b, d, f, g
5	c, e, g
6	a, b, c, d, g
7	a, b, c, d
8	a, b, c, e
9	b, c, d
10	a, c, e, g
11	a, b, d, f
12	a, b, d, g

ภาพที่ 2-1 แบ่งช่วงของฐานข้อมูลรายการด้วยค่าขีดแบ่งความสม่ำเสมอ

ซึ่งเทคนิคในการแบ่งฐานข้อมูลรายการจะเปรียบเทียบการปรากฏร่วมกันของเซตรูปแบบออกเป็นส่วน ๆ จะสามารถลดปริมาณการเปรียบเทียบเซตรูปแบบที่ปรากฏร่วมกันได้ ทำให้เวลาในการประมวลผลลัพธ์มีประสิทธิภาพมากขึ้น และเทคนิคการประมาณค่าสนับสนุนจะช่วยกำจัดการเปรียบเทียบเซตรูปแบบในส่วนที่ไม่จำเป็นออกไปได้บางส่วน แต่ถึงอย่างไรก็ตามผลลัพธ์ของรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอจากอัลกอริทึม MTKPP จะมีผลลัพธ์ของเซตรูปแบบที่มีขนาดเป็นหนึ่งรายการ (1-pattern) จำนวนไม่น้อยกว่า 30 เปอร์เซ็นต์ของผลลัพธ์ทั้งหมด ซึ่งเซตรูปแบบที่มีขนาดรายการเป็นหนึ่ง จะไม่สามารถนำไปสร้างกฎความความสัมพันธ์และไม่สามารถนำผลลัพธ์ไปวิเคราะห์หาความรู้จากผลลัพธ์ดังกล่าวได้ ทางผู้วิจัยจึงได้นำเสนออัลกอริทึม ETFRP เพื่อมาแก้ปัญหาดังกล่าว โดยมีขนาดของเซตรูปแบบที่มีขนาดตั้งแต่สองรายการขึ้นไป เพื่อให้ได้ผลลัพธ์รูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอที่นำไปสร้างกฎความสัมพันธ์และวิเคราะห์หาความรู้ต่าง ๆ ได้ ซึ่งจะกล่าวถึงขั้นตอนการทำงานของอัลกอริทึม ETFRP ในบทที่ 3 ต่อไป

บทที่ 3
วิธีดำเนินการวิจัย

จากบทก่อนหน้าได้ทราบแล้วว่าการกำหนดค่าขีดแบ่งสนับสนุน (ที่มีค่าที่เหมาะสม) เพื่อทำการค้นหารูปแบบปรากฏบ่อยและปรากฏอย่างสม่ำเสมอเป็นเรื่องที่ทำได้ยาก จากปัญหาดังกล่าวจึงได้มีนักวิจัยได้นำเสนอการกำหนดจำนวนผลลัพธ์ที่ต้องการ (k) แทนการกำหนดค่าขีดแบ่งสนับสนุน ซึ่งจะทำให้ปัญหาการค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ (ภายใต้ค่าขีดแบ่งสนับสนุนและค่าขีดแบ่งความสม่ำเสมอ) ได้กลายเป็นการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอที่ซึ่งจะคืนค่าผลลัพธ์เป็นเซตรูปแบบตามจำนวนที่ผู้ใช้กำหนด แต่อย่างไรก็ตามปัญหาการค้นหารูปแบบในลักษณะดังกล่าวมักจะคืนค่าผลลัพธ์เป็นเซตรูปแบบที่มีขนาดเล็ก (เซตรูปแบบที่ประกอบไปด้วยรายการเดียว, 1-pattern) ซึ่งจะทำให้ผู้ที่ต้องการวิเคราะห์ข้อมูลไม่สามารถสกัดความสัมพันธ์ระหว่างรายการหรือไม่สามารถค้นหาองค์ความรู้ที่น่าสนใจได้ จากปัญหาข้างต้น งานวิจัยนี้จึงมีความคิดที่จะปรับเปลี่ยนการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอโดยการกำหนดให้ขนาดของเซตรูปแบบที่เป็นผลลัพธ์จะต้องประกอบด้วยรายการอย่างน้อย 2 รายการขึ้นไป ซึ่งจะทำให้สามารถวิเคราะห์ความสัมพันธ์ของรายการต่าง ๆ และยังนำมาสร้างกฎความสัมพันธ์ของข้อมูลได้อีกด้วย ดังนั้น ในบทนี้จะกล่าวถึงนิยามของผลลัพธ์ที่งานวิจัยนี้จะทำการพิจารณาและกล่าวถึงขั้นตอนวิธีที่นำเสนอสำหรับการค้นหาผลลัพธ์ในลักษณะดังกล่าว

การค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอ

จากที่ได้กล่าวข้างต้น การค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอมักจะคืนค่าผลลัพธ์เป็นเซตรูปแบบที่มีขนาดเล็ก (เซตรูปแบบที่ประกอบไปด้วยรายการ 1 รายการ) เป็นจำนวนมาก เพื่อที่จะยืนยันข้อเท็จจริงข้างต้น งานวิจัยนี้จึงได้ทำการพิจารณาผลลัพธ์จากอัลกอริทึม MTKPP (Komate Amphawan, et al., 2009) ที่ซึ่งเป็นอัลกอริทึมสำหรับค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอ โดยทำการพิจารณาผลลัพธ์จากแฟ้มข้อมูลทั้งสิ้น 5 แฟ้มข้อมูลที่เป็นที่รู้จักกันอย่างแพร่หลาย (สามารถดาวน์โหลดได้ที่ http://fimi.ua.ac.be/data/) ดังนี้

1. Chess ประกอบด้วยข้อมูลทั้งสิ้น 3,196 ทรานแซกชัน
2. Mushroom ประกอบด้วยข้อมูลทั้งสิ้น 8,124 ทรานแซกชัน
3. Conect ประกอบด้วยข้อมูลทั้งสิ้น 67,557 ทรานแซกชัน
4. Retail ประกอบด้วยข้อมูลทั้งสิ้น 88,162 ทรานแซกชัน
5. T10I4D100K ประกอบด้วยข้อมูลทั้งสิ้น 100,000 ทรานแซกชัน

โดยในการทดสอบจะทำการกำหนดค่าขีดแบ่งความสม่ำเสมอให้มีค่าเป็น 2, 5 และ 10 เปอร์เซ็นของจำนวนทรานแซกชันทั้งหมดในชุดข้อมูล และทำการกำหนดจำนวนผลลัพธ์ที่ต้องการ (k) เป็น100, 200, 500, 1000, 1,500 และ 2,000 ตามลำดับ โดยในการพิจารณาผลลัพธ์ที่ได้จากการพิจารณาอัลกอริทึม MTKPP จะพิจารณาที่จำนวนรูปแบบที่มีขนาดเล็ก ดังตารางที่ 3-1-3-5

ตารางที่ 3-1 เซตรูปแบบขนาดเล็กที่ได้จากอัลกอรึทึม MTKPP ดำเนินการกับแฟ้มข้อมูล Chess

ขนาดเซตรูปแบบขนาดเล็ก Chess(%)			
จำนวน k	ค่าขีดแบ่งความสม่ำเสมอ		
	10%	20%	30%
100	11%	11%	11%
200	6%	6%	6%
500	3%	3%	3%
1000	2%	2%	2%
1500	2%	2%	2%
2000	1%	1%	1%

ตารางที่ 3-2 เซตรูปแบบขนาดเล็กที่ได้จากอัลกอรึทึม MTKPP ดำเนินการกับแฟ้มข้อมูล Mushroom

ขนาดเซตรูปแบบขนาดเล็ก Mushroom(%)			
จำนวน k	ค่าขีดแบ่งความสม่ำเสมอ		
	10%	20%	30%
100	11%	11%	11%
200	8%	8%	8%
500	5%	5%	5%
1000	3%	3%	3%
1500	2%	2%	2%
2000	2%	2%	2%

ตารางที่ 3-3 เซตรูปแบบขนาดเล็กที่ได้จากอัลกอรึทึม MTKPP ดำเนินการกับแฟ้มข้อมูล Connect

ขนาดเซตรูปแบบขนาดเล็ก Connect(%)			
จำนวน k	ค่าขีดแบ่งความสม่ำเสมอ		
	2%	5%	10%
100	10%	10%	10%
200	5%	5%	5%
500	3%	3%	3%
1000	2%	2%	2%
1500	2%	2%	2%
2000	1%	1%	1%

ตารางที่ 3-4 เซตรูปแบบขนาดเล็กที่ได้จากอัลกอรึทึม MTKPP ดำเนินการกับแฟ้มข้อมูล T10I4D100K

ขนาดเซตรูปแบบขนาดเล็ก T10I4D100K(%)			
จำนวน k	ค่าขีดแบ่งความสม่ำเสมอ		
	2%	5%	10%
100	100%	100%	100%
200	100%	100%	100%
500	89%	89%	89%
1000	56%	56%	56%
1500	41%	41%	41%
2000	31%	31%	31%

ตารางที่ 3-5 เซตรูปแบบขนาดเล็กที่ได้จากอัลกอรึทึม MTKPP ดำเนินการกับแฟ้มข้อมูล Retail

ขนาดเซตรูปแบบขนาดเล็ก Retail(%)			
จำนวน k	ค่าขีดแบ่งความสม่ำเสมอ		
	2%	5%	10%
100	38%	43%	43%
200	39%	45%	45%
500	35%	42%	41%
1000	35%	39%	40%
1500	35%	39%	41%
2000	35%	37%	39%

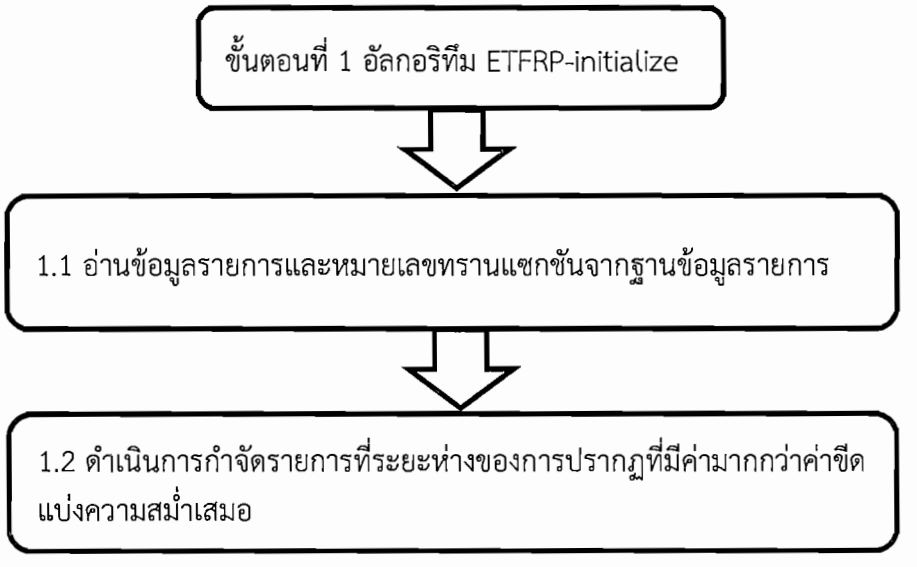
จากตารางที่ 3-1 ถึง 3-5 เป็นการแสดงเปอร์เซ็นของจำนวนรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอที่มีขนาดเป็นหนึ่งรายการที่ถูกค้นหาจากอัลกอริทึม MTKPP เมื่อสังเกตพบว่าผู้ใช้ระบุจำนวนผลลัพธ์ (k) ที่จำนวน 100 ในตารางที่ 3-1 ถึง 3-3 จะพบเซตรูปแบบที่มีขนาดหนึ่งรายการประมาณ 10 เปอร์เซ็นต์ขึ้นไป แต่เมื่อค่า k เพิ่มขึ้นเซตรูปแบบที่เป็นหนึ่งรายการก็เริ่มลดลง เนื่องจากเมื่อค่า k มากขึ้นโอกาสที่รายการที่จะเกิดขึ้นพร้อมกันก็จะมากขึ้นตามไปด้วย เพราะมีจำนวนรายการที่นำมาเปรียบเทียบมากขึ้น เมื่อค่า k มีค่าน้อย ค่า k จะมีจำนวนน้อยกว่าจำนวนเซตรายการที่ปรากฏอย่างสม่ำเสมอ ทำให้ในเซตรายการ top-k ลิสต์ถูกบรรจุด้วยรายการที่เกิดขึ้นเดี่ยว ๆ เพราะมีค่าสบับสนุนจำนวนมาก ทำให้การเปรียบไม่พบรายการที่ปรากฏขึ้นร่วมกัน ในตารางที่ 3-4 และตารางที่ 3-5 จะเห็นได้ว่าอัลกอริทึม MTKPP มีการแสดงผลลัพธ์ที่มีขนาดของเซตรูปแบบเป็นหนึ่งรายการจำนวนมาก สังเกตได้จากจำนวนเปอร์เซ็นของผลลัพธ์ เมื่อผู้ใช้ระบุค่า k ขนาด 100 และ 200 จะแสดงผลลัพธ์เซตรูปแบบที่มีขนาดหนึ่งรายการ 100 เปอร์เซ็น สำหรับไฟล์ T10I4D100K และสำหรับค่า k อื่น ๆ ก็แสดงจำนวนผลลัพธ์ที่มีขนาดเป็นหนึ่งรายการไม่ต่ำกว่าร้อยละ 30 ของจำนวนเซตรูปแบบทั้งหมด ซึ่งเซตรายการที่มีขนาดเป็นหนึ่งรายการจะไม่สามารถวิเคราะห์ความสัมพันธ์ของการเกิดขึ้นร่วมกันของรายการและไม่สามารถนำไปวิเคราะห์หาความรู้อื่น ๆ จากเซตรูปแบบที่ค้นหาได้ ดังนั้น งานวิจัยนี้จึงมีความคิดที่จะปรับเปลี่ยนการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอโดยการกำหนดให้ขนาดของรูปแบบที่เป็นผลลัพธ์จะต้องประกอบด้วยรายการอย่างน้อย 2 รายการขึ้นไปที่ซึ่งผลลัพธ์ที่จะทำการค้นหาจะสามารถนิยามได้ดังนิยามที่ 3.1

นิยามที่ 3.1 เซตรูปแบบใด ๆ จะเป็นรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏแบบสม่ำเสมอก็ต่อเมื่อรูปแบบนั้น ๆ มีคุณสมบัติดังนี้: (i) มีค่าสนับสนุนการปรากฏสูงสุดเคอันดับแรก (ii) มีค่าการปรากฏอย่างสม่ำเสมอน้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอ (σ_r) (iii) มีขนาดของเซตรูปแบบตั้งแต่สองรายการขึ้นไป

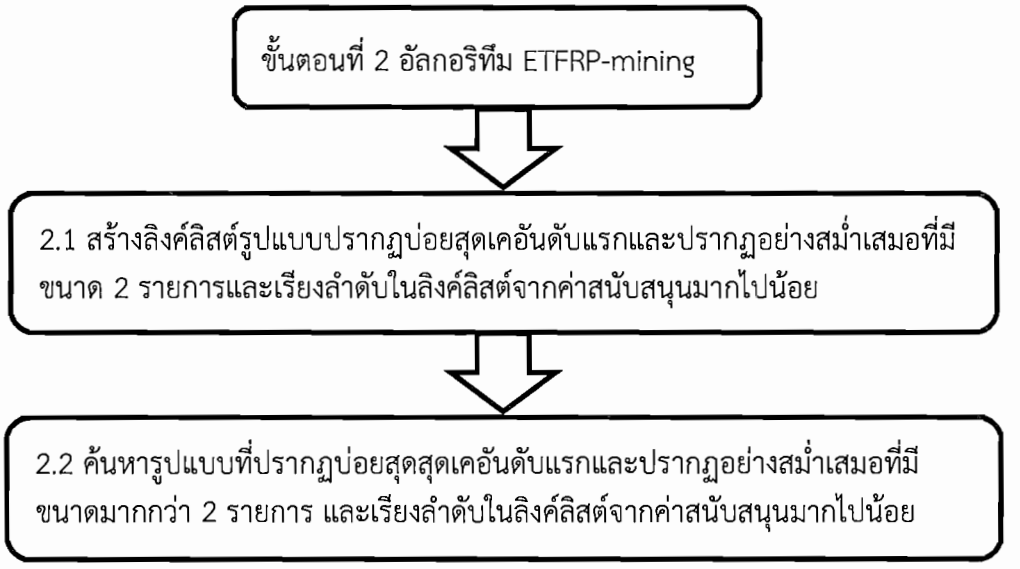
จากนิยามข้างต้น งานวิจัยนี้จะทำการค้นหารูปแบบที่น่าสนใจภายใต้การกำหนดจำนวนผลลัพธ์ที่ต้องการ (k) และค่าขีดแบ่งความสม่ำเสมอ (σ_r) โดยผลลัพธ์ที่ได้จะมีคุณสมบัติดังนิยามที่ 3.1

ขั้นตอนวิธีที่นำเสนอ

จากนิยามของผลลัพธ์ที่ต้องการภายใต้นิยาม 3.1 งานวิจัยนี้ได้นำเสนอขั้นตอนวิธีเพื่อทำการค้นหาเซตรูปแบบดังกล่าวที่มีชื่อว่า ETFRP (Enhancing of Mining Top-k Frequent-regular Patterns) ที่จะประกอบไปด้วย 2 ขั้นตอนหลักคือ 1) ETFRP-initialize เป็นขั้นตอนการอ่านข้อมูลจากฐานข้อมูลเพื่อทำการพิจารณารายการเดี่ยว ๆ (single items) ที่ปรากฏอย่างสม่ำเสมอภายใต้การกำหนดค่าขีดแบ่งความสม่ำเสมอ และ 2) ETFRP-mining ทำการค้นหาผลลัพธ์ที่มีคุณสมบัติตรงกับนิยาม 3.1 โดยการพิจารณาผลลัพธ์ที่ได้จากขั้นตอน ETFRP-initialize ตามลำดับ โดยแต่ละขั้นตอนมีการทำงานดังแสดงในภาพที่ 3-1 และ 3-2 ตามลำดับ



ภาพที่ 3-1 การค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอขั้นตอนที่หนึ่ง



ภาพที่ 3-2 การค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอขั้นตอนที่สอง

กระบวนการทำงานของ ETFRP-initailization

ETFRP-initailization เป็นขั้นตอนแรกที่ดำเนินการกับรายการเดี่ยว ๆ เพื่อค้นหาเซตรายการที่ปรากฏย่างสม่ำเสมอ โดยเริ่มอ่านข้อมูลจากฐานข้อมูลรายการมาเก็บไว้ในบัฟเฟอร์ โดยในบัฟเฟอร์มีการจัดเก็บข้อมูลดังนี้ (i) ชื่อรายการ (ii) ค่าสนับสนุน (iii) ทรานแซกชันไอดี (iiii) ค่าความสม่ำเสมอในการปรากฏซ้ำของรายการที่เป็นรายการเดี่ยว ๆ จะเป็นรายการที่ปรากฏอย่างสม่ำเสมอได้นั้นค่าความสม่ำเสมอในการปรากฏซ้ำของรายการ ต้องมีค่าน้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด หากรายการใดมีค่าความสม่ำเสมอเกินค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนดข้อมูลรายการนั้นจะถูกลบทิ้ง และเมื่อได้เซตรายการที่มีการปรากฏอย่างสม่ำเสมอทั้งหมดแล้ว จะทำการเรียงเซตรายการตามลำดับค่าสนับสนุนจากมากไปน้อยตามขั้นตอนที่ 1 ดังนี้

ขั้นตอนที่ 1: ETFRP-initailization

Input: A transactional database *TDB*, a number desired pattern *k*, and a regularity threshold σ_r

Output: A list of sorted items with regular appearance

(1) create and initialize a buffer for all items

(2) for each transaction t_q in *TDB*

(3) for each item i_x in transaction t_q

(4) update the value of support, regularity and collect t_q in the tidset of i_x's entry

(5) for each item i_y in the buffer
(6) compute lr_y^i and r_y^i
(7) if $r_y^i \leq \sigma_r$
(8) remove i_y out of the buffer
(9) sort all items in the buffer by support descending order

กระบวนการทำงานของ ETFRP-mining

ขั้นตอนที่สอง ETFRP-mining เป็นขั้นตอนในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอ (Top-k list) ที่มีขนาดสองรายการขึ้นไป โดยเริ่มต้นจากการนำเซตรายการที่ปรากฏอย่างสม่ำเสมอมาสร้างรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอที่มีขนาดสองรายการ โดยใช้การค้นหาที่ดีที่สุด (Best first search) ในการพิจารณารายการแต่ละคู่จากเซตรูปแบบที่ปรากฏอย่างสม่ำเสมอ ซึ่งจะพิจารณาจากรายการคู่ที่มีค่าสนับสนุนสูงสุดก่อน โดยนำทรานแซกชันไอดีแต่ละรายการที่จับคู่กันมาอินเตอร์เซคชันกัน ซึ่งจะได้ข้อมูลหมายเลขทรานแซคชันที่ปรากฏร่วมกันของรายการทั้งสอง และคำนวณค่าขีดแบ่งความสม่ำเสมอของเซตรายการคู่นั้น หากรายการคู่ใดมีค่าความสม่ำเสมอมากกว่าค่าขีดแบ่งความสม่ำเสมอ ($\geq \sigma_r$) ที่ผู้ใช้กำหนด หรือมีค่าสนับสนุนน้อยกว่าค่าสนับสนุนเคอันดับแรกก็จะถูกตัดข้อมูลทิ้งไป หากพบรายการคู่ใดที่มีคุณสมบัติตามนิยามที่ 3.1 ก็จะทำการเพิ่มข้อมูลเซตรูปแบบเข้าไปใน Top-k ลิสต์ตามลำดับของค่าสนับสนุนของรายการคู่นั้นโดยเรียงลำดับจากมากไปน้อยมีการเพิ่มใน Tok-k ลิสต์ 2 รูปแบบ

1. เซตรูปแบบคู่นั้นมีค่าสนับสนุนมากกว่าเซตรายการตัวสุดท้ายลำดับที่เคของ Top-k ลิสต์ เมื่อเพิ่มเซตรายการคู่นั้นแล้ว ลำดับที่เกินเคอันดับของเซตรายการตัวสุดท้ายก็จะถูกลบออกจาก Top-k ลิสต์

2. เซตรูปแบบคู่นั้นมีค่าสนับสนุนมีค่าสนับสนุนเท่ากับค่าสนับสนุนข้อมูลตัวสุดท้ายของ Top-k ลิสต์ เซตรูปแบบคู่นั้นจะนำมาต่อท้ายใน Top-k ลิสต์ถึงแม้ข้อมูลใน Top-k ลิสต์จะเกินลำดับที่เคแล้วก็ตาม

จากนั้นทำการวนซ้ำจนครบทุกคู่รายการก็จะได้ Top-k ลิสต์ของเซตรายการที่ปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอที่มีขนาด 2 รายการ ขั้นตอนต่อมาทำการค้นหาเซตรายการที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอที่มีขนาดมากกว่า 2 รายการ โดยพิจารณาเซตรูปแบบขนาด 2 รายการที่อยู่ใน Top-k ลิสต์ ภายใต้สองเงื่อนไขต่อไปนี้ (i) ขนาดของเซตรูปแบบทั้งสองที่นำมาพิจารณาจะต้องเท่ากัน (ii) เซตรูปแบบทั้งสองที่นำมาพิจารณาจะต้องมีรายการด้านหน้าเหมือนกัน n-1 ตัว เมื่อ n คือ ขนาดของเซตรูปแบบ (โดยแต่ละเซตรูปแบบจะต้องมีสมาชิกด้านหน้าเหมือนกันทุกรายการยกเว้นสมาชิกตัวสุดท้าย) เมื่อเซตรูปแบบที่นำมาพิจารณามีคุณสมบัติทั้งสองข้อข้างต้นจะนำมาพิจารณาการปรากฏขึ้นร่วมกัน เพื่อสร้างรูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอที่มีขนาดมากกว่า 2 รายการ โดยจะเพิ่มหรือลดข้อมูลของเซตรูปแบบในลิงค์ลิสต์ด้วยค่าสนับสนุนและค่าความสม่ำเสมอของเซตรูปแบบเป็นเกณฑ์ โดยทำการวนซ้ำจนไม่พบเซตรายการคู่ใดที่เข้าเงื่อนไขดังกล่าว เมื่อดำเนินการ

ครบทุกเซตรูปแบบแล้ว จะได้รูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอที่มีขนาดตั้งแต่ 2 รายการขึ้นไป โดยแสดงขั้นตอนการทำงานดังขั้นตอนที่ 2

ขั้นตอนที่ 2: ETFRP-mining

Input: A buffer containing all single items with regular appearance, the number of desired patterns *k*, and the regularity threshold σ_r

Output: A top-*k* list containing the sorted *k* regular patterns with highest support and contains more than one items.

(1) for each item *x* in the buffer

(2) for each item *y* in the buffer

(3) sequentially intersect T^x and T^y and then collect the result in T^{xy}

(4) calculate s^{xy} and r^{xy}

(5) if $r^{xy} \leq \sigma_r$ and $s^{xy} \geq s_k$

(6) remove the entry of the k^{th} pattern out of the top-*k* list

(7) create an entry for pattern *xy* with its support, regularity and tidset and then insert into the top-*k* list by support descending order

(8) for each entry of pattern *P* in the top-*k* list

(9) for each entry of pattern *Q* in the top-*k* list

(10) if $|P| = |Q|$ and $p_1 = q_1, p_2 = q_2, \ldots, p_{|P|-1} = q_{|Q|-1}$

(11) sequentially intersect T^P and T^Q and then collect the result in T^{PQ}

(12) calculate s^{PQ} and r^{PQ}

(13) if $r^{PQ} \leq \sigma_r$ and $s^{PQ} \geq s_k$

(14) create an entry for pattern *PQ* with its support, regularity and tidset and then insert into the top-*k* list by support descending order

ตัวอย่างการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอด้วยอัลกอริทึม ETFRP

การค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอที่มีขนาดตั้งแต่สองรายการขึ้นไปของอัลกอริทึม ETFRP ใช้โครงสร้างลิงค์ลิสต์ในการจัดเก็บข้อมูล และใช้การค้นหาที่ดีที่สุด (Best first search) ในในการคัดเลือกคู่รายการเพื่อเปรียบเทียบการเกิดขึ้นร่วมกันในแต่ละทรานแซกชัน ซึ่งแต่ละโหนดของเซตรูปแบบจะมีการเก็บข้อมูลดังภาพที่ 3-3

ภาพที่ 3-3 โครงสร้างของการจัดเก็บข้อมูลรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอ

โดยที่ I แทน เซตรูปแบบที่ประกอบไปด้วยรายการตั้งแต่สองรายการขึ้นไป
F แทน ค่าสนับสนุนที่เซตรูปแบบปรากฏ
P แทน ค่าขีดแบ่งความสม่ำเสมอที่มากที่สุดในการปรากฏซ้ำของเซตรูปแบบนั้น ๆ
Tid แทน เซตของทรานแซกชันไอดีที่มีเซตรูปแบบปรากฏ

ตารางที่ 3-6 ฐานข้อมูลตัวอย่างที่ใช้ในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอ

Tid	Item
1	a, b, d, e, g
2	c, d, e
3	b, c, d, f, g
4	a, b, d, f, g
5	c, e, g
6	a, b, c, d, g
7	a, b, c, d
8	a, b, c, e
9	c, d
10	a, c, e, g
11	a, b, d, f
12	a, b, d, g

จากฐานข้อมูลตัวอย่างมีทั้งหมด 12 ทรานแซกชัน และมีรายการทั้งหมด 6 รายการ โดยกำหนดค่าขีดแบ่งความสม่ำเสมอ 40% (ประมาณ 4 ทรานแซกชัน ต้องมีรายการปรากฏอย่างน้อย 1 ครั้ง) จำนวนผลลัพธ์ที่ต้องการ (k) เท่ากับ 5 เซตรูปแบบ โดยเริ่มต้นการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากอย่างสม่ำเสมอโดยดำเนินการขั้นตอนที่ 1 ETFRP–initialization แบ่งออก

ได้เป็น 2 ขั้นตอน คือ ขั้นตอนหนึ่งอ่านข้อมูลรายการจากฐานข้อมูลรายการตัวอย่าง โดยเก็บชื่อรายการ ค่าสนับสนุน ค่าความสม่ำเสมอในการปรากฏซ้ำของรายการ หมายเลขทรานแซกชันที่รายการนั้น ๆ ปรากฏ ดังตารางที่ 3-7

ตารางที่ 3-7 การเก็บข้อมูลที่ได้จากการอ่านฐานข้อมูลรายการ

Item	Support	Max period	Tid
a	8	3	1, 4, 6, 7, 8, 10, 11, 12
b	8	3	1, 3, 4, 6, 7, 8, 11, 12
c	8	2	2, 3, 5, 6, 7, 8, 9, 10
d	9	2	1, 2, 3, 4, 6, 7, 9, 11,
e	5	3	1, 2, 5, 8, 10
f	3	7	3, 4, 11
g	7	4	1, 3, 4, 5, 6, 10, 12

การคำนวณค่าความสม่ำเสมอในการปรากฏซ้ำของรายการนั้น ๆ หาได้จากสมการ $Max\ period = max\ (fr^X, rtt^X_2, rtt^X_3, ..., rtt^X_{|T^X|}, lr^X)$ โดยที่

fr^X เท่ากับ ทรานแซกชันไอดีแรกที่รายการปรากฏลบด้วยทรานแซกชันไอดีแรกของฐานข้อมูลรายการ

rtt^X_2 เท่ากับ ทรานแซกชันไอดีตัวที่สองที่รายการปรากฏลบด้วยทรานแซกชันไอดีแรกที่รายการปรากฏ

rtt^X_3 เท่ากับ ทรานแซกชันไอดีตัวที่สามที่รายการปรากฏลบด้วยทรานแซกชันไอดีที่สองที่รายการปรากฏ

$rtt^X_{|T^X|}$ เท่ากับ ทรานแซกชันไอดีสุดท้ายที่รายการปรากฏลบด้วยทรานแซกชันไอดีรองสุดท้ายที่รายการปรากฏ

lr^X เท่ากับ ทรานแซกชันไอดีสุดท้ายที่รายการปรากฏลบด้วยทรานแซกชันไอดีสุดท้ายของฐานข้อมูลรายการ

ขั้นตอนที่สองดำเนินการกำจัดรายการที่มีค่าความสม่ำเสมอในการปรากฏซ้ำของรายการที่มีค่ามากกว่าค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด(ค่าขีดแบ่งสม่ำเสมอไม่เกิน 4) โดยจะเหลือข้อมูลรายการที่เป็นรายการที่ปรากฏอย่างสม่ำเสมอดังตารางที่ 3-8

ตารางที่ 3-8 รายการที่ปรากฏอย่างสม่ำเสมอ

Item	Support	Max period	Tid
d	9	2	1,2,3,4,6,7,9,11,12
a	8	3	1,4,6,7,8,10,11,12
b	8	3	1,3,4,6,7,8,11,12
c	8	2	2,3,5,6,7,8,9,10
g	7	4	1,3,4,5,6,10,12
e	5	3	1,2,5,8,10

จากตารางที่ 3-8 รายการ f ถูกกำจัดออกไปเนื่องจากมีค่าขีดแบ่งความสม่ำเสมอของรายการเท่ากับ 7 ซึ่งมีค่ามากกว่าค่าขีดแบ่งความสม่ำเสมอที่กำหนดไว้คือ 4 จึงต้องทำการตัดข้อมูลรายการ f ออกจากเซตรายการที่ปรากฏอย่างสม่ำเสมอ โดยจะเหลือรายการที่ปรากฏอย่างสม่ำเสมอที่ใช้ในการพิจารณาค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอทั้งหมด 6 รายการ

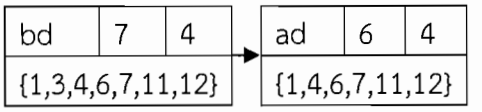
ขั้นตอนต่อมาเป็นขั้นตอนที่สอง ETFRP–mining ซึ่งแบ่งการทำงานออกเป็นสองขั้นตอนด้วยกัน คือ (i) สร้างลิงค์ลิสต์รูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอที่มีความถี่ในการปรากฏสูงสุดเคอันดับแรกที่มีขนาด 2 รายการโดยเรียงลำดับจากมากไปน้อย (ii) ค้นหารูปแบบปรากฏบ่อยและปรากฏอย่างสม่ำเสมอโดยมีความถี่ในการปรากฏสูงสุดเคอันดับแรกที่มีขนาดมากกว่า 2 รายการโดยเรียงลำดับจากมากไปน้อย

ขั้นตอน ETFRP–mining เริ่มจากพิจารณารายการที่มีค่าสนับสนุนสูงสุดสองรายการแรกก่อนคือรายการ d และรายการ a เพื่อค้นหาทรานแซกชันไอดีที่ปรากฏขึ้นร่วมกัน พบว่ารายการ d และ a ปรากฏขึ้นร่วมกันในทรานแซกชันต่อไปนี้ T^{ad}= {1,4,6,7,11,12} โดยมีค่าสนับสนุน S^{ad}=6 และมีค่าความสม่ำเสมอในการปรากฏซ้ำ rtt^{ad}=4 โดยทำการเก็บข้อมูลในลิงค์ลิสต์ได้ดังภาพที่ 3-4

ad	6	4
{1,4,6,7,11,12}		

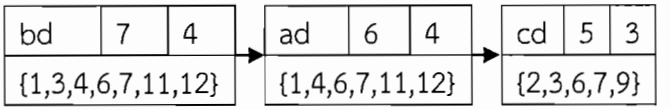
ภาพที่ 3-4 โครงสร้างในการจัดเก็บข้อมูลรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอในโครงสร้างลิงค์ลิสต์ภาพที่ 1

พิจารณารายการ d และรายการ b เพื่อค้นหาทรานแซกชันไอดีที่ปรากฏขึ้นร่วมกัน จะพบว่ารายการ d และ b จะปรากฏขึ้นร่วมกันในทรานแซกชันต่อไปนี้ T^{bd}= {1,3,4,6,7,11,12} โดยมีค่าสนับสนุน S^{bd}=7 และมีค่าความสม่ำเสมอในการปรากฏซ้ำ rtt^{bd}=4 โดยทำการเก็บข้อมูลในลิงค์ลิสต์ได้ดังภาพที่ 3-5



ภาพที่ 3-5 โครงสร้างในการจัดเก็บข้อมูลรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอในโครงสร้างลิงค์ลิสต์ภาพที่ 2

พิจารณารายการ d และรายการ c เพื่อค้นหาทรานแซกชันไอดีที่ปรากฏขึ้นร่วมกัน จะพบว่ารายการ d และ c จะปรากฏขึ้นร่วมกันในทรานแซกชันต่อไปนี้ T^{cd}= {2,3,6,7,9} โดยมีค่าสนับสนุน S^{cd}=5 และมีค่าความสม่ำเสมอในการปรากฏซ้ำ rtt^{cd}=3 โดยทำการเก็บข้อมูลในลิงค์ลิสต์ได้ดังภาพที่ 3-6

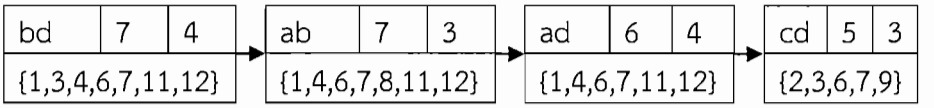


ภาพที่ 3-6 โครงสร้างในการจัดเก็บข้อมูลรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอในโครงสร้างลิงค์ลิสต์ภาพที่ 3

พิจารณารายการ d และรายการ g เพื่อค้นหาทรานแซกชันไอดีที่ปรากฏขึ้นร่วมกัน จะพบว่ารายการ d และ g มีค่าความสม่ำเสมอในการปรากฏซ้ำ rtt^{dg}=6 ซึ่งมีค่าเกินค่าขีดแบ่งความสม่ำเสมอ ข้อมูลนี้ก็จะถูกทิ้งไป

พิจารณารายการ d และรายการ e เพื่อค้นหาทรานแซกชันไอดีที่ปรากฏขึ้นร่วมกัน จะพบว่ารายการ d และ e มีค่าความสม่ำเสมอในการปรากฏซ้ำ rtt^{ed}=10 ซึ่งมีค่าเกินค่าขีดแบ่งความสม่ำเสมอ ข้อมูลนี้ก็จะถูกทิ้งไป

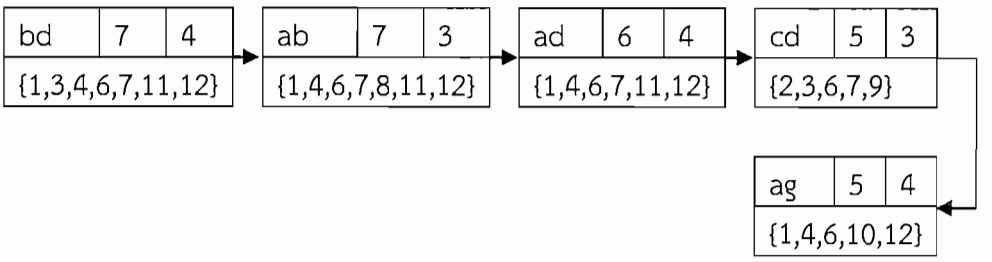
พิจารณารายการ a และรายการ b เพื่อค้นหาทรานแซกชันไอดีที่ปรากฏขึ้นร่วมกัน จะพบว่ารายการ a และ b จะปรากฏขึ้นร่วมกันในทรานแซกชันต่อไปนี้ T^{ab}= {1,4,6,7,8,11,12} โดยมีค่าสนับสนุน S^{ab}=7 และมีค่าความสม่ำเสมอในการปรากฏซ้ำ rtt^{ab}=3 เนื่องจากเซตรูปแบบ ab มีค่าสนับสนุนมากกว่าเซตรูปแบบ ad จึงเรียงเซตรูปแบบ ab ไว้ด้านหน้า โดยทำการเก็บข้อมูลในลิงค์ลิสต์ได้ดังภาพที่ 3-7



ภาพที่ 3-7 โครงสร้างในการจัดเก็บข้อมูลรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอในโครงสร้างลิงค์ลิสต์ภาพที่ 4

พิจารณารายการ a และรายการ c เพื่อค้นหาทรานแซกชันไอดีที่ปรากฏขึ้นร่วมกัน จะพบว่ารายการ a และ c มีค่าความสม่ำเสมอในการปรากฏซ้ำ rtt^{ac}=6 ซึ่งมีค่าเกินค่าขีดแบ่งความสม่ำเสมอ ข้อมูลนี้ก็จะถูกทิ้งไป

พิจารณารายการ a และรายการ g เพื่อค้นหาทรานแซกชันไอดีที่ปรากฏขึ้นร่วมกัน จะพบว่ารายการ a และ g จะปรากฏขึ้นร่วมกันในทรานแซกชันต่อไปนี้ T^{ag}= {1,4,6,10,12} โดยมีค่าสนับสนุน S^{ag}=5 และมีค่าความสม่ำเสมอในการปรากฏซ้ำ rtt^{ag}=4 โดยทำการเก็บข้อมูลในลิงค์ลิสต์ได้ดังดังภาพที่ 3-8



ภาพที่ 3-8 โครงสร้างในการจัดเก็บข้อมูลรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอในโครงสร้างลิงค์ลิสต์ภาพที่ 5

พิจารณารายการ a และรายการ e เพื่อค้นหาทรานแซกชันไอดีที่ปรากฏขึ้นร่วมกัน จะพบว่ารายการ a และ e มีค่าความสม่ำเสมอในการปรากฏซ้ำ rtt^{ae}=7 ซึ่งมีค่าเกินค่าขีดแบ่งความสม่ำเสมอ ข้อมูลนี้ก็จะถูกทิ้งไป

พิจารณารายการ b และรายการ c เพื่อค้นหาทรานแซกชันไอดีที่ปรากฏขึ้นร่วมกัน จะพบว่ารายการ b และ c จะปรากฏขึ้นร่วมกันในทรานแซกชันต่อไปนี้ T^{bc}= {3,6,7,8} โดยมีค่าสนับสนุน S^{bc}=4 และมีค่าความสม่ำเสมอในการปรากฏซ้ำ rtt^{bc}=4 แต่เนื่องจากเซตรูปแบบลำดับสุดท้ายของลิงค์ลิสต์มีขนาดค่าสนับสนุนเท่ากับ 5 ข้อมูลนี้ก็จะถูกทิ้งไป

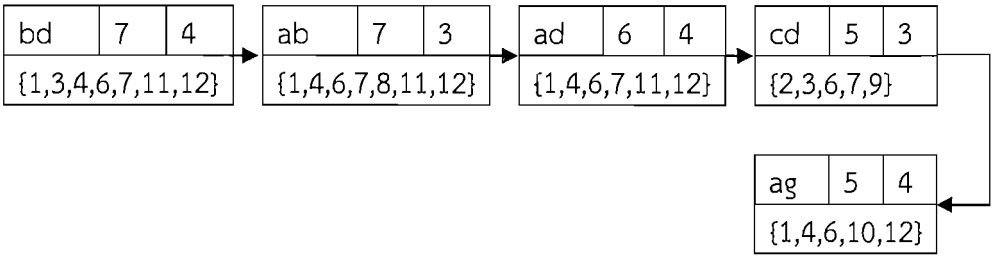
พิจารณารายการ b และรายการ g เพื่อค้นหาทรานแซกชันไอดีที่ปรากฏขึ้นร่วมกัน จะพบว่ารายการ b และ g มีค่าความสม่ำเสมอในการปรากฏซ้ำ rtt^{bg}=6 ซึ่งมีค่าเกินค่าขีดแบ่งความสม่ำเสมอ ข้อมูลนี้ก็จะถูกทิ้งไป

พิจารณารายการ b และรายการ e เพื่อค้นหาทรานแซกชันไอดีที่ปรากฏขึ้นร่วมกัน จะพบว่ารายการ b และ e มีค่าความสม่ำเสมอในการปรากฏซ้ำ rtt^{be}=7 ซึ่งมีค่าเกินค่าขีดแบ่งความสม่ำเสมอ ข้อมูลนี้ก็จะถูกทิ้งไป

พิจารณารายการ c และรายการ g เพื่อค้นหาทรานแซกชันไอดีที่ปรากฏขึ้นร่วมกัน จะพบว่ารายการ c และ g จะปรากฏขึ้นร่วมกันในทรานแซกชันต่อไปนี้ T^{cg}= {3,5,6,10} โดยมีค่าสนับสนุน S^{cg}=4 และมีค่าความสม่ำเสมอในการปรากฏซ้ำ rtt^{cg}=4 แต่เนื่องจากเซตรูปแบบลำดับสุดท้ายของลิงค์ลิสต์มีขนาดค่าสนับสนุนเท่ากับ 5 ข้อมูลนี้ก็จะถูกทิ้งไป

พิจารณารายการ c และรายการ e เพื่อค้นหาทรานแซกชันไอดีที่ปรากฏขึ้นร่วมกัน จะพบว่ารายการ c และ e จะปรากฏขึ้นร่วมกันในทรานแซกชันต่อไปนี้ T^{ce}= {2,5,8,10} โดยมีค่าสนับสนุน S^{ce}=4 และมีค่าความสม่ำเสมอในการปรากฏซ้ำ rtt^{ce}=3 แต่เนื่องจากเซตรูปแบบลำดับสุดท้ายของลิงค์ลิสต์มีขนาดค่าสนับสนุนเท่ากับ 5 ข้อมูลนี้ก็จะถูกทิ้งไป

พิจารณารายการ g และรายการ e เพื่อค้นหาทรานแซกชันไอดีที่ปรากฏขึ้นร่วมกัน จะพบว่ารายการ g และ e มีค่าความสม่ำเสมอในการปรากฏซ้ำ rtt^{eg}=5 ซึ่งมีค่าเกินค่าขีดแบ่งความสม่ำเสมอ ข้อมูลนี้ก็จะถูกทิ้งไป เมื่อดำเนินพิจารณาครบทุกรายการแล้วจะได้รูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอที่มีค่าสนับสนุนสูงสุดเคอันดับแรกที่มีขนาดเท่ากับ 2 รายการดังภาพที่ 3-9

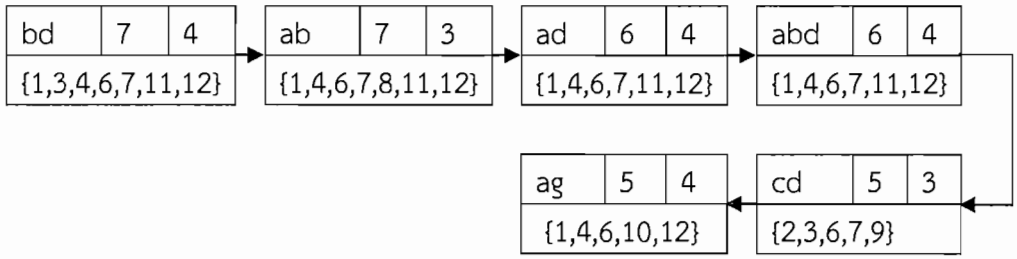


ภาพที่ 3-9 โครงสร้างในการจัดเก็บข้อมูลรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอในโครงสร้างลิงค์ลิสต์ภาพที่ 6

ขั้นตอนต่อไปนำเซตรูปแบบในลิงค์ลิสต์มาค้นหาเซตรูปแบบที่มีขนาดมากกว่า 2 รายการ โดยเซตรูปแบบที่นำมาพิจารณาต้องมีคุณสมบัติ 2 ข้อด้วยกัน คือ (i) เซตรูปแบบที่นำมาพิจารณาต้องมีขนาดเท่ากัน (ii) เซตรูปแบบที่นำมาพิจารณาต้องมี prefix เหมือนกันยกเว้นตัวสุดท้ายเท่านั้น

พิจารณาเซตรูปแบบ ab และเซตรูปแบบ ad เพื่อค้นหาทรานแซกชันไอดีที่ปรากฏขึ้นร่วมกัน จะพบว่ารายการ ab และ ad มีการปรากฏขึ้นร่วมกันในทรานแซกชันต่อไปนี้

T^{abd} = {1,4,6,7,11,12} โดยมีค่าสนับสนุน S^{abd}=6 และมีค่าความสม่ำเสมอในการปรากฏซ้ำ rtt^{abd}=4 เมื่อตรวจสอบค่าสนับสนุนในลิงค์ลิสต์แล้วทำการเพิ่มข้อมูลในลิงค์ลิสต์ แต่เนื่องจากเซตรูปแบบลำดับที่ห้าและลำดับที่หกของลิงค์ลิสต์มีค่าสนับสนุนเท่ากันข้อมูลในลิงค์ลิสต์จึงไม่ถูกลบออก โดยจัดเก็บข้อมูลในลิงค์ลิสต์ได้ดังภาพที่ 3-10

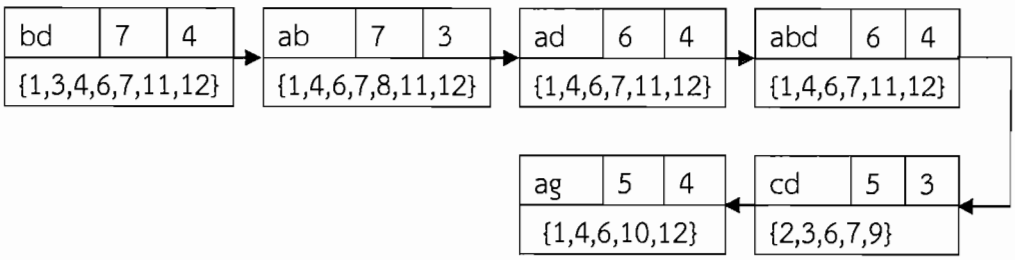


ภาพที่ 3-10 โครงสร้างในการจัดเก็บข้อมูลรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอในโครงสร้างลิงค์ลิสต์ภาพที่ 7

พิจารณาเซตรูปแบบ ab และเซตรูปแบบ ag เพื่อค้นหาทรานแซกชันไอดีที่ปรากฏขึ้นร่วมกันจะพบว่ารายการ ab และ ag มีค่าความสม่ำเสมอในการปรากฏซ้ำ rtt^{abg}=6 ซึ่งมีค่าเกินค่าขีดแบ่งความสม่ำเสมอ ข้อมูลนี้ก็จะถูกทิ้งไป

พิจารณาเซตรูปแบบ ad และเซตรูปแบบ ag เพื่อค้นหาทรานแซกชันไอดีที่ปรากฏขึ้นร่วมกัน จะพบว่ารายการ ad และ ag มีค่าความสม่ำเสมอในการปรากฏซ้ำ rtt^{adg}=6 ซึ่งมีค่าเกินค่าขีดแบ่งความสม่ำเสมอ ข้อมูลนี้ก็จะถูกทิ้งไป

เมื่อดำเนินวนลูปพิจารณาครบทุกเซตรูปแบบแล้ว จะได้รูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอที่มีค่าสนับสนุนสูงสุดเคอันดับแรกดังภาพที่ 3-11



ภาพที่ 3-11 โครงสร้างในการจัดเก็บข้อมูลรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอในโครงสร้างลิงค์ลิสต์ภาพที่ 8

จากผลลัพธ์ของเซตรูปแบบที่ได้นั้นจะมีจำนวน 6 เซตรูปแบบ ซึ่งเกินจำนวนเซตรูปแบบที่ผู้ใช้กำหนดไว้คือ 5 เซตรูปแบบ เนื่องจากเซตรูปแบบตัวที่ 6 นั้นมีค่าสนับสนุนจำนวนเท่ากับเซตรูปแบบตัวที่ 5 จึงไม่ได้ตัดเซตรูปแบบตัวที่ 6 ออกจากลิสต์ข้อมูลได้ จึงแสดงข้อมูลออกมาทั้งหมด 6 เซตรูปแบบ ซึ่งการปรับปรุงผลลัพธ์ของรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอด้วยอัลกอริทึม ETFRP ทำให้ผลลัพธ์ของเซตรูปแบบมีขนาดยาวขึ้นสามารถวิเคราะห์ความสัมพันธ์ระหว่างรายการที่ปรากฏขึ้นร่วมกันได้และสามารถนำไปสร้างเป็นกฎความสัมพันธ์และวิเคราะห์ความรู้อื่น ๆ ได้

บทที่ 4
ผลการวิจัย

เนื้อหาในบทนี้จะแสดงผลลัพธ์ในการทดสอบประสิทธิภาพในการทำงานของอัลกอริทึม ETFRP และอัลกอริทึม MTKPP โดยจะทำการทดสอบในเชิงเวลาและหน่วยความจำที่ใช้ในการประมวลผล ซึ่งแฟ้มข้อมูลที่ใช้ทดสอบประกอบไปด้วยแฟ้มข้อมูลแบบจำลอง (synthetic dataset) และแฟ้มข้อมูลจริง (real dataset) ที่นิยมนำมาใช้ในการทดสอบการทำเหมืองข้อมูล (สามารถดาวน์โหลดแฟ้มข้อมูลได้จากเว็บไซต์ http://fimi.ua.ac.be/data) ที่ประกอบไปด้วย 5 ชุดข้อมูล โดยแต่ละชุดข้อมูลจะมีรายละเอียดดังนี้

1. Chess ประกอบด้วยข้อมูลทั้งสิ้น 3,196 ทรานแซกชัน
2. Mushroom ประกอบด้วยข้อมูลทั้งสิ้น 8,124 ทรานแซกชัน
3. Conect ประกอบด้วยข้อมูลทั้งสิ้น 67,557 ทรานแซกชัน
4. Retail ประกอบด้วยข้อมูลทั้งสิ้น 88,162 ทรานแซกชัน
5. T10I4D100K ประกอบด้วยข้อมูลทั้งสิ้น 100,000 ทรานแซกชัน

ภายใต้การดำเนินการทดลอง ผู้วิจัยได้กำหนดจำนวน k ให้มีค่าเป็น 100, 200, 500, 1000, และ 2000 ตามลำดับ และทำการกำหนดค่าขีดแบ่งความสม่ำเสมอ(α_p) ให้ค่าอยู่ระหว่าง 2% และ 30% การทดลองได้ดำเนินการบนเครื่อง Linux Ubuntu version 12.04.2 LTS ประกอบด้วย

1. CPU AMD A6-3420M 1.5 GHz
2. Memory 4 GB
3. Hard disk 500 GB

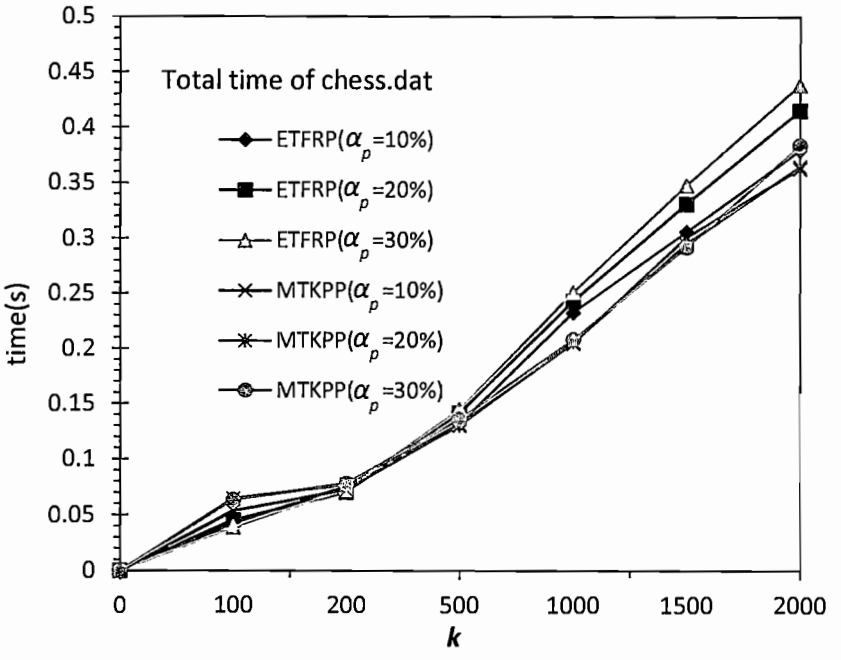
การวิเคราะห์ประสิทธิภาพของ ETFRP ในเชิงสัญกรณ์ทางคณิตศาสตร์

อัลกอริทึม ETFRP จะแสดงผลลัพธ์ของเซตรูปแบบที่มีขนาดตั้งแต่สองรายการขึ้นไป โดยแบ่งขั้นตอนการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอเป็น 2 ขั้นตอน ขั้นตอนแรกคือการค้นหาเซตรายการขนาดหนึ่งรายการที่ปรากฏอย่างสม่ำเสมอโดยมีค่าความซับซ้อนในการค้นหา คือ O(mn) และขั้นตอนที่สองนำเซตรายการที่ปรากฏอย่างสม่ำเสมอมาค้นหาเซตรูปแบบที่ปรากฏบ่อยสุดเคอันดับแรก โดยมีค่าความซับซ้อนในการค้นหา คือ $O(mn^2 + mk^2)$ เมื่อ m เป็นจำนวนทรานแซกชันทั้งหมดของฐานข้อมูลรายการ, n เป็นจำนวนสมาชิกรายการสินค้าที่ปรากฏในฐานข้อมูล และ k เป็นจำนวนผลลัพธ์ที่ผู้ใช้ต้องการ

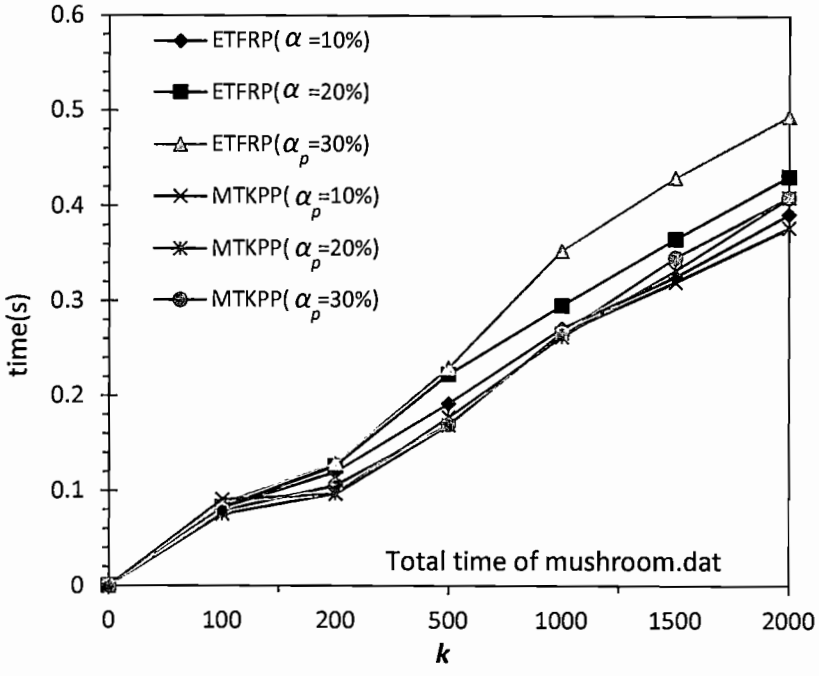
ผลการทดลองเวลาในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอ

ผลลัพธ์ของรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอของอัลกอริทึม ETFRP จะเป็นเซตรูปแบบที่มีขนาดสองรายการขึ้นไป ในการกำหนดค่าขีดแบ่งความสม่ำเสมอ (α_p) และจำนวนผลลัพธ์ที่ต้องการ (k) แบบหลาย ๆ ค่า เพื่อทดสอบเสถียรภาพการทำงานของอัลกอริทึม

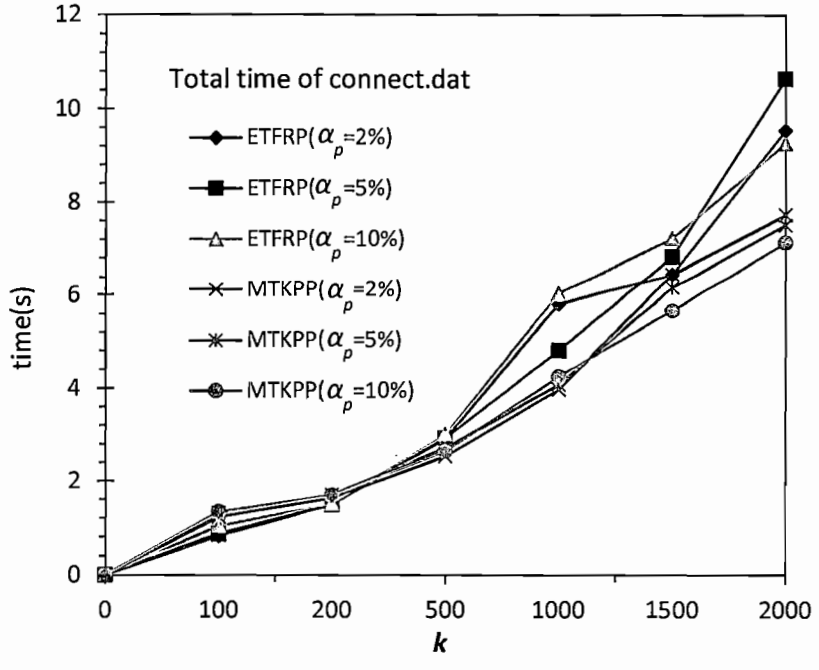
ETFRP โดยเปรียบเทียบเวลาในการประมวลผลของการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอตามจำนวนผลลัพธ์ที่ต้องการ (k) กับอัลกอริทึม MTKPP ซึ่งแสดงออกมาในรูปจำนวนของเซตรูปแบบที่ค้นหาได้กับจำนวนเวลาที่ใช้ในการประมวลผล ดังภาพที่ 4-1 ถึงภาพที่ 4-5



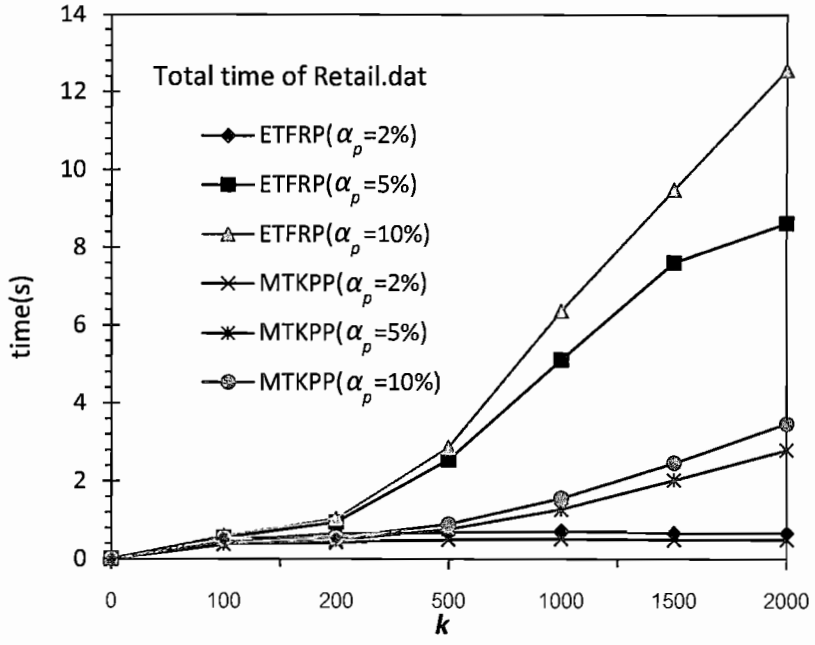
ภาพที่ 4-1 เวลาเปรียบเทียบในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอของอัลกอริทึม ETFRP และอัลกอริทึม MTKPP ของแฟ้มข้อมูล Chess



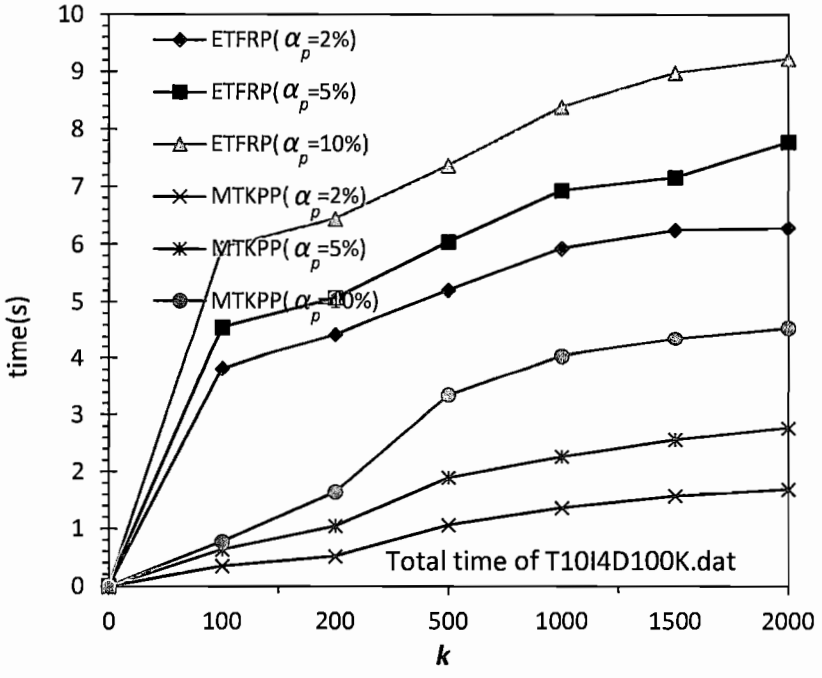
ภาพที่ 4-2 เวลาเปรียบเทียบในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอของอัลกอริทึม ETFRP และอัลกอริทึม MTKPP ของแฟ้มข้อมูล Mushroom



ภาพที่ 4-3 เวลาเปรียบเทียบในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอของอัลกอริทึม ETFRP และอัลกอริทึม MTKPP ของแฟ้มข้อมูล Connect



ภาพที่ 4-4 เวลาเปรียบเทียบในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอของอัลกอริทึม ETFRP และอัลกอริทึม MTKPP ของแฟ้มข้อมูล Retail



ภาพที่ 4-5 เวลาเปรียบเทียบในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอของอัลกอริทึม ETFRP และอัลกอริทึม MTKPP ของแฟ้มข้อมูล T10I4D100K

จากภาพที่ 4-1 ถึงภาพที่ 4-5 จะพบว่าเมื่อจำนวนผลลัพธ์ที่ต้องการ (k) และค่าขีดแบ่งความสม่ำเสมอ (α_p) มีค่าเพิ่มขึ้น เวลาในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอก็ใช้เวลาเพิ่มขึ้นตามลำดับทั้งอัลกอริทึม ETFRP และอัลกอริทึม MTKPP เนื่องจากเมื่อ α_p เพิ่มขึ้นจะทำให้มีจำนวนเซตรายการที่ปรากฏอย่างสม่ำเสมอมีจำนวนเพิ่มขึ้น เพราะช่วงการเกิดขึ้นซ้ำของรายการมีระยะห่างเพิ่มขึ้นนั่นเอง ทำให้มีเซตรายการที่ต้องเปรียบเทียบการเกิดขึ้นพร้อมกันเพิ่มขึ้น และเมื่อค่า k เพิ่มขึ้นจะมีการเปรียบเทียบเซตรายการที่ปรากฏอย่างสม่ำเสมอเพื่อค้นหาจำนวนผลลัพธ์ที่เพิ่มมากขึ้น แต่จะพบว่าอัลกอริทึม ETFRP จะใช้เวลาในการประมวลผลมากกว่าอัลกอริทึม MTKPP เนื่องจากอัลกอริทึม ETFRP มีการสร้างเซตรูปแบบปรากฏบ่อยเคอันดับแรกขนาดสองรายการก่อน ซึ่งจะต้องเปรียบเทียบทรานแซกชันทั้งหมดที่เป็นเซตรายการที่ปรากฏอย่างสม่ำเสมอทุกรายการเพื่อค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกที่มีขนาดสองรายการและมีความถี่ในการปรากฏสูงสุด แล้วนำไปเซตรูปแบบที่มีขนาดสองรายการไปค้นหาเซตรูปแบบที่มีขนาดใหญ่มากขึ้นจนไม่มีเซตรูปแบบใดสามารถนำมาเปรียบเทียบได้อีก ซึ่งต่างจากอัลกอริทึม MTKPP ที่ทำการกำหนดรูปแบบปรากฏบ่อยสุดเคอันดับแรกที่มีค่าสนับสูงสุดจากเซตรายการที่ปรากฏอย่างสม่ำเสมอได้ทันที เพราะเริ่มต้นด้วยเซตรายการที่มีขนาดหนึ่งรายการ แล้วนำมาเปรียบเทียบการเกิดขึ้นร่วมกันของเซตรายการภายในเซตรายการเคอันดับ จึงทำให้เวลาในการประมวลผลใช้เวลาน้อยกว่าอัลกอริทึม ETFRP

ผลการทดลองขนาดของรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอที่ค้นหาได้เปรียบเทียบระหว่างอัลกอริทึม ETFRP และอัลกอริทึม MTKPP

เนื้อหาในส่วนนี้จะแสดงการเปรียบเทียบขนาดของเซตรูปแบบที่ค้นหาจากอัลกอริทึม ETFRP และอัลกอริทึม MTKPP ซึ่งอัลกอริทึม ETFRP จะมีเซตรูปแบบขนาดตั้งแต่สองรายการขึ้นไป โดยแสดงผลการเปรียบเทียบเป็นข้อมูลตารางดังตารางที่ 4-1 ถึงตารางที่ 4-15

ตารางที่ 4-1 จำนวนรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอตามขนาดเซตรูปแบบของอัลกอริทึม ETFRP และ MTKPP โดยใช้แฟ้มข้อมูล Chess ที่ค่าขีดแบ่งความสม่ำเสมอที่ 10% ของจำนวนทรานแซกชันทั้งหมด

K	Chess Regularity 10%														
	ETFRP Pattern size							*MTKPP* Pattern size							
	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
100	32	40	25	5	0	0	0	11	28	37	22	4	0	0	0
200	46	74	57	22	3	0	0	12	45	70	53	19	2	0	0
500	64	147	172	98	22	1	0	13	62	144	167	92	21	1	0
1000	88	229	326	253	95	14	0	16	86	224	320	248	93	14	0
1500	94	274	451	424	211	48	3	16	94	273	450	416	205	46	3
2000	101	316	558	574	344	110	13	16	101	312	550	568	339	105	11

ตารางที่ 4-2 จำนวนรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอตามขนาดเซตรูปแบบของอัลกอริทึม ETFRP และ MTKPP โดยใช้แฟ้มข้อมูล Chess ที่ค่าขีดแบ่งความสม่ำเสมอที่ 20% ของจำนวนทรานแซกชันทั้งหมด

K	Chess Regularity 20%														
	ETFRP Pattern size							*MTKPP* Pattern size							
	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
100	32	40	25	5	0	0	0	11	28	37	22	4	0	0	0
200	46	74	57	22	3	0	0	12	45	70	53	19	2	0	0
500	64	147	172	98	22	1	0	13	62	144	167	92	21	1	0
1000	88	229	326	253	95	14	0	16	86	224	320	248	93	14	0
1500	94	274	451	424	211	48	3	16	94	273	450	416	205	46	3
2000	101	316	558	574	344	110	13	16	101	312	550	568	339	105	11

ตารางที่ 4-3 จำนวนรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอตามขนาดเซตรูปแบบของอัลกอริทึม ETFRP และ MTKPP โดยใช้แฟ้มข้อมูล Chess ที่ค่าขีดแบ่งความสม่ำเสมอที่ 30% ของจำนวนทรานแซกชันทั้งหมด

K	Chess Regularity 30%														
	ETFRP Pattern size							*MTKPP* Pattern size							
	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
100	32	40	25	5	0	0	0	11	28	37	22	4	0	0	0
200	46	74	57	22	3	0	0	12	45	70	53	19	2	0	0
500	64	147	172	98	22	1	0	13	62	144	167	92	21	1	0
1000	88	229	326	253	95	14	0	16	86	224	320	248	93	14	0
1500	94	274	451	424	211	48	3	16	94	273	450	416	205	46	3
2000	101	316	558	574	344	110	13	16	101	312	550	568	339	105	11

จากตารางที่ 4-1 ถึงตารางที่ 4-3 แสดงขนาดของเซตรูปแบบที่ค้นหาจากแฟ้มข้อมูลรายการ Chess โดยใช้ค่าขีดแบ่งความสม่ำเสมอ 10%, 20% และ 30% ตามลำดับ จากการสังเกตจะพบว่าเซตรูปแบบที่มีขนาดเท่ากันของอัลกอริทึม ETFRP จะมีจำนวนมากกว่าอัลกอริทึม MTKPP เกือบทุกขนาดของเซตรูปแบบ แต่จำนวนเซตรูปแบบที่ได้จะมีจำนวนเท่ากันทั้งสามตารางในแต่ละอัลกอริทึม เนื่องจากเซตรายการที่ปรากฏบ่อยของค่าขีดแบ่งความสม่ำเสมอ 10% เป็นสับเซตของค่าขีดแบ่งความสม่ำเสมอ 20% และ 30% ทำให้การเปรียบเทียบการเกิดขึ้นของเซตรายการที่ปรากฏขึ้นอย่างสม่ำเสมอมีผลลัพธ์ของ 10% ด้วย ถึงแม้ว่าเซตรายการที่ปรากฏอย่างสม่ำเสมอที่ค่าขีดแบ่งความสม่ำเสมอ 20% และ 30% จะมีจำนวนเซตรายการมากกว่าที่ค่าขีดแบ่งความสม่ำเสมอ 10% แต่เซตรูปแบบผลลัพธ์จะจัดเก็บ

เฉพาะเซตรูปแบบที่มีค่าสนับสนุนสูงสุดเคอันดับแรกเท่านั้นและประกอบกับเซตรายการของแฟ้มข้อมูล Chess มีจำนวนรายการเพียง 75 รายการ จึงส่งผลให้การค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอที่ค่าขีดแบ่งความสม่ำเสมอที่ 10% , 20% และ 30% มีจำนวนขนาดของรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากกอย่างสม่ำเสมอมีจำนวนที่เท่ากันในแต่ละอัลกอริทึม

ตารางที่ 4-4 จำนวนรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอตามขนาดเซตรูปแบบของอัลกอริทึม ETFRP และ MTKPP โดยใช้แฟ้มข้อมูล Mushroom ที่ค่าขีดแบ่งความสม่ำเสมอที่ 10% ของจำนวนทรานแซกชันทั้งหมด

K	Mushroom Regularity 10%																
	ETFRP Pattern size								*MTKPP* Pattern size								
	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
100	38	42	18	3	0	0	0	0	11	33	37	17	3	0	0	0	0
200	54	78	53	15	1	0	0	0	16	54	74	48	13	1	0	0	0
500	101	187	165	69	11	0	0	0	22	98	173	146	58	9	0	0	0
1000	131	296	338	203	59	6	0	0	24	128	287	324	190	53	5	0	0
1500	142	359	472	346	144	34	4	0	24	142	356	466	342	143	34	4	0
2000	162	424	593	486	245	79	17	2	27	162	424	593	485	240	78	17	2

ตารางที่ 4-5 จำนวนรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอตามขนาดเซตรูปแบบของอัลกอริทึม ETFRP และ MTKPP โดยใช้แฟ้มข้อมูล Mushroom ที่ค่าขีดแบ่งความสม่ำเสมอที่ 20% ของจำนวนทรานแซกชันทั้งหมด

K	Mushroom Regularity 20%															
	ETFRP Pattern size								*MTKPP* Pattern size							
	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8
100	38	42	18	3	0	0	0	0	11	33	37	17	3	0	0	0
200	54	78	53	15	1	0	0	0	16	54	74	48	13	1	0	0
500	101	188	166	69	11	0	0	0	22	96	170	144	58	9	0	0
1000	129	293	331	193	54	5	0	0	24	128	288	322	186	52	5	0
1500	143	361	483	363	154	36	4	0	25	141	354	471	347	139	30	3
2000	155	407	587	506	271	91	19	2	27	155	407	587	506	271	91	17

ตารางที่ 4-6 จำนวนรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอตามขนาดเซตรูปแบบของอัลกอริทึม ETFRP และ MTKPP โดยใช้แฟ้มข้อมูล Mushroom ที่ค่าขีดแบ่งความสม่ำเสมอที่ 30% ของจำนวนทรานแซกชันทั้งหมด

K	Mushroom Regularity 30%															
	ETFRP Pattern size								*MTKPP* Pattern size							
	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8
100	35	41	20	4	0	0	0	0	11	33	39	19	4	0	0	0
200	54	76	53	18	2	0	0	0	15	49	70	50	17	2	0	0
500	93	169	152	70	15	1	0	0	21	91	164	146	66	14	1	0
1000	126	292	340	209	66	8	0	0	24	125	286	324	189	55	6	0
1500	142	363	492	371	155	34	3	0	25	141	358	479	350	139	28	2
2000	152	410	602	513	261	82	15	1	25	152	410	602	513	261	74	9

จากตารางที่ 4-4 ถึงตารางที่ 4-6 แสดงขนาดของเซตรูปแบบที่ค้นหาจากแฟ้มข้อมูล Mushroom ที่ค่าขีดแบ่งความสม่ำเสมอ 10%, 20% และ 30% ตามลำดับ จากการสังเกตจะพบว่าเซตรูปแบบที่มีขนาดเท่ากันจะมีจำนวนมากกว่าอัลกอริทึม MTKPP เกือบทุกขนาด เมื่อค่า k มีค่าเพิ่มมากขึ้นความแตกต่างของขนาดเซตรูปแบบที่ได้จากอัลกอริทึม MTKPP และอัลกอริทึม ETPFRP ก็จะน้อยลง เนื่องจากมีการจัดเก็บเซตผลลัพธ์ที่มากขึ้นและการตัดทิ้งของเซตรูปแบบก็จะลดลง ดังจะสังเกตได้จากค่า k ที่จำนวน 2,000 มีขนาดของเซตรูปแบบเท่ากันเกือบทุกขนาดเพราะเซตรายการที่ปรากฏอย่างสม่ำเสมอที่นำมาเปรียบเทียบการเกิดขึ้นของเซตรายการเป็นชุดเดียวกันทั้งสองอัลกอริทึม

ตารางที่ 4-7 จำนวนรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอตามขนาดเซตรูปแบบของอัลกอริทึม ETFRP และ MTKPP โดยใช้แฟ้มข้อมูล Connect ที่ค่าขีดแบ่งความสม่ำเสมอที่ 2% ของจำนวนทรานแซกชันทั้งหมด

K	Connect Regularity 2%														
	ETFRP Pattern size							*MTKPP* Pattern size							
	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
100	32	45	20	4	0	0	0	10	32	37	18	3	0	0	0
200	43	79	66	15	0	0	0	10	43	75	58	15	0	0	0
500	52	125	168	120	36	0	0	11	52	125	168	114	30	0	0
1000	77	193	291	266	138	34	1	15	76	191	289	265	136	31	0
1500	92	261	412	399	243	85	13	16	92	257	407	396	240	84	13
2000	105	316	536	548	345	128	22	17	105	315	533	541	340	128	22

ตารางที่ 4-8 จำนวนรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอตามขนาดเซตรูปแบบของอัลกอริทึม ETFRP และ MTKPP โดยใช้แฟ้มข้อมูล Connect ที่ค่าขีดแบ่งความสม่ำเสมอที่ 5% ของจำนวนทรานแซกชันทั้งหมด

K	Connect Regularity 5%														
	ETFRP Pattern size							*MTKPP* Pattern size							
	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
100	32	45	20	4	0	0	0	10	32	37	18	3	0	0	0
200	43	79	66	15	0	0	0	10	43	75	58	15	0	0	0
500	52	125	168	120	36	0	0	11	52	125	168	114	30	0	0
1000	77	193	291	266	138	34	1	15	76	191	289	265	136	31	0
1500	92	261	412	399	243	85	13	16	92	257	407	396	240	84	13
2000	105	316	536	548	345	128	22	17	105	315	533	541	340	128	22

ตารางที่ 4-9 จำนวนรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอตามขนาดเซตรูปแบบของอัลกอริทึม ETFRP และ MTKPP โดยใช้แฟ้มข้อมูล Connect ที่ค่าขีดแบ่งความสม่ำเสมอที่ 10% ของจำนวนทรานแซกชันทั้งหมด

K	Connect Regularity 10%														
	ETFRP Pattern size							*MTKPP* Pattern size							
	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
100	32	45	20	4	0	0	0	10	32	37	18	3	0	0	0
200	43	79	66	15	0	0	0	10	43	75	58	15	0	0	0
500	52	125	168	120	36	0	0	11	52	125	168	114	30	0	0
1000	77	193	291	266	138	34	1	15	76	191	289	265	136	31	0
1500	92	261	412	399	243	85	13	16	92	257	407	396	240	84	13
2000	105	316	536	548	345	128	22	17	105	315	533	541	340	128	22

จากตารางที่ 4-7 ถึงตารางที่ 4-9 แสดงขนาดของเซตรูปแบบที่ได้จากไฟล์ Connect ที่ใช้ค่าขีดแบ่งความสม่ำเสมอ 2%, 5% และ 10% ในการค้นหาตามลำดับ จากการสังเกตจะพบว่าอัลกอริทึม ETFRP มีจำนวนเซตรูปแบบที่มากกว่าอัลกอริทึม MTKPP แต่จะแตกต่างกันไม่มาก ซึ่งเซตรูปแบบของค่าขีดแบ่งที่ 2%, 5% และ 10% ของแต่ละอัลกอริทึม มีจำนวนเซตรูปแบบแต่ละขนาดมีจำนวนเท่ากัน เนื่องจากเซตรายการที่ปรากฏบ่อยของค่าขีดแบ่งความสม่ำเสมอ 2% และ 5% เป็นสับเซตของเซตรายการที่ปรากฏบ่อยของค่าขีดแบ่งความสม่ำเสมอ 10% แต่ในการเปรียบเทียบการเกิดขึ้นร่วมกันของเซตรายการนั้นต้องมีค่าสนับสนุนสูงสุดเคอันดับแรกเท่านั้นทำให้เซตรายการที่ปรากฏแบบสม่ำเสมอ

ที่เพิ่มขึ้นมาไม่แสดงผลลัพธ์เพราะค่าสนับสนุนมีค่าไม่เพียงพอ ทำให้ผลลัพธ์ของรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอมีผลเหมือนกันที่ค่าขีดแบ่งความสม่ำเสมอดังกล่าว

ตารางที่ 4-10 จำนวนรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอตามขนาดเซตรูปแบบของอัลกอริทึม ETFRP และ MTKPP โดยใช้แฟ้มข้อมูล Retail ที่ค่าขีดแบ่งความสม่ำเสมอที่ 2% ของจำนวนทรานแซกชันทั้งหมด

Retail Regularity 2%							
K	*ETFRP* Pattern size			*MTKPP* Pattern size			
	2	3	4	1	2	3	4
100	71	25	4	38	40	19	3
200	139	54	7	77	91	28	4
500	209	87	11	160	209	87	11
1000	209	87	11	160	209	87	11
1500	209	87	11	160	209	87	11
2000	209	87	11	160	209	87	11

ตารางที่ 4-11 จำนวนรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอตามขนาดเซตรูปแบบของอัลกอริทึม ETFRP และ MTKPP โดยใช้แฟ้มข้อมูล Retail ที่ค่าขีดแบ่งความสม่ำเสมอที่ 5% ของจำนวนทรานแซกชันทั้งหมด

Retail Regularity 5%									
K	*ETFRP* Pattern size				*MTKPP* Pattern size				
	2	3	4	5	1	2	3	4	5
100	71	25	4	-	43	38	17	2	0
200	145	51	4	-	89	79	28	4	0
500	351	133	16	-	208	211	74	8	0
1000	677	286	38	3	385	431	167	21	0
1500	1026	426	62	3	585	623	264	34	2
2000	1328	577	95	3	740	866	358	51	3

ตารางที่ 4-12 จำนวนรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอตามขนาดเซตรูปแบบของอัลกอริทึม ETFRP และ MTKPP โดยใช้แฟ้มข้อมูล Retail ที่ค่าขีดแบ่งความสม่ำเสมอที่ 10% ของจำนวนทรานแซกชันทั้งหมด

Retail Regularity 10%									
K	*ETFRP* Pattern size				*MTKPP* Pattern size				
	2	3	4	5	1	2	3	4	5
100	71	25	4	0	42	38	17	2	0
200	146	50	4	0	89	78	29	4	0
500	361	128	13	0	207	214	75	8	0
1000	681	287	34	1	393	428	160	21	0
1500	1038	417	55	3	605	615	261	30	0
2000	1386	546	76	3	772	835	345	46	3

จากตารางที่ 4-10 ถึงตารางที่ 4-12 แสดงขนาดของเซตรูปแบบที่ได้จากไฟล์ Retail ที่ใช้ค่าขีดแบ่งความสม่ำเสมอ 2%, 5% และ 10% ตามลำดับ จากการสังเกตจะพบว่าตารางที่ 4.10 ใช้ค่าขีดแบ่งความสม่ำเสมอ 2% จะมีจำนวนขนาดของเซตรายการที่แตกต่างกันที่ค่า k เท่ากับ 100 และ 200 เท่านั้น เนื่องจากมีเซตรายการที่ปรากฏอย่างสม่ำเสมอเพียง 160 รายการ เมื่อค่า k มีจำนวนมากขึ้นทำให้จำนวนเซตรายการที่ค้นหามีจำนวนน้อยกว่าค่า k จึงไม่มีการตัดเซตรูปแบบใดออกทำให้ทั้งสองอัลกอริทึมมีจำนวนเซตรูปแบบที่เท่ากันตั้งแต่ค่า k เท่ากับ 500 จนถึงค่า k เท่ากับ 2,000 ส่วนที่ค่าขีดแบ่งความสม่ำเสมอที่ 5% และ 10% จำนวนของเซตรูปแบบที่มีขนาดเท่ากันของอัลกอริทึม ETFRP จะมีจำนวนมากกว่าอัลกอริทึม MTKPP อย่างเห็นได้ชัด เพราะมีเซตรูปแบบที่ปรากฏอย่างสม่ำเสมอในการเปรียบเทียบจำนวนมาก ทำให้ผลลัพธ์ของเซตรูปแบบที่ได้มีปริมาณที่แตกต่างกันอย่างชัดเจน

ตารางที่ 4-13 จำนวนรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอตามขนาดเซตรูปแบบของอัลกอริทึม ETFRP และ MTKPP โดยใช้แฟ้มข้อมูล T10I4D100K ที่ค่าขีดแบ่งความสม่ำเสมอที่ 2% ของจำนวนทรานแซกชันทั้งหมด

K	T10I4D100K Regularity 2%											
	ETFRP Pattern size						*MTKPP* Pattern size					
	2	3	4	5	6	7	1	2	3	4	5	6
100	73	22	5	0	0	0	100	0	0	0	0	0
200	143	42	13	2	0	0	200	0	0	0	0	0
500	342	110	43	9	0	0	445	47	9	0	0	0
1000	613	293	87	11	1	0	560	294	106	37	3	0
1500	835	411	186	60	11	1	605	571	252	61	11	1
2000	1108	570	250	67	12	1	630	755	378	178	59	9

ตารางที่ 4-14 จำนวนรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอตามขนาดเซตรูปแบบของอัลกอริทึม ETFRP และ MTKPP โดยใช้แฟ้มข้อมูล T10I4D100K ที่ค่าขีดแบ่งความสม่ำเสมอที่ 5% ของจำนวนทรานแซกชันทั้งหมด

K	T10I4D100K Regularity 5%											
	ETFRP Pattern size						*MTKPP* Pattern size					
	2	3	4	5	6	7	1	2	3	4	5	6
100	73	22	5	0	0	0	100	0	0	0	0	0
200	143	42	13	2	0	0	200	0	0	0	0	0
500	342	110	43	9	0	0	445	47	9	0	0	0
1000	617	289	82	11	1	0	560	294	106	37	3	0
1500	836	414	187	59	11	1	606	576	252	61	11	1
2000	1128	557	253	65	12	1	629	760	378	174	56	3

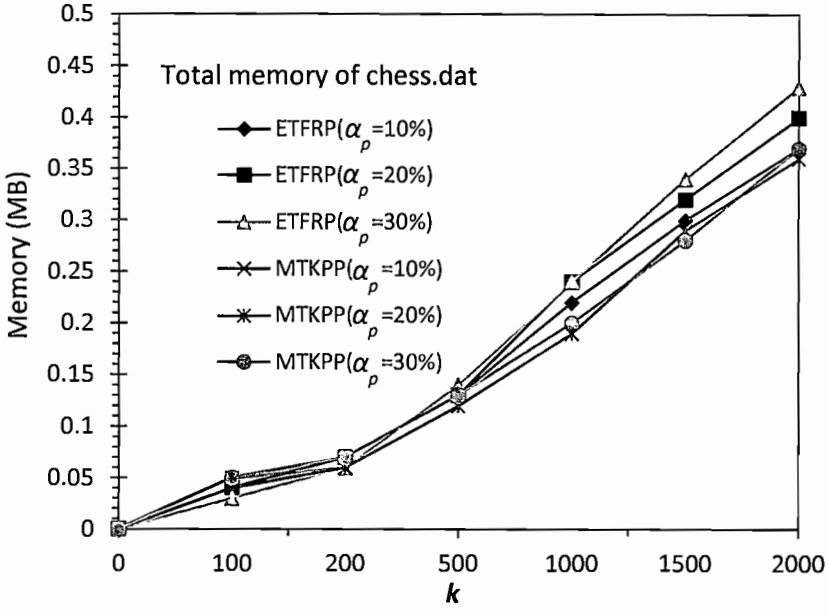
ตารางที่ 4-15 จำนวนรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอตามขนาดเซตรูปแบบของอัลกอริทึม ETFRP และ MTKPP โดยใช้แฟ้มข้อมูล T10I4D100K ที่ค่าขีดแบ่งความสม่ำเสมอที่ 10% ของจำนวนทรานแซกชันทั้งหมด

K	T10I4D100K Regularity 10%											
	ETFRP Pattern size						*MTKPP* Pattern size					
	2	3	4	5	6	7	1	2	3	4	5	6
100	73	22	5	0	0	0	100	0	0	0	0	0
200	143	42	13	2	0	0	200	0	0	0	0	0
500	342	110	43	9	0	0	445	47	9	0	0	0
1000	617	289	82	11	1	0	560	294	106	37	3	0
1500	836	414	187	59	11	1	606	576	252	61	11	1
2000	1128	557	253	65	12	1	629	760	378	174	56	3

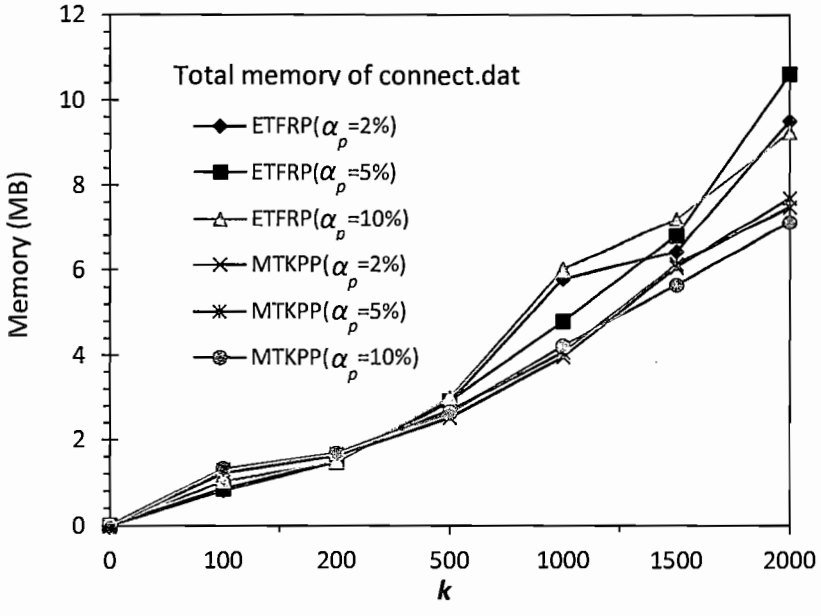
จากตารางที่ 4-13 ถึงตารางที่ 4-15 แสดงขนาดของเซตรูปแบบที่ค้นหาได้จากแฟ้มข้อมูล T10I4D100K ที่มีค่าขีดแบ่งความสม่ำเสมอ 2%, 5% และ 10% ตามลำดับ จากการสังเกตจะพบว่าอัลกอริทึม MTKPP เมื่อกำหนดค่า k ที่ขนาด 100 และ 200 เซตรูปแบบที่ค้นหาได้มีขนาดหนึ่งรายการทั้งหมด และเมื่อค่า k เพิ่มขึ้นเซตรูปแบบที่มีขนาดเป็นหนึ่งรายการจะมีจำนวนไม่น้อยกว่า 30 เปอร์เซ็นของจำนวนเซตรูปแบบทั้งหมดที่ค้นหาได้ ซึ่งอัลกอริทึม ETFRP จะมีขนาดของเซตรูปแบบที่ค้นหาได้ตั้งแต่สองรายการขึ้นไป แต่ผลลัพธ์ของเซตรูปแบบที่ค่าขีดแบ่งความสม่ำเสมอ 5% และ 10% ของแต่ละอัลกอริทึม มีจำนวนเซตรูปแบบแต่ละขนาดมีจำนวนเท่ากัน เนื่องจากเซตรายการที่ปรากฏอย่างสม่ำเสมอของค่าขีดแบ่งความสม่ำเสมอ 5% เป็นสับเซตของเซตรายการที่ปรากฏอย่างสม่ำเสมอของค่าขีดแบ่งความสม่ำเสมอ 10% แต่ในการเปรียบเทียบการเกิดขึ้นร่วมกันของเซตรายการนั้นต้องมีค่าสนับสนุนสูงสุดเคอันดับแรกเท่านั้นทำให้เซตรายการที่ปรากฏอย่างสม่ำเสมอที่เพิ่มขึ้นมาไม่แสดงผลลัพธ์เพราะค่าสนับสนุนมีค่าไม่เพียงพอ ทำให้ผลลัพธ์ของเซตรายการที่ปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอมีผลเหมือนกันที่ค่าขีดแบ่งความสม่ำเสมอดังกล่าว

จากตารางที่ 4-1 ถึงตารางที่ 4-15 การเปรียบเทียบขนาดของรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอของแฟ้มต่าง ๆ ที่นำมาทดสอบจะพบว่าอัลกอริทึม ETFRP ตัดผลลัพธ์ของเซตรูปแบบที่มีขนาดเป็นหนึ่งรายการออกไป ซึ่งรูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอจะมีขนาดขั้นต่ำเป็นสองรายการขึ้นไป และผลลัพธ์ของเซตรูปแบบที่มีขนาดต่าง ๆ สามารถนำไปวิเคราะห์หากฎความสัมพันธ์และนำไปวิเคราะห์หาความรู้ขั้นต่อไปได้

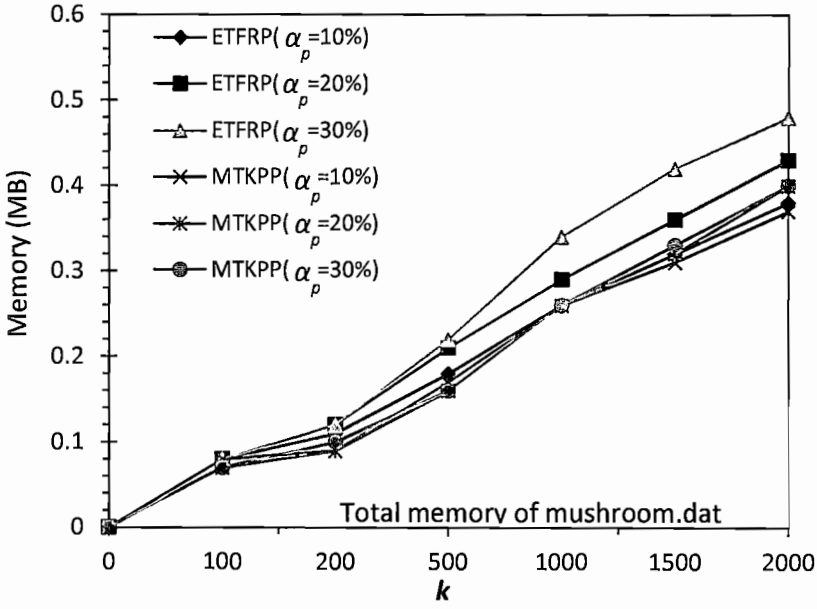
ผลการทดลองหน่วยความจำที่ใช้ในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอของอัลกอริทึม ETFRP



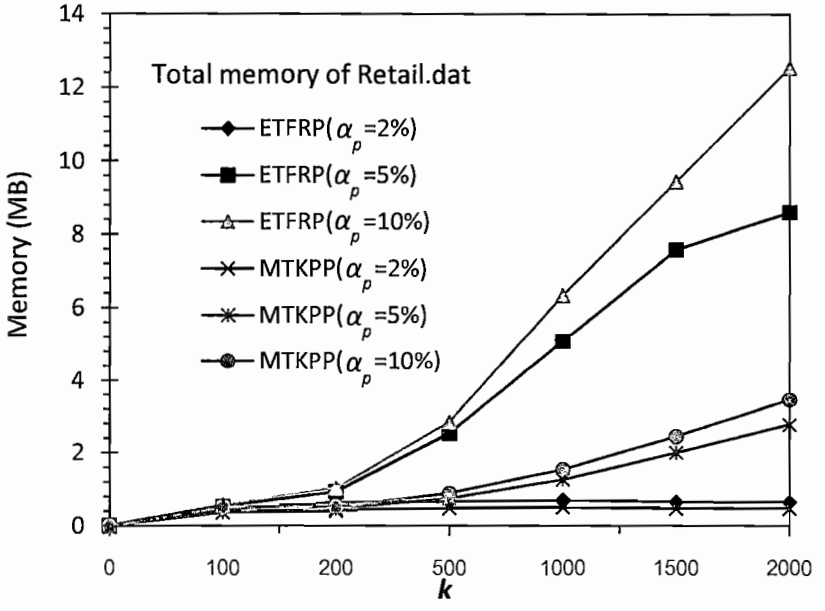
ภาพที่ 4-6 การใช้หน่วยความจำในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและและปรากฏอย่างสม่ำเสมอของแฟ้มข้อมูล Chess



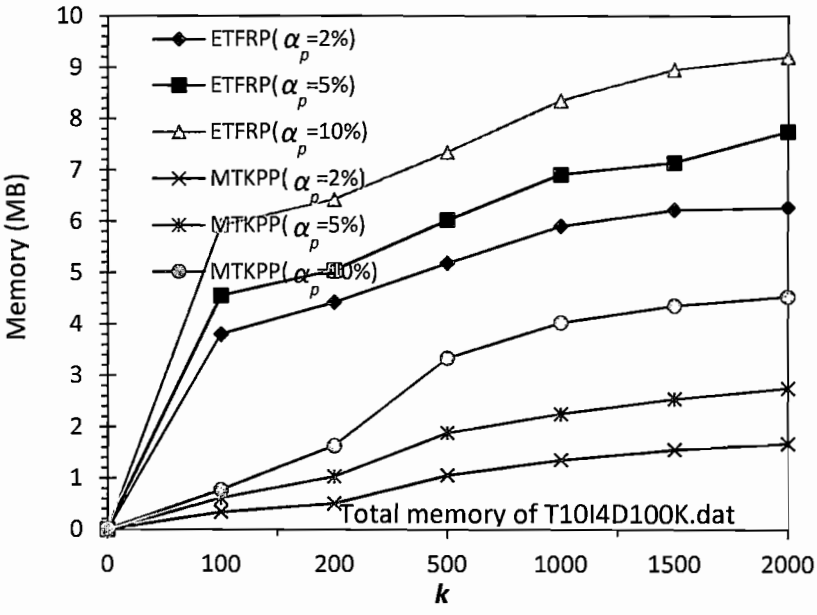
ภาพที่ 4-7 การใช้หน่วยความจำในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอของแฟ้มข้อมูล Connect



ภาพที่ 4-8 การใช้หน่วยความจำในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอของแฟ้มข้อมูล Mushroom



ภาพที่ 4-9 การใช้หน่วยความจำในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอของแฟ้มข้อมูล Retail



ภาพที่ 4-10 การใช้หน่วยความจำในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอของแฟ้มข้อมูล T10I4D100K

จากภาพที่ 4-6 ถึงภาพที่ 4-10 จะพบว่าเมื่อจำนวนผลลัพธ์ของเซตรูปแบบที่ต้องการ (k) และค่าขีดแบ่งความสม่ำเสมอ (α_p) เมื่อมีค่าเพิ่มขึ้นหน่วยความจำที่ใช้ในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอก็ใช้เพิ่มขึ้นตามอันดับทั้งอัลกอริทึม ETFRP และอัลกอริทึม MTKPP เนื่องจากค่า k เพิ่มขึ้นจะมีการเปรียบเทียบเซตรายการที่ปรากฏอย่างสม่ำเสมอเพื่อค้นหาจำนวนผลลัพธ์ที่เพิ่มขึ้นมีการจัดเก็บจำนวนผลลัพธ์เซตของแฟ้มข้อมูลรูปแบบที่มากขึ้น และเมื่อ α_p เพิ่มขึ้นจะทำให้มีจำนวนเซตรายการที่ปรากฏอย่างสม่ำเสมอเพิ่มจำนวนขึ้น ทำให้มีเซตรายการที่ต้องเปรียบเทียบมากขึ้น ซึ่งอัลกอริทึม MTKPP จะมีการใช้หน่วยความจำในการค้นหาที่มีจำนวนน้อยกว่าอักอริทึม ETFRP โดยเฉพาะภาพที่ 4-9 และ 4-10 จะมีการใช้หน่วยความจำแตกต่างกันเห็นได้ชัด เนื่องจากจำนวนทรานแซกชันที่มีปริมาณมากและการทำงานที่แตกต่างกันของทั้งสองอัลกอริทึมดังที่ได้กล่าวไว้ในช่วงของผลการทดลองของการใช้เวลาในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอในหัวข้อที่ 4.2

การวิเคราะห์ประสิทธิภาพอัลกอริทึม

การวิเคราะห์ความซับซ้อนของอัลกอริทึม ETFRP จะวิเคราะห์ออกมาในรูปของเวลาและพื้นที่หน่วยความจำสูงสุดในการประมวลผล

ข้อเสนอที่ 4.1 ความซับซ้อนของเวลาในการสร้าง regular periodic ลิสต์ O(nm) เมื่อ m เป็นจำนวนทรานแซกชันทั้งหมดของฐานข้อมูลรายการ, n เป็นจำนวนสมาชิกรายการสินค้าในแต่ละทรานแซกชัน

พิสูจน์ข้อเสนอที่ 4.1 การสร้าง regular periodic ลิสต์ เริ่มต้นจากขั้นตอนวิธีการที่สแกนรายการสินค้าที่เกิดขึ้นในแต่ละทรานแซกชันจากฐานข้อมูลเพียงครั้งเดียว เพื่อเก็บ Tid เป็น tidset ใช้เวลา O(nm) การเรียงลำดับของรายการสินค้าที่มีค่าสนับสนุนสูงสุดเรียงจากมากไปน้อยใช้เวลา O(nlog(n)) ดังนั้นความซับซ้อนของเวลาในการสร้าง regular periodic ลิสต์ คือ O(nm + nlog(n)) แต่ในความเป็นจริงจำนวนรายการ (n) จะมีค่าน้อยกว่าหรือเท่ากับจำนวนของทรานแซกชัน (m) จึงทำให้ค่า nlog(n) มีค่าน้อยกว่า nm ดังนั้นเราจึงสามารถสรุปความซับซ้อนของเวลาในการสร้างรายการ regular periodic ลิสต์ คือ O(nm)

ข้อเสนอที่ 4.2 ความซับซ้อนของเวลาในการการสร้าง Top-k ลิสต์ ที่มีขนาด 2 รายการขึ้นไป $O(mn^2 + mk^2)$ เมื่อ k เป็นจำนวนผลลัพธ์ของเซตรูปแบบที่ผู้ใช้ต้องการ, m เป็นจำนวนทรานแซกชันทั้งหมดของฐานข้อมูลรายการ, n เป็นจำนวนสมาชิกรายการสินค้าในแต่ละทรานแซกชันที่มีการปรากฏซ้ำอย่างสม่ำเสมอ

พิสูจน์ข้อเสนอที่ 4.2 ขั้นตอนนี้จะเริ่มต่อจากข้อเสนอที่ 4.1 นำ regular periodic ลิสต์มาทำการสร้าง Top-k ลิสต์ที่มีขนาด 2 รายการ โดยนำรายการสินค้าที่เป็น regular periodic มาทำการเปรียบเทียบการปรากฏขึ้นร่วมกันในแต่ละทรานแซกชัน ซึ่งจำนวนครั้งในการจับคู่รายการเพื่อเปรียบเทียบจะอยู่รูป $C_{(n,2)}$ จะได้จำนวนการเปรียบเทียบมากที่สุดเท่ากับ n(n-1)/2 ซึ่งในแต่ละคู่ของรายการจะมีการเปรียบเทียบทรานแซกชันจำนวนมากที่สุด m รอบ ซึ่งจะได้จำนวนรอบในการประมวลผลสูงสุดเท่ากับ $m(n^2-n)/2$ ซึ่งสามารถสรุปความซับซ้อนในการประมวลผลในส่วนนี้เท่ากับ $O(mn^2)$ ส่วนขั้นตอนถัดไปนำ Top-k ลิสต์ที่มีขนาด 2 รายการ โดยนำรายการสินค้าที่มีใน Top-k ลิสต์มาทำการเปรียบเทียบการปรากฏขึ้นร่วมกันในแต่ละทรานแซกชัน ซึ่งจำนวนครั้งในการจับคู่รายการเพื่อเปรียบเทียบจะอยู่รูป $C_{(k,2)}$ โดยค่า $k \leq (n^2-n)/2$ จะได้จำนวนการเปรียบเทียบมากที่สุดเท่ากับ k(k-1)/2 ซึ่งในแต่ละคู่ของรายการจะมีการเปรียบเทียบทรานแซกชันจำนวนมากที่สุด m รอบ จะได้จำนวนรอบในการประมวลผลสูงสุดเท่ากับ $m(k^2-k)/2$ ซึ่งสามารถสรุปความซับซ้อนในการประมวลผลในส่วนนี้เท่ากับ $O(mk^2)$ เมื่อครบทั้ง 2 ขั้นตอนเราจึงสามารถสรุปความซับซ้อนในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอเท่ากับ $O(mn^2 + mk^2)$

ข้อเสนอที่ 4.3 พื้นที่หน่วยความจำที่ใช้ในการจัดเก็บใน Top-k ลิสต์ใช้ O(km) โดยที่ m คือ จำนวนทรานแซกชันทั้งหมดที่มีในฐานข้อมูล และ k คือจำนวนเซตผลลัพธ์ที่ผู้ใช้ต้องการ

พิสูจน์ข้อเสนอที่ 4.3 เนื่องจาก top-k ลิสต์ จะเก็บจำนวนเซตรูปแบบที่ผู้ใช้ต้องการ k ตัว ซึ่งแต่ละตัวจะเก็บหมายเลขทรานแซกชัน โดยจำนวนสูงสุดของทรานแซกชันที่เป็นไปได้ คือ m ตัว เพราะฉะนั้นพื้นที่หน่วยความจำที่ใช้ในการจัดเก็บ top-k ลิสต์ มีค่าความซับซ้อน O(km)

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

สรุปผลการวิจัย

วิทยานิพนธ์ฉบับนี้ได้นำเสนอการพัฒนาการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอ โดยผลลัพธ์ในการค้นหาเซตรูปแบบมีขนาดเป็นสองรายการขึ้นไป ซึ่งผู้ใช้ต้องเป็นผู้กำหนดจำนวนผลลัพธ์ที่ต้องการ (k) และค่าขีดแบ่งความสม่ำเสมอ (regularity threshold (σ_r)) ซึ่งพัฒนามาจากการค้นหาเซตรูปแบบของอัลกอริทึม MTKPP เนื่องจากอัลกอริทึม MTKPP มีผลลัพธ์เซตรูปแบบที่มีขนาดหนึ่งรายการจำนวนมาก โดยอัลกอริทึม ETFRP สามารถลดจำนวนเซตรูปแบบที่มีขนาดหนึ่งรายการจากอัลกอริทึม MTKPP ได้ทั้งหมด ซึ่งขนาดของเซตรูปแบบที่ได้จากอัลกอริทึม ETFRP จะมีขนาดใหญ่ขึ้นและมีขนาดตั้งแต่สองรายการขึ้นไป ทำให้สามารถวิเคราะห์ความสัมพันธ์ของการปรากฏขึ้นร่วมกันของรายการสินค้าได้ โดยมีขั้นตอนในการทำงานดังหัวข้อ 5.1 และ 5.2

5.1 ขั้นตอนการค้นหารายการที่ปรากฏอย่างสม่ำเสมอ

ขั้นตอนนี้เป็นการค้นหาเซตรายการสินค้าที่มีการปรากฏอย่างสม่ำเสมอ เริ่มต้นจากการอ่านรายการสินค้าที่เกิดขึ้นในแต่ละทรานแซกชันจากฐานข้อมูลเพียงครั้งเดียว เพื่อเก็บหมายเลขทรานแซกชันที่เกิดขึ้น ตามค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้เป็นผู้กำหนด และจัดการเรียงลำดับของรายการสินค้าที่มีค่าสนับสนุนสูงสุดเรียงจากมากไปน้อย โดยมีค่าความซับซ้อนในการประมวลผล คือ O (nm) เมื่อ m เป็นจำนวนทรานแซกชันทั้งหมดของฐานข้อมูลรายการ, n เป็นจำนวนสมาชิกรายการสินค้าในฐานข้อมูลรายการ

5.2 ขั้นตอนการค้นหาเซตรูปแบบที่มีขนาดตั้งแต่สองรายการขึ้นไป

ขั้นตอนนี้จะดำเนินการนำเซตรายการสินค้าที่มีการปรากฏอย่างสม่ำเสมอมาสร้าง Top-k ลิสต์ที่มีขนาด 2 รายการ โดยนำเซตรายการสินค้าที่มีการปรากฏอย่างสม่ำเสมอมาทำการเปรียบเทียบการปรากฏขึ้นร่วมกันในแต่ละทรานแซกชัน ซึ่งจะได้เซตรูปแบบที่ปรากฏบ่อยสุดและปรากฏสม่ำเสมอจำนวน k ตัว จากนั้นนำเซตรูปแบบที่ปรากฏบ่อยสุดและปรากฏสม่ำเสมอจำนวน k ตัว มาจับคู่โดยต้องมี prefix ของเซตรูปแบบที่เหมือนกัน n-1 ตัว เพื่อเปรียบเทียบการปรากฏขึ้นร่วมกันในแต่ละทรานแซกชัน จนไม่มีเซตรูปแบบตัวใดที่สามารถเปรียบเทียบได้อีก ซึ่งสามารถสรุปความซับซ้อนในการประมวลผลในส่วนนี้เท่ากับ O ($mn^2 + mk^2$) เมื่อ m เป็นจำนวนทรานแซกชันทั้งหมดของฐานข้อมูลรายการ, n เป็นจำนวนสมาชิกรายการสินค้าในฐานข้อมูลรายการ และ k เป็นจำนวนผลลัพธ์ที่ผู้ใช้เป็นผู้กำหนด

ข้อเสนอแนะ

ในการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอ ยังใช้เวลาในการเปรียบเทียบเซตรายการที่มีทรานแซกชันที่เกิดขึ้นร่วมกันค่อนข้างมาก และการเรียงลำดับข้อมูลในลิงค์ลิสต์ของเคอันดับยังต้องใช้การเปรียบเทียบที่ละค่าสนับสนุนจากตัวท้าย เพื่อทำการแทรกข้อมูลในลิงค์

ลิสต์เคอันดับแรก จึงอาจจะต้องทำการพัฒนาและปรับปรุงโครงสร้างข้อมูลเพื่อลดเวลาในการประมวลผลและลดความซับซ้อนของขั้นตอนในการประมวลผลเพิ่มขึ้น

งานที่จะพัฒนาต่อไปในอนาคต

1. หาวิธีการเปรียบเทียบการเกิดขึ้นของรายการที่มีการเกิดขึ้นพร้อมกันในแต่ละทรานแซกชันให้รวดเร็วกว่าเดิม

2. วิเคราะห์ความต้องการของผู้ใช้ว่าต้องการข้อมูลผลลัพธ์ในลักษณะไหน เพื่อให้การค้นหาเซตรูปแบบผลลัพธ์ตรงกับความต้องการของผู้ใช้มากยิ่งขึ้น

3. พัฒนาขั้นตอนวิธีหรือปรับโครงสร้างข้อมูลเพื่อลดเวลาในการประมวลผลและเพิ่มประสิทธิภาพในการค้นหาเซตผลลัพธ์ที่ต้องการ

บรรณานุกรม

Ada Wai-chee Fu, Renfrew Wang-wai Kwong & Jian Tang. (2000). *Miing N-most Interesting Itemsets*. Foundations of Intelligent Systems, 1932, 2000. pp.59–67.

Doug Burdick, Manuel Calimlim, Johaannes Gehrke. (2001). *MAFIA: A Maximal Frequent Itemset Algorithm for Transactional Databases*. Cornell University.

G¨osta Grahne, Jianfei Zhu. (2005). *Fast Algorithms for Frequent Itemset Mining using FP-Trees*. IEEE Transactions on Knowledge and Data Engineering, Vol.17, No.11, November,2005. pp.1347–1362.

Joong Hyuk Chang. (2011). *Mining Weighted Patterns in a Sequence Database with a Time-interval Weight*. Knowledge-based Systems, 24, 2011. pp.1–9.

Jianyong Wang, Jiawei Han, Ying Lu, Petre Tzvetkov. (2005). *TFP: An Efficient Algorithm for Mining Top-k Frequent Closed Itemsets*. IEEE Transactions on Knowledge and Data Engineering, 17, 2005. pp.652–663.

Jiawei Han, Jian Pei, Yiwen Yin. (2000). *Mining Frequent Patterns without Candidate Generation*. Proceedings of the 2000 ACM SIGMOD international conference on Management of data. Dallas, Texas, USA. pp.1-12.

Jiawei Han, Jian Pei, Yiwen Yin, Runying Mao. (2004). *Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach*. Data Mining Knowledge Discovery, 8, 2004. pp.53–87.

Jiawei Han, Jianyong Wang, Ying Lu, Petre Tzvetkov. (2002). *Mining Top-k Frequent Closed Patterns without Minimum Support*. Proceedings IEEE International Conference on Data Mining (ICDM). pp.211–218.

Komate Amphawan, Philippe Lenca, Athasit Surarerks. (2009). *Mining Top-k Periodic-Frequent Pattern from Transactional Databases without Support Threshold*. Proceedings of the 3rd International Conference on Advances in Information Technology (IAIT). Bangkok, Thailand, 2009. pp.18–29.

Komate Amphawan, Philippe Lenca, Athasit Surarerks. (2012). *Efficient Mining Top-k Regular-frequent Itemset Using Compressed Tidsets*. New Frontiers in Applied Data Mining. 2012. pp.124–135.

Komate Amphawan, Philippe Lenca, Athasit Surarerks. (2012). *Mining Top-k Regular-frequent Itemsets Using Database Partitioning and Support Extimation*. Expert Systems with Applications, 39, 2012 . pp.1924–1936.

Komate Amphawan, Philippe Lenca. (2013). *Mining Top-k Frequent-regular Patterns Based on User-given Trade-off Between Frequency and Regularity*. Proceedings of the 6th International Conference on Advances in Information Technology (IAIT). Bangkok, Thailand, 2013.

Petre Tzvetkov, Xifeng Yan, Jiawei Han. (2005). *TSP: Mining Top-k Closed Sequential Patterns*. Knowledge and Information Systems, 7, 2005. pp.438–457.

Rakesh Agrawal, Ramakrishnan Srikant. (1994). *Fast Algorithms for Mining Association Rules in Large Databases*. Proceedings of 20th International Conference on Very Large Data Bases (VLDB). Santiago, Chile, 1994. pp.487–499.

Rakesh Agrawal, Tomasz Imielinski, Arun Swami. (1993). *Mining Association Rules between Sets of Items in Large Databases*. Proceedings of the 1993 ACM SIGMOD Conference on Management of Data. Washington DC, USA. 207–16.

Syed Khairuzzaman Tanbeer, Chowdhury Farhan Ahmed, Byeong-Soo Jeong, Young-Koo Lee. (2009). *Discovering Periodic-Frequent Patterns in Transactional Databases*. Advances in Knowledge Discovery and Data Mining, 5476, 2009. pp.242–253.

Sze-Chung Ngan, Tsang Lam, Raymond Chi-Wing Wong, Ada Wai-Chee Fu. (2005). *Mining N-most Interesting Itemsets without Support Threshold by the COFI-tree*. International Journal of Business Intelligence and Data Mining, 1 ,2005. pp.88–106.

Yin-Ling Cheung, Ada Wai-Chee Fu. (2002). *Fp-tree Approach for Mining N-most Interesting Itemsets*. Data Mining and Knowledge Discovery; Theory, Tools, and Technology IV4730. pp.460–471.

Yin-Ling Cheung, Ada Wai-Chee Fu. (2004). *Mining Frequent Itemsets without Support Threshold:With and without Item Constraints*. IEEE Transactions on Knowledge and Data Engineering, Vol.16, No.9, September, 2004. pp.1052–1069.

ภาคผนวก

ภาคผนวก ก

เอกสารรับรองผลการพิจารณาจริยธรรมการวิจัยในมนุษย์

ที่ ๐๐๓/๒๕๕๘

เอกสารรับรองผลการพิจารณาจริยธรรมการวิจัยในมนุษย์
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

คณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา ได้พิจารณาเค้าโครงวิทยานิพนธ์

เรื่อง การพัฒนาการค้นหารูปแบบปรากฏบ่อยสุดเคอันดับแรกและปรากฏอย่างสม่ำเสมอจากฐานข้อมูลรายการ

หัวหน้าโครงการวิจัย นายปรีชา สิทธิชัยทวีกุล นิสิตระดับบัณฑิตศึกษา

คณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา ได้พิจารณาแล้วเห็นว่า เค้าโครงวิทยานิพนธ์ดังกล่าวเป็นไปตามหลักจริยธรรมการวิจัยในมนุษย์ โดยที่ผู้วิจัยเคารพสิทธิและศักดิ์ศรีในความเป็นมนุษย์ ไม่มีการล่วงละเมิดสิทธิ สวัสดิภาพ และไม่ก่อให้เกิดภยันตรายแก่ตัวอย่างการวิจัยกลุ่มตัวอย่างและผู้เข้าร่วมในโครงการวิจัย

จึงเห็นสมควรให้ดำเนินการวิจัยในขอบข่ายของเค้าโครงวิทยานิพนธ์ที่เสนอได้ ตั้งแต่วันที่ออกเอกสารรับรองผลการพิจารณาจริยธรรมการวิจัยในมนุษย์ฉบับนี้จนถึงวันที่ ๓๐ เมษายน พ.ศ. ๒๕๕๙

ออกให้ ณ วันที่ ๑๙ ตุลาคม พ.ศ. ๒๕๕๘

ลงนาม

(ผู้ช่วยศาสตราจารย์ ดร.กฤษณะ ชินสาร)
ประธานคณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

ภาคผนวก ข

การเผยแพร่ผลงานวิทยานิพนธ์

Preecha Sittichaitaweekul and Komate Amphawan, "Enhancing quality of results on Top-k Frequent-Regular Pattern mining". In proceedings of 1st international conference on Engineering Science and Innovative Technology (ESIT-2014), 2014, p.64.

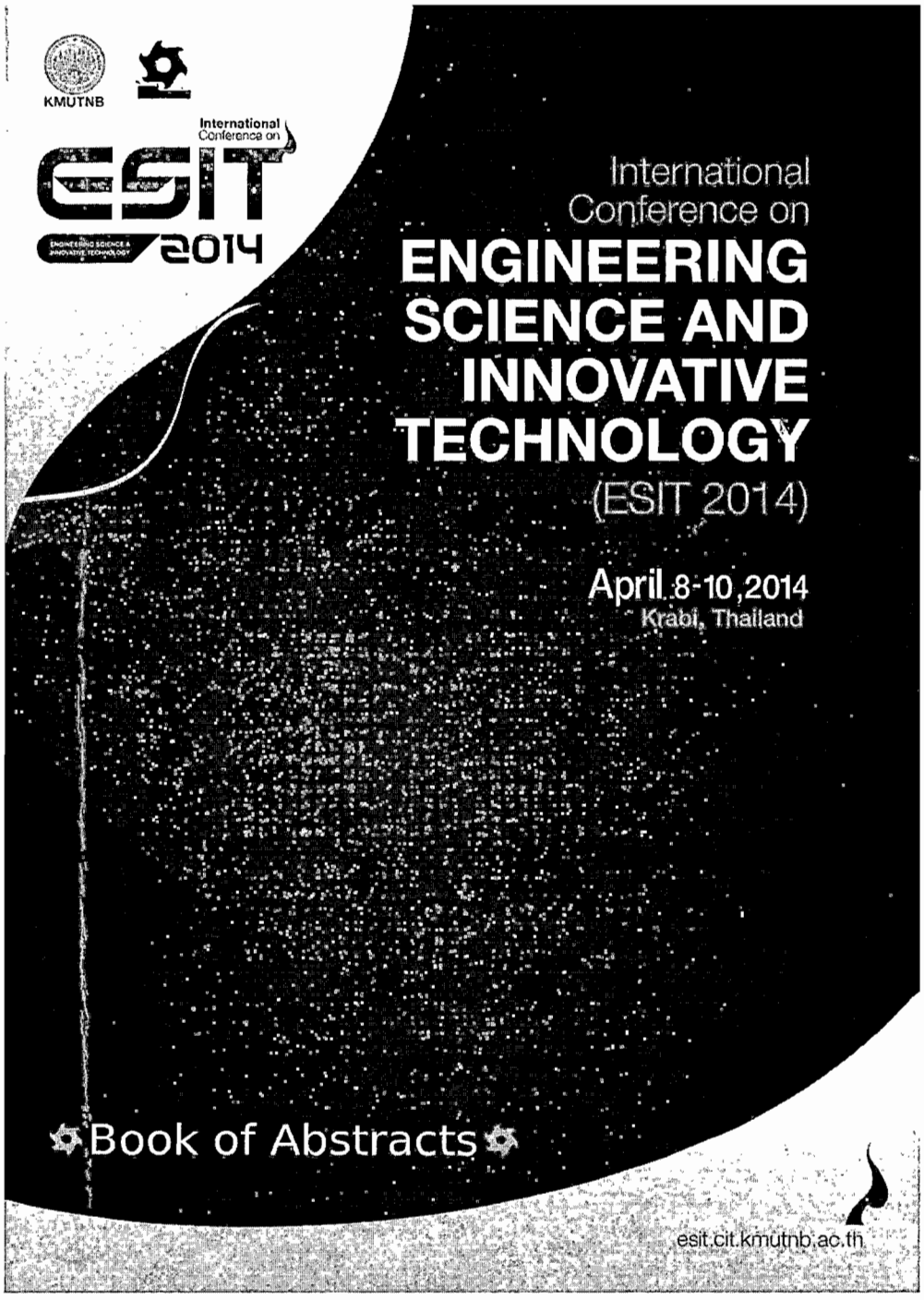

KMUTNB
International Conference on
ESIT 2014
ENGINEERING SCIENCE & INNOVATIVE TECHNOLOGY
International Conference on
ENGINEERING SCIENCE AND INNOVATIVE TECHNOLOGY
(ESIT 2014)
April 8-10, 2014
Krabi, Thailand
Book of Abstracts
esit.cit.kmutnb.ac.th

Table of Contents

Electrical and Electronics Engineering

Computer Science

Enhancing quality of results on Top-*k* Frequent-Regular Pattern Mining

Preecha Sittichaitaweekul and Komate Amphawan*

Abstract

In this paper, we study the problem of frequent-regular pattern mining for extracting patterns based on their occurrence behavior. This requires two parameters, support and regularity thresholds, to measure the significant or the important of patterns based on the frequency and regularity of occurrence. However, it is well-known that setting of support threshold is typically difficult. Hence, the top-*k* frequent-regular pattern mining has been introduced to avoid these difficulties by allowing users to specify the number of desired patterns (*k*) instead of support threshold. Nevertheless, this approach usually returns a large amount of 1-patterns (patterns with only one item) in which (*i*) it may cause redundancy on the set of results and (*ii*) it may not well identify relationships between objects or events appearing together. Therefore, in this paper, we introduce an alternative approach to mine top-*k* frequent-regular patterns without 1-patterns included in the results set. This can help to alleviate redundancy of results and to gain longer patterns in which users can better discover relationship or knowledge from the discovered patterns. To mine such patterns, we propose an efficient single-pass algorithm, called *ETFRP*, applying best-first search strategy to quickly discover the results and employing a linked-list structure to maintain patterns during mining process. Experimental studies show that *ETFRP* can effectively and efficiently discover patterns that meet the users' interest.

Keywords : Data mining, Frequent-regular pattern, Top-*k* frequent-regular pattern, Quality of results

Computational Innovation Laboratory, Faculty of Informatics, Burapha University, Thailand.
* Corresponding author, E-mail: komate@buu.ac.th

Paper ID #27

1st International Conference on Engineering Science and Innovative Technology (ESIT 2014), Krabi, Thailand, April 9 - 10 2014

Enhancing quality of results on Top-*k* Frequent-Regular Pattern Mining

Preecha Sittichaitaweekul and Komate Amphawan*

Abstract

In this paper, we study the problem of frequent-regular pattern mining for extracting patterns based on their occurrence behavior. This requires two parameters, support and regularity thresholds, to measure the significant or the important of patterns based on the frequency and regularity of occurrence. However, it is well-known that setting of support threshold is typically difficult. Hence, the top-*k* frequent-regular pattern mining has been introduced to avoid these difficulties by allowing users to specify the number of desired patterns (*k*) instead of support threshold. Nevertheless, this approach usually returns a large amount of 1-pattern (pattern with only one item) in which (*i*) it may cause redundancy on the set of results and (*ii*) it may not well identify relationships between objects or events appearing together. Therefore, in this paper, we introduce an alternative approach to mine top-*k* frequent-regular patterns without 1-patterns included in the results set. This can help to alleviate redundancy of results and to gain longer patterns in which users can better discover relationship or knowledge from the discovered patterns. To mine such patterns, we propose an efficient single-pass algorithm, called *ETFRP*, applying best-first search strategy to quickly discover the results and employing a linked-list structure to maintain patterns during mining process. Experimental studies show that *ETFRP* can effectively and efficiently discover patterns that meet the users' interest.

Keywords : Data mining, Frequent-regular pattern, Top-*k* frequent-regular pattern, Quality of results

Computational Innovation Laboratory, Faculty of Informatics, Burapha University, Thailand.

* Corresponding author, E-mail: komate@buu.ac.th

International Conference on Engineering Science and Innovative Technology 2014 Conference Programme Overview

Day 1: April 8, 2014			
8.00-8.30	Registration and Upload presentation files (In front of Andaman Grand Room)		
8.30-9.00	Welcome Chairperson and VIP Guests		
9.00-10.00	*Opening Ceremony (Andaman Grand Room)* Welcome addresses speech by Dr. Rattanakorn Phadungthin (Chair of ESIT International Conference) Prof. Dr. Juergen van der List (Honorary Chair of ESIT International Conference) Opening Remark by Prof. Dr. Teeravuti Boonyasopol (President of KMUTNB)		
	Photo session and coffee break		
10.00-10.40	Keynote lecture by Prof. Dr. Stefan Gast (Coburg University of Applied Science, Germany)		
10.40-11.20	Keynote lecture by Dr. Athanasios Tsolakis (University of Birmingham, United Kingdom)		
11.20-12.00	Keynote lecture by Prof. Dr. Reiner Dudziak (Bochum University of Applied Science, Germany)		
12.00-13.00	**Lunch break**		
Oral Presentation Session 1			
Track / Room	**Room 1: "Andaman II"**	**Room 2: "Krabi I"**	**Room 3: "Krabi II"**
13.00-13.20	**Paper ID.8** Efficient and Effective Testing of Automotive Software Product Lines	**Paper ID.7** The Performance Study of a Thin Film Electrode Solid Oxide Fuel Cell	**Paper ID.22** Self-Scaling Platform as a Service
13.20-13.40	**Paper ID.20** New Approaches for the Thermal Management of the Cabin of Battery Electric Vehicles	**Paper ID.9** Relationship between Customer's Perception and Design Parameter in Wood Furniture	**Paper ID.26** Traveling Salesman Problem Solver using Genetics Algorithm on CUDA Framework
13.40-14.00	**Paper ID.25** Reforming Exhaust Gas Recirculation of Diesel Fuel by Non-Thermal Plasma for NOx and PM Reduction in Diesel Engine	**Paper ID.11** Mechanical Properties of Woven Flax Fiber Biocomposites Based on Poly(Butylene Adipate-co-Terephthalate) (PBAT)	**Paper ID.27** Enhancing Quality of Results on Top-k Frequent-Regular Pattern mining
14.00-14.20	**Paper ID.29** Lightweight Design with Integrated Functions	**Paper ID.15** Case Study of Rigid Coupling Failure Analysis and Redesigned	**Paper ID.44** An Arduino Network Design for Multipoint Power Monitoring System
14.20-14.50	**Coffee break**		

KMUTNB

ESIT 2014

Certificate to Participation

This is to certify that

Preecha Sittichaitaweekul and Komate Amphawan

have participated

The 1st International Conference on Engineering Science and Innovative Technology

April 8-10, 2014 Sheraton Krabi Beach Resort, Krabi, THAILAND

Prof. Dr. Jürgen van der List
Honorary Chair of the ESIT Conference 2014

Asst. Prof. Preecha Ong-aree
Dean of College of Industrial Technology
King Mongkut's University of Technology North Bangkok

Dr. Rattanakorn Phadungthin
Chair of the ESIT Conference 2014

Enhancing quality of results on Top-*k* Frequent-Regular Pattern Mining

Preecha Sittichaitaweekul and Komate Amphawan*

Abstract

In this paper, we study the problem of frequent-regular pattern mining for extracting patterns based on their occurrence behavior. This requires two parameters, support and regularity thresholds, to measure the significant or the important of patterns based on the frequency and regularity of occurrence. However, it is well-known that setting of support threshold is typically difficult. Hence, the top-*k* frequent-regular pattern mining has been introduced to avoid these difficulties by allowing users to specify the number of desired patterns (*k*) instead of support threshold. Nevertheless, this approach usually returns a large amount of 1-patterns (patterns with only one item) in which (*i*) it may cause redundancy on the set of results and (*ii*) it may not well identify relationships between objects or events appearing together. Therefore, in this paper, we introduce an alternative approach to mine top-*k* frequent-regular patterns without 1-patterns included in the results set. This can help to alleviate redundancy of results and to gain longer patterns in which users can better discover relationship or knowledge from the discovered patterns. To mine such patterns, we propose an efficient single-pass algorithm, called *ETFRP*, applying best-first search strategy to quickly discover the results and employing a linked-list structure to maintain patterns during mining process. Experimental studies show that *ETFRP* can effectively and efficiently discover patterns that meet the users' interest.

Keywords : Data mining, Frequent-regular pattern, Top-*k* frequent-regular pattern, Quality of results

Computational Innovation Laboratory, Faculty of Informatics, Burapha University, Thailand.

* Corresponding author, E-mail: komate@buu.ac.th

1. Introduction

Frequent pattern mining is an important data-mining task that has been extensively studied [1-3]. It has a broad range of applications including analysis of customer purchase patterns, web-access, DNA sequences, etc. However, as pointed out in [4] that the use only support threshold to measure the interestingness or significant of patterns may not be sufficient. Then, Tanbeer et al. proposed the problem of mining frequent-regular patterns based on their occurrence behavior (*i.e.* whether a pattern occurs regularly, irregularly, or mostly in a specific time interval). This approach requires two parameters, support and regularity thresholds, to measure the significant of patterns which can be an important criteria in various applications (e.g. retail marketing, stock marketing, elderly daily habits' monitoring, etc.)

However, it is well-known that setting of support threshold is typically difficult and it is more reasonable to ask for the number of patterns to be mined. Then, there are several approaches that try to alleviate these difficulties such as mining top-k frequent patterns [5-7], mining N-most interesting frequent patterns [8-9], etc. Under this framework, there is an approach that mines patterns under support and regularity values that is top-k frequent-regular frequent patterns mining [10-13] which allows users to assign the number of desired patterns. Nevertheless, this approach is often return 1-patterns (*i.e.* patterns contain only one item) in which (*i*) it may causes redundancy on the set of results and (*ii*) it may not well identify relationships between objects or events appearing together. Thus, in this paper, we introduce an alternative approach to mine top-k frequent-regular patterns without 1-patterns included in the set of results. This can help users to gain the longer patterns in which they can better discover relation on objects. To mine such patterns, we propose an efficient single-pass algorithm, called *ETFRP*, which applies best-first search strategy to quickly discover the results and employs a linked-list structure to maintain patterns during mining process. The performance of the proposed technique is investigated via simulation experiments. From the results, we can notice that our proposed approach can effectively and efficiently discover patterns that meet the users' interest.

The rest of this paper is organized as follows. Section 2 gives the notations and definitions of the top-k frequent-regular pattern mining. Section 3 presents the *ETFRP* algorithm to quickly discover a set of desired patterns. Several experimental studies are conducted and investigated in section 4. We conclude our paper in section 5. Finally, the important references are shown in section 6.

2. Problem statements

Let $I = \{i_1, i_2, ..., i_m\}$ be the set of literals (also called *items*). A set $X \subseteq I$ is called a pattern (itemset), or a l-pattern when X contains l items. A transactional database $TDB = \{t_1, t_2, ..., t_n\}$ is a set of n transactions in which each transaction $t_q = (q, Y)$ is a 2-tuple containing: (*i*) a unique transaction identifier(tid) equal to q and (*ii*) a set of items Y. For t_q, if $X \subseteq Y$, it can be said that X occurs in t_q or t_q contains X which can be denoted as t^X_q. Then, a set of all *ordered tids* in which X occurs, $T^X = \{t^X_p, t^X_{p+1}, ..., t^X_q\}$ also called *tidset,* can be defined. To investigate the occurrence's frequency of X, we compute the support value of X by $s^X = |T^X|$ (*i.e.* the number of tids/transactions that contain X).

To observe the regularity of occurrence of X, any two consecutive tids in T^X, t^X_p and t^X_{p+1}, are considered. The regularity between the two consecutive tids (*i.e.* the number of tids not containing X between t^X_p and t^X_{p+1}) is calculated as $rtt^X_{p+1} = t^X_{p+1} - t^X_p$. In addition, the first regularity—the number of tids not containing X before its first appearance, $fr^X = t^X_1$, and the last regularity—the number of tids not containing X from the last occurrence of X to the end of database, $lr^X = |TDB| - t^X_{|T^X|}$, are also

calculated. Based on the regularity values mentioned above, we can define the total regularity of X as $r^X = max(fr^X, rtt^X_2, rtt^X_3, ..., rtt^X_{|T^X|}, lr^X)$, which is the maximum number of tids that X disappears from database.

With the regularity value, we can guarantee that X appears at least once in every set of r^X consecutive transactions. Thus, frequent-regular pattern mining is to discover patterns that frequently and regularly appear in a database. However, frequent-regular patterns mining still suffer from setting of appropriate thresholds. Hence, the top-k frequent-regular pattern mining [10] was proposed to avoid the difficulties of setting an appropriate support threshold in which it can be defined as follow:

Definition A pattern X is a top-k frequent-regular pattern if (*i*) its regularity is no greater than regularity threshold σ_r, and (*ii*) there is no more that k - *1* patterns that have support greater than that of X.

To mine a complete set of top-k frequent-regular patterns, it requires two parameters: (*i*) the number of desired results k, and (*ii*) regularity threshold σ_r. However, this may generate redundant patterns. Thus, in this paper, we aim to alleviate redundancy of patterns to be mined by eliminated 1-patterns from the set of results. Then, the results must have three properties as follows: (i) regularly appear in database (ii) have highest support and (iii) containing more than one item, respectively.

3. *ETFRP* algorithm

We here introduce an efficient single-pass algorithm namely *ETFRP* to quickly discover the k patterns with three properties as mentioned above. *ETFRP* employs a buffer and a top-k list structure to maintain all patterns during mining process and its mining process can be separated into two main steps: (*i*) ETFRP-intialization—scan the given transactional database to collect and maintain the regularity, support and occurrence information of all items in the buffer, and (*ii*) ETFRP-mining—mine the complete set of top-k frequent-regular patterns (without 1-patterns in the results set) using best-first search strategy and applying top-k list to maintain all the desired patterns.

A top-k list is a simple linked-list in where each entry consists of four tuples: (*i*) set of items containing more than one item (*i*) regularity value (*iii*) support value, and (*iv*) tidset, the set of tids in which the itemset occurs.

3.1 ETFRP-initailization

To capture the content of single items, a buffer for all items is created and then initialized. After that, each transaction of *TDB* is sequentially scanned and then each items in the transaction is regarded. Next, the regularity, support and tidset of the regarded item are updated in the buffer. After scanning all transactions, the exact regularity of all items are calculated. Then, items with regularity greater than the regularity threshold are eliminated from the buffer and all the items are sorted by descending order of support. At the end, we gain a list of sorted items with regular appearance that can be utilized to mine patterns in the next step. Details of ETFRP-initialization are shown in Algorithm 1.

Algorithm 1: ETFRP-initailization
Input: A transactional database *TDB*, a number desired pattern k, and a regularity threshold σ_r
Output: A list of sorted items with regular appearance
(1) create and initialize a buffer for all items
(2) for each transaction t_q in *TDB*
(3) for each item i_x in transaction t_q
(4) update the value of support, regularity and collect t_q in the tidset of i_x's entry
(5) for each item i_y in the buffer
(6) compute lr^i_y and r^i_y
(7) if $r^i_y < \sigma_r$
(8) remove i_y out of the buffer
(9) sort all items in the buffer by support descending order

3.2 ETFRP-mining

As shown the details in Algorithm 2, ETFRP-mining starts to create a set of 2-pattern with regular occurrence and have highest support. To do that, the best-first search strategy is applied where it considers a pair of items in the buffer from the most frequent to the least one (the most frequent items tend to generate the most frequent itemsets). Then, the tidset of the two considered items are sequentially intersected in order to calculate support, regularity and the collect the set of tids that the two items appearing together. In the case that the regularity value of the new generated 2-pattern is no greater than regularity threshold and its support is no less than the support of the k^{th} pattern in the sorted top-k list, the entry of the k^{th} is eliminated from the top-k list, since it cannot be the results (based on downward closure property[1]). Then, an entry of the new generated 2-pattern is created (with its information: (i) itemset, (ii) regularity, (iii) support, and (iv) tidset) and inserted into the top-k list by support descending order. After consider all pairs of items in the buffer, we gain a set of sorted 2-patterns contained in the top-k list.

Next, a complete set of top-k frequent-regular pattern without 1-patterns is generated. With the using of best-first search strategy, a pair of patterns in the top-k list with highest support is firstly considered. Then, the two patterns are merged together to generate longer pattern if (i) they have the same number of items, and (ii) they have same the same prefix, i.e. they have the same items except only the last item. When the two patterns meet the two conditions above, their tidsets are sequentially intersected and collected. Consequently, the regularity and support of the new generated pattern are calculated. If the regularity of the new pattern is no greater than regularity threshold and the its support is no less than that of the k^{th} pattern, the entry of the k^{th} is eliminated out of the top-k list and the entry of the new generated pattern is created (with its information) and then inserted into the top-k list by support descending order. Finally, we gain the set of sorted k patterns with three properties.

Algorithm 2: ETFRP-mining

Input: A buffer containing all single items with regular appearance, the number of desired patterns k, and the regularity threshold σ_r

Output: A top-k list containing the sorted k regular patterns with highest support and contains more than one items.

(1) for each item x in the buffer
(2) for each item y in the buffer
(3) sequentially intersect T^x and T^y and then collect the result in T^{xy}
(4) calculate s^{xy} and r^{xy}
(5) if $r^{xy} \leq \sigma_r$ and $s^{xy} > s_k$
(6) remove the entry of the k^{th} pattern out of the top-k list
(7) create an entry for pattern xy with its support, regularity and tidset and then insert into the top-k list by support descending order
(8) for each entry of pattern P in the top-k list
(9) for each entry of pattern Q in the top-k list
(10) if $|P| = |Q|$ and $p_1 = q_1, p_2 = q_2, \ldots, p_{|P|-1} = q_{|Q|-1}$
(10) sequentially intersect T^P and T^Q and then collect the result in T^{PQ}
(11) calculate s^{PQ} and r^{PQ}
(12) if $r^{PQ} \leq \sigma_r$ and $s^{PQ} > s_k$
(13) create an entry for pattern PQ with its support, regularity and tidset and then insert into the top-k list by support descending order

4. Performance study

In this section, we here report experimental studies done to investigate the performance of the proposed

ETFRP. From the best of our knowledge, there is no approach that aims to improve the quality of results on top-k frequent-regular pattern mining. Then, there is no comparative study in this paper. However, we can use *ETFRP* as the base line on this approach.

Due to limitation of space, we used two well-known datasets downloaded from http://fimi.ua.ac.be/data/ to investigate the performance of *ETFRP*, that is *T10I4D100K* (with 100,000 transactions, 1,000 items and average length of transaction equal to 10), and retail (collected from real super market with 88,163 transactions, 16,470 items and average length of transaction equal to 10.3) Three types of experiments that aim to investigate computational time and memory consumption and then to observe length of discovered results were conducted, respectively.

As shown in table 1 and 2, the runtime of *ETFRP* with three specific regularity thresholds (*i.e.* σ_r = 2%, 5%, and 10%). From these figures, we can observe that runtime increases as the value of k and/or σ_r increases. With the increasing of k, *ETFRP* has to mine more results, then runtime increases as well. Meanwhile, in the case that σ_r increases, it causes the increasing of the number of patterns that have regularity value less than σ_r. Then, *ETFRP* has to consider a larger group of patterns since it cannot prune patterns by using only the threshold σ_r.

Table 1. Runtime of *ETFRP* on T10I4D100K

T10I4D100K			
Amount	Time(s)		
K	αp=2%	αp=5%	αp=10%
100	3.894	4.675	5.885
200	4.317	5.085	6.371
500	5.152	6.329	7.764
1000	5.897	6.975	8.338
1500	6.218	7.164	8.700
2000	6.290	7.528	9.056

The investigation of memory consumption is shown in table 3. Within the table, we can observe, it increases as the value of the value of k increases since we have to store and maintain more patterns in memory. In addition, memory usage also increases as the σ_r increases. This is due to the fact that there will be a group of patterns with high support and high regularity values included in the set of results. Then, *ETFRP* needs more memory to maintain itemset's information (*i.e.* tidsets) of the group of patterns.

Table 2. Runtime on *ETFRP* on *Retail*

Retail			
Amount	Time(s)		
k	αp=2%	αp=5%	αp=10%
100	0.445	0.541	0.608
200	0.640	0.928	1.012
500	0.679	2.606	2.827
1000	0.678	5.107	6.273
1500	0.682	7.698	9.269
2000	0.676	0.421	0.490

Table 3. Memory consumption of *ETFRP*

Memory usage (MB)						
k	Regularity(*Retail*)			Regularity(*T10I4D100K*)		
	2%	5%	10%	2%	5%	10%
100	1.73	2.60	3.17	4.06	4.16	4.18
200	1.94	2.83	3.40	4.32	4.43	4.44
500	2.08	3.27	3.86	4.98	5.08	5.09
1000	2.08	3.73	4.34	5.86	5.95	5.97
1500	2.08	4.07	4.70	6.65	6.76	6.77
2000	2.08	4.31	4.99	7.35	7.48	7.50

Table 4. Length of results on *T10I4D100K*

T10I4D100K Regularity 5%												
K	*ETFRP* Pattern size						*MTKPP* Pattern size					
	2	3	4	5	6	7	1	2	3	4	5	6
100	73	22	5	-	-	-	100	0	0	0	0	0
200	143	42	13	2	-	-	200	0	0	0	0	0
500	342	110	43	9	-	-	445	47	9	0	0	0
1000	617	289	82	11	1	-	560	294	106	37	3	0
1500	836	414	187	59	11	1	606	576	252	61	11	1
2000	1128	557	253	65	12	1	629	760	378	174	56	3

Table 5. Length of results on *Retail*

Retail Regularity 5%										
K	*ETFRP* Pattern size				*MTKPP* Pattern size					
	2	3	4	5	1	2	3	4	5	
100	71	25	4	-	43	38	17	2	0	
200	145	51	4	-	89	79	28	4	0	
500	351	133	16	-	208	211	74	8	0	
1000	677	286	38	3	385	431	167	21	0	
1500	1026	426	62	3	585	623	264	34	2	
2000	1328	577	95	3	740	866	358	51	3	

Lastly, size of patterns in the set of results is considered. As illustrated in table 4 and 5, we can see that top-k frequent-regular patterns mining algorithm, *MTKPP*, mostly generates short patterns. In all cases, there is at

least 30% of 1-itemset included in the set of results. Meanwhile, our proposed can discover longer patterns in which can help user to gain more knowledge and information.

5. Conclusion

In this paper, we study the problem of top-*k* frequent-regular pattern mining and then try to eliminate the set of 1-patterns out of the results set. This may help to alleviate redundancy of patterns and yield longer patterns in which users can better discover relationship and can gain more knowledge. To mine such patterns, an efficient single-pass with best-first search strategy, called *ETFRP*, is also introduced. The experimental results demonstrate that the proposed *ETFRP* can provide time and memory efficiency during mining process. Moreover, it is highly scalable in terms of runtime and memory usage.

6. Reference

[1] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases", Proceedings of 20th International Conference on Very Large Data Bases (VLDB), 1994, pp. 487 – 499.

[2] J. Han, J. Pei, Y. Yin and R. Mao, "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach", Data Mining Knowledge Discovery, 8, 2004, pp. 53 – 87.

[3] J. H. Change, "Mining weighted patterns in a sequence database with a time-interval weight", Knowledge-based Systems, 24, 2011, pp. 1 – 9.

[4] S. Tanbeer, C. Ahmed, B. Jeong and Y. Lee, "Discovering Periodic-Frequent Patterns in Transactional Databases", Advances in Knowledge Discovery and Data Mining, 5476, 2009, pp. 242 – 253.

[5] J. Han, J. Wang, Y. Lu and P. Tzvetkov, "Mining top-k frequent closed patterns without minimum support", Proceedings IEEE International Conference on Data Mining (ICDM), 2002, pp. 211 – 218.

[6] P. Tzvetkov, X. Yan and J. Han, "TSP: Mining top-k closed sequential patterns", Knowledge and Information Systems, 7, 2005, 438 – 457.

[7] J. Wang, J. Han Y. Lu and P. Tzvetkov, "TFP: an efficient algorithm for mining top-k frequent closed itemsets", IEEE transactions on Knowledge and Data Engineering, 17, 2005, 652 – 663.

[8] A. W. Fu, R. W. Kwong and J. Tang, "Miing N-most Interesting Itemsets", Foundations of Intelligent Systems, 1932, 2000, pp. 59 – 67.

[9] S. Ngan, T. Lam, R. C. Wong and A. W. Fu, "Mining N-most interesting itemsets without support threshold by the COFI-tree", International Journal of Business Intelligence and Data Mining, 1, 2005, pp. 88 – 106.

[10] K. Amphawan, P. Lenca, and A. Surarerks, "Efficient mining top-*k* regular-frequent itemset using compressed tidsets", New Frontiers in Applied Data Mining, 2012, 124 – 135.

[11] K. Amphawan, P. Lenca, and A. Surarerks, "Mining top-*k* regular-frequent itemsets using database partitioning and support extimation", Expert Systems with Applications, 39, 2012, pp. 1924 – 1936.

[12] K. Amphawan, P. Lenca, and A. Surarerks, "Mining top-k Periodic-Frequent Pattern from Transactional Databases without Support Threshold", Proceedings of the 3rd International Conference on Advances in Information Technology (IAIT), Bangkok, Thailand, 2009, pp. 18 – 29.

[13] K. Amphawan and P. Lenca, "Mining top-*k* frequent-regular patterns based on user-given trade-off between frequency and regularity", Proceedings of the 6th International Conference on Advances in Information Technology (IAIT), Bangkok, Thailand, 2013.