การค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ

กิตติพา คลังวิสาร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา
กรกฎาคม พ.ศ. 2561

MINING WEIGHTED-FREQUENT-REGULAR ITEMSETS FROM TRANSACTIONAL DATABASE

KITTIPA KLANGWISAN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE MASTER DEGREE OF SCIENCE IN COMPUTER SCIENCE
FACULTY OF INFORMATICS BURAPHA UNIVERSITY
2018.

คณะกรรมการควบคุมวิทยานิพนธ์และคณะกรรมการสอบวิทยานิพนธ์ได้พิจารณาวิทยานิพนธ์ของ กิตติพา คลังวิสาร ฉบับนี้แล้ว เห็นสมควรรับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ของมหาวิทยาลัยบูรพาได้

คณะกรรมการควบคุมวิทยานิพนธ์

.....อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ ดร.โกเมศ อัมพวัน)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธาน
(ผู้ช่วยศาสตราจารย์ ดร.อนุชิต จิตพัฒนกุล)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.โกเมศ อัมพวัน)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.อุรีรัฐ สุขสวัสดิ์ชน)

คณะวิทยาการสารสนเทศ อนุมัติให้รับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ของมหาวิทยาลัยบูรพา

.....คณบดีคณะวิทยาการสารสนเทศ
(ผู้ช่วยศาสตราจารย์ ดร.กฤษณะ ชินสาร)
วันที่ 23 เดือน กรกฎาคม พ.ศ. 2561

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยความกรุณาจาก ผู้ช่วยศาสตราจารย์ ดร.โกเมศ อัมพวัน อาจารย์ที่ปรึกษา ที่กรุณาให้คำปรึกษาแนะนำแนวทางที่ถูกต้อง ตลอดจนแก้ไขข้อบกพร่องต่าง ๆ ด้วยความละเอียดถี่ถ้วนและเอาใจใส่ด้วยดีเสมอมา ผู้วิจัยรู้สึกซาบซึ้งเป็นอย่างยิ่ง จึงขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

เนื่องจากวิทยานิพนธ์นี้ได้รับทุนอุดหนุนวิทยานิพนธ์และดุษฎีนิพนธ์จากมหาวิทยาลัยบูรพา ปีงบประมาณ ๒๕๖๐ จึงขอขอบพระคุณ ณ ที่นี้ด้วย

ขอกราบขอบพระคุณ คุณพ่อสุพจน์ คุณแม่มาลี คลังวิสาร และพี่ ๆ ทุกคนที่ให้กำลังใจและสนับสนุนผู้วิจัยเสมอมา

คุณค่าและประโยชน์ของวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบเป็นกตัญญูกตเวทิตาแด่ บุพการี บูรพาจารย์ และผู้มีพระคุณทุกท่านทั้งในอดีตและปัจจุบัน ที่ทำให้ข้าพเจ้าเป็นผู้มีการศึกษาและประสบความสำเร็จมาจนตราบเท่าทุกวันนี้

กิตติพา คลังวิสาร

56910495: สาขาวิชา: วิทยาการคอมพิวเตอร์; วท.ม. (วิทยาการคอมพิวเตอร์)
คำสำคัญ: การทำเหมืองข้อมูล/ การค้นหากฎความสัมพันธ์/ การค้นหาเซตรายการ/ เซตรายการที่ปรากฏบ่อยและสม่ำเสมอ/ ค่าน้ำหนักความสำคัญ

กิตติพา คลังวิสาร: การค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ (MINING WEIGHTED-FREQUENT-REGULAR ITEMSETS FROM TRANSACTIONAL DATABASE) คณะกรรมการควบคุมวิทยานิพนธ์: โกเมศ อัมพวัน, Ph.D., 109 หน้า. ปี พ.ศ. 2561

การค้นหาเซตรายการ/รูปแบบที่ปรากฏบ่อยและสม่ำเสมอได้ถูกค้นคว้าและนำเสนออย่างแพร่หลายในการค้นหาเซตรายการที่มีความน่าสนใจจากฐานข้อมูล ซึ่งวิธีการดั้งเดิมสำหรับการค้นหาเซตรายการดังกล่าวจะพิจารณาถึงพฤติกรรม/รูปแบบในการปรากฏของข้อมูล โดยพิจารณาเพียงความถี่หรือจำนวนครั้งและความสม่ำเสมอในการปรากฏขึ้นของข้อมูล แต่อย่างไรก็ตามในการประยุกต์ใช้งานจริงนั้นข้อมูลหรือเซตรายการที่ต้องการค้นหาจากฐานข้อมูลสามารถมีความสำคัญหรือความน่าสนใจที่แตกต่างกันส่งผลให้วิธีการดั้งเดิมไม่สามารถตอบสนองต่อความต้องการดังกล่าวได้ ดังนั้นงานวิทยานิพนธ์นี้จึงได้มีการนำเสนอปัญหาและวิธีการสำหรับการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ (Weighted-Frequent-Regular Itemsets Miner, WFRIM) ซึ่งจะสามารถค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอในฐานข้อมูลภายใต้เงื่อนไขที่เซตรายการมีความสำคัญหรือความน่าสนใจที่แตกต่างกัน โดยวิธีการ WFRIM ใช้โครงสร้างต้นไม้ที่เรียกว่า WFRI-tree ในการจัดเก็บข้อมูลและใช้เทคนิค WFRIM-growth ในการค้นหาเซตรายการที่เป็นผลลัพธ์ในขนาดต่าง ๆ ต่อมาได้ทำการพัฒนาปรับปรุงขั้นตอนวิธีในการค้นหาเซตรายการที่นำเสนอให้มีประสิทธิภาพเพิ่มมากขึ้นที่เรียกว่า Weighted-Frequent-Regular Itemset Miner using Inverval Word Segment structure (WFRIM-IWS) โดยใช้การจัดเก็บข้อมูลในรูปแบบไดนามิกบิตเวกเตอร์ (dynamic bit-vector) ที่เรียกว่า โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง (interval word segment, IWS) ซึ่งทั้งสองขั้นตอนวิธีได้มีการประยุกต์ใช้เทคนิคการคำนวณหาค่าน้ำหนักที่มากสุด (Global maximum weight) และค่าน้ำหนักที่มากสุดของเซตรายการที่พิจารณา (Local maximum weight) เพื่อทำการลดทอนปริภูมิสถานะและเวลาในการประมวลผล จากผลการทดลองในฐานข้อมูลสังเคราะห์และฐานข้อมูลจริงแสดงให้เห็นว่าทั้งสองขั้นตอนวิธีสามารถค้นหาเซตรายการ/รูปแบบที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการได้อย่างมีประสิทธิภาพ โดยขั้นตอนวิธี WFRIM-IWS สามารถประมวลผลได้รวดเร็วและใช้หน่วยความจำน้อยกว่าขั้นตอนวิธี WFRIM

56910495: MAJOR: COMPUTER SCIENCE; M.Sc. (COMPUTER SCIENCE)
KEYWORD: DATA MINING/ ASSOCIATION RULE/ ITEMSETS MINING/ FREQUENT-REGULAR ITEMSETS/ WEIGHT OF IMPORTANCE
KITTIPA KLANGWISAN: MINING WEIGHTED-FREQUENT-REGULAR ITEMSETS FROM TRANSACTIONAL DATABASE. ADVISORY COMMITTEE: KOMATE AMPHAWAN, Ph.D., 109 P. 2018.

Frequent-regular itemsets/petterns mining has been explored and proposed to find interesting itemsets in a database based on their own occurrence behavior. Traditionally, an itemset can be identified as interesting by considering only frequency and regularity of an itemset occurred in the database. However, itemsets can have different degree of importance which traditional approach may affect the missing of important/interesting knowledge in real-world applications. In this thesis, we introduce approaches on mining weighted-frequent-regular itemsets (also called mining WFRIs). in which the first approach is called Weighted-Frequent-Regular Itemsets Miner (WFRIM) by using FP-tree like structure named WFRI-tree to maintain candidate itemsets during mining process and using WFRIM-growth technique to mine WFRIs. To improve WFRIM, the second approach is proposed called Weighted-Frequent-Regular Itemset Miner using Interval Word Segment structure (WFRIM-IWS). The dynamic bit vector is utilized for maintaining occurrence information of each itemset named interval word segments structure (IWS). The both approaches apply the concept of overestimated weighted-frequency and global/local maximum weights to early prune search space and reduce computational time. From experimental results on synthetic and real datasets, the both approaches can exhibit to discover weighted-frequent-regular itemsets efficiently. In addition, WFRIM-IWS outperforms WFRIM in the terms of computational time and memory consumptio

สารบัญ

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญภาพ

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

บทที่ 1
บทนำ

ที่มาและความสำคัญของวิทยานิพนธ์

ในยุคปัจจุบันเป็นยุคที่เทคโนโลยีและระบบสารสนเทศเข้ามามีบทบาทและมีความสำคัญในการดำเนินชีวิตประจำวันของมนุษย์ กอปรกับการดำเนินธุรกิจประยุกต์ใช้เทคโนโลยีเพื่อขับเคลื่อนองค์กร สร้างการแข่งขันทางการตลาดและอื่น ๆ จึงเป็นเหตุให้ยุคปัจจุบันที่ซึ่งเป็นยุคที่แต่ละองค์กรมีการจัดเก็บข้อมูลเป็นจำนวนมาก และยังเป็นยุคที่องค์กรต่าง ๆ มีการประมวลผล/วิเคราะห์ข้อมูลที่ถูกจัดเก็บไว้เพื่อสนับสนุนการตัดสินใจในการวางแผน ปรับกลยุทธ์ทางการแข่งขันทางการตลาดและสามารถทำการวิเคราะห์พฤติกรรมของผู้บริโภค ซึ่งการดำเนินการดังกล่าวจะสามารถช่วยเพิ่มผลกำไร ความเชื่อมั่น และความน่าสนใจของธุรกิจนั้น ๆ ได้

จากความต้องการในการประมวลผล/วิเคราะห์ข้อมูลข้างต้น การค้นหากฎความสัมพันธ์ (Association rule mining) (Agrawal, Imielinski, & Swami, 1993) เป็นกระบวนการหนึ่งที่ได้รับความนิยมในการสกัดข้อมูลเพื่อให้ได้สารสนเทศที่น่าสนใจซึ่งการค้นหากฎความสัมพันธ์นั้นมีขั้นตอนในการทำงาน 2 ขั้นตอน ประกอบด้วย 1) ขั้นตอนการค้นหาเซตรายการที่ปรากฏบ่อย (Frequent itemset/pattern mining) และ 2) ขั้นตอนการสร้างกฎความสัมพันธ์ โดยในเริ่มแรกนั้นการค้นหากฎความสัมพันธ์ถูกคิดค้นขึ้นเพื่อค้นหาความสัมพันธ์ของข้อมูลในธุรกิจการค้าปลีก (Retail business) (Agrawal, & Srikant, 1994) ซึ่งจะทำการวิเคราะห์สินค้าในตะกร้าสินค้า (Market basket analysis) เพื่อศึกษาพฤติกรรมการซื้อสินค้าของลูกค้าโดยจะทำการค้นหากลุ่มของสินค้าที่ลูกค้าซื้อร่วมกันบ่อยครั้งจากนั้นนำผลลัพธ์ที่ได้ไปใช้ประโยชน์ในการจัดทำโปรโมชันสินค้า การจัดวางสินค้าและใช้สำหรับการวางแผนทางการตลาด เป็นต้น ต่อมาได้มีการนำการค้นหากฎความสัมพันธ์และการค้นหาเซตรายการที่ปรากฏบ่อยไปประยุกต์ใช้กับงานอื่น ๆ เช่น 1) ในการวิเคราะห์ข้อมูลทางพันธุกรรม (DNA analysis) (Cong, Tung, Xu, Pan, & Yang, 2004) จะทำการค้นหาคู่ของ DNA ที่ปรากฏร่วมกันบ่อยครั้งแล้วเป็นสาเหตุที่ก่อให้เกิดโรค 2) ในทางการแพทย์ (Serban, Czibula, & Campan, 2006) จะทำการวินิจฉัยผู้ป่วยหรือทำนายโรคของผู้ป่วยจากกลุ่มอาการที่ปรากฏร่วมกันบ่อย ๆ 3) ในการวิเคราะห์ข้อมูลสำมะโนประชากร (Census data) (Malerba, Esposito, & Lisi, 2001) โดยการพิจารณาความสัมพันธ์ของลักษณะภูมิประเทศและลักษณะประชากรที่อาศัยอยู่ในพื้นที่นั้น ๆ เช่น อาชีพ เพศ อายุ รายได้ การศึกษา และอื่น ๆ เพื่อใช้ประกอบการตัดสินใจในการวางแผนการบริการสาธารณะ (การศึกษา การขนส่ง และสุขภาพ) รวมทั้งการประกอบธุรกิจ (โรงงานอุตสาหกรรม ห้างสรรพสินค้า และธนาคาร) เป็นต้น

จากการประยุกต์ใช้กับงานในด้านต่าง ๆ ที่กล่าวข้างต้นส่งผลให้การค้นหาเซตรายการที่ปรากฏบ่อยโดยการพิจารณากลุ่มของข้อมูลที่ปรากฏร่วมกันจะใช้จำนวนครั้งในการปรากฏหรือความถี่ของการปรากฏเพียงอย่างเดียว ซึ่งการพิจารณาดังกล่าวอาจไม่เพียงพอหรือตอบสนองต่อความต้องการสำหรับงานที่จะนำไปประยุกต์ใช้ จึงเป็นเหตุให้มีนักวิจัยคิดค้นและพัฒนาการค้นหาเซตรายการที่ปรากฏบ่อยในหลากหลายแง่มุม เช่น การค้นหาเซตรายการที่ปรากฏบ่อยในข้อมูลกระแส (Frequent itemsets mining in stream data) (Lin, Chiu, Wu, & Chen, 2005) การค้นหาเซตรายการแบบใกล้ชิด (Closed frequent itemsets) (Yan, Han, & Afshar, 2003) การค้นหาเซตรายการที่มีค่าคุณประโยชน์สูง (High utility itemsets mining) (Li, Huang, Chen, Liu, & Lee, 2008) การค้นหาเซตรายการที่มีการปรากฏร่วมกันอย่างเป็นลำดับ (Sequential itemsets mining) (Agrawal, & Srikant, 1995) และอื่น ๆ ต่อมาในปี ค.ศ. 2009 Tanbeer, Ahmed, Jeong, and Lee (2009) นำเสนอการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอ (Frequent-regular itemsets mining) เป็นวิธีการหนึ่งในการค้นหาเซตรายการที่น่าสนใจซึ่งได้รับความสนใจเป็นอย่างมาก โดยจะค้นหาเซตรายการที่มีลักษณะการปรากฏบ่อย ๆ และปรากฏอย่างสม่ำเสมอในฐานข้อมูล จากนั้นมีหลายงานวิจัยได้นำการค้นหาเซตรายการดังกล่าวไปพัฒนาต่อยอด เช่น การค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอในฐานข้อมูลที่เพิ่มขึ้นหรือในฐานข้อมูลกระแส (Frequent-regular itemsets mining on incremental database or data stream) (Tanbeer, Ahmed, & Jeong, 2010a, 2010b) การค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอที่มีค่าคุณประโยชน์สูง (High utility-regular itemsets mining) (Fournier-Viger, Lin, Duong, & Dam, 2016) และอื่น ๆ

จากแนวคิดพื้นฐานของการค้นหาเซตรายการที่ปรากฏบ่อยและเซตรายการที่ปรากฏบ่อยและสม่ำเสมอที่กล่าวมาข้างต้นจะทำการค้นหาเซตรายการโดยที่แต่ละรายการมีความสำคัญหรือความน่าสนใจเท่ากัน แต่สำหรับในการประยุกต์ใช้งานจริงหลาย ๆ แอปพลิเคชันแต่ละรายการสามารถมีความสำคัญ/ความน่าสนใจที่แตกต่างกัน ตัวอย่างเช่น ในธุรกิจค้าปลีกหรือห้างสรรพสินค้า สินค้าแซลมอนและไวน์มีความสำคัญ/น่าสนใจมากกว่าสินค้าโยเกิร์ตและกราโนลา ถ้าหากสินค้าดังกล่าวมีจำนวนครั้งในการซื้อเท่ากันหรือสินค้าแซลมอนและไวน์มีจำนวนครั้งในการซื้อน้อยกว่าสินค้าโยเกิร์ตและกราโนลา เนื่องจากแซลมอนและไวน์มีผลกำไรและมูลค่าทางการตลาดมากกว่าโยเกิร์ตและกราโนลา จากกรณีดังกล่าวแสดงให้เห็นถึงความสำคัญหรือความน่าสนใจที่แตกต่างกันระหว่างกลุ่มของสินค้าทั้งสอง ด้วยเหตุนี้การค้นหาเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ (Frequent weighted itemsets mining) (Cai, Fu, Cheng, & Kwong, 1998) (Wang, Yang, & Yu, 2004) (Ahmed, Tanbeer, Jeong, & Lee, 2008) (Vo & Coenen, 2013) จึงได้ถูกนำเสนอขึ้นในหลายงานวิจัย อย่างไรก็ตามงานวิจัยที่มีอยู่มีจุดประสงค์หลัก

คือการค้นหาเซตรายการโดยพิจารณาจำนวนครั้ง/ความถี่ในการปรากฏภายใต้เงื่อนไขที่แต่ละรายการมีความสำคัญ/ความน่าสนใจที่แตกต่างกัน ซึ่งอาจไม่เพียงพอในการพิจารณาถึงลักษณะหรือพฤติกรรมในการปรากฏของเซตรายการที่น่าสนใจในฐานข้อมูล

ดังนั้นวิทยานิพนธ์นี้จึงได้นำเสนอปัญหาการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ ที่ซึ่งจะสามารถค้นหาเซตรายการที่มีความสำคัญ/ความน่าสนใจที่แตกต่างกันและมีการปรากฏบ่อย ๆ อย่างสม่ำเสมอในฐานข้อมูล โดยทำการกำหนดนิยามและขอบเขตของปัญหา (ดังแสดงรายละเอียดในบทที่ 3) จากนั้นได้นำเสนอขั้นตอนวิธีสำหรับการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการที่มีชื่อเรียกว่า *Weighted-Frequent-Regular Itemsets Miner (WFRIM)* โดยจัดเก็บข้อมูลไว้ในโครงสร้างต้นไม้ที่เรียกว่า *WFRI-tree* จากนั้นทำการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการขนาดต่าง ๆ ด้วยเทคนิค *WFRIM-growth* ผสมผสานกับการประยุกต์ใช้เทคนิคการคำนวณหาค่าน้ำหนักที่มากสุด (Global maximum weight) และค่าน้ำหนักที่มากสุดของเซตรายการที่พิจารณา (Local maximum weight) เพื่อทำการลดทอนการพิจารณาเซตรายการที่ไม่สามารถเป็นผลลัพธ์ได้ อันนำมาซึ่งการลดทอนปริภูมิสถานะและเวลาในการประมวลผลได้ จากนั้นผู้วิจัยได้ทำการศึกษาแนวทางการเพิ่มประสิทธิภาพการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการเพื่อให้มีประสิทธิภาพที่เพิ่มขึ้นจาก *WFRIM* ที่นำเสนอก่อนหน้า โดยผู้วิจัยได้นำเสนอขั้นตอนวิธีใหม่ที่เรียกว่า *Weighted-Frequent-Regular Itemsets Miner using Interval Word Segment (WFRIM-IWS)* โดยขั้นตอนดังกล่าวได้ประยุกต์ใช้โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง (interval word segment, IWS) ที่ซึ่งมีลักษณะเป็นแบบไดนามิกบิตเวกเตอร์ (dynamic bit-vector) เพื่อจัดเก็บข้อมูลทรานแซกชันที่มีรายการหรือเซตรายการหนึ่ง ๆ ปรากฏ โดยขั้นตอนวิธี *WFRIM-IWS* ได้ทำการประยุกต์ใช้โครงสร้างต้นไม้ที่เรียกว่า *WFRIM-tree* เพื่อทำการจัดเก็บเซตรายการต่าง ๆ ระหว่างการประมวลผล และเทคนิคการคำนวณหาค่าน้ำหนักที่มากสุด (Global maximum weight) และค่าน้ำหนักที่มากสุดของเซตรายการที่พิจารณา (Local maximum weight) เพื่อทำการลดทอนปริภูมิสถานะ โดยจากการทดสอบประสิทธิภาพของสองขั้นตอนวิธีที่นำเสนอจะทำให้ทราบได้ว่าทั้งสองขั้นตอนวิธีสามารถค้นหาผลลัพธ์ได้ถูกต้องครบถ้วน และขั้นตอนวิธี *WFRIM-IWS* จะสามารถประมวลผลได้รวดเร็วและใช้หน่วยความจำน้อยกว่าขั้นตอนวิธี *WFRIM*

วัตถุประสงค์ของวิทยานิพนธ์

1. เพื่อศึกษา วิเคราะห์ปัญหา และหาแนวทางการพัฒนาเพิ่มเติมจากปัญหาการค้นหาเซตรายการปรากฏบ่อย ปัญหาการค้นหาเซตรายการปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ และปัญหาการค้นหาเซตรายการปรากฏบ่อยและปรากฏสม่ำเสมอ
2. เพื่อนำเสนอปัญหาการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการที่ซึ่งสามารถให้ผลลัพธ์ที่แตกต่างและให้ข้อมูล สารสนเทศ และองค์ความรู้เพิ่มเติมจากปัญหาก่อนหน้าได้
3. เพื่อพัฒนาขั้นตอนวิธีในการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการให้มีประสิทธิภาพในการลดหน่วยความจำและใช้เวลาในการคำนวณที่รวดเร็วและมีความเหมาะสมสำหรับการใช้งานจริง
4. เพื่อวิเคราะห์และศึกษารูปแบบในการปรากฏขึ้นของเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ ดังเช่น รูปแบบการเลือกซื้อสินค้าของผู้บริโภคในธุรกิจค้าปลีก รูปแบบการคลิกเพื่อเลือกซื้อสินค้าออนไลน์ในธุรกิจพาณิชย์อิเล็กทรอนิกส์ โดยการศึกษารูปแบบที่ปรากฏขึ้นเหล่านี้สามารถช่วยเจ้าของธุรกิจในการเพิ่มผลกำไรและความเป็นที่นิยมในทางการตลาดเป็นต้น
5. เพื่อให้ผู้สนใจหรือนักวิจัยสามารถประยุกต์ใช้ขั้นตอนวิธีการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการในการค้นหารูปแบบของข้อมูลภายใต้เงื่อนไขนี้และสามารถพัฒนาต่อยอดงานวิจัยได้

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้นิยามและขอบเขตของปัญหาการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ
2. ได้ขั้นตอนวิธีในการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการที่มีประสิทธิภาพในด้านเวลาที่ใช้ในการคำนวณและหน่วยความจำที่ใช้ในการจัดเก็บข้อมูล
3. ได้องค์ความรู้ใหม่ในมุมมองของการปรากฏขึ้นของรูปแบบหรือเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ

4. ได้ผลงานวิจัยที่สามารถตีพิมพ์ในงานประชุมวิชาการหรือวารสารวิชาการ

ขอบเขตของวิทยานิพนธ์

1. ฐานข้อมูลที่ใช้ทดสอบประสิทธิภาพของขั้นตอนวิธีที่นำเสนอในวิทยานิพนธ์นี้จะเป็นฐานข้อมูลรายการ (transactional database) ที่ซึ่งดาวน์โหลดมาจากเว็บไซต์ fimi[1] โดยฐานข้อมูลดังกล่าวมีลักษณะข้อมูลทั้งหนาแน่นและเบาบาง
2. ค่าน้ำหนักความสำคัญของแต่ละรายการ (weight) จะถูกกำหนดโดยการสุ่ม ที่ซึ่งจะมีค่าอยู่ระหว่าง 0.1 - 0.9 โดยวิธีการสุ่มจะดำเนินการด้วยวิธีการเดียวกันกับงานวิจัยก่อนหน้า อาทิเช่น (Yun & Leggett, 2005) (Wang, Yang, & Yu, 2004) และ (Tao, 2003)
3. ผู้ใช้ต้องทำการกำหนดค่าขีดแบ่งน้ำหนักสนับสนุน (weighted-support threshold) และค่าขีดแบ่งความสม่ำเสมอ (regularity threshold) เพื่อใช้เป็นเกณฑ์สำหรับพิจารณาความน่าสนใจของเซตรายการ
4. การทดสอบประสิทธิภาพของขั้นตอนวิธีที่นำเสนอจะดำเนินการใน 3 แง่มุม คือ เวลาที่ใช้ในการประมวลผล หน่วยความจำที่ใช้ในการประมวลผล และจำนวนของเซตรายการผลลัพธ์ที่ได้รับ

แผนการดำเนินงานวิทยานิพนธ์

แผนการดำเนินงานและขั้นตอนการดำเนินงานของวิทยานิพนธ์นี้ สามารถแสดงรายละเอียดได้ ดังตารางที่ 1-1

[1] http://fimi.ua.ac.be/data/

ตารางที่ 1-1 แผนการดำเนินงานวิทยานิพนธ์

การดำเนินงาน	2557	2558	2559	2560
	ต.ค. – ธ.ค.	ม.ค. – ธ.ค.	ม.ค. – ธ.ค.	ม.ค. – ธ.ค.
1. รวบรวมข้อมูลและศึกษาปัญหางานวิจัย	◄———	———►		
2. ศึกษาขั้นตอน วิเคราะห์ข้อมูลและคิดค้นขั้นตอนวิธีสำหรับการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ		◄———►		
3. เขียนโปรแกรมและทดสอบ		◄———►		
4. เผยแพร่งานวิจัยที่ 1		◄———►		
5. ศึกษาปรับปรุงประสิทธิภาพของขั้นตอนวิธีสำหรับการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ			◄———►	
6. เผยแพร่งานวิจัยที่ 2			◄———►	
7. จัดทำวิทยานิพนธ์			◄———►	

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การค้นหากฎความสัมพันธ์ (Association rule) เป็นเทคนิคหนึ่งในการทำเหมืองข้อมูลที่ได้รับความนิยมและมีการศึกษาอย่างแพร่หลาย ซึ่งการค้นหากฎความสัมพันธ์ประกอบด้วย 2 ขั้นตอนคือ 1) การค้นหาเซตรายการ/รูปแบบที่ปรากฏบ่อย และ 2) การสร้างกฎความสัมพันธ์จากเซตรายการที่ได้รับในขั้นตอนแรก โดยงานวิจัยส่วนมากมุ่งเน้นที่จะเพิ่มประสิทธิภาพทางด้านเวลาและหน่วยความจำในการประมวลผลของขั้นตอนการค้นหาเซตรายการ/รูปแบบที่ปรากฏบ่อย เนื่องจากขั้นตอนนี้เป็นงานที่ยากและมีความท้าทายรวมถึงพัฒนาต่อยอดการค้นหาเซตรายการที่ปรากฏบ่อยในหลายแง่มุมเพื่อให้ได้เซตรายการที่มีความเหมาะสมกับความต้องการของแอปพลิเคชัน ซึ่งแง่มุมที่วิทยานิพนธ์นี้สนใจ ได้แก่ 1) การค้นหาเซตรายการที่ปรากฏบ่อย 2) การค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอและ 3) การค้นหาเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ ดังนั้นในบทนี้จะกล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องรวมถึงคุณลักษณะของฐานข้อมูลที่ใช้ทดสอบงานวิจัยที่วิทยานิพนธ์นี้นำเสนอ ซึ่งจะแบ่งออกเป็นส่วนต่าง ๆ ดังนี้

1. การค้นหาเซตรายการที่ปรากฏบ่อย (Frequent Itemsets Mining)
2. การค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอ (Frequent Regular Itemsets Mining)
3. การค้นหาเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ (Weighted Frequent Itemsets Mining)
4. คุณลักษณะของฐานข้อมูลที่ใช้ทดสอบในงานวิจัยที่วิทยานิพนธ์นี้นำเสนอ

2.1 การค้นหาเซตรายการที่ปรากฏบ่อย (Frequent Itemsets/Pattern Mining)

การค้นหาเซตรายการ/รูปแบบที่ปรากฏบ่อย (Frequent Itemsets/Pattern Mining, FIM) เป็นกระบวนการหนึ่งในการสกัดข้อมูลที่น่าสนใจจากฐานข้อมูล ถูกนำเสนอครั้งแรกโดย Agrawal, Imielinski, and Swami (1993) ซึ่งเซตรายการที่ปรากฏบ่อย คือ เซตรายการ (itemsets) ที่มีจำนวนครั้ง/ความถี่ของการปรากฏในฐานข้อมูลไม่น้อยกว่าเกณฑ์ที่ผู้ใช้กำหนด ต่อมาการค้นหาเซตรายการที่ปรากฏบ่อยได้ถูกนำไปประยุกต์ใช้การวิเคราะห์รูปแบบหรือพฤติกรรมการซื้อสินค้าในตะกร้าสินค้า (Market basket analysis) (Agrawal & Srikant, 1994) โดยขั้นตอนวิธีการที่นำเสนอนี้จะทำการค้นหาความสัมพันธ์ระหว่างสินค้าที่แตกต่างกันภายในตะกร้าสินค้า เช่น ถ้าลูกค้าซื้อ

ผงซักฟอกแล้วมักจะซื้อน้ำยาปรับผ้านุ่มร่วมด้วย ซึ่งข้อมูลที่ได้การวิเคราะห์นี้สามารถส่งผลประโยชน์ต่อนักวิเคราะห์การตลาดหรือผู้จัดการในการกำหนดนโยบาย วางแผนหรือจัดทำโปรโมชันสินค้า ที่ซึ่งสามารถช่วยในการเพิ่มผลกำไร กระตุ้นยอดขายและลดต้นทุนให้กับธุรกิจนั้น ๆ เป็นต้น

สำหรับคำจำกัดความและนิยามต่าง ๆ ของการค้นหาเซตรายการ/รูปแบบที่ปรากฏบ่อยที่ซึ่ง Agrawal and Srikant (1994) ได้ระบุไว้สามารถแสดงดังต่อไปนี้

กำหนดให้ $I = \{i_1, i_2, \dots, i_n\}$ เมื่อ $n \geq 1$ เป็นเซตของตัวเลขหรือตัวอักษร โดยเรียกแต่ละ $i_j \in I$ ว่า “รายการ” (item) เซต $X = \{i_1, \dots, i_k\} \subseteq I$ คือเซตรายการ (itemset) (รูปแบบ (pattern) หรือ k-เซตรายการ (k-itemset) เมื่อ X มีสมาชิก k รายการที่แตกต่างกัน) ฐานข้อมูลทรานแซกชัน (transaction database) $TDB = \{t_1, t_2, \dots, t_m\}$ เป็นเซตของทรานแซกชัน ที่ซึ่งแต่ละทรานแซกชัน $t_j \in TDB$ บรรจุไปด้วย 2 ข้อมูลคือ j เป็นหมายเลขทรานแซกชัน (tid) และเซตรายการ Y จากข้อมูลในทรานแซกชัน t_j ถ้าเซตรายการ $X \subseteq Y$ สามารถอธิบายได้ว่าเซตรายการ X ปรากฏในทรานแซกชัน t_j หรือ ทรานแซกชัน t_j มีเซตรายการ X บรรจุอยู่ ดังนั้นเมื่อทำการพิจารณาการปรากฏขึ้นของเซตรายการ X หนึ่ง ๆ จะทำให้ทราบถึงเซต $T^X = \{t_j^X, \dots, t_k^X\}$ ซึ่งคือเซตของหมายเลขทรานแซกชันที่เซตรายการ X ปรากฏอยู่ เมื่อ $j, k \in [1, m]$ ซึ่ง $j \leq k$ ได้ (หมายเหตุ เซตรายการ X การเรียงลำดับของสมาชิกนั้นไม่มีความสำคัญ เซตรายการ $\{a, b, c\}$ และเซตรายการ $\{c, a, b\}$ จะมีค่าเทียบเท่ากัน สำหรับการเขียนระบุหรืออ้างถึงเซตรายการ X เพื่อความสะดวกในการใช้งานสามารถเขียนโดยไม่ใช้เครื่องหมายเซตได้ เช่น เซตรายการ $\{a, b, c\}$ สามารถเขียนได้เป็น abc)

นิยามที่ 2.1 (ค่าสนับสนุน/ค่าความถี่ของเซตรายการ X) ค่าสนับสนุน/ค่าความถี่ (support/frequency) ของเซตรายการ X คือ จำนวนทรานแซกชันที่เซตรายการ X ปรากฏในฐานข้อมูลฐานแซกชัน TDB สามารถเขียนระบุได้เป็น s^X หรือ f^X และคำนวณได้โดย $s^X = |T^X|$ หรือ $f^X = |T^X|$ เมื่อ $|T^X|$ คือขนาดของ T^X

นิยามที่ 2.2 (เซตรายการที่ปรากฏบ่อย) เซตรายการ X จะถูกระบุว่าเป็นเซตรายการที่ปรากฏบ่อยก็ต่อเมื่อค่าสนับสนุน s^X หรือค่าความถี่ f^X มีค่ามากกว่าหรือเท่ากับค่าขีดแบ่งสนับสนุนขั้นต่ำ (minimum support threshold, $minsup$) ($s^X \geq minsup$) หรือ ($f^X \geq minsup$) ซึ่งค่าขีดแบ่งสนับสนุนขั้นต่ำถูกกำหนดโดยผู้ใช้

ตัวอย่าง 2.1 กำหนดให้ฐานข้อมูลทรานแซกชันมีทั้งหมด 6 ทรานแซกชันซึ่งประกอบด้วยรายการ a, b, c, d และ e ดังภาพที่ 2-1

tid	items
1	a,b,d,e
2	b,c,e
3	a,b,d,e
4	a,b,c,e
5	a,b,c,d,e
6	b,c,d

ภาพที่ 2-1 ฐานข้อมูลทรานแซกชัน

จากฐานข้อมูลทรานแซกชันตัวอย่างในภาพที่ 2-1 กำหนดให้ค่าขีดแบ่งสนับสนุนขั้นต่ำ ($minsup$) มีค่าเท่ากับ 3 จะเห็นได้ว่าเซตรายการ b, c และ e ปรากฏในทรานแซกชันที่ 2 4 และ 5 (T^{bce} = {2, 4, 5}) ซึ่งค่าสนับสนุนของเซตรายการดังกล่าวคำนวณได้เป็น s^{bce} = 3 ดังนั้นเซตรายการ b, c และ e เป็นเซตรายการที่ปรากฏบ่อยในฐานข้อมูล เนื่องจากมีค่าสนับสนุนคือ 3 ซึ่งค่าสนับสนุนของเซตรายการ b, c และ e มีค่าเท่ากับค่าขีดแบ่งสนับสนุนขั้นต่ำที่ถูกกำหนดไว้

2.1.1 วิธีการในการค้นหาเซตรายการที่ปรากฏบ่อย

การค้นหาเซตรายการที่ปรากฏบ่อยภายในฐานข้อมูลเป็นงานที่ยากและมีความท้าทายเป็นอย่างยิ่ง เนื่องจากความเป็นไปได้ทั้งหมดของเซตรายการที่ปรากฏร่วมกัน (search space) มีจำนวนเป็น $2^{|I|}$ เมื่อ I คือรายการที่ปรากฏภายในฐานข้อมูล ถ้าหากฐานข้อมูลมีขนาดใหญ่และรายการมีจำนวนมากวิธีการพื้นฐาน (naive approach) ในการค้นหาเซตรายการที่ปรากฏร่วมกันและการคำนวณค่าสนับสนุนของเซตรายการที่ปรากฏในฐานข้อมูลจะใช้เวลาในการประมวลผลที่นานรวมถึงใช้หน่วยความจำมาก ส่งผลให้หลายงานวิจัยได้มีการศึกษาและพัฒนาวิธีการในการค้นหาเซตรายการที่ปรากฏบ่อยเพื่อแก้ไขปัญหาดังกล่าว ซึ่งวิธีการที่ได้รับความนิยมในการค้นหาเซตรายการที่ปรากฏบ่อยสามารถแบ่งได้เป็น 3 กลุ่ม ดังนี้

1) วิธีการสร้างเซตรายการแข่งขัน (Candidate generation) วิธีการนี้จะพิจารณาเซตรายการที่คาดว่าจะเป็นผลลัพธ์ของเซตรายการที่ปรากฏบ่อยเป็นรอบ ๆ ตามขนาดของเซตรายการ (Level-wise approach) โดยเริ่มจากอ่านฐานข้อมูลครั้งแรกเพื่อค้นหาเซตรายการที่เป็นผลลัพธ์ของเซตรายการที่ปรากฏบ่อยขนาด 1-เซตรายการ จากนั้นนำ 1-เซตรายการ ที่ได้จากขั้นตอนแรกไปสร้าง “เซตรายการแข่งขัน” (candidate itemset) สำหรับ 2-เซตรายการ แล้วอ่านฐานข้อมูลอีกครั้งเพื่อ

ค้นหาผลลัพธ์ของเซตรายการขนาด 2-เซตรายการ จากนั้นค้นหาเซตรายการ 3-เซตรายการ หรือขนาดอื่น ๆ ด้วยวิธีการเดียวกันจนกระทั่งไม่สามารถสร้างเซตรายการแข่งขันได้จึงจบการทำงาน ซึ่ง Agrawal and Srikant (1994) นำเสนอขั้นตอนวิธีที่มีชื่อว่า Apiori ขั้นตอนวิธีนี้ใช้วิธีการการสร้างเซตรายการแข่งขันร่วมกับคุณสมบัติ downward closure property สำหรับการลดทอนเซตรายการที่ไม่เป็นผลลัพธ์ออกจากการพิจารณา โดยถ้าเซตรายการที่พิจารณาไม่เป็นเซตรายการปรากฏบ่อยแล้วซูเปอร์เซตของเซตรายการนั้นจะไม่เป็นเซตรายการที่ปรากฏบ่อยด้วย จากคุณสมบัตินี้สามารถช่วยลดเวลาที่ใช้ในการประมวลผลและลดหน่วยความจำที่ใช้ในการจัดเก็บได้อย่างมีประสิทธิภาพ หลังจากที่ขั้นตอนวิธี Apiori ถูกนำเสนอขึ้นได้มีหลายงานวิจัยนำขั้นตอนวิธีดังกล่าวมาปรับปรุงหรือพัฒนาต่อยอด เช่น Park, Chen, and Yu (1995) พัฒนาขั้นตอนวิธี Apiori โดยใช้เทคนิค hashing ที่เรียกว่า Direct hashing and pruning (DHP) สำหรับการค้นหาเซตรายการที่ปรากฏบ่อย ที่ซึ่งวิธีการนี้สามารถลดจำนวนเซตรายการแข่งขันของสองเซตรายการและส่งผลไปยังการลดเวลาและหน่วยความจำที่ใช้ในการประมวลผลและ Savasere, Omiecinski and Navathe (1995) ได้พัฒนาขั้นตอนวิธี Partitioning สำหรับการค้นหาเซตรายการที่ปรากฏบ่อย โดยขั้นตอนวิธีการที่นำเสนอนี้ได้ทำการอ่านข้อมูลจากฐานข้อมูลเพียง 2 ครั้งซึ่งส่งผลให้ลดเวลาในการอ่านข้อมูลทรานแซกชันที่มีปริมาณมาก ต่อมา Lin, Lee และ Hsueh (2012) ได้นำเสนอขั้นตอนวิธีสำหรับการค้นหาเซตรายการที่ปรากฏบ่อยที่เรียกว่า Single Pass Counting (SPC), Fixed Passes Combined-counting (FPC) และ Dynamic Passes Combined-counting (DPC) บนพื้นฐานของขั้นตอนวิธี Apiori ร่วมกับการประยุกต์ใช้โครงสร้างการทำงานแบบ MapReduce ที่ซึ่งส่งผลให้วิธีการที่นำเสนอนี้สามารถใช้เวลาในการคำนวณที่รวดเร็วและค้นหาผลลัพธ์เซตรายการได้อย่างมีประสิทธิภาพบนฐานข้อมูลที่มีขนาดใหญ่ เป็นต้น

2) วิธีการแบ่งแยกและเอาชนะ (Divide-and-conquer) จากวิธีการสร้างเซตรายการแข่งขัน ถ้าในกรณีที่ฐานข้อมูลมีขนาดใหญ่จะส่งผลให้มีการสร้างเซตรายการแข่งขันจำนวนมากและมีการอ่านฐานข้อมูลทุก ๆ รอบสำหรับค้นหาเซตรายการที่เป็นเซตรายการที่ปรากฏบ่อย Han, Pei, and Yin (2000) ได้นำเสนอขั้นตอนวิธีที่เรียกว่า FP-Growth เพื่อค้นหาเซตรายการที่ปรากฏบ่อยโดยที่ไม่ต้องสร้างสร้างเซตรายการแข่งขันและลดจำนวนครั้งในการอ่านฐานข้อมูล ซึ่งขั้นตอนวิธี FP-Growth มีกระบวนการทำงานแบบแบ่งแยกและเอาชนะ (Divide-and-conquer) โดยจะเริ่มจากอ่านฐานข้อมูลหนึ่งครั้งเพื่อให้ได้ซึ่งเซตรายการที่ปรากฏบ่อยขนาด 1-เซตรายการ แล้วเรียงลำดับเซตรายการดังกล่าวจากค่าสนับสนุนมากไปน้อย จากนั้นอ่านฐานข้อมูลอีกครั้งเพื่อจัดเก็บข้อมูลของรายการที่มีความสัมพันธ์กันไว้ในโครงสร้างต้นไม้ที่เรียกว่า Frequent pattern tree หรือ FP-tree สำหรับการค้นหาเซตรายการที่ปรากฏบ่อยในขนาดต่าง ๆ ของขั้นตอนวิธีนี้จะค้นหาเซตรายการที่

ปรากฏร่วมกันกับแต่ละเซตรายการขนาดที่ 1-เซตรายการ ภายในโครงสร้างต้นไม้ FP-tree แล้วสร้าง conditional pattern base (ฐานข้อมูลย่อย จะบรรจุเซตของของรายการที่ปรากฏร่วมกันกับรายการที่กำลังพิจารณาภายใน FP-tree (Prefix path)) จากนั้นสร้าง FP-tree สำหรับเซตรายการที่กำลังพิจารณา แล้วทำการค้นหาเซตรายการแบบวนซ้ำต่อไปบน FP-tree ดังกล่าว ต่อมาหลายงานวิจัยได้นำขั้นตอนวิธีดังกล่าวมาปรับปรุงและพัฒนาต่อยอด อาทิเช่น Giannella, Han, Pei, Yan and Yu (2003) ได้พัฒนาขั้นตอนวิธีการค้นหาเซตรายการที่ปรากฏบ่อยในฐานข้อมูลกระแสที่เรียกว่า FP-stream ภายใต้การจัดเก็บข้อมูลบนโครงสร้างต้นไม้ที่เรียกว่า Pattern-tree ซึ่งได้พัฒนาต่อยอดจากขั้นตอนวิธี FP-growth และใช้การพิจารณาข้อมูลตามช่วงเวลาโดยใช้เทคนิค Tilted-time Window โดยในวิธีการที่นำเสนอนี้สามารถค้นหาผลลัพธ์ของเซตรายการที่ปรากฏบ่อยได้อย่างมีประสิทธิภาพในฐานข้อมูลแบบกระแส ต่อมา Grahne and Zhu (2005) ได้นำเสนอขั้นตอนวิธีการค้นหาเซตรายการที่ปรากฏบ่อย โดยการใช้เทคนิค FP-array และขั้นตอนวิธีที่เรียกว่า FPgrowth* ที่ซึ่งใช้โครงสร้างอาเรย์ในการจัดเก็บข้อมูลเพื่อช่วยลดเวลาในการค้นหาผลลัพธ์ของเซตรายการในโครงสร้างต้นไม้ โดยขั้นตอนวิธีที่นำเสนอนี้สามารถค้นหาเซตรายการผลลัพธ์ได้อย่างรวดเร็วในฐานข้อมูลที่มีขนาดใหญ่ ต่อมา Xia, Zhou, Rong and Zhang (2013) ได้พัฒนาขั้นตอนวิธีสำหรับการค้นหาเซตรายการที่ปรากฏบ่อยที่เรียกว่า Improved Parallel FP-Growth (IPFP) ที่ซึ่งจะทำการค้นหาผลลัพธ์ของเซตรายการโดยใช้การประมวลผลแบบขนานบนแพลตฟอร์ม Hadoop โดยวิธีการที่นำเสนอนี้สามารถใช้เวลาในการคำนวณที่รวดเร็วและลดหน่วยความจำที่ใช้ในการจัดเก็บข้อมูล เป็นต้น

3) วิธีการแบบผสมผสาน (Hybrid) จากวิธีการสร้างเซตรายการแข่งขันและวิธีการแบ่งแยกและเอาชนะที่กล่าวมาข้างต้นนั้นจะค้นหาเซตรายการที่ปรากฏบ่อยภายใต้ฐานข้อมูลที่มีลักษณะเป็นโครงสร้างข้อมูลแบบแนวนอน (Horizontal data format) ซึ่งมีลักษณะเป็น $\{tid: itemset\}$ เมื่อ tid คือ หมายเลขทรานแซกชัน และ $itemset$ คือเซตของรายการที่ปรากฏในทรานแซกชัน สำหรับวิธีการค้นหาเซตรายการที่ปรากฏบ่อยแบบผสมผสานนั้นจะแปลงฐานข้อมูลให้อยู่ในโครงสร้างแบบแนวตั้ง (Vertical data format) $\{item: tidset\}$ เมื่อ $item$ คือ รายการ และ $tidset$ คือ เซตของหมายเลขทรานแซกชันที่รายการดังกล่าวปรากฏภายในฐานข้อมูล ซึ่ง Zaki (2000) นำเสนอขั้นตอนวิธีการค้นหาเซตรายการที่ปรากฏบ่อยที่เรียกว่า Eclat ที่ซึ่งทำการอ่านข้อมูลจากฐานข้อมูลแล้วทำการแปลงข้อมูลให้อยู่ในรูปแบบแนวตั้ง (Vertical data format) แล้วใช้พื้นฐานแนวคิด divide-and-conquer ในการค้นหาเซตรายการที่ปรากฏบ่อยโดยใช้วิธี intersection เซตหมายเลขทรานแซกชันของสองเซตรายการภายใต้โครงสร้างต้นไม้ที่เรียกว่า IT-tree หลังจากที่ Zaki (2000) นำเสนอขั้นตอนวิธี Eclat แล้ววิธีการดังกล่าวแสดงให้เห็นว่าสามารถใช้เวลาในการคำนวณที่

รวดเร็วกว่าสองวิธีการก่อนหน้าแต่ในกรณีที่ฐานข้อมูลที่มีขนาดใหญ่จะส่งผลให้การจัดเก็บข้อมูลหมายเลขทรานแซกชันของแต่ละรายการใช้หน่วยความจำในการจัดเก็บจำนวนมาก จึงได้มีหลายงานวิจัยมุ่งเน้นที่จะพัฒนาวิธีการจัดเก็บโครงสร้างข้อมูลแบบแนวนอนให้ใช้หน่วยความจำที่น้อยลงและเวลาที่ใช้ในการประมวลผลให้รวดเร็วยิ่งขึ้น อาทิเช่น Zaki and Gouda (2003) นำเสนอขั้นตอนวิธี dEclat ที่ซึ่งปรับปรุงและต่อยอดจากขั้นตอนวิธีการ Eclat (Zaki, 2000) โดยการจัดเก็บหมายเลขทรานแซกชันที่ไม่ปรากฏในฐานข้อมูลของแต่ละรายการที่เรียกว่า diffsets แทนที่การจัดเก็บหมายเลขทรานแซกชันที่ปรากฏในฐานข้อมูลของแต่ละรายการ โดยโครงสร้างในการจัดเก็บนี้สามารถค้นหาเซตรายการที่ปรากฏบ่อยได้อย่างมีประสิทธิภาพที่ซึ่งสามารถลดเวลาที่ใช้ในการจัดเก็บและลดเวลาในการคำนวณได้อย่างสูงในฐานข้อมูลที่มีความหนาแน่น ต่อมา Shenoy, Haritsa, Sudarshan, Bhalotia, Bawa, and Shah (2000) นำเสนอขั้นตอนวิธี VIPER ที่ซึ่งทำการอ่านข้อมูลจากฐานข้อมูลแล้วทำการแปลงข้อมูลให้อยู่ในรูปแบบแนวตั้งจากนั้นทำการบีบอัดข้อมูลโดยใช้โครงสร้างบิท-เวกเตอร์ (bit-vectors) ที่เรียกว่า snakes จากโครงสร้างการจัดเก็บนี้สามารถใช้หน่วยความจำในการจัดเก็บที่น้อยกว่าวิธีการก่อนหน้า จากนั้นขั้นตอนวิธี DBV-Miner ได้ถูกนำเสนอขึ้นสำหรับการค้นหาเซตรายกรที่ปรากฏบ่อยแบบใกล้ชิดภายใต้โครงสร้างการจัดเก็บข้อมูลที่เรียกว่าไดนามิกบิท-เวกเตอร์ (Dynamic Bit-Vector, DBV) ซึ่งจัดเก็บข้อมูลในรูปแบบบิทเวกเตอร์โดยไม่จัดเก็บบิท-เวกเตอร์ที่เป็นศูนย์ที่หัวและท้ายของสายบิท-เวกเตอร์ โดยขั้นตอนวิธี DBV-Miner สามารถใช้เวลาในการคำนวณที่รวดเร็วและลดหน่วยความจำที่ใช้ในการจัดเก็บอย่างมีประสิทธิภาพเป็นต้น

2.2 การค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอ (Frequent Regularity Itemsets Mining)

การค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอ (Frequent Regular/Periodic Itemsets Mining, FRIM) เป็นการค้นหาเซตรายการที่ถูกพัฒนามาจากการค้นหาเซตรายการที่ปรากฏบ่อย โดยจะมีการพิจารณาลักษณะหรือพฤติกรรมการปรากฏของเซตรายการในฐานข้อมูลที่ซึ่งจะค้นหาเซตรายการที่ปรากฏขึ้นบ่อยและสม่ำเสมอหรือปรากฏในทุก ๆ ช่วงเวลาในฐานข้อมูลภายใต้ผู้ใช้เป็นผู้กำหนดค่าขีดแบ่งสนับสนุน (support threshold) และค่าขีดแบ่งความสม่ำเสมอ (regularity threshold) ค่าขีดแบ่งความสม่ำเสมอสามารถแสดงถึงระยะห่างหรือช่วงเวลามากที่สุดที่เซตรายการปรากฏ/ไม่ปรากฏ การค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอนั้นถูกนำเสนอโดย Tanbeer, Ahmed, Jeong, and Lee (2009) ซึ่งได้รับความสนใจและเป็นที่นิยมอย่างมากในการค้นหาความสัมพันธ์ของข้อมูลส่งผลให้หลายงานวิจัยนำการค้นหาเซตรายการดังกล่าวไปพัฒนา ดังนี้

Amphawan, Lenca, and Surarerks (2009) นำเสนอวิธีการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอจากฐานข้อมูลโดยที่ผู้ใช้จะต้องระบุจำนวนเซตรายการที่ปรากฏบ่อยและสม่ำเสมอร่วมกับค่าขีดแบ่งความสม่ำเสมอโดยไม่ต้องกำหนดค่าขีดแบ่งสนับสนุน และในปี ค.ศ. 2012 Amphawan, Lenca, and Surarerks (2012) ได้มีการพัฒนาวิธีการดังกล่าวด้วยการบีบอัดหมายเลขทรานแซกชัน (tid-sets) ซึ่งทั้งสองวิธีการมีการอ่านฐานข้อมูลเพียงครั้งเดียวและใช้วิธีการ best-first search ในการค้นหาเซตรายการที่เป็นผลลัพธ์ทั้งหมด ต่อมา Tanbeer, Ahmed, and Jeong (2010b) ได้นำเสนอการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอในฐานข้อมูลกระแส (data streams) โดยใช้วิธี pattern-growth ในการค้นหาเซตรายการที่เป็นผลลัพธ์และจัดเก็บข้อมูลในโครงสร้างต้นไม้ที่เรียกว่า “single-pass” ในปี ค.ศ. 2010 Tanbeer, Ahmed, and Jeong (2010a) ยังได้ใช้โครงสร้างต้นไม้และวิธี pattern-growth สำหรับค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอในฐานข้อมูลทรานแซกชันที่เพิ่มขึ้น (incremental transactional databases) โดยจะกำหนดเพียงค่าขีดแบ่งความสม่ำเสมอ ถัดมา Rashid, Karim, Jeong, and Choi (2012) นำเสนอการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอ โดยการวัดช่วงของความสม่ำเสมอจากความแปรปรวนระหว่างช่วงเวลาการปรากฏของเซตรายการและใช้วิธีการ pattern-growth สำหรับการค้นหาเซตรายการที่เป็นผลลัพธ์ จากนั้น Kumar and Kumari (2012) ได้นำเสนอวิธีการสำหรับค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอในฐานข้อมูลกระแส โดยฐานข้อมูลที่ใช้เป็นฐานข้อมูลแนวตั้ง (Vertical database) และใช้เทคนิค sliding window ซึ่งวิธีการนี้ใช้คุณสมบัติ downward closure property ได้อย่างมีประสิทธิภาพเนื่องจากลักษณะการเกิดขึ้นของเซตรายการอาจจะมีการเปลี่ยนแปลงจากการอัพเดตฐานข้อมูล

การค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอที่ซึ่ง Tanbeer et al. (2009) ได้นิยามและให้คำจำกัดความแสดงดังต่อไปนี้

กำหนดให้ $I = \{i_1, i_2, \dots, i_n\}$ เมื่อ $n \geq 1$ เป็นเซตของตัวเลขหรือตัวอักษร โดยเรียกแต่ละ $i_j \in I$ ว่า “รายการ” เซต $X = \{i_1, \dots, i_k\} \subseteq I$ คือเซตของรายการ (รูปแบบหรือ k-เซตรายการ เมื่อ X มีสมาชิก k รายการที่แตกต่างกัน) ฐานข้อมูลทรานแซกชัน (transaction database) $TDB = \{t_1, t_2, \dots, t_m\}$ เป็นเซตของทรานแซกชัน ที่ซึ่งแต่ละทรานแซกชัน $t_j \in TDB$ บรรจุไปด้วย 2 ข้อมูลคือ j เป็นหมายเลขทรานแซกชัน (tid) และเซตรายการ Y จากข้อมูลในทรานแซกชัน t_j ถ้าเซตรายการ $X \subseteq Y$ สามารถอธิบายได้ว่าเซตรายการ X ปรากฏในทรานแซกชัน t_j หรือ ทรานแซกชัน t_j มีเซตรายการ X บรรจุอยู่ ดังนั้นเมื่อทำการพิจารณาการปรากฏขึ้นของเซตรายการ X หนึ่ง ๆ จะทำให้ทราบถึงเซต $T^X = \{t_j^X, \dots, t_k^X\}$ ซึ่งคือเซตของหมายเลขทรานแซกชันที่เซตรายการ X ปรากฏอยู่ เมื่อ $j, k \in [1, m]$ ซึ่ง $j \leq k$ ได้

นิยามที่ 2.3 (ค่าความสม่ำเสมอของเซตรายการ X ที่ปรากฏในทรานแซกชัน) ค่าความสม่ำเสมอของเซตรายการ X ที่ปรากฏในทรานแซกชัน คือระยะห่างระหว่างทรานแซกชันที่เซตรายการ X ปรากฏขึ้น กำหนดให้ทรานแซกชัน t_k เป็นทรานแซกชันที่มีเซตรายการ X ปรากฏอยู่ ค่าความสม่ำเสมอของเซตรายการ X ที่ปรากฏในทรานแซกชัน t_k ($r_{t_k}^X$) สามารถคำนวณได้ 3 กรณีดังนี้

1) ถ้า t_k คือทรานแซกชันที่เซตรายการ X ปรากฏขึ้นในฐานข้อมูลทรานแซกชันเป็นครั้งแรกค่าความสม่ำเสมอ $r_{t_k}^X$ มีค่าเท่ากับ k ดังสมการที่ 2.1

$$r_{t_k}^X = k \tag{2.1}$$

(หมายเหตุ ค่า $r_{t_k}^X$ แสดงถึงระยะห่างระหว่างทรานแซกชันแรก (เริ่มต้นด้วย 0) จนถึงทรานแซกชัน k ที่เซตรายการ X ปรากฏขึ้นเป็นครั้งแรก)

2) ถ้า t_k คือทรานแซกชันที่ปรากฏขึ้นหลังจากทรานแซกชัน t_j ซึ่งทรานแซกชัน t_k และ t_j มีเซตรายการ X ปรากฏขึ้นทั้งคู่ ดังนั้นค่า $r_{t_k}^X$ สามารถคำนวณได้ ดังสมการที่ 2.2

$$r_{t_k}^X = k - j \tag{2.2}$$

(หมายเหตุ ค่า $r_{t_k}^X$ แสดงถึงระยะห่างระหว่างทรานแซกชัน t_j และทรานแซกชัน t_k ที่เซตรายการ X ปรากฏขึ้นต่อเนื่องกัน)

3) ถ้า t_k คือทรานแซกชันที่เซตรายการ X ปรากฏขึ้นในฐานข้อมูลทรานแซกชันเป็นครั้งสุดท้าย ดังนั้นค่า $r_{t_k}^X$ สามารถคำนวณได้ ดังสมการที่ 2.3

$$r_{t_k}^X = m - k \tag{2.3}$$

เมื่อ m คือจำนวนของทรานแซกชันทั้งหมดในฐานข้อมูล

(หมายเหตุ ค่า $r_{t_k}^X$ แสดงถึงระยะห่างระหว่างทรานแซก t_k ที่เซตรายการ X ปรากฏขึ้นจนถึงทรานแซกชันสุดท้ายของฐานข้อมูล)

นิยามที่ 2.4 (ค่าความสม่ำเสมอของเซตรายการ X) ค่าความสม่ำเสมอของเซตรายการ X คือระยะห่างมากที่สุดระหว่างทรานแซกชันที่เซตรายการ X ปรากฏขึ้นอย่างน้อยหนึ่งครั้งในฐานข้อมูลทรานแซกชัน TDB สามารถคำนวณได้สามารถคำนวณได้โดยสมการที่ 2.4

$$r^X = max\{r^X_{t_j}, r^X_{t_k}, \dots, r^X_{t_l}, r^X_{t_l}\} \quad (2.4)$$

เมื่อ $j, k, l \in [1, m]$ ตามลำดับ

(หมายเหตุ ค่าความสม่ำเสมอ $r^X_{t_l}$ มีการคำนวณทั้งในกรณีที่ 2 และ 3)

นิยามที่ 2.5 (ค่าสนับสนุนของเซตรายการ X) ค่าสนับสนุนของเซตรายการ X คือจำนวนทรานแซกชันที่เซตรายการ X ปรากฏในฐานข้อมูลฐานแซกชัน TDB สามารถเขียนระบุได้เป็น s^X และคำนวณได้โดย $s^X = |T^X|$

นิยามที่ 2.6 (เซตรายการที่ปรากฏบ่อยและสม่ำเสมอ) เซตรายการ X จะถูกระบุว่าเป็นเซตรายการที่ปรากฏบ่อยและสม่ำเสมอก็ต่อเมื่อ 1) ค่าสนับสนุนของเซตรายการ X มีค่ามากกว่าหรือเท่ากับค่าขีดแบ่งสนับสนุน (minimum support threshold, σ_s) ($s^X \geq \sigma_s$) และ 2) ค่าความสม่ำเสมอของเซตรายการ X มีค่าน้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอ (maximum regularity threshold, σ_r) ($r^X \leq \sigma_r$) โดยที่ค่าขีดแบ่งทั้งสองจะถูกกำหนดจากผู้ใช้

ตัวอย่าง 2.2 กำหนดให้ฐานข้อมูลประกอบด้วย 10 ทรานแซกชันที่มีรายการ a, b, c, d, e และ f บรรจุอยู่ดังภาพที่ 2-2

tid	item
1	a, c, d, e
2	a, d, e, f
3	a, c, e
4	c, d, e
5	a, c, e, f
6	b, f
7	b, c, d, e
8	b, c, d, e
9	a, b, c, d
10	a, b, e, f

ภาพที่ 2-2 ฐานข้อมูลทรานแซกชัน

จากฐานข้อมูลทรานแซกชันที่แสดงในภาพที่ 2-2 กำหนดให้ค่าขีดแบ่งความสม่ำเสมอเป็น 4 (σ_r=4) และค่าขีดแบ่งสนับสนุนขั้นต่ำเป็น 3 (σ_s=3) ตามลำดับ เซตรายการ de ปรากฏขึ้นในเซต

ของหมายเลขทรานแซกชันดังนี้ T^{de}= {1,2,4,7,8} ซึ่งค่าความสม่ำเสมอของเซตรายการ de สามารถคำนวณได้โดย $r^{de} = max\{r_1^{de}, r_2^{de}, r_4^{de}, r_7^{de}, r_8^{de}, r_8^{de}\}$ โดยที่ r_1^{de}=1, r_2^{de}=1 (2-1), r_4^{de}=2 (4-2), r_7^{de}=3 (7-4), r_8^{de}=1 (8-7) และ r_8^{de}=2 (10-8) ดังนั้น r^{de}=3 (max {1, 1, 2, 3, 1, 2}) แล้วค่าสนับสนุนของเซตรายการ s^{de} มีค่าเป็น 5 ซึ่งคำนวณได้จาก s^{de}=$|T^{de}|$

ดังนั้นเซตรายการ de จึงเป็นเซตรายการที่ปรากฏบ่อยและสม่ำเสมอ เนื่องจากเซตรายการดังกล่าวมีค่าความสม่ำเสมอน้อยกว่าค่าขีดแบ่งความสม่ำเสมอ ($3 \leq \sigma_r$) และค่าสนับสนุนมากกว่าค่าขีดแบ่งสนับสนุน ($5 \geq \sigma_s$)

2.3 การค้นหาเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ (Weighted frequent itemsets/pattern mining)

การค้นหาเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ (Weighted-Frequent Itemsets Mining, WFIM) เป็นการค้นหาเซตรายการที่พัฒนาต่อยอดมาจากการค้นหาเซตรายการที่ปรากฏบ่อย (Frequent Itemsets Mining, FIM) ที่ซึ่งจะค้นหาเซตรายการที่ปรากฏขึ้นบ่อย ๆ หรือถี่ ๆ ในฐานข้อมูล โดยแนวคิดพื้นฐานของการค้นหาเซตรายการที่ปรากฏบ่อยจะกำหนดความสำคัญของแต่ละรายการเท่ากันซึ่งอาจจะยังไม่เพียงพอและไม่สามารถสะท้อนได้ถึงลักษณะหรือพฤติกรรมการปรากฏของเซตรายการได้ทุกแง่มุม ตัวอย่างเช่น ในธุรกิจค้าปลีกหรือห้างสรรพสินค้า สินค้าแซลมอนและไวน์มีจำนวนในการซื้อน้อยครั้งกว่าสินค้าขนมปังและนม แต่ในแง่มุมของมูลค่าทางการตลาดหรือผลกำไรที่ได้จากการขายสินค้าทั้งสองนั้น สินค้าแซลมอนและไวน์มีมูลค่ามากกว่าสินค้าขนมปังและนม ดังนั้นจากกรณีดังกล่าวแสดงให้เห็นถึงความสำคัญหรือความน่าสนใจที่แตกต่างกันระหว่างกลุ่มของสินค้าทั้งสอง เพื่อแก้ไขปัญหานี้ (Cai et al., 1998) ได้นำเสนอวิธีการการค้นหากฎความสัมพันธ์ภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ โดยจะกำหนดค่าน้ำหนักที่แตกต่างกันให้แต่ละราย (ค่าน้ำหนักระบุหรือบ่งชี้ถึงความสำคัญหรือความน่าสนใจของรายการนั้น ๆ เช่น ในทางธุรกิจค้าปลีกความสำคัญหรือความน่าสนใจอาจจะแสดงถึงมูลค่า ผลกำไร ความเสี่ยง) แล้วมีการระบุค่าน้ำหนักสนับสนุน (ค่าสนับสนุนคูณกับค่าเฉลี่ยน้ำหนักของเซตรายการ) รวมถึงกำหนดค่าขีดแบ่งน้ำหนักสนับสนุน (weighted support threshold) เพื่อเป็นเกณฑ์ในการค้นหาเซตรายการดังกล่าว และใช้ขั้นตอนวิธี Apriori ในการค้นหาเซตรายการที่เป็นผลลัพธ์ แต่เนื่องจากแต่ละรายการมีค่าน้ำหนักไม่เท่ากันส่งผลให้ไม่สามารถใช้คุณสมบัติ downward closure property ในการลดทอนการพิจารณาเซตรายการที่ไม่เป็นผลลัพธ์ได้จึงได้นำเสนอขอบเขตค่าน้ำหนักมากที่สุดที่เป็นไปได้ของ k-เซตรายการ ที่เรียกว่า k-สนับสนุน โดยที่ค่าน้ำหนักสนับสนุนของแต่ละรายการที่พิจารณาในขนาด k ต้องมีค่ามากกว่าหรือเท่ากับค่า k-สนับสนุน แต่วิธีการนี้มี

การสร้างรายการแข่งขันที่ไม่เป็นเซตรายการที่เป็นผลลัพธ์มากเกินไป Tao, Murtagh and Farid (2003) จึงได้ประยุกต์ใช้คุณสมบัติ downward closure property ในการประมาณค่าน้ำหนักสนับสนุนของแต่ละรายการโดยใช้ค่าสนับสนุนร่วมกับค่าน้ำหนักของแต่ละรายการ ต่อมา Yun and Leggett, (2005) นำเสนอวิธีการจัดเก็บข้อมูลในโครงสร้างต้นไม้ที่เรียกว่า FP-tree เป็นครั้งแรกในขั้นตอนวิธีสำหรับการค้นหาเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการด้วยการอ่านฐานข้อมูลสองครั้ง มีการใช้ค่าน้ำหนักขั้นต่ำร่วมกับช่วงค่าน้ำหนักโดยการสุ่มค่าน้ำหนักให้กับรายการภายในช่วงค่าน้ำหนักที่กำหนดไว้ ซึ่งมีการจัดเก็บข้อมูลภายใน FP-Tree โดยมีการเรียงลำดับตามค่าน้ำหนักความสำคัญภายในโครงสร้างต้นไม้มากไปหาน้อยร่วมกับพิจารณาคุณสมบัติ downward closure property เพื่อลดการพิจารณาเซตรายการทำให้ช่วยลดเวลาในการคำนวณ Vo and Coenen (2013) นำเสนอขั้นตอนวิธีสำหรับค้นหาเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการโดยใช้โครงสร้างต้นไม้ที่เรียกว่า WIF-tree (Weighted Itemset Tidset tree) ร่วมกับการจัดเก็บหมายเลขทรานแซกชันแบบ Diffset ในหลาย ๆ งานวิจัยได้มีการนำการค้นหาเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการไปพัฒนาต่อยอดในแง่มุมอื่น ๆ เช่น การค้นหาความสัมพันธ์ของข้อมูลโดยไม่ต้องมีการกำหนดค่าน้ำหนักล่วงหน้า (Weighted association rules without preassigned weights) (Sun & Bai, 2008) การค้นหาเซตรายการที่มีการกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการอย่างเป็นลำดับ Weight sequential pattern (Lan, Hong, & lee, 2014) (Yun & Leggett, 2006) และ อื่น ๆ

การค้นหาเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการมีนิยามและคำจำกัดความแสดงดังต่อไปนี้

กำหนดให้ $I = \{i_1, i_2, \dots, i_n\}$ เป็นเซตของรายการ (items) โดยที่แต่ละรายการ $i_p \in I$ มีค่าน้ำหนัก w_{i_p} ที่บ่งชี้ถึงความสำคัญ/ความน่าสนใจของรายการนั้น ๆ (กล่าวคือ สำหรับเซต $I = \{i_1, i_2, \dots, i_n\}$ จะมีเซตค่าน้ำหนัก $W = \{w_1, w_2, \dots, w_n\}$ ที่ซึ่งแต่ละ $w_i \in W$ แสดงถึงค่าน้ำหนักความสำคัญของรายการ $i_j \in I$) เซต $X = \{i_p, \dots, i_q\}$ เมื่อ $1 \leq p \leq q \leq n$ คือเซตของรายการ (รูปแบบหรือ k-เซตรายการ ถ้า X ประกอบไปด้วย k รายการที่แตกต่างกันเช่น “ab” คือ 2-เซตรายการ และ “abc” คือ 3-เซตรายการ) ฐานข้อมูลทรานแซกชัน $TDB = \{t_1, t_2, \dots, t_m\}$ เป็นเซตของทรานแซกชัน ที่ซึ่งแต่ละทรานแซกชัน $t_j \in TDB$ บรรจุไปด้วย 2 ข้อมูลคือ j เป็นหมายเลขทรานแซกชัน (tid) และเซตรายการ Y จากข้อมูลในทรานแซกชัน t_j ถ้าเซตรายการ $X \subseteq Y$ สามารถอธิบายได้ว่าเซตรายการ X ปรากฏในทรานแซกชัน t_j หรือ ทรานแซกชัน t_j มีเซตรายการ X บรรจุอยู่ ดังนั้นเมื่อทำการพิจารณาการปรากฏขึ้นของเซตรายการ X หนึ่ง ๆ จะทำให้

ทราบถึงเซต $T^X = \{t_j^X, \dots, t_k^X\}$ ซึ่งคือเซตของหมายเลขทรานแซกชันที่เซตรายการ X ปรากฏอยู่ เมื่อ $j, k \in [1, m]$ ซึ่ง $j \leq k$ ได้

นิยามที่ 2.7 (ค่าน้ำหนักของเซตรายการ X) ค่าน้ำหนักของเซตรายการ $X = \{i_p, \dots, i_q\} \subseteq I$ คือ ค่าน้ำหนักเฉลี่ยของรายการทั้งหมดภายในเซตรายการ X สามารถคำนวณได้โดยสมการ 2.5

$$w^X = \frac{\sum_{p=1}^{|X|} w^{ip}}{|X|} \tag{2.5}$$

นิยามที่ 2.8 (ค่าสนับสนุนของเซตรายการ X) ค่าสนับสนุนของเซตรายการ X คือ จำนวนทรานแซกชันที่เซตรายการ X ปรากฏในฐานข้อมูลฐานแซกชัน TDB สามารถเขียนระบุได้เป็น s^X และคำนวณได้โดย $s^X = |T^X|$

นิยามที่ 2.9 (ค่าน้ำหนักสนับสนุนของเซตรายการ X) ค่าน้ำหนักสนับสนุนของเซตรายการ X (ws^X) คือผลลัพธ์ที่ได้จากการคูณระหว่างค่าน้ำหนัก (w^X) และค่าสนับสนุน (s^X) ของเซตรายการ X ซึ่งสามารถคำนวณได้จากสมการที่ 2.6

$$ws^X = s^X \times w^X \tag{2.6}$$

นิยามที่ 2.10 (เซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ) เซตรายการ X จะถูกระบุว่าเป็นเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ ก็ต่อเมื่อค่าน้ำหนักสนับสนุนของเซตรายการ X มีค่ามากกว่าหรือเท่ากับค่าขีดแบ่งน้ำหนักสนับสนุนที่ผู้ใช้กำหนด (weighted-support threshold, σ_{ws}) ($ws^X \geq \sigma_{ws}$)

สำหรับการลดจำนวนการพิจารณาหาผลลัพธ์ของเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ (prune search space) นั้นไม่สามารถใช้คุณสมบัติ downward closure (กล่าวคือ ถ้าเซตรายการที่พิจารณาไม่เป็นเซตรายการปรากฏบ่อยแล้วซูเปอร์เซตของเซตรายการนั้นจะไม่เป็นเซตรายการที่ปรากฏบ่อยด้วย) ได้เนื่องจากแต่ละรายการมีค่าน้ำหนักไม่เท่ากัน อาทิเช่น รายการ a มีค่าน้ำหนัก 0.6 และมีค่าสนับสนุน 5 ส่วนรายการ d มีค่า

น้ำหนัก 0.35 มีค่าสนับสนุนเป็น 4 เซตรายการ ad มีค่าสนับสนุน 4 จากสมการที่ 2.5 ค่าน้ำหนักของเซตรายการ ad คือ (0.6+0.35)/2=0.475 และจากสมการ 2.6 ค่าน้ำหนักสนับสนุนของเซตรายการ ad คือ 1.9 (0.475×4) สำหรับค่าน้ำหนักสนับสนุนของเซตรายการ a เป็น 3.0 (0.6×5) และ d เป็น 1.4 (0.35×4) ถ้าผู้ใช้กำหนดค่าขีดแบ่งน้ำหนักสนับสนุนเป็น 1.5 จะส่งผลให้เซตรายการ d ไม่เป็นเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการแต่เซตรายการ ad เป็นผลลัพธ์ของเซตรายการดังกล่าว ดังนั้น Tao et al. (2003) (Tao et al., 2003) และ Yun and Leggett (2005) (Yun & Leggett, 2005) จึงประยุกต์ใช้คุณสมบัติ downward closure โดยการประมาณค่าน้ำหนักสนับสนุนของแต่ละเซตรายการจากค่าน้ำหนักที่มากสุดที่เรียกว่า Global maximum weight ($GMAXW$) หรือค่าน้ำหนักที่มากสุดของเซตรายการที่พิจารณา Local maximum weight ($LMAXW$)

นิยามที่ 2.11 (ค่าน้ำหนักที่มากที่สุด) ค่าน้ำหนักที่มากสุด (Global maximum weight, $GMAXW$) คือค่าน้ำหนักที่มากสุดของทุกรายการที่เป็นสมาชิกของ I สามารถคำนวณได้จากสมการที่ 2.7

$$GMAXW = max\{w_1, w_2, \dots, w_n\} \tag{2.7}$$

นิยามที่ 2.12 (ค่าน้ำหนักที่มากสุดของเซตรายการ X) ค่าน้ำหนักที่มากสุดของเซตรายการ X (Local maximum weight, $LMAXW$) คือ ค่าเฉลี่ยผลรวมของค่าน้ำหนักของทุกรายการ $i_j \in X$ รวมกับค่าน้ำหนักที่มากสุดของรายการ $i_j \notin X$ (หมายเหตุ รายการที่ไม่ได้เป็นสมาชิกของ X สามารถนิยามได้เป็น $Y = I - X$ ซึ่ง $Y = \{i_r, i_s, \dots, i_u\}$ ดังนั้น ค่าน้ำหนักที่มากสุดของรายการที่ไม่เป็นสมาชิกของ X สามารถคำนวณได้โดย $max\{w_{i_r}, w_{i_s}, \dots, w_{i_u}\}$) ที่ซึ่งสามารถคำนวณได้ดังนี้

$$LMAXW^X = \frac{\sum_{i_j \in X} w^{i_j} + max\{w^{i_r}, w^{i_s}, \dots, w^{i_u}\}}{|X|+1} \tag{2.8}$$

นิยามที่ 2.13 (การประมาณค่าน้ำหนักสนับสนุนของรายการ $i_j \in I$) สำหรับการประมาณค่าน้ำหนักสนับสนุน (overestimated-weight-frequency, ows) ของรายการ $i_j \in I$ นั้น

สามารถประมาณค่าได้จากประยุกต์ใช้ค่าน้ำหนักที่มากสุด ($GMAXW$) ร่วมกับค่าสนับสนุนของเซตรายการ i_j ซึ่งจะแสดงถึงค่าน้ำหนักสนับสนุนที่มากสุดที่เป็นไปได้ของรายการ i_j ดังสมการที่ 2.9

$$ows^{i_j} = GMAXW \times s^{i_j} \tag{2.9}$$

(หมายเหตุ จากนิยามที่ 2.10 ถ้าค่า ows^{i_j} ของรายการ $i_j \in I$ มีค่าน้อยกว่าค่าขีดแบ่งน้ำหนักสนับสนุน ($ows^{i_j} < \sigma_{ws}$) จะสามารถสรุปได้ว่ารายการ i_j ไม่เป็นเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ ส่งผลให้ซุปเปอร์เซตของ i_j จะไม่ถูกพิจารณาไปด้วย)

นิยามที่ 2.14 (การประมาณค่าน้ำหนักสนับสนุนของเซตรายการ $Y = X \cup i_j \in I$) สำหรับการประมาณค่าน้ำหนักสนับสนุน (overestimated-weight-frequency, ows) ของเซตรายการ $Y = X \cup i_j \in I$ นั้นสามารถประมาณค่าได้จากประยุกต์ใช้ค่าน้ำหนักที่มากสุดของเซตรายการ X ($LMAXW$) ร่วมกับค่าสนับสนุนของเซตรายการ X ซึ่งจะแสดงถึงค่าน้ำหนักสนับสนุนที่มากสุดที่เป็นไปได้ของเซตรายการ Y ดังสมการที่ 2.10

$$ows^{Y} = LMAXW^{X} \times s^{X} \tag{2.10}$$

(หมายเหตุ จากนิยามที่ 2.10 ถ้าค่า ows^{Y} ของเซตรายการ Y มีค่าน้อยกว่าค่าขีดแบ่งน้ำหนักสนับสนุน ($ows^{Y} < \sigma_{ws}$) จะสามารถสรุปได้ว่าเซตรายการ Y ไม่เป็นเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ ส่งผลให้ซุปเปอร์เซตของ Y จะไม่ถูกพิจารณาไปด้วย)

ตัวอย่าง 2.3 กำหนดให้ฐานข้อมูลประกอบด้วย 6 ทรานแซกชันที่มีทั้งหมด 8 รายการ และค่าน้ำหนักที่บ่งชี้ถึงความสำคัญ/ความน่าสนใจของแต่ละรายการกำหนดให้ดังภาพที่ 2-3

tid	item
1	a, c, d, g, h
2	a, e, f
3	b, e, f, g, h
4	a, b, c, d
5	a, b, d, h
6	a, b, d, e

ฐานข้อมูลทรานแซกชัน

item	weight
a	0.6
b	0.5
c	0.2
d	0.35
e	0.5
f	0.3
g	0.4
h	0.38

ตารางค่าน้ำหนัก

ภาพที่ 2-3 ฐานข้อมูลทรานแซกชันและตารางค่าน้ำหนักของแต่ละรายการ

จากฐานข้อมูลในภาพที่ 2-3 ค่าน้ำหนักของเซตรายการ a และ d เท่ากับ w^{ad}= 0.475 ((0.6+0.35)/2) ตามสมการที่ 2.5 แล้วค่าสนับสนุนของเซตรายการ s^{ad} เป็น 4 เนื่องจากเซตรายการดังกล่าวปรากฏในฐานข้อมูล 4 ครั้ง ค่าน้ำหนักสนับสนุนของเซตรายการ ws^{ad} จึงมีค่าเท่ากับ 1.9 (0.475×4) ตามสมการที่ 2.6 ถ้าผู้ใช้กำหนดให้ค่าขีดแบ่งน้ำหนักสนับสนุนเป็น 1.5 ดังนั้นจึงสามารถสรุปได้ว่าเซตรายการ ad เป็นเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ

จากตัวอย่างข้างต้นรายการ a มีค่าน้ำหนัก w^a=0.6 ค่าสนับสนุน s^a=5 ค่าน้ำหนักสนับสนุนเป็น ws^a=3.0 (0.6×5) ส่วนรายการ d มีค่าน้ำหนัก w^d=0.35 ค่าสนับสนุน s^d=4 ค่าน้ำหนักสนับสนุนเป็น ws^d=1.4 (0.35×4) จากค่าขีดแบ่งน้ำหนักสนับสนุนข้างต้นรายการ d จะไม่เป็นผลลัพธ์ของการค้นหาเซตรายการและจะไม่พิจารณาเซตรายการที่เกิดร่วมกับรายการดังกล่าวตามคุณสมบัติ downward closure แต่เซตรายการ ad เป็นผลลัพธ์เพื่อที่จะค้นหาเซตรายการขนาดต่าง ๆ จึงได้มีการประมาณค่าน้ำหนักสนับสนุนที่เป็นไปได้จากค่าน้ำหนักที่มากสุดในฐานข้อมูล ดังนั้น ows^d= 2.4 (0.6×4) จากการประมาณค่าดังกล่าวจะทำให้เราสามารถค้นหาเซตรายการ ad ได้แต่รายการ d จะไม่เป็นผลลัพธ์ของการค้นหาเซตรายการจากค่าน้ำหนักสนับสนุนที่แท้จริง

2.4 คุณลักษณะของฐานข้อมูลที่ใช้ทดสอบในงานวิจัยที่วิทยานิพนธ์นี้นำเสนอ

ฐานข้อมูลที่ใช้สำหรับทดสอบประสิทธิภาพของขั้นตอนวิธีที่นำเสนอในวิทยานิพนธ์นี้เป็นฐานข้อมูลมาตรฐานที่นิยมใช้ในการทดสอบประสิทธิภาพของขั้นตอนวิธีการค้นหากฎความสัมพันธ์ ซึ่งประกอบไปด้วยฐานข้อมูล 2 ประเภทได้แก่ 1) ฐานข้อมูลที่เป็นข้อมูลจริง (Accidents Chess Connect Kosarak Mushroom Pumsb Pumsb* และ Retail) 2) ฐานข้อมูลที่ข้อมูลถูกสังเคราะห์

ขึ้นซึ่งจัดทำและเผยแพร่โดย IBM Almaden[2] (T10I4D100K และ T40I10D100K) โดยฐานข้อมูลทั้งหมดสามารถดาวน์โหลดได้จากเว็บไซต์ fimi[3] ซึ่งคุณลักษณะของฐานข้อมูลที่ใช้ทดสอบในงานวิจัยที่วิทยานิพนธ์นี้นำเสนอสามารถแสดงรายละเอียดได้ ดังตารางที่ 2-1 สำหรับค่าน้ำหนักซึ่งบงชี้ถึงความสำคัญ/ความน่าสนใจของข้อมูลในแต่ละฐานข้อมูลได้รับจากการสุ่มแบบแจกแจงปกติ (Normal Distribution) โดยที่ช่วงของการสุ่มอยู่ที่ 0.1-0.9 ซึ่งดำเนินการตามงานวิจัยก่อนหน้า (Tao, 2003) (Wang et al., 2004) (Yun & Leggett, 2005) (Vo & Coenen, 2013)

ตารางที่ 2.1 คุณลักษณะของฐานข้อมูลที่ใช้ทดสอบในงานวิจัยที่วิทยานิพนธ์นี้นำเสนอ

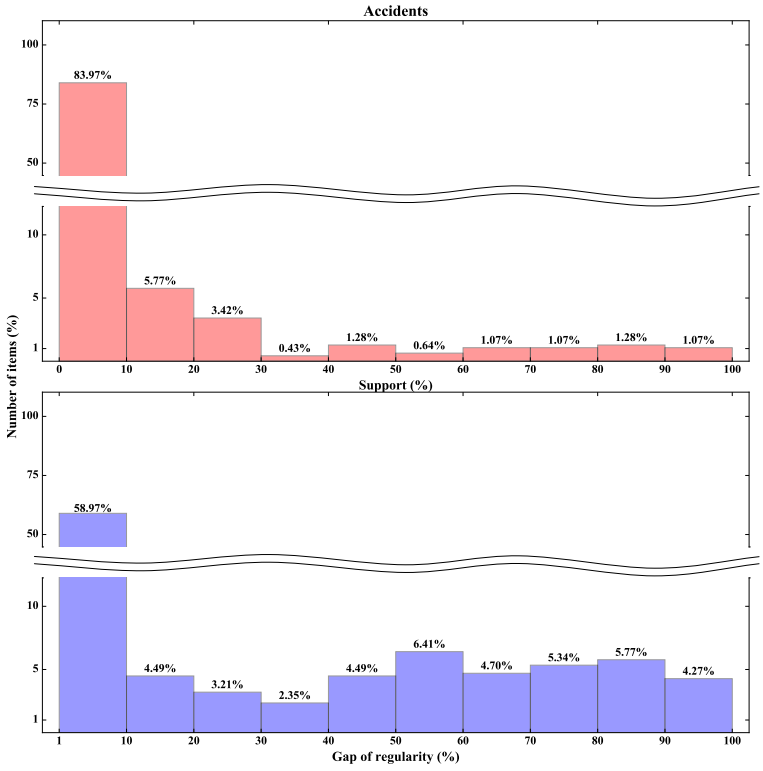
ฐานข้อมูล	จำนวนรายการ	จำนวนทรานแซกชันเฉลี่ย	จำนวนทรานแซกชันทั้งหมด	ลักษณะข้อมูล
Accidents	468	33.8	340,138	หนาแน่น
Chess	75	37	3,196	หนาแน่น
Connect	129	43	67,557	หนาแน่น
Mushroom	119	23	8,124	หนาแน่น
Pumsb	7,117	74	49,046	หนาแน่น
Pumsb*	7,117	50	49,046	หนาแน่น
Kosarak	41,270	8.1	990,002	เบาบาง
Retail	16,470	10.3	88,126	เบาบาง
T10I4D100k	1,000	10	100,000	เบาบาง
T40I10D100K	1,000	40	100,000	เบาบาง

จากตารางที่ 2.1 แสดงให้เห็นถึงคุณลักษณะของฐานข้อมูลที่จะใช้ในการทดสอบประสิทธิภาพของขั้นตอนวิธีที่นำเสนอในวิทยานิพนธ์ ซึ่งจะแสดงให้เห็นถึงจำนวนของรายการ จำนวนความยาวเฉลี่ยของทรานแซกชัน และจำนวนทรานแซกชันทั้งหมดในแต่ละฐานข้อมูล นอกจากนี้ตารางที่ 2.1 ยังแสดงให้เห็นถึงลักษณะของข้อมูลภายในแต่ละฐานข้อมูลว่ามีลักษณะหนาแน่นหรือเบาบางอีกด้วย

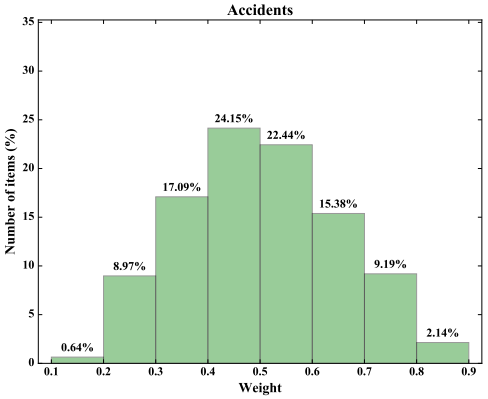
[2] http://www.almaden.ibm.com/cs/quest/syndata.html

[3] http://fimi.cs.helsinki.fi/data/

ฐานข้อมูล Accidents คือฐานข้อมูลที่รวบรวมข้อมูลการเกิดอุบัติเหตุในการจราจรทางบกบริเวณแถบ Flanders (พื้นที่ทางตอนเหนือของประเทศเบลเยี่ยม) ในช่วงปี ค.ศ. 1991-2000 ซึ่งจัดเก็บโดยสถาบันสถิติแห่งชาติ (National Institute of Statistics, NIS) โดยที่รายการ (item) ส่วนมากถึง 83.97% มีจำนวนครั้งของการปรากฏ (support) ในฐานข้อมูลอยู่ในช่วง 0-10% ของทรานแซกชันทั้งหมดภายในฐานข้อมูล (ฐานข้อมูล Accidents มีจำนวนทรานแซกชันทั้งหมด 340,183 ดังนั้น 0-10% ของ 340,183 เท่ากับ 0-34018) และรายการส่วนมากถึง 58.97% มีความสม่ำเสมอในการปรากฏ (regularity) ที่ซึ่งปรากฏห่างกันอย่างต่อเนื่องในช่วง 1-10% (ความสม่ำเสมอช่วง 1-10% คือปรากฏอย่างน้อยหนึ่งทรานแซกชันห่างกันอยู่ในช่วงที่ไม่เกิน 34,018) แสดงดังภาพที่ 2-4 และค่าน้ำหนักของฐานข้อมูล Accidents แสดงดังภาพที่ 2-5 ซึ่งรายการส่วนมากถึง 46.59% มีช่วงน้ำหนักอยู่ที่ 0.4-0.6



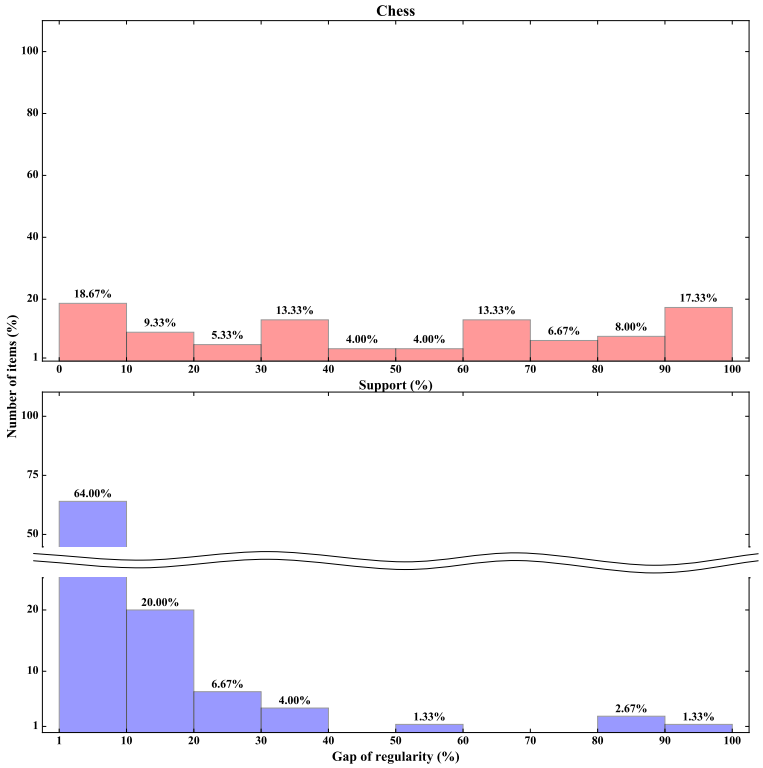
ภาพที่ 2-4 กราฟแสดงจำนวนครั้งและความสม่ำเสมอในการปรากฏของรายการภายในฐานข้อมูล Accidents



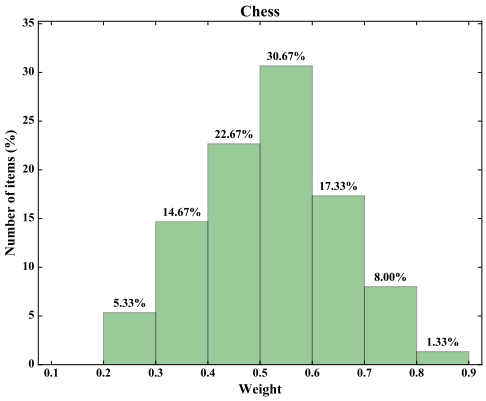
ภาพที่ 2-5 กราฟแสดงค่าน้ำหนักของฐานข้อมูล Accidents

ฐานข้อมูล Chess และ Connect ถูกรวบรวมโดย UCI Machine Learning Repository[4] ซึ่งแต่ละทรานแซกชันได้จัดเก็บวิธีการเดินหมากในระหว่างการแข่งขัน โดยที่รายการภายในฐานข้อมูล Chess มีจำนวนครั้งของการปรากฏในฐานข้อมูลที่ใกล้เคียงกันในทุก ๆ ช่วง และรายการส่วนมากถึง 64 % มีการปรากฏอย่างสม่ำเสมออยู่ในช่วง 1-10 % ของทรานแซกชันทั้งหมดภายในฐานข้อมูล (ฐานข้อมูล Chess มีจำนวนทรานแซกชันทั้งหมด 3,196 ความสม่ำเสมอ 1-10% คือปรากฏอย่างน้อยหนึ่งทรานแซกชันห่างกันอยู่ในช่วงที่ไม่เกิน 319) แสดงดังภาพที่ 2-6 สำหรับค่าน้ำหนักของฐานข้อมูล Chess ซึ่งรายการส่วนมาก 30.67% มีช่วงน้ำหนักอยู่ที่ 0.5-0.6 แสดงดังภาพที่ 2-7

[4] http://www.ics.uci.edu/ mlearn/MLRepository.html



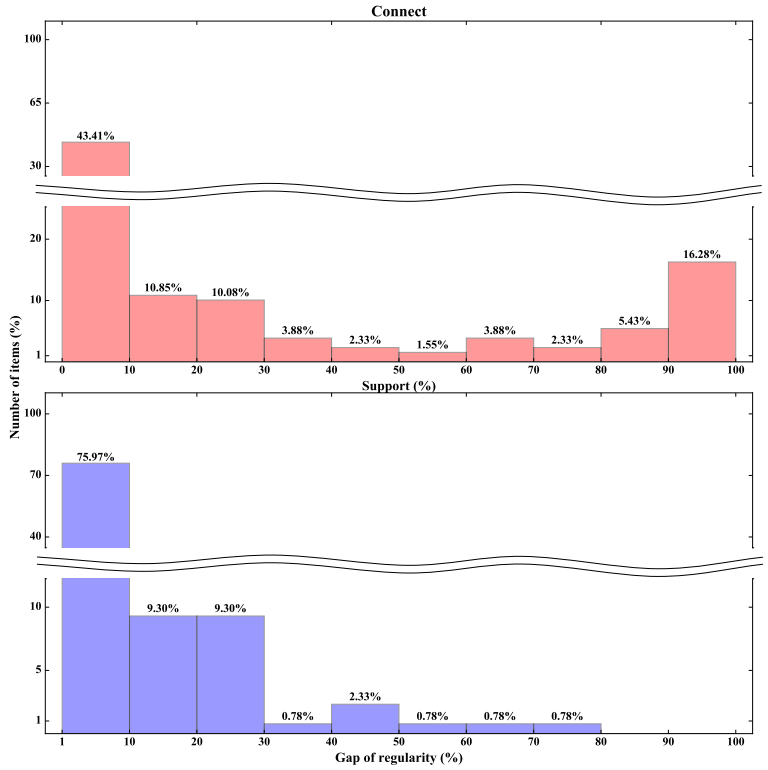
ภาพที่ 2-6 กราฟแสดงจำนวนครั้งและความสม่ำเสมอในการปรากฏของรายการภายในฐานข้อมูล Chess



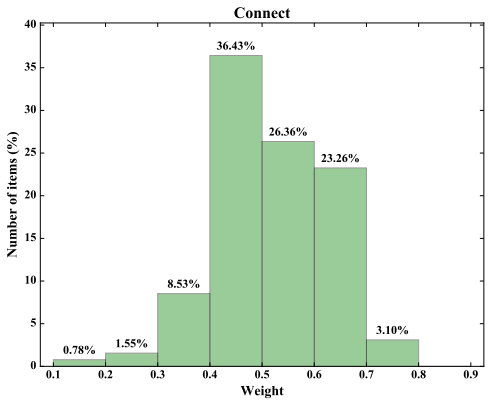
ภาพที่ 2-7 กราฟแสดงค่าน้ำหนักของฐานข้อมูล Chess

ฐานข้อมูล Connect รายการส่วนมากถึง 43.41% มีจำนวนครั้งของการปรากฏในฐานข้อมูลอยู่ที่ช่วง 0-10 % ของทรานแซกชันทั้งหมดภายในฐานข้อมูล (ฐานข้อมูล Connect มี

จำนวนทรานแซกชันทั้งหมด 67,557 ดังนั้น 0-10% ของ 67,557 เท่ากับ 0-6,756) และรายการที่เหลือมีจำนวนครั้งในการเกิดในแต่ละช่วงมากน้อยอย่างต่อเนื่องกัน แต่รายการส่วนมากถึง 75.97% ปรากฏอย่างสม่ำเสมอในช่วง 1-10% ของทรานแซกชันทั้งหมดในฐานข้อมูล (ความสม่ำเสมอ 1-10% คือปรากฏอย่างน้อยหนึ่งทรานแซกชันห่างกันอยู่ในช่วงที่ไม่เกิน 6,756) แสดงดังภาพที่ 2-8 และค่าน้ำหนักของฐานข้อมูล Connect แสดงดังภาพที่ 2-9 ซึ่งรายการถึง 36.43 มีน้ำหนักที่แสดงถึงความสำคัญ/ความน่าสนใจอยู่ในช่วง 0.4-0.5

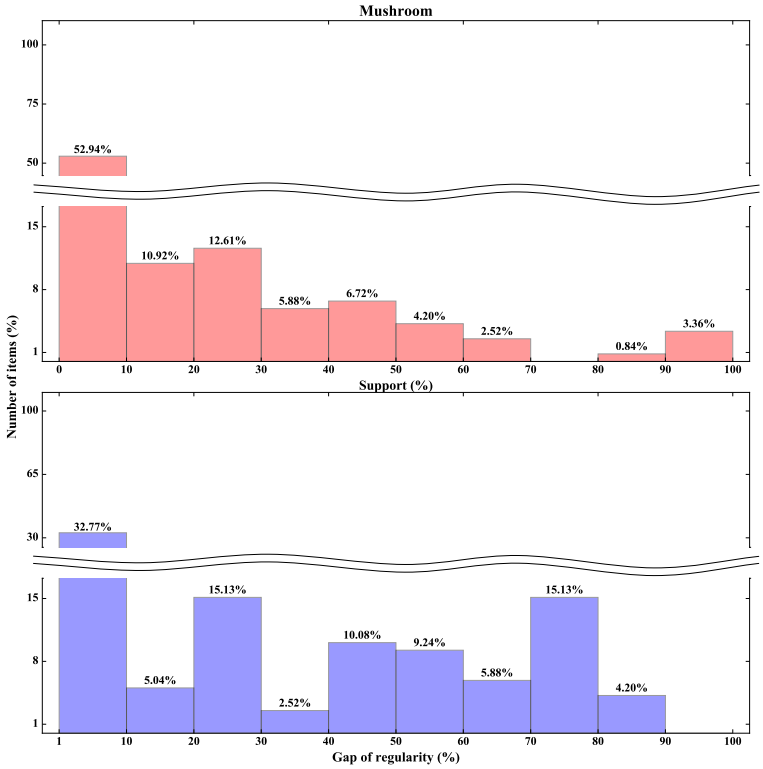


ภาพที่ 2-8 กราฟแสดงจำนวนครั้งและความสม่ำเสมอในการปรากฏของรายการภายในฐานข้อมูล Connect

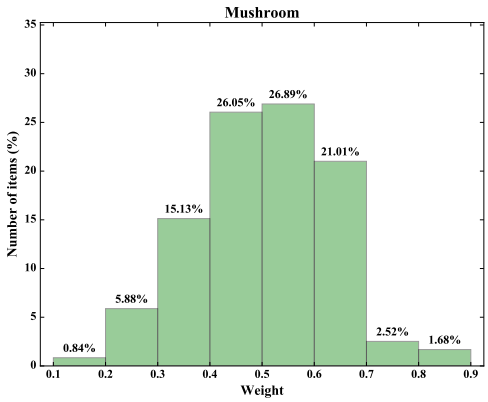


ภาพที่ 2-9 กราฟแสดงค่าน้ำหนักของฐานข้อมูล Connect

ฐานข้อมูล Mushroom เป็นฐานข้อมูลที่จัดเก็บคุณลักษณะสายพันธุ์ของเห็ดแต่ละชนิดซึ่งรายการส่วนมากปรากฏห่างกันอย่างสม่ำเสมอในช่วง 1-10% ของฐานข้อมูลทั้งหมด (ฐานข้อมูล Mushroom มีจำนวนทรานแซกชันทั้งหมด 8,124 ความสม่ำเสมอ 1-10% คือปรากฏอย่างน้อยหนึ่งทรานแซกชันห่างกันอยู่ในช่วงที่ไม่เกิน 812) และรายการส่วนมากถึง 52.94% มีจำนวนครั้งในการปรากฏภายในฐานข้อมูลอยู่ในช่วง 0-10% ของฐานข้อมูลทรานแซกชัน (0-10% ของ 8,124 เท่ากับ 0-812) แสดงดังภาพที่ 2-10 และภาพที่ 2-11 แสดงค่าน้ำหนักของฐานข้อมูล Mushroom ที่ซึ่งรายการส่วนมากมีค่าน้ำหนักอยู่ในช่วง 0.4-0.6

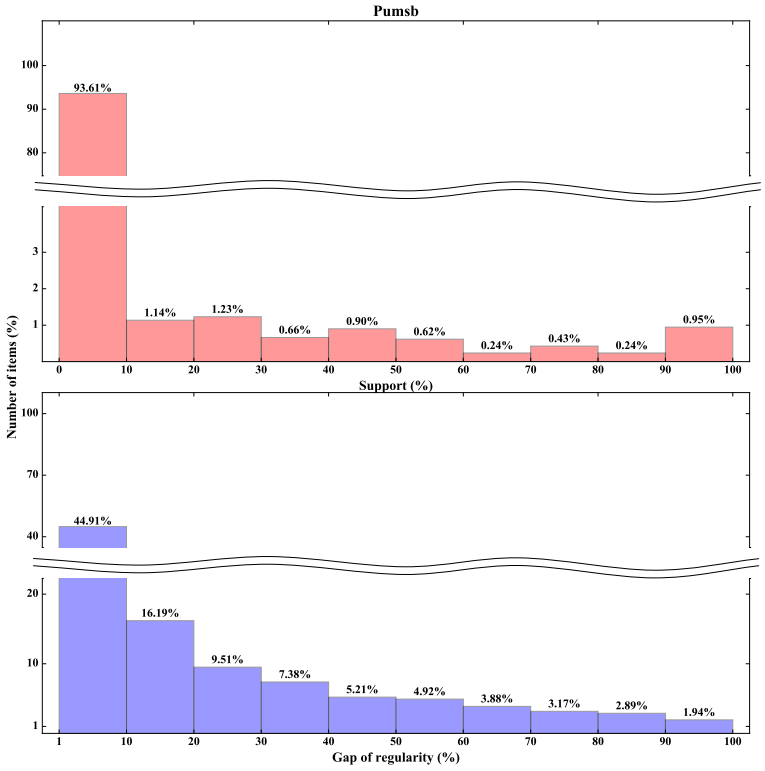


ภาพที่ 2-10 กราฟแสดงจำนวนครั้งและความสม่ำเสมอในการปรากฏของรายการภายในฐานข้อมูล Mushroom

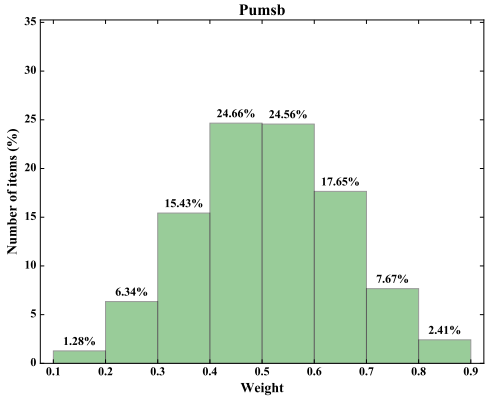


ภาพที่ 2.11 กราฟแสดงค่าน้ำหนักของฐานข้อมูล Mushroom

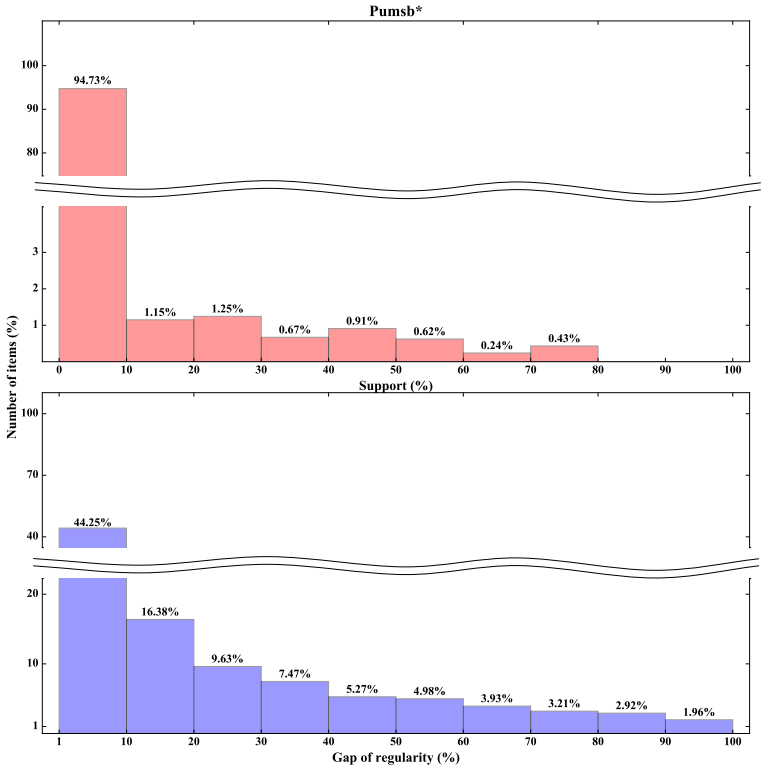
ฐานข้อมูล Pumsb และฐานข้อมูล Pumsb* เป็นฐานข้อมูลที่จัดเก็บข้อมูลสำมะโนประชากร ซึ่งรายการส่วนมาถึง 93.61% ในฐานข้อมูล Pumsb และรายการ 94.73% ในฐานข้อมูล Pumsb* มีจำนวนครั้งในการปรากฏอยู่ในช่วง 0-10% ของฐานข้อมูลทั้งหมด (ฐานข้อมูล Pumsb และฐานข้อมูล Pumsb* มีจำนวนทรานแซกชันทั้งหมด 49,046 ดังนั้น 0-10% ของ 49,046 เท่ากับ 0-4,905) และรายการส่วนมากประมาณ 40% ของฐานข้อมูลทั้งสองปรากฏห่างกันอย่างสม่ำเสมอในช่วง 1-10% ของฐานข้อมูลทั้งหมด (ความสม่ำเสมอ 1-10% คือปรากฏอย่างน้อยหนึ่งทรานแซกชันห่างกันอยู่ในช่วงที่ไม่เกิน 4,905) แสดงดังภาพที่ 2-12 และ 2-14 และภาพที่ 2-13 และ 2-15 แสดงค่าน้ำหนักของฐานข้อมูล Pumsb และ Pumsb* ซึ่งค่าน้ำหนักส่วนใหญ่ของรายการในฐานข้อมูลทั้งสองนั้นมีช่วงน้ำหนักอยู่ที่ 0-4-0.6



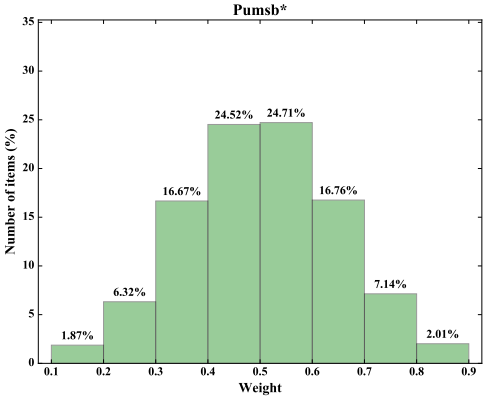
ภาพที่ 2-12 กราฟแสดงจำนวนครั้งและความสม่ำเสมอในการปรากฏของรายการภายในฐานข้อมูล Pumsb



ภาพที่ 2-13 กราฟแสดงค่าน้ำหนักของฐานข้อมูล Pumsb

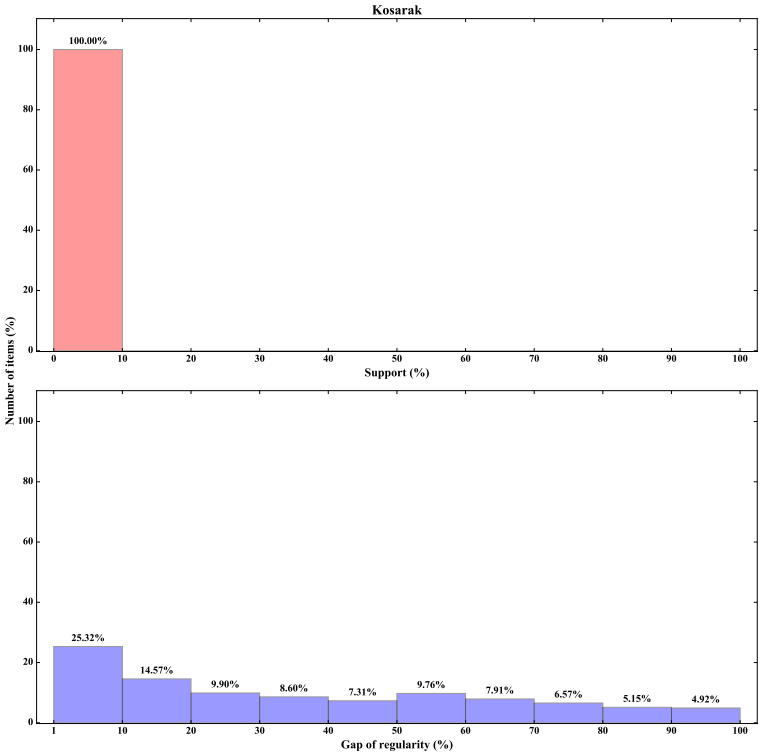


ภาพที่ 2-14 กราฟแสดงจำนวนครั้งและความสม่ำเสมอในการปรากฏของรายการภายในฐานข้อมูล Pumsb*

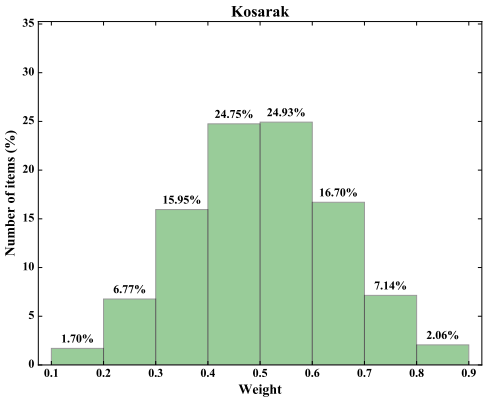


ภาพที่ 2-15 กราฟแสดงค่าน้ำหนักของฐานข้อมูล Pumsb*

ฐานข้อมูล Kosarak คือฐานข้อมูลที่จัดเก็บข้อมูลการคลิกในเว็บไซต์ข่าวออนไลน์ของประเทศฮังการี โดยที่รายการทั้งหมดในฐานข้อมูลมีจำนวนครั้งในการปรากฏอยู่ในช่วง 0-10% ของฐานข้อมูลทั้งหมด (ฐานข้อมูล Kosarak มีจำนวนทรานแซกชันทั้งหมด 990,002 ดังนั้น 0-10% ของ 990,002 เท่ากับ 0-99,000) และรายการส่วนมากถึง 25.32% ปรากฏห่างกันอย่างสม่ำเสมอในช่วง 1-10% ของฐานข้อมูลทั้งหมด (ความสม่ำเสมอ 1-10% คือปรากฏอย่างน้อยหนึ่งทรานแซกชันห่างกันอยู่ในช่วงที่ไม่เกิน 99,000) ส่วนรายการที่เหลือปรากฏห่างกันอย่างสม่ำเสมอในทุกช่วงของฐานข้อมูลดังภาพที่ 2-16 ค่าน้ำหนักของฐานข้อมูล Kosarak แสดงดังภาพที่ 2-17 ซึ่งค่าน้ำหนักส่วนใหญ่ของรายการในฐานข้อมูล Kosarak นั้นมีช่วงน้ำหนักอยู่ที่ 0-4-0.6



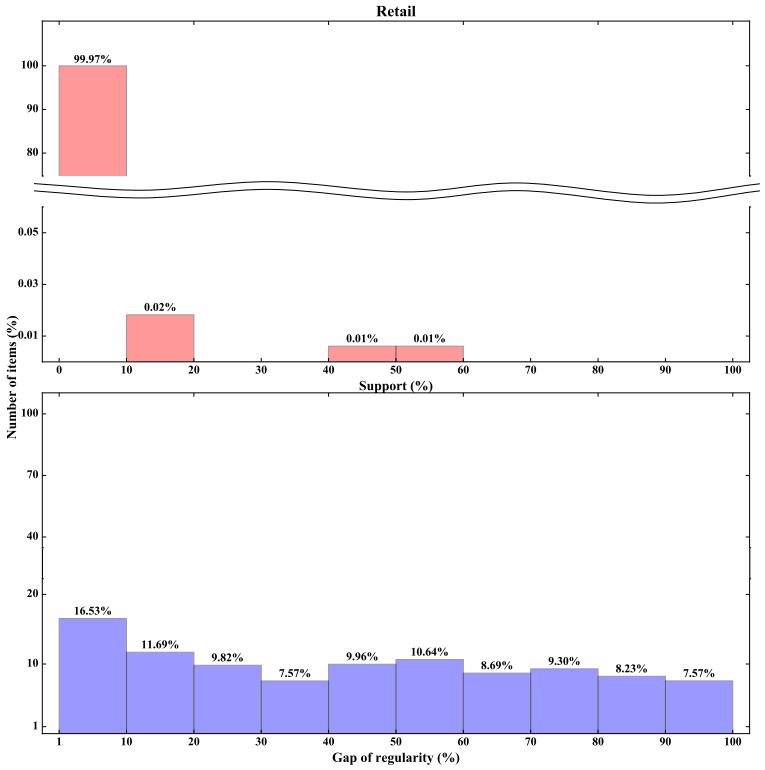
ภาพที่ 2-16 กราฟแสดงจำนวนครั้งและความสม่ำเสมอในการปรากฏของรายการภายในฐานข้อมูล Kosarak



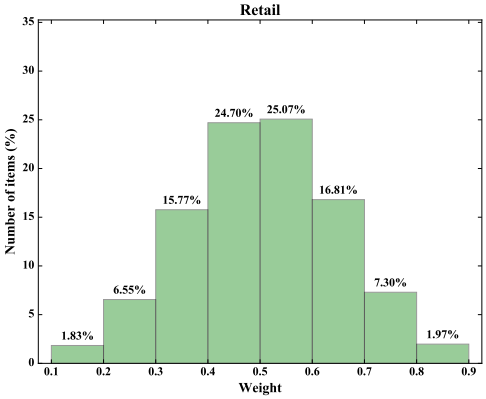
ภาพที่ 2-17 กราฟแสดงค่าน้ำหนักของฐานข้อมูล Kosarak

ฐานข้อมูล Retail จัดเก็บข้อมูลการซื้อสินค้าของลูกค้าในประเทศเบลเยี่ยมตั้งแต่เดือนธันวาคม ปี ค.ศ. 1999 ถึง เดือนพฤศจิกายน ปี ค.ศ. 2000 โดยที่แต่ละทรานแซกชันแสดงถึงข้อมูล

การซื้อสินค้าภายในตระกร้าสินค้า ซึ่งรายการส่วนใหญ่ในฐานข้อมูลถึง 99.79% มีจำนวนครั้งในการปรากฏอยู่ในช่วง 0-10% ของฐานข้อมูลทั้งหมด (ฐานข้อมูล Retail มีจำนวนทรานแซกชันทั้งหมด 88,126 ดังนั้น 0-10% ของ 88,126 เท่ากับ 0-8,813) และรายการทั้งหมดทั้งหมดภายในฐานข้อมูลปรากฏห่างกันอย่างสม่ำเสมอในทุกช่วงของฐานข้อมูลดังภาพที่ 2-18 และภาพที่ 2-19 แสดงค่าน้ำหนักของฐานข้อมูล Retail ซึ่งค่าน้ำหนักส่วนใหญ่ของรายการในฐานข้อมูล Retail นั้นมีช่วงน้ำหนักอยู่ที่ 0-4-0.6

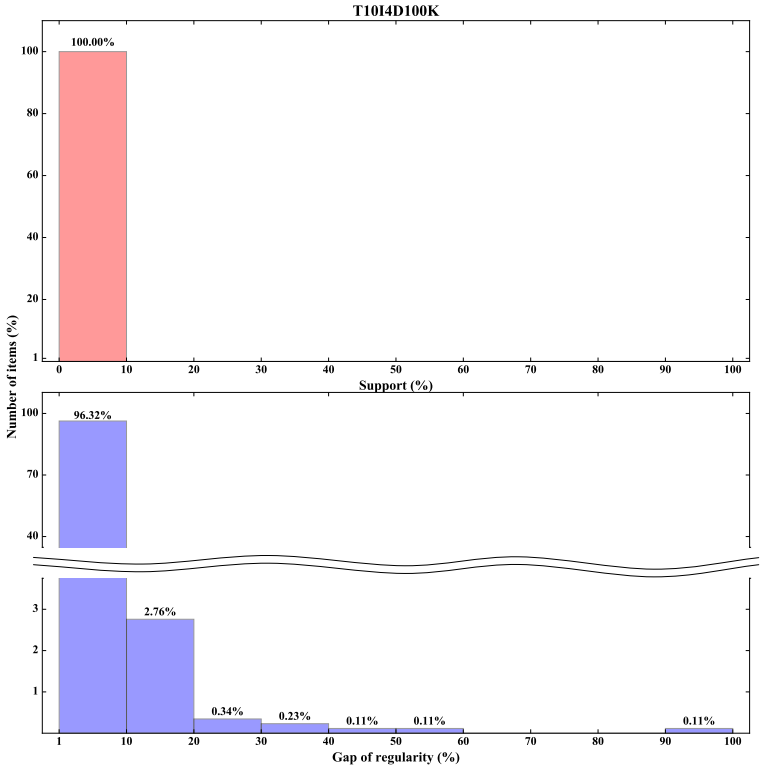


ภาพที่ 2-18 กราฟแสดงจำนวนครั้งและความสม่ำเสมอในการปรากฏของรายการภายในฐานข้อมูล Retail

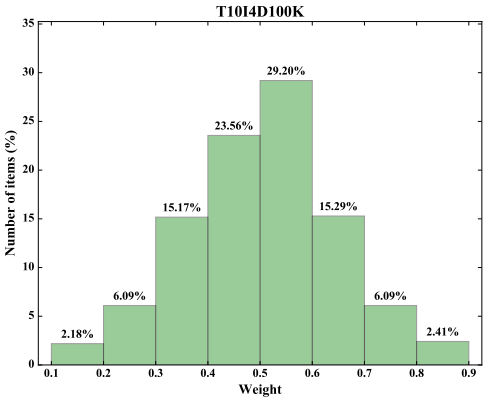


ภาพที่ 2-19 กราฟแสดงค่าน้ำหนักของฐานข้อมูล Retail

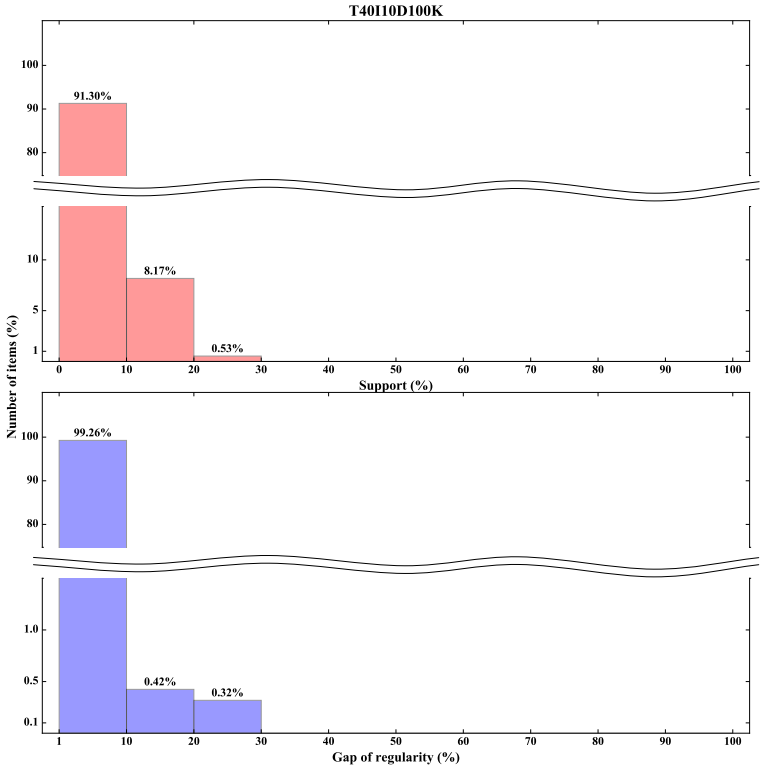
ฐานข้อข้อมูล T10I4D100 และ T40D100K เป็นฐานข้อมูลที่สังเคราะห์ข้อมูลขึ้นมาเลียนแบบทรานแซกชันของการซื้อสินค้าในธุรกิจค้าปลีก ซึ่งรายการส่วนมากในฐานข้อมูลทั้งสองมีจำนวนครั้งของการปรากฏในฐานข้อมูลอยู่ในช่วง 0-10% ของทรานแซกชันทั้งหมด (ฐานข้อมูล T10I4D100 และ T40I10D100K มีจำนวนทรานแซกชันทั้งหมด 100,000 ดังนั้น 0-10% ของ 100,000 เท่ากับ 0-10,000) และความสม่ำเสมอในการปรากฏของรายการส่วนใหญ่ถึง 96.32% ในฐานข้อมูล T10I4D100 และ 99.26% ในฐานข้อมูล T40I10D100K อยู่ในช่วง 1-10% ของทรานแซกชันทั้งหมดในฐานข้อมูล (ความสม่ำเสมอ 1-10% คือปรากฏอย่างน้อยหนึ่งทรานแซกชันห่างกันอยู่ในช่วงที่ไม่เกิน 10,000) แสดงดังภาพที่ 2-20 และ 2-22 สำหรับค่าน้ำหนักของฐานข้อมูล T10I4D100 และ T40I10D100K แสดงดังภาพที่ 2-21 และ 2-22 ซึ่งค่าน้ำหนักส่วนใหญ่ของรายการในฐานข้อมูลทั้งสองนั้นมีช่วงน้ำหนักอยู่ที่ 0-4-0.6



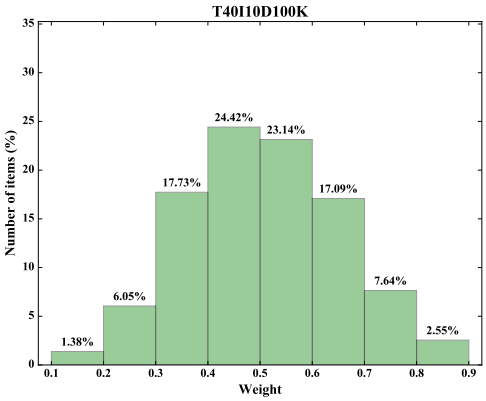
ภาพที่ 2-20 กราฟแสดงจำนวนครั้งและความสม่ำเสมอในการปรากฏของรายการภายในฐานข้อมูล T10I4D100K



ภาพที่ 2-21 กราฟแสดงค่าน้ำหนักของฐานข้อมูล T10I4D100K



ภาพที่ 2-22 กราฟแสดงจำนวนครั้งและความสม่ำเสมอในการปรากฏของรายการภายในฐานข้อมูล T40I10D100K



ภาพที่ 2-23 กราฟแสดงค่าน้ำหนักของฐานข้อมูล T40I10D100K

บทที่ 3
การค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ

การค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ (Mining weighted-frequent-regular itemsets from transactional database, WFRIM) เป็นปัญหาการค้นหาเซตรายการที่น่าสนใจจากฐานข้อมูล ซึ่งจะค้นหาเซตรายการที่มีลักษณะการปรากฏในฐานข้อมูลบ่อยและสม่ำเสมอโดยที่แต่ละรายการมีความสำคัญ/ความน่าสนใจที่แตกต่างกัน การค้นหาเซตรายการดังกล่าวถูกพัฒนามาจากการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอ (Tanbeer, et al., 2009) และเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ (Cai et al., 1998) (Ahmed et al., 2012) ดังนั้นในบทนี้จะกล่าวถึงนิยามและขอบเขตของปัญหารวมถึงขั้นตอนวิธีสำหรับการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการที่มีชื่อเรียกว่า *Weighted-Frequent-Regular Itemsets Miner (WFRIM)* โดยแบ่งออกเป็นส่วนต่าง ๆ ดังต่อไปนี้

1. นิยามและขอบเขตของปัญหาของการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ
2. วิธีการสำหรับการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ
3. ตัวอย่างสำหรับวิธีการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ
4. การวิเคราะห์ความซับซ้อนของขั้นตอนวิธี

3.1 การค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ (Mining weighted-frequent-regular itemsets mining)

ในส่วนนี้จะกล่าวถึงนิยามและคำจำกัดความของเซตรายการที่ปรากฏบ่อยและสม่ำเสมอ (Tanbeer et al., 2009) และเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ (Ahmed et al., 2012) จากนั้นจะกล่าวถึงปัญหาและขอบเขตรวมถึงนิยามและคำจำกัดความของเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ แสดงดังต่อไปนี้

กำหนดให้ $I = \{i_1, i_2, \dots, i_n\}$ เป็นเซตของรายการ (items) โดยที่แต่ละรายการ $i_p \in I$ มีค่าน้ำหนัก w_{i_p} ที่บ่งชี้ถึงความสำคัญ/ความน่าสนใจของรายการนั้น ๆ (กล่าวคือ สำหรับเซต $I = \{i_1, i_2, \dots, i_n\}$ จะมีเซตค่าน้ำหนัก $W = \{w_1, w_2, \dots, w_n\}$ ที่ซึ่งแต่ละ $w_i \in W$ แสดงถึงค่าน้ำหนักความสำคัญของรายการ $i_j \in I$) เซต $X = \{i_p, \dots, i_q\}$ เมื่อ $1 \leq p \leq q \leq n$ คือเซตของรายการ (รูปแบบหรือ k-เซตรายการ ถ้า X ประกอบไปด้วย k รายการที่แตกต่างกัน) ฐานข้อมูลทรานแซกชัน $TDB = \{t_1, t_2, \dots, t_m\}$ เป็นเซตของทรานแซกชัน ที่ซึ่งแต่ละทรานแซกชัน $t_j \in TDB$ บรรจุไปด้วย 2 ข้อมูลคือ j เป็นหมายเลขทรานแซกชัน (tid) และเซตรายการ Y จากข้อมูลในทรานแซกชัน t_j ถ้าเซตรายการ $X \subseteq Y$ สามารถอธิบายได้ว่าเซตรายการ X ปรากฏในทรานแซกชัน t_j หรือ ทรานแซกชัน t_j มีเซตรายการ X บรรจุอยู่ ดังนั้นเมื่อทำการพิจารณาการปรากฏขึ้นของเซตรายการ X หนึ่ง ๆ จะทำให้ทราบถึงเซต $T^X = \{t_j^X, \dots, t_k^X\}$ ซึ่งคือเซตของหมายเลขทรานแซกชันที่เซตรายการ X ปรากฏอยู่ เมื่อ $j, k \in [1, m]$ ซึ่ง $j \leq k$ ได้

3.1.1 ค่าความสม่ำเสมอของเซตรายการและเซตรายการที่ปรากฏบ่อยและสม่ำเสมอ

นิยามที่ 3.1 (ค่าความสม่ำเสมอของเซตรายการ X ที่ปรากฏในทรานแซกชัน) ค่าความสม่ำเสมอของเซตรายการ X ที่ปรากฏในทรานแซกชัน คือระยะห่างระหว่างทรานแซกชันที่เซตรายการ X ปรากฏขึ้น กำหนดให้ทรานแซกชัน t_k เป็นทรานแซกชันที่มีเซตรายการ X ปรากฏอยู่ ค่าความสม่ำเสมอของเซตรายการ X ที่ปรากฏในทรานแซกชัน t_k ($r_{t_k}^X$) สามารถคำนวณได้ 3 กรณีดังนี้

1) ถ้า t_k คือทรานแซกชันที่เซตรายการ X ปรากฏขึ้นในฐานข้อมูลทรานแซกชันเป็นครั้งแรกค่าความสม่ำเสมอ $r_{t_k}^X$ มีค่าเท่ากับ k ดังสมการที่ 3.1

$$r_{t_k}^X = k \tag{3.1}$$

(หมายเหตุ ค่า $r_{t_k}^X$ แสดงถึงระยะห่างระหว่างทรานแซกชันแรก (เริ่มต้นด้วย 0) จนถึทรานแซกชัน k ที่เซตรายการ X ปรากฏขึ้นเป็นครั้งแรก)

2) ถ้า t_k คือทรานแซกชันที่ปรากฏขึ้นหลังจากทรานแซกชัน t_j ซึ่งทรานแซกชัน t_k และ t_j มีเซตรายการ X ปรากฏขึ้นทั้งคู่ ดังนั้นค่า $r_{t_k}^X$ สามารถคำนวณได้ ดังสมการที่ 3.2

$$r_{t_k}^X = k - j \tag{3.2}$$

(หมายเหตุ ค่า $r_{t_k}^X$ แสดงถึงระยะห่างระหว่างทรานแซกชัน t_j และทรานแซกชัน t_k ที่เซตรายการ X ปรากฏขึ้นต่อเนื่องกัน)

3) ถ้า t_k คือทรานแซกชันที่เซตรายการ X ปรากฏขึ้นในฐานข้อมูลทรานแซกชันเป็นครั้งสุดท้าย ดังนั้นค่า $r_{t_k}^X$ สามารถคำนวณได้ ดังสมการที่ 3.3

$$r_{t_k}^X = m - k \tag{3.3}$$

เมื่อ m คือจำนวนของทรานแซกชันทั้งหมดในฐานข้อมูล

(หมายเหตุ ค่า $r_{t_k}^X$ แสดงถึงระยะห่างระหว่างทรานแซก t_k ที่เซตรายการ X ปรากฏขึ้นจนถึงทรานแซกชันสุดท้ายของฐานข้อมูล)

นิยามที่ 3.2 (ค่าความสม่ำเสมอของเซตรายการ X) ค่าความสม่ำเสมอของเซตรายการ X คือระยะห่างมากที่สุดระหว่างทรานแซกชันที่เซตรายการ X ปรากฏขึ้นอย่างน้อยหนึ่งครั้งในฐานข้อมูล ทรานแซกชัน TDB สามารถคำนวณได้โดยสมการที่ 3.4

$$r^X = max\{r_{t_j}^X, r_{t_k}^X, \dots, r_{t_l}^X, r_{t_l}^X\} \tag{3.4}$$

เมื่อ $j, k, l \in [1, m]$ ตามลำดับ

(หมายเหตุ ค่าความสม่ำเสมอ $r_{t_l}^X$ มีการคำนวณทั้งในกรณีที่ 2 และ 3)

นิยามที่ 3.3 (ค่าสนับสนุนของเซตรายการ X) ค่าสนับสนุนของเซตรายการ X คือจำนวนทรานแซกชันที่เซตรายการ X ปรากฏในฐานข้อมูลฐานแซกชัน TDB สามารถเขียนระบุได้เป็น s^X และคำนวณได้โดย $s^X = |T^X|$

นิยามที่ 3.4 (เซตรายการที่ปรากฏบ่อยและสม่ำเสมอ) เซตรายการ X จะถูกระบุว่าเป็นเซตรายการที่ปรากฏบ่อยและสม่ำเสมอ ก็ต่อเมื่อ 1) ค่าสนับสนุนของเซตรายการ X มีค่ามากกว่าหรือเท่ากับค่าขีดแบ่งสนับสนุนขั้นต่ำ (minimum support threshold, σ_s) ($s^X \geq \sigma_s$) และ 2) ค่าความสม่ำเสมอของเซตรายการ X มีค่าน้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอ (maximum regularity threshold, σ_r) ($r^X \leq \sigma_r$) โดยที่ค่าขีดแบ่งทั้งสองจะถูกกำหนดจากผู้ใช้

3.1.2 ค่าน้ำหนักสนับสนุนของเซตรายการและเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ

นิยามที่ 3.5 (ค่าน้ำหนักของเซตรายการ X) ค่าน้ำหนักของเซตรายการ $X = \{i_p, \dots, i_q\} \subseteq I$ คือ ค่าเฉลี่ยน้ำหนักของรายการทั้งหมดภายในเซตรายการ X สามารถคำนวณได้โดยสมการ 3.5

$$w^X = \frac{\sum_{p=1}^{|X|} w^{ip}}{|X|} \tag{3.5}$$

นิยามที่ 3.6 (ค่าน้ำหนักสนับสนุนของเซตรายการ X) ค่าน้ำหนักสนับสนุนของเซตรายการ X คือผลลัพธ์ที่ได้จากการคูณระหว่างค่าน้ำหนัก (w^X) และค่าสนับสนุน (s^X) ของเซตรายการ X ซึ่งสามารถคำนวณได้จากสมการที่ 3.6

$$ws^X = s^X \times w^X \tag{3.6}$$

นิยามที่ 3.7 (เซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ) เซตรายการ X จะถูกระบุว่าเป็นเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ ก็ต่อเมื่อค่าน้ำหนักสนับสนุนของเซตรายการ X มีค่ามากกว่าหรือเท่ากับค่าขีดแบ่งน้ำหนักสนับสนุนที่ผู้ใช้กำหนด (weighted-support threshold, σ_{ws}) ($ws^X \geq \sigma_{ws}$)

นิยามที่ 3.8 (ค่าน้ำหนักที่มากสุด) ค่าน้ำหนักที่มากสุดในฐานข้อมูล (Global maximum weight, $GMAXW$) คือค่าน้ำหนักที่มากที่สุดของทุกรายการที่เป็นสมาชิกของ I สามารถคำนวณได้จากสมการที่ 3.7

$$GMAXW = max\{w_1, w_2, \dots, w_n\} \tag{3.7}$$

นิยามที่ 3.9 (ค่าน้ำหนักที่มากสุดของเซตรายการ X) ค่าน้ำหนักที่มากสุดของเซตรายการ X (Local maximum weight, $LMAXW$) คือ ค่าเฉลี่ยผลรวมของค่าน้ำหนักของทุกรายการ $i_j \in X$ รวมกับค่าน้ำหนักที่มากสุดของรายการ $i_j \notin X$ (หมายเหตุ รายการที่ไม่ได้เป็นสมาชิกของ X สามารถนิยามได้เป็น $Y = I - X$ ซึ่ง $Y = \{i_r, i_s, \dots, i_u\}$ ดังนั้น ค่าน้ำหนักที่มากสุด

ของรายการที่ไม่เป็นสมาชิกของ X สามารถคำนวณได้โดย $max\{w_{i_r}, w_{i_s}, \dots, w_{i_u}\}$) ที่ซึ่งสามารถคำนวณได้ดังนี้

$$LMAXW^X = \frac{\sum_{i_j \in X} w^{i_j} + max\{w^{i_r}, w^{i_s}, \dots, w^{i_u}\}}{|X|+1} \quad (3.8)$$

นิยามที่ 3.10 (การประมาณค่าน้ำหนักสนับสนุนของรายการ $i_j \in I$) สำหรับการประมาณค่าน้ำหนักสนับสนุน (overestimated-weight-frequency, ows) ของรายการ $i_j \in I$ นั้นสามารถประมาณค่าได้จากประยุกต์ใช้ค่าน้ำหนักที่มากสุด ($GMAXW$) ร่วมกับค่าสนับสนุนของเซตรายการ i_j ซึ่งจะแสดงถึงค่าน้ำหนักสนับสนุนที่มากสุดที่เป็นไปได้ของรายการ i_j ดังสมการที่ 3.9

$$ows^{i_j} = GMAXW \times s^{i_j} \quad (3.9)$$

(หมายเหตุ จากนิยามที่ 3.7 ถ้าค่า ows^{i_j} ของรายการ $i_j \in I$ มีค่าน้อยกว่าค่าขีดแบ่งน้ำหนักสนับสนุน ($ows^{i_j} < \sigma_{ws}$) จะสามารถสรุปได้ว่ารายการ i_j ไม่เป็นเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ ส่งผลให้ซุปเปอร์เซตของ i_j จะไม่ถูกพิจารณาไปด้วย)

นิยามที่ 3.11 (การประมาณค่าน้ำหนักสนับสนุนของเซตรายการ $Y = X \cup i_j \in I$) สำหรับการประมาณค่าน้ำหนักสนับสนุน (overestimated-weight-frequency, ows) ของเซตรายการ $Y = X \cup i_j \in I$ นั้นสามารถประมาณค่าได้จากประยุกต์ใช้ค่าน้ำหนักที่มากสุดของเซตรายการ X ($LMAXW$) ร่วมกับค่าสนับสนุนของเซตรายการ X ซึ่งจะแสดงถึงค่าน้ำหนักสนับสนุนที่มากสุดที่เป็นไปได้ของเซตรายการ Y ดังสมการที่ 3.10

$$ows^Y = LMAXW^X \times s^X \quad (3.10)$$

(หมายเหตุ จากนิยามที่ 3.7 ถ้าค่า ows^Y ของเซตรายการ Y มีค่าน้อยกว่าค่าขีดแบ่งน้ำหนักสนับสนุน ($ows^Y < \sigma_{ws}$) จะสามารถสรุปได้ว่าเซตรายการ Y ไม่เป็นเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ ส่งผลให้ซุปเปอร์เซตของ Y จะไม่ถูกพิจารณาไปด้วย)

3.1.3 เซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความของแต่ละรายการ

จากแนวคิดของเซตรายการที่ปรากฏบ่อยและสม่ำเสมอและเซตรายการที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการที่กล่าวมาข้างต้น ทำให้สามารถนิยามเซตรายการที่ปรากฏบ่อยและสม่ำเสมอในฐานข้อมูลภายใต้เงื่อนไขที่แต่ละรายการมีค่าน้ำหนักความสำคัญที่แตกต่างกันได้ดังนี้

นิยามที่ 3.12 (เซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ) เซตรายการ X จะถูกระบุว่าเป็นเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ ก็ต่อเมื่อ 1) ค่าน้ำหนักสนับสนุนของเซตรายการ X มีค่ามากกว่าหรือเท่ากับค่าขีดแบ่งน้ำหนักสนับสนุน ($ws^X \geq \sigma_{ws}$) และ 2) ค่าความสม่ำเสมอของเซตรายการ X มีค่าน้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอ ($r^X \leq \sigma_r$) โดยที่ค่าขีดแบ่งทั้งสองจะถูกกำหนดจากผู้ใช้

นิยามปัญหา (problem statement) กำหนดให้มีฐานข้อมูลทรานแซกชัน TDB พร้อมด้วยเซตค่าน้ำหนักของแต่ละรายการ $W = \{w_1, w_2, \dots, w_n\}$ ค่าขีดแบ่งน้ำหนักสนับสนุน σ_{ws} และค่าขีดความแบ่งสม่ำเสมอ σ_r สำหรับปัญหาการค้นหาเซตรายการที่เป็นผลลัพธ์ของเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการนั้น แต่ละรายการจะต้องมีค่าน้ำหนักสนับสนุนไม่น้อยกว่า σ_{ws} และค่าความสม่ำเสมอไม่มากกว่า σ_r

3.2 วิธีการสำหรับค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ

ในส่วนนี้จะนำเสนอขั้นตอนวิธีการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ (Weighted-Frequent-regular Itemsets Miner, WFRIM) ซึ่งขั้นตอนวิธีดังกล่าวใช้สำหรับค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้เงื่อนไขที่แต่ละรายการมีความสำคัญหรือความน่าสนใจแตกต่างกัน โดยที่ขั้นตอนวิธี WFRIM ใช้โครงสร้างในการจัดเก็บข้อมูลสำหรับการค้นหาเซตรายการขนาดต่าง ๆ ที่คล้ายกับ *FP-tree* ที่เรียกว่า *WFRI-tree* รวมถึงมีการประยุกต์ใช้คุณสมบัติ downward closure property ร่วมกับค่าน้ำหนักที่มากสุด (global maximum weight) และค่าน้ำหนักที่มากสุดของเซตรายการ X ที่พิจารณา (local maximum weight) เพื่อลดทอนการพิจารณาเซตรายการที่ไม่เป็นผลลัพธ์

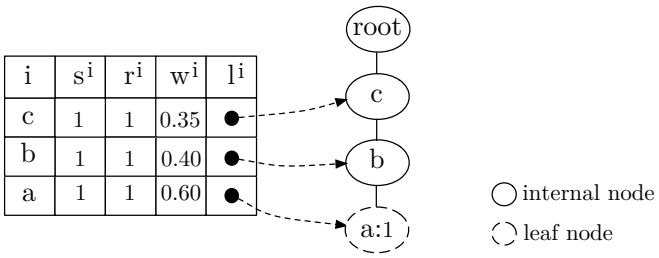
3.2.1 โครงสร้างของตารางรายการและ *WFRI-tree*

3.2.1.1 ตารางรายการ (Header table)

ตารางรายการ (Header table) คือโครงสร้างข้อมูลที่ใช้สำหรับจัดเก็บรายการและข้อมูลที่สำคัญของรายการต่าง ๆ ซึ่งตารางรายการดังกล่าวจะมีการเชื่อมโยง (link) กับ *WFRI-tree* โดยที่แต่ละรายการภายในตารางรายการจะต้องเป็นรายการที่ปรากฏสม่ำเสมอและค่าน้ำหนักสนับสนุนโดยประมาณ (ows) มากกว่าค่าขีดแบ่งน้ำหนักสนับสนุนที่ผู้ใช้กำหนดไว้ ซึ่งตารางรายการประกอบไปด้วย 5 ส่วน ดังนี้ 1) ชื่อรายการ (i) 2) ค่าสนับสนุนของรายการ i (s^i) 3) ค่าความสม่ำเสมอของรายการ i (r^i) 4) ค่าน้ำหนักของรายการ i (w^i) และ 5) การเชื่อมโยงแบบแนวนอน (horizontal link) ของรายการ i ภายในตารางรายการและโหนด (node) ของรายการ i ใน WFRI-tree (l^i) แสดงดังภาพที่ 3-1

3.2.1.2 *WFRI-Tree*

WFRI-tree คือ โครงสร้างต้นไม้ที่ใช้จัดเก็บเซตรายการในขณะที่ทำการค้นหาเซตรายการขนาดต่าง ๆ ที่เป็นผลลัพธ์ ซึ่งภายใน *WFRI-tree* จะประกอบไปด้วยหลายเส้นทาง (path) โดยที่แต่ละเส้นทางบ่งบอกถึงเซตของรายการที่ปรากฏขึ้นภายในทรานแซกชันนั้น ๆ สำหรับแต่ละเส้นทางจะประกอบด้วยโหนด 2 ประเภทคือ 1) โหนดภายใน (internal node) ที่ซึ่งทำการจัดเก็บชื่อรายการหนึ่ง ๆ ในเซตรายการและการเชื่อมโยงถึงโหนดพ่อแม่ (parent node) และโหนดลูก (children node) และ 2) โหนดสุดท้าย (leaf node) ทำการจัดเก็บรายการสุดท้ายในเซตรายการและทำการจัดเก็บเซตของหมายเลขทรานแซกชันที่เซตรายการปรากฏขึ้น แสดงดังภาพที่ 3-1 จากภาพแสดงให้เห็นเส้นทางของเซตรายการที่ปรากฏขึ้นภายในทรานแซกชัน t_1 ซึ่งแต่ละโหนดภายในเส้นทางจะเรียงลำดับค่าน้ำหนักของแต่ละรายการจากน้อยไปมาก จากภาพโหนด c โหนด b เป็นโหนดภายในและโหนด a เป็นโหนดสุดท้ายจัดเก็บหมายเลขทรานแซกชัน 1 ไว้ ซึ่งแต่ละโหนดใน *WFRI-tree* จะมีการเชื่อมโยงกับแต่ละรายการในตารางรายการอีกด้วย



ภาพที่ 3-1 โครงสร้างของตารางรายการและโครงสร้างต้นไม้ *WFRI-tree*

3.2.2 ขั้นตอนวิธี WFRIM (WFRIM algorithm)

ขั้นตอนวิธี WFRIM ประกอบไปด้วย 2 ขั้นตอนคือ 1) WFRIM-initialization ที่จะทำการอ่านฐานข้อมูลเพื่อสร้างตารางรายการ (header table) และสร้าง *WFRI-tree* และ 2) WFRIM-growth เป็นขั้นตอนการค้นหาเซตรายการที่เป็นผลลัพธ์ของเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการบน *WFRI-tree* ที่ได้รับจากขั้นตอน WFRIM-initialization โดยใช้แนวคิดพื้นฐานของ pattern growth

3.2.2.1 WFRIM-Initialization

WFRIM-initialization เป็นขั้นตอนวิธีสำหรับค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการและสร้าง *WFRI-tree* เพื่อใช้สำหรับจัดเก็บเซตรายการเพื่อการค้นหาเซตรายการในขนาดต่าง ๆ ซึ่ง WFRIM-initialization เริ่มต้นด้วยการสร้างตารางรายการ H สำหรับเก็บข้อมูลแต่ละรายการ $i_p \in I$ โดยกำหนดค่าสนับสนุนและค่าความสม่ำเสมอ ให้มีค่าเริ่มต้นเป็น 0 (ขั้นตอนวิธีที่ 3.1 บรรทัดที่ 1) จากนั้นอ่านฐานข้อมูลแต่ละทรานแซกชัน t_j และอ่านแต่ละรายการ $i_p \in t_j$ เพื่อทำการอัพเดทค่าสนับสนุน s^{i_p} และค่าความสม่ำเสมอ r^{i_p} (ขั้นตอนวิธีที่ 3.1 บรรทัดที่ 2-5) จนกระทั่งอ่านฐานข้อมูลครบทุกทรานแซกชัน เมื่ออ่านฐานข้อมูลครบแล้วทำการลบรายการที่มีค่าความสม่ำเสมอมากกว่าค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนดออกจากตารางรายการ H (ขั้นตอนวิธีที่ 3.1 บรรทัดที่ 6-8) (หมายเหตุ รายการที่ถูกลบออกจะถือว่ารายการนั้นเป็นรายการที่ปรากฏไม่สม่ำเสมอหรือไม่สามารถเป็นผลลัพธ์ของเซตรายการที่ปรากฏบ่อยและสม่ำเสมอ) หลังจากนั้นเรียงลำดับแต่ละรายการ i_p ภายในตารางรายการ H จากค่าน้ำหนักน้อยไปมาก (ขั้นตอนวิธีที่ 3.1 บรรทัดที่ 9) แล้วคำนวณหาค่าน้ำหนักที่มากสุด ($GMAXW$) เพื่อใช้สำหรับประมาณค่าน้ำหนักสนับสนุนของแต่ละรายการ $i_p \in H$ จากขั้นตอนนี้รายการ i_p ใดมีค่าน้ำหนักสนับสนุนที่ได้รับจากการประมาณค่า (ows^{i_p}) น้อยกว่าค่าขีดแบ่งน้ำหนักสนับสนุนที่ผู้ใช้กำหนดจะถูกลบออกจากตารางรายการ (ขั้นตอนวิธีที่ 3.1 บรรทัดที่ 10-16) ซึ่งขั้นตอนวิธีการค้นหา $GMAXW$ และการประมาณค่า ows^{i_p} จะมีการทำงานแบบวนซ้ำจนกระทั่งรายการทั้งหมดในตารางรายการ H ที่มีค่าน้ำหนักเท่ากับ $GMAXW$ ถูกลบออกจากตารางรายการ H สำหรับแต่ละรายการ i_p ในตารางรายการ H จะถูกกำหนดให้เป็นเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการจากค่าน้ำหนักที่แท้จริงคูณกับค่าสนับสนุนของรายการดังกล่าวแล้วมีค่าไม่น้อยกว่าค่าขีดแบ่งน้ำหนักสนับสนุนที่ถูกกำหนดโดยผู้ใช้ (ขั้นตอนวิธีที่ 3.1 บรรทัดที่ 17-19)

สำหรับการอ่านฐานข้อมูลรอบที่ 2 (ขั้นตอนวิธีที่ 3.1 บรรทัดที่ 20-31) โหนดราก (root node) ของ *WFRI-tree* จะถูกสร้างขึ้นเป็นอันดับแรก โดยในการอ่านแต่ละทรานแซกชัน t_j

จะทำการพิจารณาแต่ละรายการ $i_p \in t_j$ และทำการตรวจสอบแต่ละรายการ i_p ว่ามีรายการดังกล่าวอยู่ในตารางรายการ H หรือไม่ ถ้ารายการดังกล่าวไม่มีอยู่ใน H จะถูกตัดออกจากทรานแซกชัน t_j เมื่อทำการพิจารณาทุกรายการ $i_p \in t_j$ แล้ว ต่อไปจะทำการเรียงลำดับแต่ละรายการ i_p ที่เหลืออยู่ภายใน t_j ตามลำดับของตารางรายการ H จากนั้นสร้างเส้นทางของทรานแซกชัน t_j ใน *WFRI-tree* แล้วเก็บหมายเลขทรานแซกชัน j ไว้ที่โหนดสุดท้าย (หมายเหตุ ถ้าเส้นทางที่มีอยู่ใน *WFRI-tree* มีเส้นทางเหมือนกับทรานแซกชัน t_j ไม่ต้องทำการสร้างเส้นทางใหม่แต่ให้เพิ่มหมายเลขทรานแซกชัน j ที่โหนดสุดท้ายของเส้นทางนั้น)

Algorithm 3.1 WFRIM-initialization

Input: TDB, σ_r, σ_{ws}
Output: *WFRI-Tree*, *WFRIs*

create a header-table H with an entry for each item $i_p \in I$
for each transaction t_j in TDB **do**
  for each item i_p in transaction t_j **do**
    add support s^{i_p} in the entry of i_p of H by 1
    calculate regularity r^{i_p} in the entry of i_p of H by t_j
for each item i_p in H **do**
  if $r^i > \sigma_r$ **then**
    remove the entry of i_p out of H
sort all items in H by ascending order of their weights
repeat
  calculate global maximum weight of all items in H, $GMAXW = max(w^{i_1}, w^{i_2}, \ldots, w^{i_{|H|}})$
  for each item i_p in H **do**
    calculate overestimated weighted-frequency of i_p, $owf^{i_p} = GMAXW \times s^{i_p}$
    if $owf^{i_p} < \sigma_{ws}$ **then**
      remove the entry of i_p out of H
until all items with weight equal to $GMAXW$ are not removed from H
for each item i_p in H **do**
  calculate weigthed-frequency $wf^{i_p} = w^{i_p} \times s^{i_p}$
  $WFRIs \leftarrow WFRIs \cup i_p$ **if** $wf^{i_p} \geq \sigma_{ws}$
create and initial *WFRI-tree* with a root node R
for each transaction t_j in TDB **do**
  remove item $i_p \in t_j$ such that $i_p \notin H$
  sort t_j as the order of H
  $temp \leftarrow R$
  for each item i_p in transaction t_j **do**
    if *temp* does not have a child node with i_p **then**
      create a new node Z for i_p, set Z as a child node of *temp*, and link Z with *node-link* of i_p in a header-table
      $temp \leftarrow Z$
    else
      $temp \leftarrow$ the child node of *temp* with i_p
  collect *tid* j in T^{i_p} of *temp* Z, $T^{i_p} \leftarrow T^{i_p} \cup j$

3.2.2.2 WFRIM-growth

สำหรับการค้นหาเซตรายการทั้งหมดที่เป็นผลลัพธ์ของเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการจากขั้นตอน WFRIM-growth จะทำการค้นหาเซตรายการขนาดต่าง ๆ แบบวนซ้ำบน *WFRI-tree* โดยเริ่มจากกำหนดให้เซตรายการ X เป็นเซตรายการที่พิจารณาก่อนหน้า โดยเริ่มแรกกำหนดให้ X เป็น $\emptyset$ (ขั้นตอนวิธีที่ 3.2 บรรทัดที่ 1) ต่อมาตรวจสอบ *WFRI-tree* ว่ามีเพียงเส้นทางเดียว (single path) หรือไม่ถ้าหากตรวจสอบแล้วพบว่า *WFRI-tree* มีเส้นทางของเซตรายการ P เพียงเส้นทางเดียว จะทำการคำนวนค่าน้ำหนักสนับสนุนของแต่ละซับเซตภายในเส้นทาง P เพื่อให้ได้ซึ่งเซตรายการที่เป็นผลลัพธ์ของเซตรายการที่ค้นหา (ขั้นตอนวิธีที่ 3.2 บรรทัดที่ 4-7) แต่ในกรณีที่ *WFRI-tree* มีหลายเส้นทาง ให้ทำการพิจารณาแต่ละรายการ i_p ในตารางรายการ H (เริ่มจากด้านล่างของตารางรายการ) แล้วสร้างตารางรายการใหม่ H^{i_p} สำหรับจัดเก็บข้อมูลของรายการที่ปรากฏร่วมกับรายการ i_p จากนั้นท่องไปยังโหนดพ่อแม่ของรายการ i_p ผ่านทางตารางรายการ H ที่เชื่อมโยงไปยังแต่ละโหนด n^{i_p} ของรายการ i_p ใน *WFRI-tree* เพื่อให้ได้ซึ่งข้อมูลของรายการที่ปรากฏร่วมกันกับรายการ i_p โดยที่ข้อมูลดังกล่าวใช้สำหรับคำนวนค่าความสม่ำเสมอและค่าสนับสนุน ในการลดจำนวนการพิจารณาเซตรายการที่เป็นผลลัพธ์สำหรับแต่ละรายการ i_q ในตารางรายการ H^{i_p} ถ้ารายการ i_q มีค่าความสม่ำเสมอ r^{i_q} มากกว่าค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนดรายการนั้นจะถูกลบออกจากตารางรายการ H^{i_p} (หมายเหตุ จากคุณสมบัติ downward closure property ถ้ารายการนั้นเป็นรายการที่ไม่ปรากฏบ่อยและสม่ำเสมอซูเปอร์เซตของรายการนั้นจะเป็นเซตรายการที่ไม่ปรากฏบ่อยและสม่ำเสมอไปด้วย) ในขั้นตอนวิธีที่ 3.2 บรรทัดที่ 20 จะทำการคำนวณหาค่าน้ำหนักที่มากสุดของเซตรายการที่พิจารณา ($LMAXW$) แล้วคำนวณค่าน้ำหนักสนับสนุนโดยการประมาณค่า ows จาก $LMAXW$ ร่วมกับค่าสนับสนุนของแต่ละรายการ $i_q \in H^{i_p}$ ($ows^{i_q} = LMAXW \times |T^{i_q}|$) ถ้าหากค่า ows^{i_q} (ที่ปรากฏร่วมกับเซตรายการ X และรายการ i_p) มีค่าน้อยกว่า σ_{ws} ให้ลบรายการ i_q นั้นออกจากตารางรายการ H^{i_p} (หมายเหตุ จากคุณสมบัติ downward closure property เซตรายการ $X \cup i_p \cup i_q$ และซุปเปอร์เซตของเซตรายการดังกล่าวจะไม่สามารถมีค่าน้ำหนักสนับสนุนที่มากกว่า ows^{i_q}) ในขั้นตอนการหาค่า $LMAXW$ และ ows^{i_q} ของแต่ละรายการ i_q จะมีการทำงานแบบวนซ้ำจนกระทั่งรายการทั้งหมดในตารางรายการ H^{i_p} ที่มีค่าน้ำหนักเท่ากับ $LMAXW$ ถูกลบออกจากตารางรายการ H^{i_p} จากนั้นพิจารณาแต่ละรายการ $i_q \in H^{i_p}$ สำหรับหาค่าน้ำหนักสนับสนุนโดนค่าน้ำหนักที่แท้จริง (ws) (ขั้นตอนวิธีที่ 3.2 บรรทัดที่ 26) ถ้าค่า ws^{i_q} มีค่าไม่น้อยกว่า σ_{ws} เซตรายการ $X \cup i_p \cup i_q$ จะเป็นผลลัพธ์ของเซตรายการที่ค้นหา

ขั้นตอนถัดมาสร้างโหนดราก (root node) สำหรับ *WFRI-tree* ใหม่ ซึ่ง *WFRI-tree* ใหม่นี้ใช้สำหรับจัดเก็บเซตรายการที่ปรากฏร่วมกับรายการ i_p สามารถเขียนระบุได้เป็น *WFRI-tree*^{i_p} โดยท่องไปยังโหนดพ่อแม่ของแต่ละโหนด n^{i_p} ใน *WFRI-tree* เพื่อตรวจสอบว่ารายการที่พบจากการท่องมีอยู่ในตารางรายการ H^{i_p} หรือไม่ ถ้าไม่มีอยู่ให้ทำการลบออกจากการพิจารณาแล้วนำรายการที่เหลืออยู่มาสร้าง *WFRI-tree*^{i_p} แล้วทำการจัดเก็บหมายเลขทรานแซกชันของรายการ i_p ไว้ที่โหนดสุดท้าย จากนั้นให้เลื่อนหมายเลขทรานแซกชันที่อยู่ในโหนดรายการ i_p ใน *WFRI-tree* ไปที่โหนดพ่อแม่แล้วลบโหนด n^{i_p} ออกจาก *WFRI-tree* (ขั้นตอนวิธีที่ 3.2 บรรทัดที่ 27-40) เมื่อได้รับ *WFRI-tree*^{i_p} แล้วขั้นตอนวิธี WFRIM-growth จะทำงานแบบวนซ้ำโดยจะทำการพิจารณาบน *WFRI-tree*^{i_p} ที่ซึ่งมีเซตรายการที่พิจารณาก่อนหน้าเป็น $X \cup i_p$ แล้วสร้างตารางรายการ H^{i_pq} และ *WFRI-tree*^{i_pq} เพื่อค้นหาเซตรายการที่เป็นผลลัพธ์ในขนาดที่ใหญ่ขึ้น ซึ่งกระบวนการทำงานแบบวนซ้ำนี้จะทำไปเรื่อย ๆ จนกระทั่ง *WFRI-tree*^{X} มีเพียงเส้นทางเดียว (single path) หรือพิจารณารายการในตารางรายการ H^{i_p} ครบทุกรายการ หลังจากจบการทำงานของขั้นตอนวิธี WFRIM-growth จะได้เซตที่เป็นผลลัพธ์ของรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ

Algorithm 3.2 WFRIM-growth

Input: *WFRI-Tree*,σ_r,σ_{ws}
Output: *WFRIs*

1: $X \leftarrow$ set of items considered from previous iterations (at beginning $X \leftarrow \emptyset$)
2: call *WFRI-growth* (*WFRI-tree*, $X = \emptyset$, σ_r, σ_{ws})
3: **Procedure** *WFRI-growth* (*WFRI-tree with* H, X, σ_r, σ_{ws})
4: **if** *WFRI-tree* contains only one path P **then**
5: **for** each sub-itemset Y of path P **do**
6: calculate weighted-frequency $wf^Y = \frac{\sum_{i_k \in Y} w^{i_k}}{|Y|} \times s^Y$ of Y
7: $WFRIs \leftarrow WFRIs \cup i_k$ **if** $wf^Y \geq \sigma_{ws}$
8: **else**
9: **for** each item i_p in the header-table H (in bottom-up manner) **do**
10: create a new header-table H^{i_p} to store all items in H except item i_p
11: **for** each node n^{i_p} linked with *node-link* of item i_p in the header-table H **do**
12: $Y \leftarrow$ the set of items in the same path with n^{i_p}
13: **for** each item $i_q \in Y$ **do**
14: update T^{i_q} of entry i_q of H^{i_p} by T^{i_p} of n_{i_p}, $T^{i_q} \leftarrow T^{i_q} \cup T^{i_p}$
15: **for** each item i_q in H^{i_p} **do**
16: compute r^{i_q} from T^{i_q} of entry i_q of H^{i_p}
17: remove the entry of i_q out of H^{i_p} **if** $r^{i_q} > \sigma_r$
18: **repeat**
19: calculate local maximum weighted as $LMAXW = \frac{\sum_{i_j \in X} w^{i_j} + w^{i_p} + max(i_y, \ldots, i_z)}{|X| + 2}$ where $i_y, \ldots, i_z \in H^{i_p}$
20: **for** each item i_q in H^{i_p} **do**
21: calculate overestimated weighted-frequency $owf^{i_q} = LMAXW \times s^{i_q}$
22: remove entry i_q out of H^{i_p} **if** $owf^{i_q} < \sigma_{ws}$
23: **until** all items with wight of importance equal to $LMAXW$ are removed from H^{i_p}
24: **for** each item i_q in H^{i_p} **do**
25: calculate weight-frequency $wf^{i_q} = \frac{\sum_{i_j \in X} w^{i_j} + w^{i_p} + w^{i_q}}{|X| + 2} \times s^{i_q}$
26: $WFRIs \leftarrow WFRIs \cup (X \cup i_p \cup i_q)$ **if** $wf^{i_q} \geq \sigma_{ws}$
27: create and initial a new *WFRI-tree* with Z as root
28: **for** each node n^{i_p} linked with *node-link* of item i_p in the header-table H **do**
29: $temp \leftarrow Z$
30: $Y \leftarrow$ the set of items in the same path with node n^{i_p} in which each item $i_q \in Y$ has an entry in H^{i_p}
31: **for** each item $i_q \in Y$ **do**
32: **if** *temp* does not have a child node with i_q **then**
33: create a new node X for i_q, set X to be a child node of *temp*, and link X with *node-link* of i_q of H^{i_p}
34: $temp \leftarrow X$
35: **else**
36: $temp \leftarrow$ the child node of X with item i_q
37: update T^{i_q} of *temp* by T^{i_p} of n^{i_p}, $T^{i_q} \leftarrow T^{i_q} \cup T^{i_p}$
38: $U \leftarrow$ the parent node of n^{i_p}
39: update T^U of U by T^{i_p} of n^{i_p}, $T^U \leftarrow T^U \cup T^{i_p}$
40: unlink n^{i_p} from *node-link* and remove node n^{i_p} out of *WFRI-tree*
41: call *WFRI-growth* (*WFRI-tree* with H^{i_p}, $X \cup i_p$, σ_r, σ_{ws})
42: remove entry of i_p out of H

3.3 ตัวอย่างสำหรับวิธีการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ

กำหนดให้ฐานข้อมูลทรานแซกชัน ตารางค่าน้ำหนัก แสดงดังภาพที่ 3-2 ค่าขีดแบ่งน้ำหนักสนับสนุน σ_{ws} เป็น 4.0 และค่าขีดแบ่งความสม่ำเสมอ σ_r เป็น 20

tid	item
1	a, b, c, d
2	c, e, f
3	a, b, e, f, g
4	a, b, c, f, g
5	d, e, g
6	a, b, c, e, g
7	a, b, c, e
8	a, b, d, e
9	b, c, e
10	a, e, g

ฐานข้อมูลทรานแซกชัน

item	weight
a	0.6
b	0.5
c	0.35
d	0.45
e	0.45
f	0.3
g	0.4

ตารางค่าน้ำหนัก

ภาพที่ 3-2 ฐานข้อมูลทรานแซกชันและตารางค่าน้ำหนัก

สำหรับการอ่านทรานแซกชันแรกในฐานข้อมูล $t_1 = \{ a, b, c, d \}$ ผลลัพธ์ของการคำนวณค่าสนับสนุนและค่าความสม่ำเสมอของแต่ละรายการ a, b, c และ d ในตารางรายการ H มีค่าเป็น 1 แสดงในภาพที่ 3-3

i	s^i	r^i	w^i
a	1	1	0.6
b	1	1	0.5
c	1	1	0.35
d	1	1	0.45
e	0	0	0.45
f	0	0	0.3
g	0	0	0.4

ภาพที่ 3-3 ตารางรายการ H หลังจากอ่านทรานแซกชันที่ 1

จากนั้นอ่านฐานข้อมูลในทรานแซกชันที่ 2 ซึ่ง t_2 = { c, e, f} ค่าสนับสนุน s^c ของรายการ c ในตารางรายการ H เพิ่มเป็น 2 และค่าความสม่ำเสมอ r^c มีค่าเป็น 1 สำหรับค่าสนับสนุน s^e, s^f และค่าความสม่ำเสมอ r^e, r^f ของรายการ e, f มีค่าเป็น 1 และ 2 เนื่องจากรายการดังกล่าวเพิ่งปรากฏขึ้นเป็นครั้งแรกในฐานข้อมูลที่ทรานแซกชัน t_2 ค่าความสม่ำเสมอที่มากสุดมีค่าเป็น 2 แสดงในภาพที่ 3-4 เมื่ออ่านฐานข้อมูลจนครบทุกทรานแซกชันแล้วรายการ f จะถูกลบออกจากตารางรายการ H และไม่ถูกพิจารณาว่าเป็นรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการเนื่องจากมีค่าความสม่ำเสมอมากกว่าค่าขีดแบ่งความสม่ำเสมอ σ_r= 4 จากนั้นเรียงลำดับแต่ละรายการในตารางรายการ H ด้วยค่าน้ำหนักจากน้อยไปมากจะได้ตารางรายการ H ดังภาพที่ 3-5

i	s^i	r^i	w^i
a	1	1	0.6
b	1	1	0.5
c	2	1	0.35
d	1	1	0.45
e	1	2	0.45
f	1	2	0.3
g	0	0	0.4

ภาพที่ 3-4 ตารางรายการ H หลังจากอ่านทรานแซกชันที่ 2

i	s^i	r^i	w^i
~~f~~	~~3~~	~~6~~	~~0.3~~
c	6	2	0.35
g	5	4	0.4
d	3	4	0.45
e	8	2	0.45
b	7	2	0.5
a	7	2	0.6

ภาพที่ 3-5 ตารางรายการ H หลังจากอ่านครบทุกทรานแซกชันและลบรายการที่มีค่าความสม่ำเสมอมากกว่าค่าขีดแบ่งความสม่ำเสมอ σ_r

ขั้นตอนต่อไปจากตารางรายการ H จะเป็นการคำนวณค่าน้ำหนักที่มากสุด $GMAXW$ คำนวณได้จาก max(0.35, 0.4, 0.45, 0.45, 0.5, 0.6) ดังนั้น $GMAXW$ มีค่าเป็น 0.6 ค่าน้ำหนัก

สนับสนุนโดยประมาณ ows ของแต่ละรายการในตารางรายการ H ได้แก่ ows^c= 3.6 (0.6×6), ows^g= 3.0 (0.6×5), ows^d= 1.8 (0.6×3), ows^e= 4.8 (0.6×8), ows^b= 4.2 (0.6×7), และ ows^a= 4.2 (0.6×7) ตามลำดับ ดังนั้นรายการ d จะถูกลบออกจากตารางรายการ H และไม่ถูกพิจารณาว่าเป็นรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการเนื่องจากค่าน้ำหนักสนับสนุนโดยประมาณ (ows) น้อยกว่าค่าขีดแบ่งน้ำหนักสนับสนุน (σ_{ws} = 2.0) ดังภาพที่ 3-6 และแสดงตารางรายการ H ในภาพที่ 3-7 ในขั้นตอนนี้จะได้เซตรายการที่เป็นผลลัพธ์ขนาดที่ 1 เป็น c, g, e, b และ a เนื่องจากแต่ละรายการมี $ws \geq \sigma_{ws}$ ซึ่งค่าน้ำหนักสนับสนุนที่แท้จริง ws ของแต่ละรายการคำนวณได้ดังนี้ ws^c=2.1 (0.35×6), ws^g=2 (0.4×5), ws^e=3.6 (0.45×8), ws^b=3.5 (0.5×7), ws^a=4.2 (0.6×7)

i	s^i	r^i	w^i
c	6	2	0.35
g	5	4	0.4
~~d~~	~~3~~	~~4~~	~~0.45~~
e	8	2	0.45
b	7	2	0.5
a	7	2	0.6

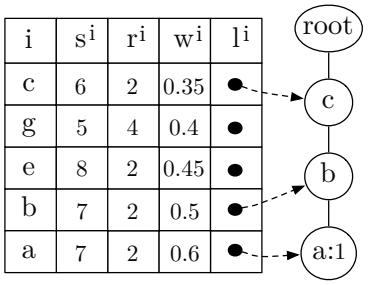
ภาพที่ 3-6 ตารางรายการ H หลังจากลบรายการที่มีค่าน้ำหนักสนับสนุนโดยการประมาณ ows น้อยกว่าค่าขีดแบ่งน้ำหนักสนับสนุน

i	s^i	r^i	w^i
c	6	2	0.35
g	5	4	0.4
e	8	2	0.45
b	7	2	0.5
a	7	2	0.6

ภาพที่ 3-7 ตารางรายการ H

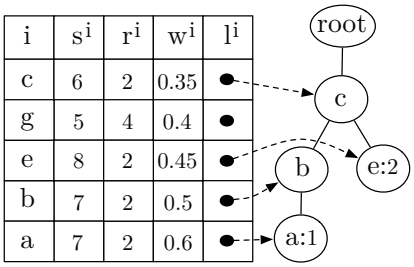
สำหรับการอ่านฐานข้อมูลทรานแซกชันครั้งที่สองเพื่อสร้าง *WFRI-Tree* ในการอ่านทรานแซกชัน t_1= { a, b, c, d } รายการ d ถูกตัดออกจากทรานแซกชัน t_1 เนื่องจากรายการ d ไม่ถูก

จัดเก็บอยู่ในตารางรายการ H แล้วเรียงลำดับรายการที่เหลือเป็น c, b, a ตามลำดับ ดังนั้นเส้นทางของ c, b, a จึงถูกสร้างใน *WFRI-Tree* โดยเก็บหมายเลขทรานแซกชัน 1 ไว้ที่โหนด a แล้วมีการเก็บการเชื่อมโยงของโหนดภายใน *WFRI-Tree* ไว้ในตารางรายการ H ตรงตามชื่อรายการ ดังภาพที่ 3-8

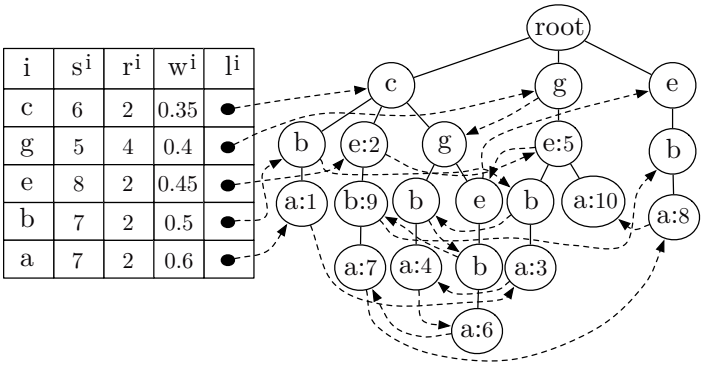


ภาพที่ 3-8 *WFRI-Tree* จากการอ่านทรานแซกชัน t_1

สำหรับการอ่านทรานแซกชันที่สอง $t_2 = \{c, e, f\}$ รายการ f ถูกตัดออกและ t_2 จะถูกเรียงลำดับเป็น c, e จากนั้นทำการสร้างเส้นทางของ c, e ใน *WFRI-Tree* และจัดเก็บหมายเลขทรานแซกชัน 2 ไว้ที่โหนด e ดังภาพที่ 3-9 ทรานแซกชันที่สาม t_3 จนถึงทรานแซกชันที่สิบ t_{10} ก็จะถูกอ่านเพื่อทำการสร้างเส้นทางของแต่ละทรานแซกชันเช่นเดียวกับข้างต้นโดยหลังจากอ่านฐานข้อมูลจนครบหมดแล้วจะได้ *WFRI-Tree* ดังภาพที่ 3-10

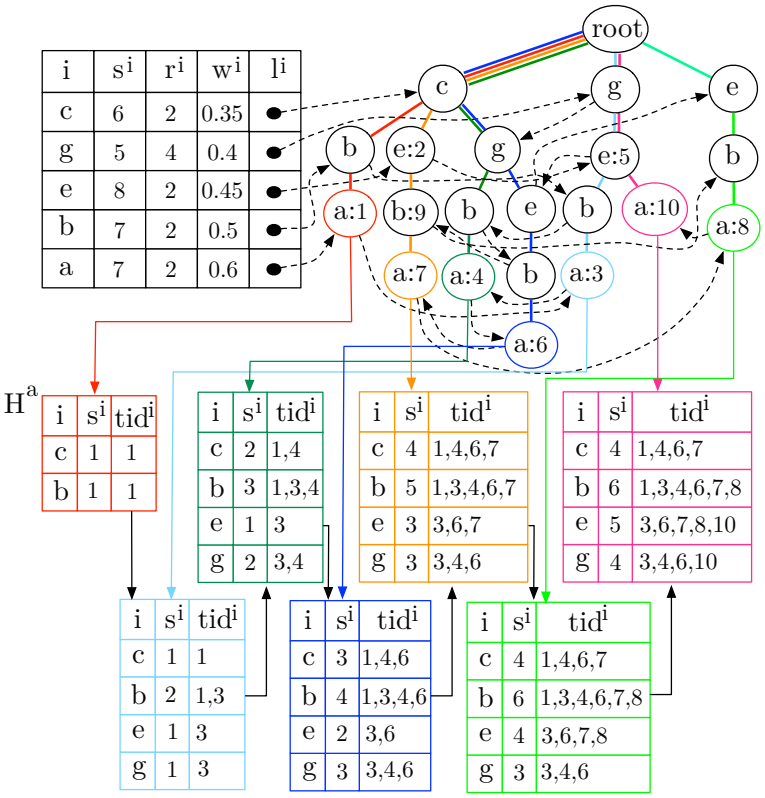


ภาพที่ 3-9 *WFRI-Tree* จากการอ่านทรานแซกชัน t_2



ภาพที่ 3-10 *WFRI-Tree* หลังจากอ่านฐานข้อมูลครบทุกทรานแซกชัน

ในขั้นตอนวิธี WFRIM-growth จะทำการค้นหาผลลัพธ์ของเซตรายการในขนาดต่าง ๆ บน *WFRI-Tree* โดยเริ่มแรกกำหนดให้เซต X เป็น $\emptyset$ จากนั้นตรวจสอบ *WFRI-Tree* ซึ่งพบว่า *WFRI-Tree* มีหลายเส้นทาง จึงได้มีการพิจารณารายการ a ในตารางรายการ H และสร้างตารางรายการใหม่สำหรับจัดเก็บข้อมูลของรายการที่ปรากฏร่วมกับรายการ a เขียนระบุเป็น H^a ซึ่งข้อมูลดังกล่าวได้รับจากการท่องไปยังโหนดพ่อแม่ของรายการ a ใน *WFRI-Tree* ผ่านทางแต่ละการเชื่อมโยง (link) ซึ่งจัดเก็บอยู่ที่ตารางรายการ H ตามลำดับ เริ่มจากการเชื่อมโยงแรกของรายการ a โดยข้อมูลที่ได้รับจากการท่องคือทรานแซกชันหมายเลข 1 ซึ่งรายการ a ปรากฏขึ้นร่วมกับรายการ c และรายการ b จึงอัพเดตค่าในตารางรายการ H^a สำหรับรายการ c และ b ดังนี้เซตของหมายเลขทรานแซกชัน (tid) เป็น T^c= {1}, T^b= {1} ค่าสนับสนุนของรายการ s^c=1 และรายการ b เป็น s^b=1 ต่อมาการเชื่อมโยงลำดับที่สองของรายการ a โดยข้อมูลที่ได้รับจากการท่องคือทรานแซกชันหมายเลข 3 ซึ่งรายการ a ปรากฏขึ้นร่วมกับรายการ g, e และ b แล้วทำการอัพเดตค่าในตารางรายการ H^a สำหรับรายการ g, e และ b โดยเซตของหมายเลขทรานแซกชัน (tid) เป็น T^g= {3} T^e= {3} T^b= {1, 3} ค่าสนับสนุนเป็น s^g=1 s^e=1 s^b=2 ทำการท่องไปทุกการเชื่อมโยงตามลำดับจนครบ ภาพที่ 3-11 แสดงตารางรายการ H^a หลังจากท่องจนครบทุกการเชื่อมโยงของรายการ a



i	s^i	r^i	w^i	l^i
c	6	2	0.35	●
g	5	4	0.4	●
e	8	2	0.45	●
b	7	2	0.5	●
a	7	2	0.6	●

i	s^i	tid^i
c	1	1
b	1	1

i	s^i	tid^i
c	2	1,4
b	3	1,3,4
e	1	3
g	2	3,4

i	s^i	tid^i
c	4	1,4,6,7
b	5	1,3,4,6,7
e	3	3,6,7
g	3	3,4,6

i	s^i	tid^i
c	4	1,4,6,7
b	6	1,3,4,6,7,8
e	5	3,6,7,8,10
g	4	3,4,6,10

i	s^i	tid^i
c	1	1
b	2	1,3
e	1	3
g	1	3

i	s^i	tid^i
c	3	1,4,6
b	4	1,3,4,6
e	2	3,6
g	3	3,4,6

i	s^i	tid^i
c	4	1,4,6,7
b	6	1,3,4,6,7,8
e	4	3,6,7,8
g	3	3,4,6

ภาพที่ 3-11 แสดงตารางรายการ H^a หลังจากที่ท่องไปยังโหนดพ่อแม่แต่ละการเชื่อมโยงของรายการ a จนครบ

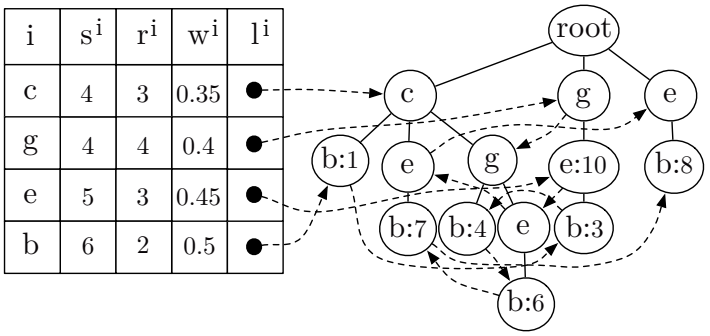
หลังจากที่ได้ตารางรายการ H^a จะทำการคำนวณค่าความสม่ำเสมอและประมาณค่าน้ำหนักสนับสนุนจาก $LMAXW$ ซึ่ง $LMAXW$ คำนวณได้จากค่าน้ำหนักเฉลี่ยของเซตรายการที่พิจารณาก่อนหน้า ($\emptyset$) เซตรายการที่กำลังพิจารณา (รายการ a) กับ ค่าน้ำหนักมากที่สุดในตารางรายการ H^a (รายการ b) ดังนั้น $LMAXW$=0.55 ((0.6+0.5)/2) จากขั้นตอนนี้รายการใดในตารางรายการ H^a ที่มีค่าความสม่ำเสมอมากกว่าค่าขีดแบ่งความสม่ำเสมอและมีการประมาณค่าน้ำหนักสนับสนุนน้อยกว่าค่าขีดแบ่งน้ำหนักสนับสนุนจะถูกลบออกจากตารางรายการ H^a ภาพที่ 3-12 แสดงตารางรายการ H^a หลังจากลดทอนรายการที่ไม่เป็นผลลัพธ์ของการค้นหาเซตรายการ สำหรับแต่ละรายการในตารางรายการ H^a จะถูกระบุให้เป็นผลลัพธ์ของเซตรายการที่ค้นหาโดยจะมีการคำนวณค่าน้ำหนักสนับสนุนที่แท้จริง ws ของแต่ละรายการที่ปรากฏร่วมกันกับรายการ a ซึ่งรายการที่เป็นผลลัพธ์คือเซตรายการ ab, ae และ ag

i	s^i	r^i	w^i
c	4	3	0.35
g	4	4	0.4
e	5	3	0.45
b	6	2	0.5

ภาพที่ 3-12 แสดงตารางรายการ H^a

ในขั้นตอนของการสร้าง *WFRI-tree*^{a} (โครงสร้างที่ใช้ในการจัดเก็บข้อมูลของรายการที่ปรากฏขึ้นร่วมกับรายการ a) จะทำการท่องไปยังโหนดพ่อแม่ของรายการ a ใน *WFRI-Tree* ผ่านทางการเชื่อมโยงที่จัดเก็บในตารางรายการ H อีกครั้งเพื่อจัดเก็บข้อมูลแต่ละเส้นทาง (path) ที่ปรากฏร่วมกับรายการ a จากนั้นนำข้อมูลที่ได้มาตรวจสอบว่ารายการภายในเส้นทางมีอยู่ในตารางรายการ H^a หรือไม่ ถ้าไม่มีอยู่ให้ทำการลบรายการนั้นออกจากการพิจารณาแล้วนำเส้นทางนั้นมาสร้างเป็น *WFRI-tree*^{a} จากนั้นทำการจัดเก็บหมายเลขทรานแซกชันที่ถูกเก็บอยู่ในโหนด a ที่ *WFRI-Tree* ไว้ที่โหนดสุดท้ายของเส้นทางนั้นใน *WFRI-tree*^{a} แสดงดังภาพที่ 3-13 สำหรับแต่ละเซตหมายเลขทรานแซกชันที่จัดเก็บอยู่ในโหนด a ที่ *WFRI-Tree* ให้นำไปจัดเก็บไว้ที่โหนดพ่อแม่จากนั้นลบโหนด a ออกจาก *WFRI-Tree*

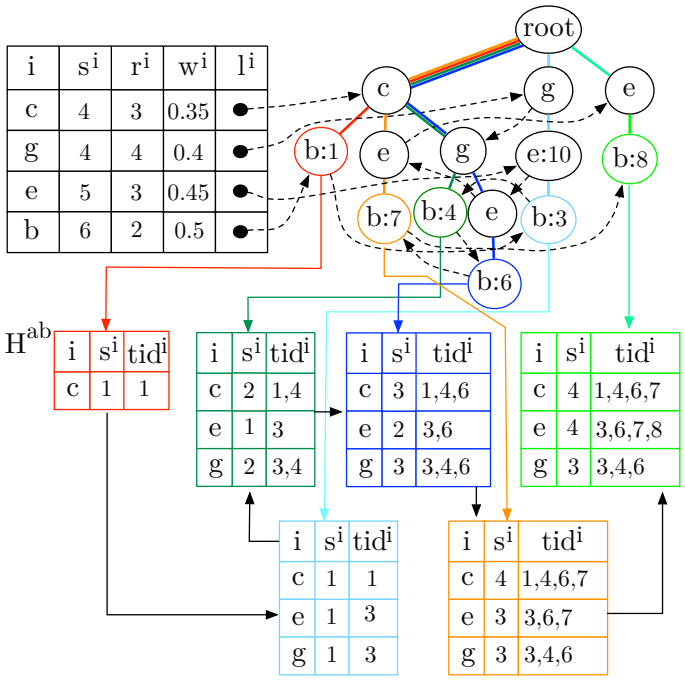
i	s^i	r^i	w^i	l^i
c	4	3	0.35	●
g	4	4	0.4	●
e	5	3	0.45	●
b	6	2	0.5	●



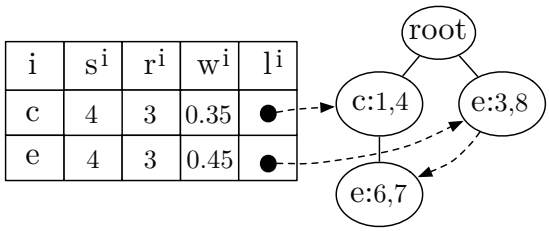
ภาพที่ 3-13 *WFRI-tree*^{a} ที่ซึ่งมีรายการ a เป็นรายการที่พิจารณาก่อนหน้า

จากนั้นขั้นตอนวิธี WFRIM-growth จะทำงานแบบวนซ้ำโดยจะทำการพิจารณาบน *WFRI-tree*^{a} ซึ่งมีเซตที่มีการพิจารณาแล้วก่อนหน้าเป็น $X = X \cup a$ แล้วสร้าง H^{ab} แสดงดังภาพที่ 3-14 และ *WFRI-tree*^{ab} แสดงดังภาพที่ 3-15 เพื่อค้นหาเซตรายการที่เป็นผลลัพธ์ในขนาดที่เพิ่มขึ้น

ซึ่งกระบวนการทำงานแบบวนซ้ำนี้จะทำไปเรื่อย ๆ จนกระทั่ง *WFRI-tree*^{X} มีเพียงเส้นทางเดียว (single path) หรือพิจารณารายการในตารางรายการ H ครบทุกรายการ



ภาพที่ 3-14 ตารางรายการ H^{ab} หลังจากที่ท่องไปยังโหนดพ่อแม่แต่ละที่อยู่ของรายการ b ใน *WFRI-tree*^{a} จนครบ



ภาพที่ 3-15 *WFRI-tree*^{ab} ที่ซึ่งมีรายการ ab เป็นรายการที่พิจารณาก่อนหน้า

จากตัวอย่างที่แสดงข้างต้น เมื่อทำการพิจารณารายการทั้งหมดจะได้ผลลัพธ์ของเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ ดังนี้ ผลลัพธ์รายการขนาดที่ 1 คือรายการ a (r^a=2, ws^a=4.2), รายการ b (r^b=2, ws^e=3.5), รายการ

e (r^e=2, ws^e=3.6) รายการ g (r^g=4, ws^g=2) และรายการ c (r^c=2, ws^c=2.1) ผลลัพธ์เซตรายการขนาดที่ 2 คือเซตรายการ ab (r^{ab}=2, ws^{ab}= 3.3), เซตรายการ ae (r^{ae}=3, ws^{ae}= 2.63), เซตรายการ ag (r^{ag}=4, ws^{ag}= 2) เซตรายการ be (r^{be}=3, ws^{be}= 2.13) และเซตรายการ bc (r^{bc}=3, ws^{bc}= 2.38) และผลลัพธ์เซตรายการขนาดที่ 3 คือเซตรายการ abc (r^{abc}=3, ws^{abc}= 2.06)

3.4 การวิเคราะห์ความซับซ้อนของขั้นตอนวิธี

ในส่วนนี้จะบรรยายถึงการวิเคราะห์ความซับซ้อนของขั้นตอนวิธี WFRIM ซึ่งจะพิจารณาอยู่ 2 ส่วนหลักได้แก่ 1) การวิเคราะห์ความซับซ้อนเชิงเวลา และ 2) การวิเคราะห์ความซับซ้อนเชิงหน่วยความจำ ดังนี้

3.4.1 การวิเคราะห์ความซับซ้อนเชิงเวลาของขั้นตอนวิธี WFRIM

สำหรับการค้นหาเซตรายการขนาด 1 เซตรายการของขั้นตอนวิธี WFRIM-Initialization จะทำการพิจารณาทุกรายการที่ปรากฏขึ้นในทรานแซกชันของฐานข้อมูล โดยกำหนดให้ n คือจำนวนรายการทั้งหมดในฐานข้อมูลและ m คือจำนวนทรานแซกชันทั้งหมดในฐานข้อมูล ดังนั้นความซับซ้อนเชิงเวลาของขั้นตอนวิธีนี้คือ $O(nm)$ จากนั้นทำการพิจารณาทุกรายการเพื่อทำการสร้าง *WFRI-tree* ซึ่งมีความซับซ้อนเชิงเวลาเป็น $O(n)$ สำหรับการค้นหาเซตรายการขนาดต่าง ๆ แบบวนซ้ำบน *WFRI-tree* ของขั้นตอนวิธี WFRIM-growth รายการที่ผ่านการพิจารณาจะถูกทำการรวมแต่ละรายการเพื่อสร้างเซตรายการที่มีขนาดใหญ่ขึ้น โดยความซับซ้อนเชิงเวลาของการค้นหารายการทั้งหมดที่คาดว่าจะเป็นผลลัพธ์ของเซตรายการคือ $O(2^p)$ ซึ่ง p คือรายการที่ผ่านการพิจารณาจากขั้นตอนวิธี WFRIM-Initialization และในแต่ละการพิจารณาเซตรายการนั้น ทรานแซกชันของเซตรายการที่ปรากฏร่วมกันจะถูกทำการเรียงลำดับเพื่อคำนวณค่าความสม่ำเสมอของเซตรายการนั้นซึ่งมีความซับซ้อนเชิงเวลาเป็น $O(m \log m)$ ดังนั้นความซับซ้อนเชิงเวลาของขั้นตอนวิธี WFRIM สามารถระบุได้เป็น $O((nm) + (n) + (2^p m \log m))$

3.4.2 การวิเคราะห์ความซับซ้อนเชิงหน่วยความจำของขั้นตอนวิธี WFRIM

ในขั้นตอนวิธี WFRIM-Initialization จะมีความซับซ้อนเชิงหน่วยความจำเป็น $O(nm)$ ที่ n คือจำนวนรายการทั้งหมดในฐานข้อมูลและ m คือจำนวนทรานแซกชันทั้งหมดในฐานข้อมูล และสำหรับขั้นตอนวิธี WFRIM-growth จะต้องทำการพิจารณาเซตรายการที่เป็นไปได้มากที่สุด p รายการ ซึ่ง p คือรายการที่ผ่านการพิจารณาจากขั้นตอนวิธี WFRIM-Initialization ซึ่งมีความซับซ้อนเชิงเวลาเป็น $O(2^{(p-1)}m)$ ซึ่ง m คือจำนวนทรานแซกชันที่ถูกจัดเก็บในแต่ละโหนดของเซตรายการ ดังนั้นความซับซ้อนเชิงหน่วยความจำของขั้นตอนวิธี WFRIM สามารถระบุได้เป็น $O((nm) + (2^{(p-1)}m))$

บทที่ 4
การค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการโดยใช้โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง

จากบทที่ 3 วิธีการในการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ จะทำการอ่านฐานข้อมูลสองครั้งเพื่อสร้างตารางรายการ (Header table) และโครงสร้างต้นไม้ *WFRI-tree* สำหรับจัดเก็บข้อมูลเพื่อใช้ในการค้นหาเซตรายการขนาดต่าง ๆ จากนั้นจะค้นหาเซตรายการที่เป็นผลลัพธ์จาก *WFRI-tree* โดยใช้แนวคิดพื้นฐานของ pattern growth อย่างไรก็ตาม ถึงแม้ว่าวิธีการที่นำเสนอดังกล่าวจะสามารถประมวลผลได้อย่างมีประสิทธิภาพแต่ยังมีส่วนที่พัฒนาประสิทธิภาพให้ดียิ่งขึ้นได้

ดังนั้น บทนี้จะนำเสนอขั้นตอนวิธีที่เพิ่มประสิทธิภาพในการประมวลผลสำหรับการค้นหาเซตรายการที่เป็นผลลัพธ์ของเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการจากฐานข้อมูลที่เรียกว่า การค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการโดยใช้โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง[5] (Efficient weighted-frequent-regular itemsets miner using interval word segments structure, WFRIM-IWS) ที่ทำการประยุกต์ใช้โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วงสำหรับจัดเก็บข้อมูลการปรากฏของรายการ/เซตรายการในขณะที่ทำการค้นหาเซตรายการทั้งหมดที่เป็นผลลัพธ์ โดยรายละเอียดของขั้นตอนที่นำเสนอสามารถแบ่งออกเป็นส่วนต่าง ๆ ดังต่อไปนี้

1 วิธีการสำหรับค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการโดยใช้โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง (WFRIM-IWS)

1.1 โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง (IWS)

1.2 ขั้นตอนวิธี WFRIM-IWS

2 ตัวอย่างสำหรับขั้นตอนวิธีค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการโดยใช้โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง

3 การวิเคราะห์ความซับซ้อนของขั้นตอนวิธี

---

[5] เวิร์ด (word) เป็นคำที่ใช้เรียกจำนวนบิตที่มากขึ้น ซึ่งในวิทยานิพนธ์นี้เวิร์ดมีจำนวน 16 บิต

4.1 วิธีการสำหรับค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการโดยใช้โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง

ในส่วนนี้จะนำเสนอขั้นตอนวิธีที่เรียกว่า วิธีการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการโดยใช้โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง (Weighted-Frequent-Regular Itemsets Miner using Interval Word Segment structure, WFRIM-IWS) โดยขั้นตอนวิธีดังกล่าวจะทำการอ่านฐานข้อมูลเพียงครั้งเดียวเพื่อค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้เงื่อนไขที่แต่ละรายการมีความสำคัญหรือความน่าสนใจแตกต่างกันจากฐานข้อมูล ขั้นตอนวิธี WFRIM-IWS มีการประยุกต์ใช้หลายแนวคิด ได้แก่ 1) ค่าน้ำหนักที่มากสุดหรือค่าน้ำหนักที่มากสุดของเซตรายการที่พิจารณา (global/local maximum weights) ในการประมาณค่าน้ำหนักสนับสนุน (overestimated weighted support) เพื่อลดทอนการพิจารณาเซตรายการที่ไม่เป็นผลลัพธ์ตามคุณสมบัติ downward closure property 2) โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง (Interval Word Segment, IWS) ในการจัดเก็บข้อมูลการปรากฏของแต่ละรายการ/เซตรายการ และ 3) ตารางค้นหา (Look-up table) เพื่อความรวดเร็วในการคำนวณค่าสนับสนุนและค่าความสม่ำเสมอจากโครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง

4.1.1 โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง (Interval word segment structure, IWS)

โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง (Interval word segment structure, IWS) (Nguyen, Vo, Nguyen, & Pedrycz, 2016) คือ โครงสร้างไดนามิกบิตเวกเตอร์ (dynamic bit-vector) ที่ใช้สำหรับจัดเก็บข้อมูลการปรากฏของรายการหรือเซตรายการหนึ่ง ๆ (กล่าวคือหมายเลขทรานแซกชันที่รายการหรือเซตรายการปรากฏ) สำหรับโครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง (IWS) ของรายการ/เซตรายการ X แสดงถึงลำดับของกลุ่มเวิร์ด (segment) สามารถเขียนระบุได้เป็น $IWS^X = \{sm_1^X, sm_2^X, \dots, sm_y^X\}$ ซึ่งแต่ละกลุ่มเวิร์ด (segment) $sm_j^X \in IWS^X$ ประกอบไปด้วยสองส่วน คือ 1) ตัวเลขที่อ้างถึงตำแหน่งแรกที่เวิร์ด (word) ไม่เป็นศูนย์ (ตำแหน่งแรก (Frist-index) สามารถเขียนระบุได้เป็น $fi^{sm_j^X}$) และ 2) ลำดับของเวิร์ด (word) ที่ไม่เป็นศูนย์ (หมายเหตุ แสดงถึงการปรากฏของรายการ/เซตรายการ X สามารถเขียนระบุได้เป็น $WO^{sm_j^X} = (wo_1^{sm_j^X}, wo_2^{sm_j^X}, \dots, wo_{|wo^{sm_j^X}|}^{sm_j^X})$) โดยที่แต่ละเวิร์ด (word) $wo_k^{sm_j^X} \in WO^{sm_j^X}$ จะบรรจุ 16 บิตโดยเริ่มจากบิตที่มีความสำคัญน้อยสุดไปจนถึงบิตที่มีความสำคัญมากสุด ซึ่งแต่ละบิต p^{th} จะมีค่าเป็น 0 ถ้าหากรายการ/เซตรายการ X ไม่มีการปรากฏในทรานแซกชัน t_p และ p^{th} จะมีค่าเป็น 1 ในกรณีที่รายการ/เซตรายการ X มีการปรากฏในทรานแซกชัน t_p ดังนั้น IWS^X สามารถเขียนแทนด้วย

$$IWS^X = \begin{Bmatrix} \langle fi^{sm_1^X}(wo_1^{sm_1^X}, wo_2^{sm_1^X}, \dots, wo_{|WO^{sm_1^X}|}^{sm_1^X})\rangle, \dots, \\ \langle fi^{sm_y^X}(wo_1^{sm_y^X}, wo_2^{sm_y^X}, \dots, wo_{|WO^{sm_y^X}|}^{sm_y^X})\rangle \end{Bmatrix}$$

4.1.1.1 ขั้นตอนวิธีในการจัดเก็บหมายเลขทรานแซกชันในรูปแบบโครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง

การปรากฏขึ้นของเซตรายการ X ในทรานแซกชัน t_p จะทำการจัดเก็บหมายเลขทรานแซกชัน p ในรูปแบบโครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง IWS^X ได้ตามตามขั้นตอนวิธีที่ 4.1 ดังนี้ ขั้นตอนแรกโดยการคำนวณตำแหน่งเวิร์ด (word index, wi_p กล่าวคือ ตำแหน่งเวิร์ดของหมายเลขทรานแซกชัน p ภายในโครงสร้าง IWS^X) และค่าเวิร์ด (word value, wo_p กล่าวคือค่าของหมายเลข ทรานแซกชัน p ที่ถูกบีบอัดให้อยู่ในรูปแบบเวิร์ด) จากหมายเลขทรานแซกชัน p (ขั้นตอนวิธีที่ 4.1บรรทัดที่ 1-2) ถ้าทรานแซกชัน t_p เป็นทรานแซกชันที่มี X ปรากฏขึ้นเป็นครั้งแรกจะสร้างกลุ่มเวิร์ด (segment) sm_1 แล้วทำการกำหนดให้ $fi^{sm_1^X}$ มีค่าเท่ากับ wi_p แล้วทำการเพิ่มค่าเวิร์ด wo_p เข้าไปในเซตของเวิร์ด ($wo_p \in WO^{sm_1^X}$) ของ sm_1^X (ขั้นตอนวิธีที่ 4.1 บรรทัดที่ 3-5) แต่ในกรณีที่ทรานแซกชัน t_p ไม่ใช่ทรานแซกชันที่ X ปรากฏขึ้นเป็นครั้งแรกจะพิจารณาทรานแซกชันสุดท้าย (last occurrence, lo^X) ที่ X ปรากฏเพื่อคำนวณหาตำแหน่งเวิร์ด ($wi_{lo}x$) สุดท้ายที่ X ปรากฏและจัดเก็บหมายเลขทรานแซกชัน p ใน IWS^X โดยสามารถดำเนินการได้ 3 กรณีดังต่อไปนี้

1) ถ้าตำแหน่งเวิร์ด $wi_{lo}x$ ของทรานแซกชันสุดท้ายที่ X ปรากฏ lo^X (ที่เป็นสมาชิกของ $WO^{sm_j^X}$ และ sm_j เป็นกลุ่มเวิร์ดกลุ่มสุดท้ายของ IWS^X) มีตำแหน่งเวิร์ดเหมือนกันกับตำแหน่งเวิร์ด wi_p ของหมายเลขทรานแซกชัน p ทำการอัพเดตค่าเวิร์ดที่ตำแหน่ง $wi_{lo}x$ ด้วย wo_p (ขั้นตอนวิธีที่ 4.1 บรรทัดที่ 8-1)

2) ถ้าตำแหน่งเวิร์ด wi_p ของหมายเลขทรานแซกชัน p มีตำแหน่งเวิร์ดที่อยู่ถัดไปจากตำแหน่งเวิร์ด $wi_{lo}x$ ของทรานแซกชันสุดท้ายที่ X ปรากฏ lo^X (ที่เป็นสมาชิกของ $WO^{sm_j^X}$ และ sm_j เป็นกลุ่มเวิร์ดกลุ่มสุดท้ายของ IWS^X) ทำการเพิ่มค่าเวิร์ด wo_p ต่อท้าย $wo_{lo}x$ (ขั้นตอนวิธีที่ 4.1 บรรทัดที่ 11-12)

3) ถ้าตำแหน่งเวิร์ด wi_p ของหมายเลขทรานแซกชัน p ไม่ใช่ตำแหน่งที่อยู่ถัดไปจากตำแหน่งเวิร์ด $wi_{lo}x$ ของทรานแซกชันสุดท้ายที่ X ปรากฏ lo^X (ที่เป็นสมาชิกของ $WO^{sm_j^X}$ และ sm_j เป็นกลุ่มเวิร์ดกลุ่มสุดท้ายของ IWS^X) จะสร้างกลุ่มเวิร์ดใหม่คือ sm_{j+1}^X (โดย ณ ปัจจุบัน IWS^X บรรจุไปด้วย $sm_1^X, \dots sm_j^X$) ทำการกำหนดให้ $fi^{sm_{j+1}^X}$ มีค่าเท่ากับ wi_p แล้วทำการเพิ่มค่า

เวิร์ด wo_p เข้าไปในเซตของเวิร์ด ($wo_p \in WO^{sm^X_{j+1}}$) ของ sm^X_{j+1} แล้วเพิ่มกลุ่มเวิร์ดดังกล่าวใน IWS^X (ขั้นตอนวิธีที่ 4.1 บรรทัดที่ 13-15)

Algorithm 4.1 *collect a tid into IWS*

Input: IWS^X, p, lo^X
Output: IWS^X

1: $wi_p \leftarrow \lfloor \frac{p-1}{wordsize} \rfloor + 1$
2: $wo_p \leftarrow 2^{((p-1) mod\ wordsize)}$
3: **if** $lo^X = 0$ **then**
4: create a new segment $sm \leftarrow (wi_p, \langle wo_p \rangle)$
5: $IWS^X \leftarrow IWS^X \cup sm$
6: **else**
7: $wi_{lo^X} \leftarrow \lfloor \frac{lo^X - 1}{wordsize} \rfloor + 1$
8: **if** $(wi_p - wi_{lo^X}) = 0$ **then**
9: $wo \leftarrow$ the last word of the last segment of IWS^X
10: $wo \leftarrow wo + wo_p$
11: **else if** $(wi_p - wi_{lo^X}) = 1$ **then**
12: add wo_p at the tail of the last segment of IWS^X
13: **else**
14: create a new segment $sm \leftarrow (wi_p, \langle wo_p \rangle)$
15: $IWS^X \leftarrow IWS^X \cup sm$

ตัวอย่าง 4.1 กำหนดให้ฐานข้อมูลทรานแซกชันประกอบไปด้วยจำนวน 80 ทรานแซกชัน แสดงดัง ภาพที่ 4-1

tid	item
1	a, c, e
...	...
15	b, c, d, f, g
16	a, b, e, f, g
...	...
32	a, b, c, e, f, g
...	...
35	b, d, g
...	...
49	a, b, c, e, g
50	a, b, c, e, g
...	...
55	a, b, d, e
...	...
70	a, b, c, d, e, g
71	a, b, d, e
...	...

A transactional database

ภาพที่ 4-1 ฐานข้อมูลทรานแซกชัน

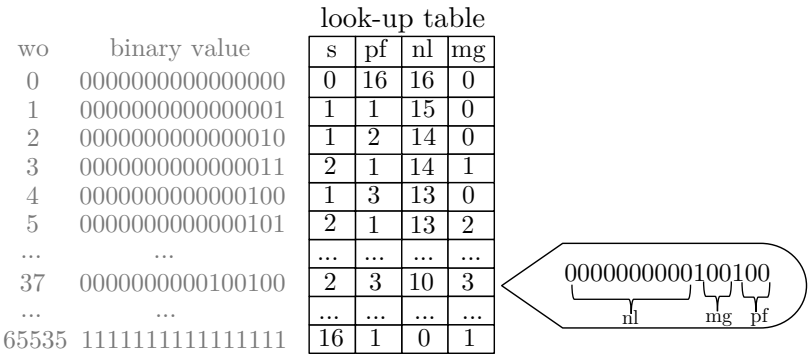
หลังจากพิจารณาฐานข้อมูลทรานแซกชันซึ่งแสดงในภาพที่ 4-1 ทำให้ทราบว่ารายการ a ปรากฏในทรานแซกชันดังต่อไปนี้ $t_1, t_{16}, t_{32}, t_{49}, t_{50}, t_{55}, t_{70}$ และ t_{71} โดยทรานแซกชันที่รายการ a ปรากฏขึ้นเป็นครั้งแรกคือ t_1 ดังนั้นตำแหน่งเวิร์ดและค่าเวิร์ดของหมายเลขทรานแซกชัน p เป็น 1 สามารถคำนวณได้โดย wi_1=$\left\lfloor\frac{1-1}{16}\right\rfloor$ +1=0+1=1 และ wo_1 = $2^{((1-1)\ mod\ 16)}$=2^0=1 จากนั้นสร้างกลุ่มเวิร์ดใหม่เป็น sm_1^a =⟨1(1)⟩ แล้วเพิ่มกลุ่มเวิร์ดดังกล่าวเข้าไปใน IWS^a=⟨1(1)⟩ ต่อมาจัดเก็บข้อมูลการปรากฏของรายการ a ซึ่งปรากฏที่ทรานแซกชัน t_{16} ตามกรณีที่ 1 ที่ซึ่งได้กล่าวไว้ข้างต้นจะเห็นได้ว่าตำแหน่งของเวิร์ดสำหรับ t_1 และ t_{16} มีค่าเป็น 1 เหมือนกันจึงทำการอัพเดตค่าเวิร์ดที่ตำแหน่งสุดท้ายของกลุ่มเวิร์ดสุดท้ายใน IWS^a ซึ่งจะอัพเดตโดย wo_{16}=32768 จากขั้นตอนนี้จะได้รับ IWS^a เป็น IWS^a=⟨1(32769)⟩ สำหรับการปรากฏของรายการ a ในทรานแซกชัน t_{32} ตำแหน่งเวิร์ดและค่าเวิร์ดของหมายเลขทรานแซกชัน p เป็น 32 คำนวณได้โดย wi_{32}=$\left\lfloor\frac{32-1}{16}\right\rfloor$ +1=1+1=2 และ wo_{32} = $2^{((32-1)\ mod\ 16)}$=2^{15}=32768 เมื่อทำการพิจารณาการปรากฏครั้งสุดท้ายของรายการ a (ปรากฏใน t_{16}) มีตำแหน่งเวิร์ดเป็น wi_{16}=1 จะเห็นได้ว่าตำแหน่งเวิร์ดของหมายเลขทรานแซกชัน p_{32} มีค่าเป็น 2 (wi_{32} =2) ซึ่งเป็นตำแหน่งเวิร์ดที่อยู่ถัดไปจากตำแหน่งเวิร์ดของการปรากฏครั้งสุดท้าย lo^a=16 (กรณีที่ 2 ที่ซึ่งได้กล่าวไว้ข้างต้น) จึงเพิ่มค่าของเวิร์ด wo_{32} =32768 ต่อท้ายที่กลุ่มเวิร์ดสุดท้ายใน IWS^a ดังนั้น IWS^a =⟨1(32769, 32768)⟩ ต่อมาการปรากฏของรายการ a ในทรานแซกชัน t_{49} ตำแหน่งเวิร์ดและค่าเวิร์ดของหมายเลขทรานแซกชัน p เป็น 49 สามารถคำนวณได้โดย $wi_{49} = \left\lfloor\frac{49-1}{16}\right\rfloor$ +1=3+1=4 และ wo_{32} = $2^{((49-1)\ mod\ 16)} = 2^0$=1 ทำการพิจารณาการปรากฏครั้งสุดท้ายของรายการ a (ปรากฏใน t_{32}) มีตำแหน่งเวิร์ดเป็น wi_{32} =2 จะเห็นได้ว่าตำแหน่งเวิร์ดของหมายเลขทรานแซกชัน p_{49} มีค่าเป็น 4 (wi_{49} =4) ซึ่งไม่ใช่ตำแหน่งที่อยู่ถัดไปจากตำแหน่งเวิร์ดในการปรากฏครั้งสุดท้ายของ lo^a=32 (กรณีที่ 3 ที่ซึ่งได้กล่าวไว้ข้างต้น) ดังนั้นจะสร้างกลุ่มเวิร์ดใหม่เป็น sm_2^a=⟨4(1)⟩ แล้วเพิ่มกลุ่มเวิร์ดดังกล่าวใน IWS^a เป็น IWS^a ={⟨1(32769, 32768)⟩, ⟨4(1)⟩} สำหรับการปรากฏของรายการ a ในทรานแซกชันอื่น ๆ ทำการพิจารณาเพื่อสร้าง IWS^a ตาม 3 กรณีที่กล่าวข้างต้นในขั้นตอนวิธีที่ 4.1 หลังจากพิจารณาจนครบแล้ว IWS^a={⟨1(32769, 32768)⟩, ⟨4(67, 96)⟩} ซึ่งประกอบไปด้วยสองกลุ่มเวิร์ดแสดงในภาพที่ 4-2

item	set of transaction containing item a				
a	1, 16, 32, 49, 50, 55, 70, 71				
bit value	1000000000000001	1000000000000000	0000000000000000	0000000010000011	0000000011000000
tid	16-1	32-17	48-33	64-49	80-65
word index	1	2	3	4	5
word value	32769 (32768+1)	32768	0	67 (64+2+1)	96 (64+32)
IWS	{<1(32769, 32768)>, <4(67, 96)>}				

ภาพที่ 4-2 ตัวอย่างการจัดเก็บหมายเลขทรานแซกชันเป็นโครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง (IWS)

4.1.1.2 การคำนวณค่าสนับสนุนและค่าความสม่ำเสมอในโครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง

การใช้โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง (IWS) ในการจัดเก็บข้อมูลการปรากฏของรายการ/เซตรายการ นั้นจะมีการบีบอัดหมายเลขทรานแซกชันให้เป็นเวิร์ดแล้วมีการแบ่งกลุ่มเวิร์ดออกเป็นช่วง ส่งผลให้ไม่สามารถคำนวณค่าสนับสนุนและค่าความสม่ำเสมอของรายการ/เซตรายการตามนิยามที่ได้กล่าวไว้ในบทที่ 3 ดังนั้นจึงได้สร้างตารางค้นหา (Look-up table) (Amphawan & Lenca, 2015) แสดงดังภาพที่ 4-3 เพื่อใช้สำหรับคำนวณค่าสนับสนุนและค่าความสม่ำเสมอของเซตรายการจากโครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง (IWS) ซึ่งตารางค้นหาดังกล่าวบรรจุ 2^{16} แถว (หมายเหตุ 16 คือขนาดของแต่ละเวิร์ดที่ใช้ในการจัดเก็บข้อมูลการปรากฏของแต่ละเซตรายการ) แต่ละแถวของตารางค้นหาประกอบไปด้วย 4 ข้อมูลดังนี้ 1) s คือค่าสนับสนุน (จำนวนครั้งของการปรากฏ) ของเวิร์ดที่พิจารณา 2) pf คือตำแหน่งที่บิต 1 ปรากฏขึ้นเป็นครั้งแรกในเวิร์ดที่พิจารณาโดยเริ่มต้นจากบิตที่มีความสำคัญน้อยสุด (แสดงถึงระยะห่างที่ X ไม่มีการปรากฏจนถึง X ปรากฏขึ้นเป็นครั้งแรกในเวิร์ดที่พิจารณา) 3) nl คือจำนวนของบิต 0 ที่ต่อเนื่องกันหลังจากบิต 1 ปรากฏครั้งสุดท้ายจนถึงบิตที่มีความสำคัญมากสุดในเวิร์ดที่พิจารณา (แสดงถึงกลุ่มของทรานแซกชันที่ X ไม่ปรากฏขึ้นในฐานข้อมูลหลังจากที่ X มีการปรากฏขึ้นครั้งสุดท้ายในเวิร์ดที่พิจารณาจนกระทั่งจบเวิร์ด) และ 4) mg คือจำนวนที่มากสุดของบิต 0 ที่อยู่ระหว่างบิต 1 สองบิต (แสดงถึง กลุ่มของทรานแซกชันที่มากสุดระหว่างสองทรานแซกชันที่ X ไม่ปรากฏขึ้นในเวิร์ดที่พิจารณา)



ภาพที่ 4-3 ตัวอย่างตารางค้นหา

นิยามที่ 4.1 (ค่าสนับสนุนของเซตรายการ X ใน IWS^X) ค่าสนับสนุนของเซตรายการ X ใน IWS^X คือผลรวมค่าสนับสนุนของแต่ละเวิร์ดในแต่ละกลุ่มเวิร์ดจาก IWS^X ซึ่งค่าสนับสนุนของแต่ละเวิร์ดสามารถได้รับจากข้อมูล s ของเวิร์ดนั้น ๆ ในตารางค้นหาโดยคำนวณได้ดังสมการ 4.4

$$s^X = \sum_{j=1}^{|IWS^X|} \sum_{k=1}^{|sm_j^X|} s^{wo_k^{sm_j^X}} \tag{4.4}$$

นิยามที่ 4.2 (ค่าความสม่ำเสมอของเซตรายการ X ใน IWS^X) ค่าความสม่ำเสมอของเซตรายการ X คือ ระยะห่างระหว่างทรานแซกชันที่เซตรายการ X ปรากฏขึ้น สำหรับค่าความสม่ำเสมอของเซตรายการ X ใน IWS^X จะพิจารณาค่าความสม่ำเสมอจากแต่ละเวิร์ด $wo^{sm_j^X}$ ภายในกลุ่มเวิร์ด sm_j^X ซึ่งค่าความสม่ำเสมอ $r^{wo^{sm_j^X}}$ สามารถคำนวณได้ 4 กรณีดังนี้

1) ถ้า $wo^{sm_j^X}$ คือ $wo_1^{sm_1^X}$ (กล่าวคือ $wo_1^{sm_1^X}$ คือเวิร์ดแรกของกลุ่มเวิร์ดแรก sm_1^X ใน IWS^X) ค่าความสม่ำเสมอ $r^{wo^{sm_j^X}}$ คำนวณได้ดังสมการ 4.5

$$r^{wo^{sm_j^X}} = max\left((((fi^{sm_1^X} - 1) \times 16) + pf(wo^{sm_j^X})), mg(wo^{sm_j^X})\right) \tag{4.5}$$

เมื่อ 1) $fi^{sm_1^X}$ คือตำแหน่งเวิร์ดเริ่มต้นของกลุ่มเวิร์ดแรก sm_1^X 2) $pf(wo^{sm_j^X})$ คือจำนวนของบิต 0 ที่ต่อเนื่องกันเริ่มต้นจากบิตที่มีความสำคัญน้อยสุดจนถึงบิตเป็น 1 ของเวิร์ดที่

พิจารณา (ได้รับจากตารางค้นหา) และ 3) $mg(wo^{sm_j^X})$ จำนวนที่มากสุดของบิต 0 ที่อยู่ระหว่างบิต 1 สองบิตของเวิร์ดที่พิจารณา (ได้รับจากตารางค้นหา)

2) ถ้า $wo^{sm_j^X}$ คือ $wo_1^{sm_j^X}$ (กล่าวคือ $wo_1^{sm_j^X}$ คือเวิร์ดแรกของกลุ่มเวิร์ด sm_j^X ซึ่งกลุ่มเวิร์ด sm_j^X มีลำดับที่ j^{th} ใน IWS^X เมื่อ $j \in [2, |IWS^X|]$ ค่าความสม่ำเสมอ $r^{wo^{sm_j^X}}$ คำนวณได้ดังสมการ 4.6

$$r^{wo^{sm_j^X}} = max\left((nl(lwo^{sm_{j-1}^X}) + ((fi^{sm_j^X} - 1 - li^{sm_{j-1}^X}) \times 16 + pf(wo^{sm_j^X})), mg(wo^{sm_j^X})\right) \quad (4.6)$$

เมื่อ 1) $nl(lwo^{sm_{j-1}^X})$ คือจำนวนบิต 0 หลังจากที่บิต 1 ปรากฏครั้งสุดท้ายจนถึงบิตที่มีความสำคัญมากสุดในเวิร์ดสุดท้าย $lwo^{sm_{j-1}^X}$ ของกลุ่มเวิร์ดลำดับก่อนหน้า $(j-1)^{th}$ (ได้รับจากตารางค้นหา) 2) $fi^{sm_j^X}$ คือตำแหน่งแรกของเวิร์ดในกลุ่มเวิร์ด j^{th} ซึ่งเป็นกลุ่มเวิร์ดที่กำลังพิจารณา 3) $li^{sm_{j-1}^X}$ คือตำแหน่งสุดท้ายของเวิร์ดในกลุ่มเวิร์ดลำดับก่อนหน้า $(j-1)^{th}$ 4) $pf(wo^{sm_j^X})$ คือจำนวนของบิต 0 ที่ต่อเนื่องกันเริ่มต้นจากบิตที่มีความสำคัญน้อยสุดจนถึงบิตเป็น 1 ของเวิร์ดที่พิจารณา (ได้รับจากตารางค้นหา) และ 5) $mg(wo^{sm_j^X})$ จำนวนมากที่สุดของบิต 0 ที่อยู่ระหว่างบิต 1 สองบิตของเวิร์ดที่พิจารณา (ได้รับจากตารางค้นหา)

3) ถ้า $wo^{sm_j^X}$ คือ $wo_k^{sm_j^X}$ (กล่าวคือ $wo_k^{sm_j^X}$ คือเวิร์ดลำดับที่ k^{th} (ไม่ใช่เวิร์ดแรก) ในกลุ่มเวิร์ดลำดับที่ j^{th}) ค่าความสม่ำเสมอ $r^{wo^{sm_j^X}}$ คำนวณได้ดังสมการ 4.7

$$r^{wo^{sm_j^X}} = max\left(((nl(wo_{k-1}^{sm_j^X}) + pf(wo^{sm_j^X})), mg(wo^{sm_j^X})\right) \quad (4.7)$$

เมื่อ 1) $nl(wo_{k-1}^{sm_j^X})$ คือจำนวนบิต 0 หลังจากที่บิต 1 ปรากฏครั้งสุดท้ายจนถึงบิตที่มีความสำคัญมากสุดในเวิร์ดลำดับก่อนหน้า $(k-1)^{th}$ ในกลุ่มเวิร์ด j^{th} ซึ่งเป็นกลุ่มเวิร์ดที่กำลังพิจารณา (ได้รับจากตารางค้นหา) 2) $pf(wo^{sm_j^X})$ คือจำนวนของบิต 0 ที่ต่อเนื่องกันเริ่มต้นจากบิตที่มีความสำคัญน้อยสุดจนถึงบิตเป็น 1 ของเวิร์ดที่พิจารณา (ได้รับจากตารางค้นหา) และ 3) $mg(wo^{sm_j^X})$ จำนวนมากที่สุดของบิต 0 ที่อยู่ระหว่างบิต 1 สองบิตของเวิร์ดที่พิจารณา (ได้รับจากตารางค้นหา)

4) ถ้า $wo^{sm_j^X}$ คือ $wo^{sm^X_{|IWS^X|}}_{|wo^{sm^X_{|IWS^X|}}|}$ (กล่าวคือ $wo^{sm^X_{|IWS^X|}}_{|wo^{sm^X_{|IWS^X|}}|}$ คือเวิร์ดสุดท้ายที่อยู่ในกลุ่มเวิร์ดลำดับสุดท้าย $sm^X_{|IWS^X|}$ ของ IWS^X) ค่าความสม่ำเสมอ $r^{wo^{sm_j^X}}$ คำนวณได้ดังสมการ 4.8

$$r^{wo^{sm_j^X}} = ((nl(wo^{sm_j^X}) + ((lwo^{|TDB|} - li^{sm^X_{|IWS^X|}}) \times 16)) \quad (4.8)$$

เมื่อ 1) $nl(wo^{sm_j^X})$ คือจำนวนบิต 0 หลังจากที่บิต 1 ปรากฏครั้งสุดท้ายจนถึงบิตที่มีความสำคัญมากสุดของเวิร์ดที่พิจารณา (ได้รับจากตารางค้นหา) 2) $lwo^{|TDB|}$ คือตำแหน่งของเวิร์ดสุดท้ายที่ซึ่งเวิร์ดดังกล่าวจัดเก็บข้อมูลการปรากฏ 16 ทรานแซกชันสุดท้ายในฐานข้อมูล TDB (คำนวณได้จาก $\left\lceil \frac{|TDB|}{wordsize} \right\rceil$) และ 3) $li^{sm^X_{|IWS^X|}}$ คือตำแหน่งของเวิร์ดสุดท้ายในกลุ่มเวิร์ดลำดับสุดท้ายของ IWS^X

ตัวอย่าง 4.2 จากฐานข้อมูลทรานแซกชันแสดงในภาพที่ 4-1 โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วงของรายการ a เป็น IWS^a={⟨1(32769, 32768)⟩, ⟨4(67, 96)⟩} ค่าสนับสนุนของรายการ a ใน IWS^a คำนวณได้โดย s^a=(($s^{wo^{32769}}$+$s^{wo^{32768}}$)+($s^{wo^{67}}$+$s^{wo^{96}}$)) ซึ่งค่าสนับสนุนของแต่ละเวิร์ดได้รับจากตารางค้นหาในข้อมูล s แสดงดังภาพที่ 4-4 ดังนั้นค่าสนับสนุนของรายการ a เป็น ((2+1)+(3+2))=8

สำหรับการคำนวณค่าความสม่ำเสมอของรายการ a (r^a) ในโครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง IWS^a={⟨1(32769, 32768)⟩, ⟨4(67, 96)⟩} นั้นสามารถคำนวณจาก 1) เวิร์ดแรกในกลุ่มเวิร์ดแรก (กรณีที่ 1) 2) เวิร์ดที่สองในกลุ่มเวิร์ดแรก (กรณีที่ 3) 3) เวิร์ดแรกในกลุ่มเวิร์ดที่สอง (กรณีที่ 2) และ 4) เวิร์ดที่สองในกลุ่มเวิร์ดที่สองโดยเวิร์ดที่สองนั้นจะมีการคำนวณในกรณีที่ 3, 4 เขียนระบุได้โดย r^a= max ($r^{wo_1^{sm_1^a}}$, $r^{wo_2^{sm_1^a}}$, $r^{wo_1^{sm_2^a}}$, $r^{wo_2^{sm_2^a}}$ (กรณีที่ 3) , $r^{wo_2^{sm_2^a}}$ (กรณีที่ 4)) ซึ่งค่าความสม่ำเสมอของแต่ละเวิร์ดคำนวณได้ดังนี้

1) $r^{wo_1^{sm_1^a}}$ = max ((((($fi^{sm_1^a}$ − 1) × 16) + $pf(wo^{sm_1^a})$), $mg(wo^{sm_1^a})$)

ดังนั้น $r^{wo_1^{sm_1^a}}$ = max (((1-1) × 16)+1), 15) =15

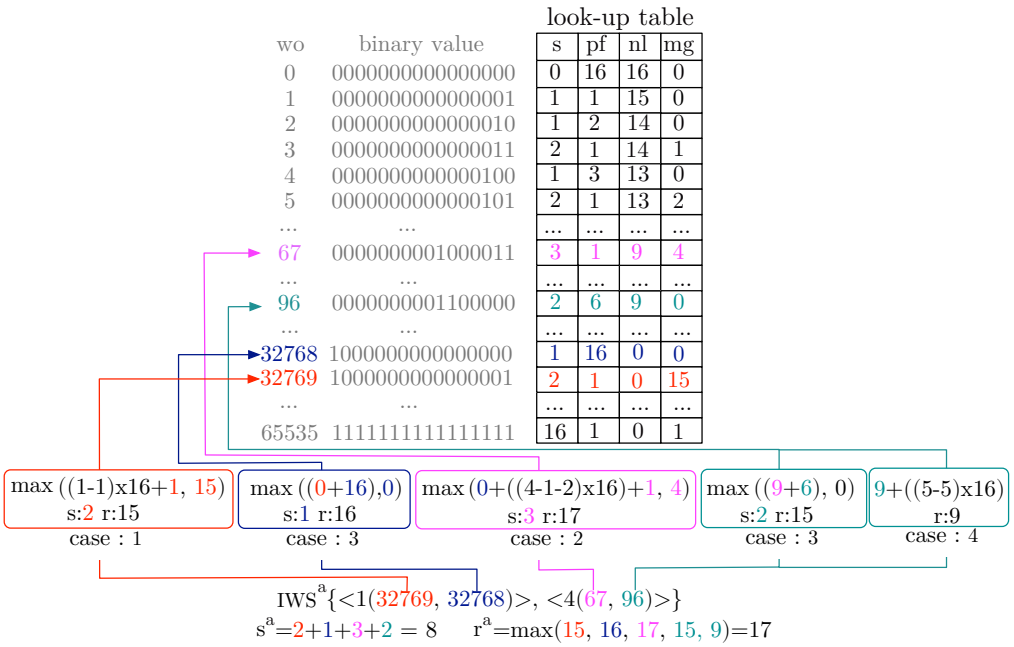
2) $r^{wo_2^{sm_1^a}}$ = max (((nl ($wo_1^{sm_1^a}$ + $pf(wo_2^{sm_1^a})$), mg ($wo_2^{sm_1^a}$)) ดังนั้น $r^{wo_2^{sm_1^a}}$ = max ((0+16), 0) =16

3) $r^{wo_1^{sm_2^a}}$ =$max((nl(lwo^{sm_1^a}) + ((fi^{sm_2^a} - 1 - li^{sm_1^a}) \times 16 + pf(wo_1^{sm_2^a})),$ $mg(wo_1^{sm_2^a}))$ ดังนั้น $r^{wo_1^{sm_2^a}} = max$ (((0+(4-1-2) ×16)+1), 4) =17

4) $r^{wo_2^{sm_2^a}} = max$ $(((nl\ (wo_1^{sm_2^a} + pf(wo_2^{sm_2^a})),\ mg(wo_2^{sm_2^a}))$ ดังนั้น $r^{wo_2^{sm_1^a}} = max$ ((9+6), 0) =15 (กรณีที่ 3)

5) $r^{wo_2^{sm_2^a}}$ =$((nl\ (wo_2^{sm_2^a}) + ((\ lwo^{|TDB|} - li^{sm_2^a}) \times 16))$ ดังนั้น $r^{wo_2^{sm_2^a}}$ = (9+(5-5) × 16) = 9 (กรณีที่ 4)

ดังนั้นค่าความสม่ำเสมอของเซตรายการ a คือ $r^a = max$ (15, 16, 17, 15, 9) = 17 ซึ่งภาพที่ 4-4 แสดงวิธีการคำนวณค่าสนับสนุนและค่าความสม่ำเสมอของรายการ a



ภาพที่ 4-4 ตัวอย่างในการคำนวณค่าสนับสนุนและค่าความสม่ำเสมอของรายการ a

4.1.1.3 การอินเตอร์เซกชัน (intersection) ในโครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง

ในการค้นหาเซตรายการขนาดต่าง ๆ นั้นจะสร้างเซตรายการที่มีขนาดใหญ่ขึ้นจากการรวมกันระหว่างสองเซตรายการโดยที่แต่ละเซตรายการจะต้องมีเซตรายการที่พิจารณาก่อนหน้าเป็นเซตรายการเดียวกัน (same prefix) (กล่าวคือ $Z = X \cup Y$ เมื่อ $X = \{i_j, \dots, i_k, i_p\}$ และ $Y = \{i_j, \dots, i_k, i_q\}$ ซึ่งทั้งสองเซตรายการมีเซตรายการที่พิจารณาก่อนหน้าเป็น $\{i_j, \dots, i_k\}$) ดังนั้นเพื่อที่จะให้ทราบถึงข้อมูลการปรากฏของเซตรายการ Z จะต้องทำการอินเตอร์เซกชันระหว่าง

IWS^X และ IWS^Y ซึ่งจะแสดงถึงทรานแซกชันที่ X และ Y ปรากฏร่วมกันแล้วจัดเก็บใน IWS^Z โดยรายละเอียดการอินเตอร์เซกชันแสดงในขั้นตอนวิธี 4.2 เริ่มจากพิจารณาแต่ละกลุ่มเวิร์ด sm_p^X ใน IWS^X และ sm_q^Y ใน IWS^Y ถ้ากลุ่มเวิร์ด sm_p^X และ sm_q^Y มีส่วนที่คาบเกี่ยวกัน (overlap) จะคำนวณหาตำแหน่งแรก (first index, fi) และตำหน่งสุดท้าย (last index, li) ของส่วนที่คาบเกี่ยวกันเพื่อให้ทราบขอบเขตที่จะทำการพิจารณา (ขั้นตอนวิธี 4.2 บรรทัดที่ 5-6) จากนั้นอินเตอร์เซกชันแต่ละเวิร์ด $wo_u^{sm_p^X}$ และ $wo_u^{sm_q^Y}$ ที่อยู่ในขอบเขต fi และ li แล้วจัดเก็บผลลัพธ์ที่ได้จากการอินเตอร์เซกชันใน wo โดยการจัดเก็บ wo จะพิจารณาเป็น 2 กรณีดังนี้

1) ถ้าค่าเวิร์ด wo เป็น 0 (กล่าวคือเซตรายการ X และ Y ไม่ได้ปรากฏร่วมกัน ขณะที่ตำแหน่งเวิร์ดคือ u^{th}) จะทำการตรวจสอบเซตของเวิร์ด WO ถ้าเซตของเวิร์ด WO เป็นเซตว่างจะเพิ่มค่า fi ขึ้น 1 ซึ่งแสดงว่าเซตรายการ Z อาจจะปรากฏเริ่มต้นที่เวิร์ด $(u+1)^{th}$ แต่ในกรณีที่เซตของเวิร์ด WO ไม่เป็นเซตว่างจะสร้างกลุ่มเวิร์ดใหม่ $sm = \langle fi(WO) \rangle$ แล้วเพิ่มกลุ่มเวิร์ดดังกล่าวต่อท้ายใน IWS^Z (ขั้นตอนวิธี 4.2 บรรทัดที่ 11-17)

2) ถ้าค่าเวิร์ด wo มีค่ามากกว่า 0 (กล่าวคือเซตรายการ X และ Y ปรากฏร่วมกัน ขณะที่ตำแหน่งเวิร์ดคือ u^{th}) ให้เพิ่มเวิร์ด wo ท่อท้ายที่เซตของเวิร์ด WO แล้วพิจารณาเวิร์ดถัดไป (เวิร์ดที่ u+1) (ขั้นตอนวิธีที่ 4.2 บรรทัดที่ 18-20)

ในการพิจารณาครั้งสุดท้ายของเวิร์ดในกลุ่มเวิร์ด sm_p^X และ sm_q^Y ถ้าเซตของเวิร์ด WO ไม่เป็นเซตว่างจะสร้างกลุ่มเวิร์ดใหม่ $sm = \langle fi(WO) \rangle$ แล้วเพิ่มกลุ่มเวิร์ดดังกล่าวต่อท้ายใน IWS^Z นอกจากนี้เมื่อไรก็ตามที่กลุ่มเวิร์ด sm_p^X และ sm_q^Y ไม่มีส่วนที่คาบเกี่ยวกันจะเพิ่มค่าตำแหน่ง (index) p ขึ้น 1 ถ้าหากตำแหน่งเวิร์ดสุดท้ายของกลุ่มเวิร์ด sm_p^X มีค่าน้อยกว่าตำแหน่งเวิร์ดสุดท้ายในกลุ่มเวิร์ด sm_q^Y แต่ในกรณีที่ตำแหน่งเวิร์ดสุดท้ายของกลุ่มเวิร์ด sm_p^X มีค่ามากกว่าตำแหน่งเวิร์ดสุดท้ายในกลุ่มเวิร์ด sm_q^Y จะเพิ่มค่าตำแหน่ง (index) q ขึ้น 1

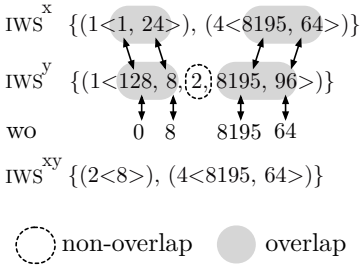
Algorithm 4.2 *IWS's intersection on a pair of IWS*

Input: IWS^X, IWS^Y
Output: IWS^Z

1: $IWS^Z \leftarrow \emptyset$
2: $p \leftarrow 1$ and $q \leftarrow 1$
3: **while** $p < |IWS^X|$ and $q < |IWS^Y|$ **do**
4: **if** sm_p^X and sm_q^Y are overlap **then**
5: $fi \leftarrow max(fi_{sm_p^X}, fi_{sm_q^Y})$
6: $li \leftarrow min(fi_{sm_p^X} + |WO_{sm_p^X}| - 1, fi_{sm_q^Y} + |WO_{sm_q^Y}| - 1)$
7: $u \leftarrow fi$
8: $WO \leftarrow \emptyset$
9: **while** $u < li$ **do**
10: $wo \leftarrow wo_u^{sm_p^X} \cap wo_u^{sm_q^Y}$
11: **if** $wo = 0$ **then**
12: **if** $WO = \emptyset$ **then**
13: $fi \leftarrow fi + 1$
14: **else**
15: $IWS^Z \leftarrow IWS^Z \cup \langle fi, WO \rangle$
16: $fi \leftarrow u + 1$
17: $WO \leftarrow \emptyset$
18: **else**
19: $WO \leftarrow WO \cup wo$
20: $u \leftarrow u + 1$
21: **if** $WO \neq \emptyset$ **then**
22: $IWS^Z \leftarrow IWS^Z \cup \langle fi, WO \rangle$
23: **else**
24: **if** $li_{sm_p^X} < li_{sm_q^Y}$ **then**
25: $p \leftarrow p + 1$
26: **else**
27: $q \leftarrow q + 1$

ตัวอย่าง 4.3 กำหนดให้โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วงของรายการ x เป็น IWS^x={⟨1(1, 24)⟩, ⟨4(8195, 64)⟩} และรายการ y เป็น IWS^y={⟨1(128, 8, 2, 8195, 96)⟩} เพื่อที่จะสร้างเซตรายการ "x, y" จะต้องอินเตอร์เซกชัน IWS^x และ IWS^y เพื่อคำนวณค่าน้ำหนักสนับสนุน ค่าความสม่ำเสมอ และจัดเก็บข้อมูลการปรากฏของเซตรายการ "x, y" โดยเริ่มจากพิจารณากลุ่มเวิร์ดแรก sm_1^x=⟨1(1, 24)⟩ และ sm_1^y=⟨1(128, 8, 2, 8195, 96)⟩ (หมายเหตุ กลุ่มเวิร์ดทั้งสองมีความคาบเกี่ยวกัน) ซึ่งขอบเขตที่จะพิจารณาเพื่ออินเตอร์เซกชันคำนวณจากตำแหน่งแรกของเวิร์ดและตำแหน่งสุดท้ายของเวิร์ดซึ่งคำนวณได้โดย $fi = max(fi_{sm_1^x}, fi_{sm_1^y})$ $= max(1,1) = 1$ และ $li = min(li_{sm_1^x}, li_{sm_1^y}) = min(fi_{sm_1^x} + |WO|^{sm_1^x} - 1, fi_{sm_1^y} + |WO|^{sm_1^y} - 1) = min(1+2-1, 1+5-1) = min(2,5) = 2$ จากนั้นอินเตอร์เซกชันแต่ละเวิร์ดใน sm_1^x และ sm_1^y ในระหว่าง $fi = 1$ และ $li = 2$ สำหรับเวิร์ดแรก $wo_1^{sm_1^x} = 1$ และ $wo_1^{sm_1^y} = 128$ ซึ่งอินเตอร์เซกชัน

ระหว่างเวิร์ดทั้งสองแล้วจะได้ wo =1 ∩ 128 =0 จึงเพิ่มค่า fi ขึ้น 1 (ในขณะนี้ fi =2) ต่อมาอินเตอร์เซกชันระหว่าง $wo_2^{sm_1^x}$=24 และ $wo_2^{sm_1^y}$=8 ได้ผลลัพธ์เป็น wo =24 ∩ 8 =8 จึงจัดเก็บเวิร์ด wo ดังกล่าวในเซตของเวิร์ด WO หลังจากพิจารณาแต่ละเวิร์ดในกลุ่มเวิร์ดภายใต้ขอบเขต fi และ li จนจบแล้ว จะสร้างกลุ่มเวิร์ดใหม่ sm=⟨2(8)⟩ และเพิ่มกลุ่มเวิร์ดดังกล่าวเข้าไปใน $IWS^{x,y}$ จากนั้นพิจารณากลุ่มเวิร์ดถัดมา sm_2^x และ sm_1^y ซึ่งกลุ่มเวิร์ดทั้งสองมีความคาบเกี่ยวกันเพื่อดำเนินการอินเตอร์เซกชันแต่ละเวิร์ดในกลุ่มเวิร์ดดังกล่าวตามกระบวนการเหมือนกับด้านบน ซึ่งหลังจากพิจารณาจนครบแต่ละกลุ่มเวิร์ดที่มีความคาบเกี่ยวกันจะได้ $IWS^{x,y}$ เป็น $IWS^{x,y}$={⟨2(8)⟩, ⟨4(8159, 64)⟩} แสดงดังภาพที่ 4-5



ภาพที่ 4-5 ตัวอย่างในการอินเตอร์เซกชันระหว่าง IWS^x และ IWS^y

4.1.2 ขั้นตอนวิธี WFRIM-IWS (WFRIM-IWS algorithm)

ขั้นตอนวิธี WFRIM-IWS (WFRIM-IWS algorithm) ประกอบไปด้วย 2 ขั้นตอน ได้แก่ 1) DBscanning จะอ่านฐานข้อมูลเพื่อสร้าง *WFRIM-tree* แล้วคำนวณค่าความสม่ำเสมอและค่าน้ำหนักสนับสนุนโดยประมาณของแต่ละรายการ ซึ่งสุดท้ายแล้ว *WFRIM-tree* จะประกอบไปด้วยรายการที่เป็นรายการแข่งขัน (candidate single item) และ 2) WFRI-mining จะค้นหาเซตรายการที่เป็นผลลัพธ์ทั้งหมดของเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ ซึ่งจะทำการค้นหาบน *WFRIM-tree* ที่ได้รับจากขั้นตอน DBscanning โดยจะบรรยายรายละเอียดดังต่อไปนี้

4.1.2.1 DBscanning

ในขั้นตอน DBscanning เริ่มแรก *WFRIM-tree* จะถูกสร้างขึ้นร่วมกับโหนดราก (root node) R และสร้างโหนดสำหรับแต่ละรายการ $i_j \in I$ โดยจะกำหนดให้แต่ละโหนด i_j ที่สร้างขึ้นเป็นโหนดลูกของโหนดราก R (ขั้นตอนวิธี 4.3 บรรทัดที่ 1-2) จากนั้นอ่านฐานข้อมูลแต่ละทรานแซกชัน $t_p \in DB$ และพิจารณาแต่ละรายการ $i_k \in t_p$ เพื่ออัพเดตข้อมูลในโหนด i_k (กล่าวคือทำ

การอัพเดตโดย 1) เพิ่มค่าสนับสนุน s^{i_k} ขึ้น 1 2) สร้าง IWS^{i_k} (รายละเอียดการจัดเก็บหมายเลขทรานแซกชันเป็นโครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง (IWS) บรรยายไว้ในขั้นตอนวิธี 4.1) 3) คำนวณค่าความสม่ำเสมอ r^{i_k} และ 4) อัพเดตการปรากฏขึ้นครั้งสุดท้าย lo^{i_k} โดย p ซึ่งเป็นหมายเลขทรานแซกชันของ t_p) (ขั้นตอนวิธี 4.3 บรรทัดที่ 3-8) หลังจากที่อ่านฐานข้อมูลครบทุกทรานแซกชันจะพิจารณาค่าความสม่ำเสมอของแต่ละรายการ i_k ถ้าค่าความสม่ำเสมอที่อัพเดตครั้งสุดท้ายของรายการ i_k ใดมีค่ามากกว่าค่าขีดแบ่งความสม่ำเสมอ (σ_r) รายการนั้นจะถูกลบออกจาก *WFRIM-tree* (ขั้นตอนวิธี 4.3 บรรทัดที่ 9-11) จากนั้นหาค่าน้ำหนักที่มากสุด ($GMAXW$) เพื่อใช้ในการคำนวณค่าน้ำหนักสนับสนุนโดยประมาณ (ows) ของแต่ละรายการ i_k ถ้าค่า ows^{i_k} มีค่าน้อยกว่าค่าขีดแบ่งน้ำหนักสนับสนุน (σ_{ws}) รายการ i_k นั้นจะถูกลบออกจาก *WFRIM-tree* แต่ในกรณีที่ค่า ows^{i_k} มีค่าไม่น้อยกว่าค่าขีดแบ่งน้ำหนักสนับสนุน (σ_{ws}) จะคำนวณค่าน้ำหนักสนับสนุนที่แท้จริงของแต่ละรายการ i_k (ws^{i_k}) ถ้ารายการ i_k ใดมีค่าน้ำหนักสนับสนุน ws^{i_k} ไม่น้อยกว่าค่าขีดแบ่งน้ำหนักสนับสนุน (σ_{ws}) รายการ i_k นั้นจะถูกระบุว่าเป็นผลลัพธ์ของเซตรายการที่ปรากฏและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ ซึ่งกระบวนการในการคำนวณค่าน้ำหนักที่มากสุด ($GMAXW$) และค่าน้ำหนักสนับสนุนโดยประมาณเพื่อลดทอนการพิจารณาเซตรายการจะมีการทำงานแบบวนซ้ำจนกระทั่งรายการทั้งหมดมีค่าน้ำหนักเท่ากับค่าน้ำหนักที่มากที่สุดถูกพิจารณา (กล่าวคือ ถ้ารายการทั้งหมดมีน้ำหนักเท่ากับค่าน้ำหนักที่มากสุดถูกตัดอกจากการพิจารณาจะทำการคำนวณค่า $GMAXW$ ใหม่) (ขั้นตอนวิธี 4.3 บรรทัดที่ 13-22) จากนั้นเรียงลำดับแต่โหนดใน *WFRIM-tree* โดยค่าน้ำหนักของแต่ละรายการจากมากไปน้อย หลังจากขั้นตอนสุดท้ายของขั้นตอนวิธี DBscanning จะได้รับ *WFRIM-tree* ซึ่งแต่ละโหนดใน *WFRIM-tree* เป็นรายการแข่งขัน (candidate single item) (กล่าวคือ รายการที่มีค่าน้ำหนักสนับสนุนโดยประมาณ $\geq \sigma_{ws}$)

Algorithm 4.3 *DBscanning*

Input: *TDB*, σ_r, σ_{ws}
Output: *WFRIM-Tree*, *WFRIs*

1: create a *WFRIM-tree* with root R.
2: create a node of item $i_j \in I$ and set to be a child of R
3: **for** each transaction t_p in TDB **do**
4: **for** each item i_k in transaction t_p **do**
5: ***collect***(IWS^{i_k}, p, lo^{i_k})
6: $s^{i_k} \leftarrow s^{i_k} + 1$
7: $r^{i_k} \leftarrow max(r^{i_k}, p - lo^{i_k})$
(if t_p is the first transaction containing i_k, $r^{i_k} \leftarrow p$)
8: $lo^{i_k} \leftarrow p$
9: **for** each node of item i_j in *WFRIM-tree* **do**
10: $r^{i_k} \leftarrow max(r^{i_k}, m - lo^{i_k})$
(where m is the tid of the last transaction of TDB)
11: **if** $r^{i_k} > \sigma_r$ **then**
12: remove node of i_j out of *WFRIM-tree*
13: **repeat**
14: $GMAXW \leftarrow max(w_{i_1}, w_{i_2}, \ldots, w_{i_{|I|}})$
15: **for** each node of item i_j in *WFRIM-tree* **do**
16: $ows^{i_k} \leftarrow GMAXW \times s^{i_k}$
17: **if** $ows^{i_k} < \sigma_{ws}$ **then**
18: remove node of i_j out of *WFRIM-tree*
19: **else**
20: $ws^{i_j} \leftarrow w_{i_j} \times s^{i_j}$
21: $WFRIs \leftarrow WFRIs \cup i_k$ **if** $ws^{i_k} \geq \sigma_{ws}$
22: **until** R does not have a child node with $w_{i_j} = GMAXW$
23: reorder child node of R by weight descending order

4.1.2.1 WFRI-mining

ในการค้นหาเซตรายการผลลัพธ์ขนาดที่ $k+1$ ขั้นตอนวิธีนี้จะมีการทำงานแบบวนซ้ำบน *WFRIM-tree* ที่ซึ่งได้รับจากขั้นตอนวิธี DBscanning (ขั้นตอนวิธีที่ 4.3) เริ่มต้นจะทำการพิจารณารายการที่ซึ่งเป็นโหนดลูกของโหนดราก R กำหนดเป็น H จากนั้นกำหนดให้ U เป็น $U = X \cdot i_p$ คือเซตรายการที่พิจารณาในปัจจุบันโดยที่ X เป็นเซตรายการที่พิจารณาก่อนหน้าและ $i_p \in H$ (ขั้นตอนวิธี 4.4 บรรทัดที่ 1-3) โดยขั้นตอนแรกจะทำการคำนวณหาค่าน้ำหนักที่มากสุดของเซตรายการที่พิจารณา ($LMAXW$) กำหนดให้ $Y = X \cdot i_q$ ที่ i_q คือเซตรายการที่ถัดไปจาก i_p และ $i_q \in H$ และกำหนดให้ $Z = i_p \cdot i_q$ คือเซตรายการผลลัพธ์ที่พิจารณา ดังนั้น $LMAXW$ สามารถคำนวณได้โดย $LMAXW = w^Z = \frac{(w^X \times |X|) + w_{i_p} + w_{i_q}}{|X|+2}$ (ขั้นตอนวิธี 4.4 บรรทัดที่ 5-7) ต่อมาทำการอินเตอร์เซกชันระหว่าง IWS^U และ IWS^Y เพื่อจัดเก็บข้อมูลการปรากฏของเซตรายการ Z ใน IWS^Z แล้วทำการคำนวณค่าสนับสนุนและค่าความสม่ำเสมอของเซตรายการ Z ที่ได้รับจากตารางการค้นหา (Lookup table) (ขั้นตอนวิธี 4.4 บรรทัดที่ 8-10) จากนั้นค่าน้ำหนักสนับสนุน ws^Z จะ

ถูกคำนวณโดย $ws^Z = w^Z \times s^Z$ ถ้าค่าความสม่ำเสมอ r^Z น้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอ (σ_r) และค่าน้ำหนักสนับสนุน ws^Z มีค่าที่มากกว่าหรือเท่ากับค่าขีดแบ่งน้ำหนักสนับสนุน (σ_{ws}) แล้วโหนดของเซตรายการ Z จะถูกสร้างขึ้นซึ่งจะทำการจัดเก็บ IWS^Z และเชื่อมโยงไปเป็นโหนดลูกของเซตรายการ U และเซตรายการ Z จะถูกจัดเก็บเป็นผลลัพธ์ของเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ อย่างไรก็ตามถ้าค่าความสม่ำเสมอ r^Z มีค่าที่มากกว่าค่าขีดแบ่งความสม่ำเสมอ (σ_r) หรือค่าน้ำหนักสนับสนุน ws^Z มีค่าที่น้อยกว่าค่าขีดแบ่งน้ำหนักสนับสนุน (σ_{ws}) แล้วรายการที่ถัดจากรายการ i_p จะถูกพิจารณาจนกว่ารายการนั้นจะมีค่าความสม่ำเสมอที่ไม่มากกว่าค่าขีดแบ่งความสม่ำเสมอ (σ_r) และค่าน้ำหนักสนับสนุนที่ไม่น้อยกว่าค่าขีดแบ่งน้ำหนักสนับสนุน (σ_{ws}) (ขั้นตอนวิธี 4.4 บรรทัดที่ 11-14)

หลังจากนั้นจะทำการพิจารณารายการถัดไปที่เหลืออยู่ใน H สามารถกำหนดได้เป็นเซตรายการ V ที่ $V = X \cdot i_q$ และเซตรายการ $Z = i_p \cdot i_q$ คือเซตรายการผลลัพธ์ที่พิจารณา (ขั้นตอนวิธี 4.4 บรรทัดที่ 15-16) ดังนั้น IWS^Z สามารถคำนวณได้โดยการอินเตอร์เซกชันระหว่าง IWS^U และ IWS^V แล้วคำนวณค่าสนับสนุนและค่าความสม่ำเสมอของเซตรายการ Z ที่ได้รับจากตารางการค้นหา (ขั้นตอนวิธี 4.4 บรรทัดที่ 17-19) จากนั้นทำการคำนวณหาค่าน้ำหนักสนับสนุนโดยประมาณ (ows) ของเซตรายการ Z ซึ่งสามารถคำนวณได้จาก $ows^Z = LMAXW \times s^Z$ (ขั้นตอนวิธี 4.4 บรรทัดที่ 20) ถ้าค่าความสม่ำเสมอ r^Z มีค่าที่น้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอ (σ_r) และค่าน้ำหนักสนับสนุนโดยประมาณ ows^Z มีค่าที่มากกว่าหรือเท่ากับค่าขีดแบ่งน้ำหนักสนับสนุน (σ_{ws}) แล้วทำการสร้างโหนดของเซตรายการ Z ซึ่งจัดเก็บ IWS^Z แล้วทำการเชื่อมโยงไปเป็นโหนดลูกของเซตรายการ U (ขั้นตอนวิธี 4.4 บรรทัดที่ 21-22) ต่อมาทำการพิจารณาผลลัพธ์ของเซตรายการโดยการคำนวณค่าน้ำหนักสนับสนุน ws^Z ซึ่งจะถูกคำนวณโดย $ws^Z = s^Z \times \frac{(w^X \times |X|) + w_{i_p} + w_{i_q}}{|X|+2}$ ถ้าค่าน้ำหนักสนับสนุน ws^Z มีค่าที่ไม่น้อยกว่าค่าขีดแบ่งน้ำหนักสนับสนุน (σ_{ws}) แล้วเซตรายการ Z จะถูกจัดเก็บเป็นผลลัพธ์ของเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ (ขั้นตอนวิธี 4.4 บรรทัดที่ 23-25) เมื่อทำการพิจารณารายการใน H ทั้งหมดแล้ว ถ้าโหนดลูกของเซตรายการ U มีจำนวนมากกว่า 1 โหนดให้ทำการพิจารณาแบบวนซ้ำในขั้นตอนวิธี WFRI-mining แต่ถ้าโหนดลูกของเซตรายการ U มีจำนวนที่น้อยกว่าหรือเท่ากับ 1 โหนดให้ทำการลบโหนดของเซตรายการ U และโหนดลูกทั้งหมดของเซตรายการ U ออกจากการพิจารณาใน *WFRIM-tree* (ขั้นตอนวิธี 4.4 บรรทัดที่ 26-29)

Algorithm 4.4 *WFRI:mine*

Input: *WFRIM-Tree* with root R, σ_r, σ_{ws}
Output: *WFRIs*

1: $X \leftarrow \emptyset$ and $w^X \leftarrow 0$
2: *mining* (node of R, X, w^X, σ_r, σ_{ws})

Procedure *mining* (node of H, X, w^X, σ_r, σ_{ws})
3: **for** each child of H with itemset $U = X \cdot i_p$ **do**
4: **repeat**
5: $Y = X \cdot i_q$ is the itemset of another child node of H
6: $Z \leftarrow X \cdot i_p \cdot i_q$
7: $LMAXW \leftarrow w^Z \leftarrow \frac{(w^X \times |X|) + w_{i_p} + w_{i_q}}{|X|+2}$
8: $IWS^Z \leftarrow intersect(IWS^U, IWS^Y)$
9: $r^Z \leftarrow lookup\text{-}r(IWS^Z)$
10: $s^Z \leftarrow lookup\text{-}f(IWS^Z)$
11: $ws^Z \leftarrow w^Z \times s^Z$
12: **until** $r^Z < \sigma_r$ and $ws^Z \geq \sigma_{ws}$
13: create a node for itemset Z with its information and then set to be a child node of U
14: $WFRIs \leftarrow WFRIs \cup Z$
15: **for** each child node of H with itemset $V = X \cdot i_q$ **do**
16: $Z \leftarrow X \cdot i_p \cdot i_q$
17: $IWS^Z \leftarrow intersect(IWS^U, IWS^V)$
18: $r^Z \leftarrow lookup\text{-}r(IWS^Z)$
19: $s^Z \leftarrow lookup\text{-}f(IWS^Z)$
20: $ows^Z \leftarrow LMAXW \times s^Z$
21: **if** $r^Z < \sigma_r$ and $ows^Z \geq \sigma_{ws}$ **then**
22: create a node for itemset Z with its information and then set to be a child node of U
23: $ws^Z \leftarrow s^Z \times \frac{((w^X \times |X|) + w_{i_p} + w_{i_q})}{|X|+2}$
24: **if** $ws^Z \geq \sigma_{ws}$ **then**
25: $WFRIs \leftarrow WFRIs \cup Z$
26: **if** U has more than one child **then**
27: *mining* (node of U, U, $\frac{w^X \times |X| + w_{i_p}}{|X|+1}$, σ_r, σ_{ws})
28: **else**
29: remove U and the child of U out of *WFRIM-tree*

4.2 ตัวอย่างสำหรับวิธีการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการโดยใช้โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง

กำหนดให้ฐานข้อมูลทรานแซกชันมีจำนวน 80 ทรานแซกชัน ซึ่งประกอบไปด้วยรายการ a, b, c, d, e และ f และตารางค่าน้ำหนักของแต่ละรายการ แสดงดังภาพที่ 4-6 ซึ่งมีค่าขีดแบ่งน้ำหนักสนับสนุน σ_{ws} เป็น 4.0 และค่าขีดแบ่งความสม่ำเสมอ σ_r เป็น 20

tid	item
1	a, c, e
...	...
15	b, c, d, f, g
16	a, b, e, f, g
...	...
32	a, b, c, e, f, g
...	...
35	b, d, g
...	...
49	a, b, c, e, g
50	a, b, c, e, g
...	...
55	a, b, d, e
...	...
70	a, b, c, d, e, g
71	a, b, d, e
...	...

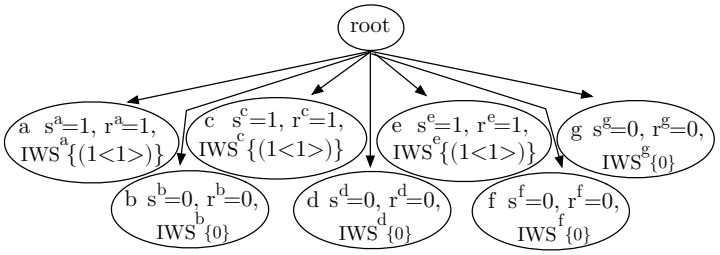
A transactional database

item	weight
a	0.75
b	0.65
c	0.65
d	0.4
e	0.7
f	0.5
g	0.6

Weight Table

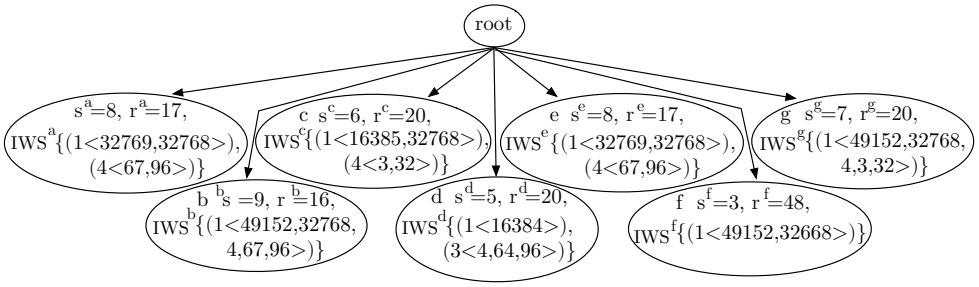
ภาพที่ 4-6 ฐานข้อมูลทรานแซกชันและตารางค่าน้ำหนัก

ในขั้นตอน DBscanning เริ่มแรก *WFRIM-tree* จะถูกสร้างร่วมกับโหนดราก R และสร้างโหนดสำหรับแต่ละรายการ a, b, c, d, e และ f โดยเชื่อมโยง (link) แต่ละโหนดของรายการให้เป็นโหนดลูกของโหนดราก R ต่อมาอ่านทรานแซกชัน t_1= { a, c, e } ทำการอัพเดต IWS^a, IWS^c และ IWS^e โดยหมายเลขทรานแซกชัน 1 (รายละเอียดการจัดเก็บหมายเลขทรานแซกชันเป็นโครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง (IWS) บรรยายไว้ในขั้นตอนวิธี 4.1) คำนวนค่าสนับสนุน ค่าความสม่ำเสมอ และจัดเก็บหมายเลขทรานแซกชันสุดท้ายที่รายการ a, c และ e ปรากฏขึ้นหลังจากอัพเดตแล้วจะได้ดังภาพที่ 4-7



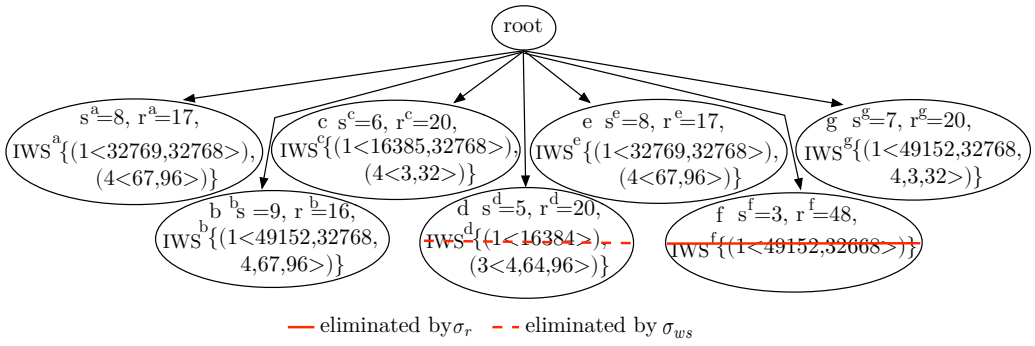
ภาพที่ 4-7 *WFRIM-tree* หลังจากอ่านทรานแซกชัน t_1

ขั้นตอนการอ่านฐานข้อมูลและอัพเดตค่าต่าง ๆ ของแต่ละรายการจะทำจนครบทุกทรานแซกชันในฐานข้อมูล หลังจากอ่านฐานข้อมูลครบแล้วจะได้รับ *WFRIM-tree* แสดงดังภาพที่ 4-8



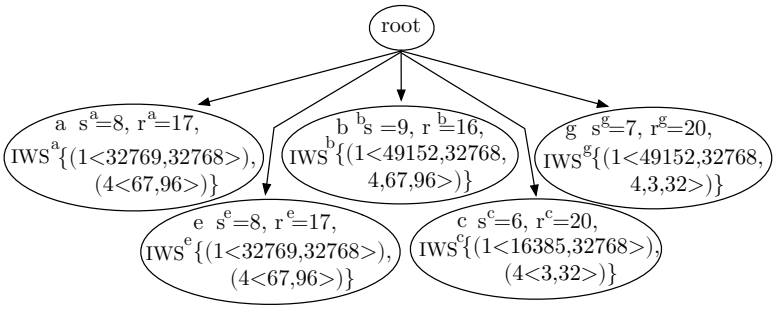
ภาพที่ 4-8 *WFRIM-tree* หลังจากอ่านฐานข้อมูลครบทุกทรานแซกชัน

จากนั้นพิจารณาค่าความสม่ำเสมอของแต่ละรายการใน *WFRIM-tree* (ค่าความสม่ำเสมอคำนวณได้จาก 4 กรณีที่กล่าวไว้ด้านบน) ในภาพที่ 4-8 จะเห็นได้ว่าค่าความสม่ำเสมอของรายการ f (r^f) มีค่าเป็น r^f=48 ซึ่งมีค่ามากกว่าค่าขีดแบ่งความสม่ำเสมอ σ_r=20 ดังนั้นจึงลบรายการ f ออกจาก *WFRIM-tree* แสดงดังภาพที่ 4-9 ขั้นตอนต่อมาคำนวณหาค่าน้ำหนักที่มากสุดซึ่งคำนวณได้จากค่าน้ำหนักที่มากสุดของรายการทั้งหมด (รายการที่เหลืออยู่หลังจากการตัดออกด้วยค่าขีดแบ่งความสม่ำเสมอ) ดังนั้นค่าน้ำหนักที่มากสุด $GMAXW=max(w^a, w^b, w^c, w^d, w^e, w^g)$ = max(0.75, 0.65, 0.65, 0.4, 0.7, 0.6) =0.75 หลังจากได้ค่าน้ำหนักที่มากสุดแล้วจะทำการคำนวณค่าน้ำหนักสนับสนุนโดยประมาณ ows ของแต่ละรายการจากภาพที่ 4-8 จะเห็นได้ว่ารายการ d มีค่า ows^d=0.75 × 5 =3.75 ซึ่งมีค่าน้อยกว่าค่าขีดแบ่งน้ำหนักสนับสนุน σ_{ws}=4.0 จึงลบรายการ d ออกจาก *WFRIM-tree* แสดงดังภาพที่ 4-9



ภาพที่ 4-9 *WFRIM-tree* หลังจากลบรายการที่มีค่า $r > \sigma_r$ และลบรายการที่มี $ows < \sigma_{ws}$

จากนั้นเรียงลำดับแต่ละรายการใน *WFRIM-tree* จากค่าน้ำหนักมากไปน้อยซึ่งจะได้ลำดับเป็น a, e, b, c, g แสดงดังภาพที่ 4-10



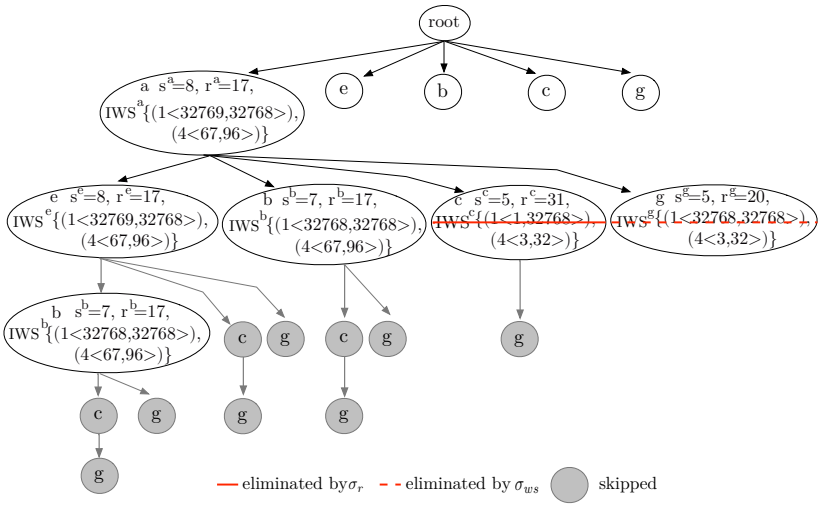
ภาพที่ 4-10 *WFRIM-tree* หลังจากจบขั้นตอน DBscanning

WFRIM-tree ที่ได้รับจากขั้นตอน DBscanning ประกอบไปด้วย 5 รายการ แสดงในภาพที่ 4-10 เพื่อที่จะค้นหาเซตรายการที่เป็นผลลัพธ์ของเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการนั้นจะดำเนินการตามขั้นตอนวิธีที่ 4.4 โดยเริ่มจากรายการ a ซึ่งเป็นรายการที่มีค่าน้ำหนักที่มากสุดและรายการ e ซึ่งเป็นรายการที่มีค่าน้ำหนักมากเป็นอันดับที่สอง แล้วสร้างเซตรายการที่มีขนาดใหญ่ขึ้นเป็นเซตรายการ “ae” จากนั้นค่าน้ำหนักที่มากสุดของเซตรายการที่พิจารณา ($LMAXW$) และค่าน้ำหนักของเซตรายการ ae (w^{ae}) สามารถคำนวณได้จาก $LMAXW = w^{ae} = \frac{w^a + w^e}{2} = \frac{0.75+0.7}{2}$ =0.725 โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วงของรายการ a เป็น IWS^a={⟨1(32769, 32768)⟩, ⟨4(67, 96)⟩} และ e เป็น IWS^e={⟨1(32769, 32768)⟩, ⟨4(67, 96)⟩} แล้วทำการอินเตอร์เซกชันระหว่าง IWS^a และ IWS^e แล้วจัดเก็บใน IWS^{ae}={⟨1(32769, 32768)⟩, ⟨4(67, 96)⟩} ที่ซึ่งค่าความสม่ำเสมอของ ae คือ r^{ae}=17 และค่าสนับสนุนของ ae คือ s^{ae}=8 (คำนวณได้จากตารางค้นหา) โดยค่าน้ำหนักสนับสนุนมีค่าเท่ากับ $ws^{ae} = w^{ae} \times s^{ae}$=0.725×8 =5.8 ดังนั้นเซตรายการ ae จึงเป็นเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ เนื่องจากมีค่าความสม่ำเสมอ r^{ae}=17 ซึ่งน้อยกว่าค่าขีดแบ่งความสม่ำเสมอ σ_r=20 และค่าน้ำหนักสนับสนุน ws^{ae}=5.8 ซึ่งมากกว่าค่าขีดแบ่งน้ำหนักสนับสนุน σ_{ws}=4 จากนั้นทำการสร้างโหนด e โดยเชื่อมโยงไปยังโหนดลูกของโหนด a (กล่าวคือโหนด e แสดงถึงเซตรายการ ae)

จากนั้นรายการ a จะถูกพิจารณาร่วมกับรายการ b เพื่อสร้างเซตรายการที่มีขนาดใหญ่ขึ้นเป็นเซตรายการ “ab” แล้วทำการอินเตอร์เซกชันระหว่าง IWS^a และ IWS^b เพื่อสร้าง IWS^{ab} สำหรับจัดเก็บข้อมูลการปรากฏของเซตรายการ ab ซึ่งสามารถคำนวณได้เป็น IWS^{ab}={⟨1(32768, 32768)⟩, ⟨4(67, 96)⟩} แล้วทำการคำนวณค่าความสม่ำเสมอและค่าสนับสนุนของเซตรายการ ab จากตารางการค้นหาได้เป็น r^{ab}=17 และ s^{ab}=7 ตามลำดับ ต่อมาทำการคำนวณหาค่าน้ำหนักสนับสนุนโดยประมาณ (ows) ของเซตรายการ ab ซึ่งสามารถคำนวณได้จาก $ows^{ab} =$

$LMAXW \times s^{ab}$ = 0.725×7 = 5.075 เนื่องจากค่าความสม่ำเสมอของเซตรายการ ab มีค่าที่น้อยกว่าค่าขีดแบ่งความสม่ำเสมอ σ_r=20 และมีค่าน้ำหนักโดยประมาณที่ ows^{ab}= 5.075 ซึ่งมากกว่าค่าขีดแบ่งน้ำหนักสนับสนุน σ_{ws}=4 ดังนั้นให้สร้างโหนด b โดยเชื่อมโยงไปยังโหนดลูกของโหนด a (กล่าวคือโหนด b แสดงถึงเซตรายการ ab) ต่อมาทำการพิจารณาผลลัพธ์ของเซตรายการ ab โดยการคำนวณค่าน้ำหนักสนับสนุนที่ซึ่งสามารถคำนวณได้เป็น $ws^{ab} = w^{ab} \times s^{ab} = \frac{0.75+0.65}{2} \times 7$ = 4.9 ด้วยเหตุนี้เซตรายการ ab จึงเป็นเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ เนื่องจากมีค่าความสม่ำเสมอ r^{ab}=17 ซึ่งน้อยกว่าค่าขีดแบ่งความสม่ำเสมอ σ_r=20 และค่าน้ำหนักสนับสนุน ws^{ab}= 4.9 ซึ่งมากกว่าค่าขีดแบ่งน้ำหนักสนับสนุน σ_{ws}=4

การพิจารณาเซตรายการที่ถัดจากเซตรายการ b จะมีการทำซ้ำดังตัวอย่างที่กล่าวมาข้างต้นต้น เมื่อทำการพิจารณาทุกรายการเสร็จสิ้นแล้ว ถ้าโหนดลูกของเซตรายการ a มีจำนวนมากกว่า 1 โหนดให้ทำการพิจารณาแบบวนซ้ำในโหนดลูกของรายการ a ตามขั้นตอนวิธี WFRI-mining แต่ถ้าโหนดลูกของเซตรายการ a มีจำนวนที่น้อยกว่าหรือเท่ากับ 1 โหนดให้ทำการลบโหนดของเซตรายการ a และโหนดลูกทั้งหมดของเซตรายการ a ออกจากการพิจารณาใน *WFRIM-tree* แล้วทำการพิจารณารายการ e และพิจารณารายการที่ถัดไปจากรายการ e แสดงดังภาพที่ 4-11



ภาพที่ 4-11 กระบวนการในการค้นหาผลลัพธ์ที่ซึ่งปรากฏร่วมกับรายการ a

จากตัวอย่างที่แสดงข้างต้น เมื่อทำการพิจารณารายการทั้งหมดภายใน *WFRIM-tree* จะได้ผลลัพธ์ของเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่

ละรายการ ดังนี้ ผลลัพธ์รายการขนาดที่ 1 คือรายการ a (r^a=17, ws^a= 6), รายการ e (r^e=17, ws^e= 5.6), รายการ b (r^b=16, ws^b= 5.85) และรายการ g (r^g=20, ws^g= 4.2) ผลลัพธ์เซตรายการขนาดที่ 2 คือเซตรายการ ae (r^{ae}=17, ws^{ae}= 5.08), เซตรายการ ab (r^{ab}=17, ws^{ab}= 4.9), เซตรายการ eb (r^{eb}=17, ws^{eb}= 4.73) และเซตรายการ bg (r^{bg}=20, ws^{bg}= 4.38) และผลลัพธ์เซตรายการขนาดที่ 3 คือเซตรายการ aeb (r^{aeb}=17, ws^{aeb}= 4.9)

4.3 การวิเคราะห์ความซับซ้อนของขั้นตอนวิธี

ในส่วนนี้จะบรรยายถึงการวิเคราะห์ความซับซ้อนของขั้นตอนวิธี WFRIM-IWS ซึ่งจะพิจารณาอยู่ 2 ส่วนหลักได้แก่ 1) การวิเคราะห์ความซับซ้อนเชิงเวลา และ 2) การวิเคราะห์ความซับซ้อนเชิงหน่วยความจำ ดังนี้

4.3.1 การวิเคราะห์ความซับซ้อนเชิงเวลาของขั้นตอนวิธี WFRIM-IWS

สำหรับสร้าง *WFRIM-tree* และสร้างเซตรายการขนาด 1 เซตรายการของขั้นตอนวิธี DBscanning จะทำการพิจารณารายการที่ปรากฏขึ้นในทรานแซกชันของฐานข้อมูล ดังนั้นความซับซ้อนเชิงเวลาของขั้นตอนวิธีนี้คือ $O(nm)$ ซึ่ง n คือจำนวนรายการทั้งหมดในฐานข้อมูลและ m คือจำนวนทรานแซกชันทั้งหมดในฐานข้อมูล จากนั้นพิจารณาทุกรายการใน *WFRIM-tree* ซึ่งมีความซับซ้อนเชิงเวลาเป็น $O(n)$ โดยรายการที่ผ่านการพิจารณา p จะถูกนำไปพิจารณาเป็นเซตรายการผลลัพธ์บน *WFRIM-tree* ในขั้นตอน WFRI-mining เพื่อทำการหาผลลัพธ์ของเซตรายการขนาดต่าง ๆ ซึ่งมีความซับซ้อนเชิงเวลาคือ $O(2^p)$ โดยแต่ละรายการจะถูกคำนวณค่าความสม่ำเสมอและค่าสนับสนุนโดยการอินเตอร์เซกชันของสองเซตรายการ ดังนั้นความซับซ้อนเชิงเวลาของขั้นตอน WFRI-mining คือ $O(2^p 2k)$ กำหนดให้ k คือขนาดของ IWS ที่ $k = m/16$ ด้วยเหตุนี้ความซับซ้อนเชิงเวลาของขั้นตอนวิธี WFRIM-IWS จึงมีค่าเท่ากับ $O((nm) + (n) + (2^p 2k)$

4.3.2 การวิเคราะห์ความซับซ้อนเชิงหน่วยความจำของขั้นตอนวิธี WFRIM-IWS

สำหรับขั้นตอนวิธี DBscanning จะมีความซับซ้อนเชิงหน่วยความจำเป็น $O(nk)$ ที่ n คือจำนวนรายการทั้งหมดในฐานข้อมูลและ k คือขนาดของ IWS ที่ $k = m/16$ เมื่อ m คือจำนวนทรานแซกชันทั้งหมดในฐานข้อมูล ในการพิจารณาเซตรายการขนาดต่างๆ ของขั้นตอน WFRI-mining จะมีความซับซ้อนเชิงเวลาเป็น $O(p^2 k)$ ที่ k คือขนาดของ IWS ที่ถูกจัดเก็บในแต่ละโหนดของเซตรายการและ p คือจำนวนรายการที่ผ่านการพิจารณาจากขั้นตอนวิธี DBscanning ดังนั้นความซับซ้อนเชิงหน่วยความจำของขั้นตอนวิธี WFRIM-IWS สามารถระบุได้เป็น $O((nk) + (p^2 k))$

บทที่ 5
ผลการดำเนินงาน

ในบทนี้จะกล่าวถึงผลการดำเนินงานในการทดสอบประสิทธิภาพของขั้นตอนวิธีสำหรับค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการจากฐานข้อมูล โดยจะทดสอบประสิทธิภาพขั้นตอนวิธี WFRIM เปรียบเทียบกับขั้นตอนวิธี WFRIM-IWS ใน 10 ฐานข้อมูล โดยที่จะมีการกำหนดค่าขีดแบ่งความสม่ำเสมอ (σ_r) และค่าขีดแบ่งน้ำหนักสนับสนุน (σ_{ws}) ที่หลากหลายในแต่ละฐานข้อมูลเพื่อศึกษาและเปรียบเทียบถึงเวลาและหน่วยความจำที่ใช้ในการประมวลผลระหว่างขั้นตอนวิธีทั้งสองรวมถึงแสดงจำนวนเซตรายการที่เป็นผลลัพธ์ในแต่ละฐานข้อมูล ซึ่งรายละเอียดของผลการดำเนินงานแสดงดังต่อไปนี้

1 การออกแบบการทดสอบประสิทธิภาพ

2 เวลาที่ใช้ในการประมวลผล

3 หน่วยความจำที่ใช้ในการประมวลผล

4 จำนวนของเซตรายการที่เป็นผลลัพธ์

5.1 การออกแบบการทดสอบประสิทธิภาพ

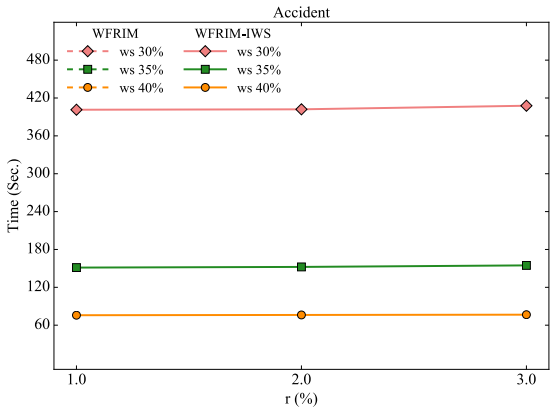
การทดสอบประสิทธิภาพของขั้นตอนวิธีสำหรับการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการนั้นจะทำการทดสอบในแต่ละฐานข้อมูลที่ได้บรรยายในบทที่ 2 ตรงส่วนของคุณลักษณะฐานข้อมูลที่ใช้ทดสอบงานวิจัยที่วิทยานิพนธ์นี้นำเสนอ ที่ซึ่งประกอบไปด้วย 8 ฐานข้อมูลที่ข้อมูลเป็นข้อมูลจริง (Accidents Chess Connect Kosarak Mushroom Pumsb Pumsb* และ Retail) และฐานข้อมูลที่ข้อมูลถูกสังเคราะห์ขึ้น (T10I4D100K และ T40I10D100K) และทำการทดสอบประสิทธิภาพของขั้นตอนวิธีที่นำเสนอบนเครื่อง Macbookpro ระบบปฏิบัติการ OSX High Sierra, CPU 2.4 GHz, RAM 8 GB ซึ่งขั้นตอนวิธี WFRIM และขั้นตอนวิธี WFRIM-IWS เขียนโปรแกรมโดยใช้ภาษาไพธอน (Python) ในการทดสอบประสิทธิภาพนั้นจะมีการกำหนดค่าขีดแบ่งความสม่ำเสมอ (σ_r) และค่าขีดแบ่งน้ำหนักสนับสนุน (σ_{ws}) ซึ่งค่าขีดแบ่งทั้งสองกำหนดจากการพิจารณาคุณลักษณะของข้อมูลในแต่ละฐานข้อมูล เมื่อพิจารณาถึงจำนวนของเซตรายการที่เป็นผลลัพธ์จะแปรผันตามการกำหนดค่าขีดแบ่งทั้งสองกล่าวคือ เมื่อค่าขีดแบ่งความสม่ำเสมอมากขึ้นและค่าขีดแบ่งน้ำหนักสนับสนุนน้อยลงเซตรายการที่เป็นผลลัพธ์ก็จะเพิ่มขึ้น สำหรับฐานข้อมูลมีคุณลักษณะหนาแน่นรายการ/เซตรายการมีการปรากฏบ่อยจึงกำหนดค่าขีดแบ่งความสม่ำเสมอที่ต่ำและค่าขีดแบ่งน้ำหนักสนับสนุนข้างสูง แต่ใน

กรณีที่ฐานข้อมูลที่มีคุณลักษณะเบาบางนั้นรายการ/เซตรายมีการปรากฏในฐานข้อมูลไม่บ่อยนักจึงทำการกำหนดค่าขีดแบ่งความสม่ำเสมอที่ค่อนข้างสูงและค่าขีดแบ่งน้ำหนักสนับสนุนค่อนข้างต่ำ

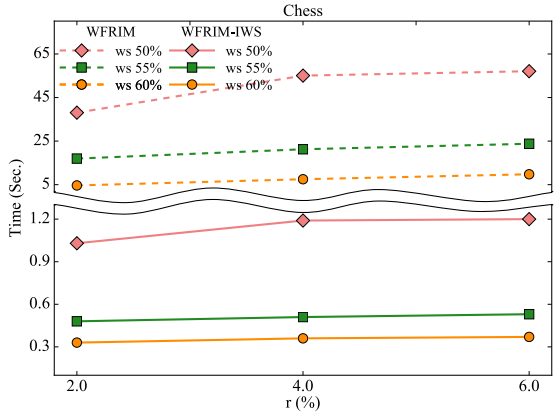
5.2 เวลาที่ใช้ในการประมวลผล

เวลาที่ใช้ในการประมวลผลของขั้นตอนวิธี WFRIM และขั้นตอนวิธี WFRIM-IWS แสดงในภาพที่ 5-1 ถึงภาพที่ 5-10 โดยแต่ละเส้นในกราฟแสดงถึงเวลาที่ใช้ในการประมวลผลของหนึ่งค่าขีดแบ่งน้ำหนักสนับสนุนในหลากหลายค่าขีดแบ่งความสม่ำเสมอ ซึ่งเวลาที่ใช้ในการประมวลผลของฐานข้อมูลที่มีคุณลักษณะข้อมูลหนาแน่น (Accidents Chess Connect Mushroom Pumsb และ Pumsb*) แสดงในภาพที่ 5-1 ถึงภาพที่ 5-6 จากภาพดังกล่าวแสดงให้เห็นถึง 1) เมื่อค่าขีดแบ่งความสม่ำเสมอเพิ่มขึ้นแล้วไม่ส่งผลกับเวลาในการประมวลผลของทั้งสองขั้นตอนวิธีแสดงในฐานข้อมูล Accidents Connect Pumsb และ Pumsb* (หมายเหตุ ฐานข้อมูล Accidents ภาพที่ 5-1 ไม่มีการแสดงเวลาที่ใช้ในการประมวลผลของขั้นตอนวิธี WFRIM เนื่องจากใช้เวลาในการประมวลผลนานกว่า 30,000 วินาที) แต่ในฐานข้อมูล Chess และ Mushroom เวลาที่ใช้ในการประมวลผลจะเพิ่มขึ้นเมื่อค่าขีดแบ่งความสม่ำเสมอมีค่าเพิ่มขึ้น 2) เวลาที่ใช้ในการประมวลผลของทั้งสองขั้นตอนวิธีจะเพิ่มขึ้นเมื่อค่าขีดแบ่งน้ำหนักสนับสนุนน้อยลงกล่าวคือ เมื่อค่าขีดแบ่งน้ำหนักสนับสนุนน้อยลงส่งผลให้มีรายการ/เซตรายการที่พิจารณาจำนวนมากขึ้นซึ่งทั้งสองขั้นตอนวิธีจะใช้เวลาในการพิจารณารายการ/เซตรายการที่เพิ่มขึ้นด้วยและ 3) ขั้นตอนวิธี WFRIM-IWS มีเวลาในการประมวลผลที่ดีกว่าขั้นตอนวิธี WFRIM ในทุกฐานข้อมูลที่มีคุณลักษณะข้อมูลหนาแน่นและทุกค่าขีดแบ่งทั้งสองที่กล่าวข้างต้น

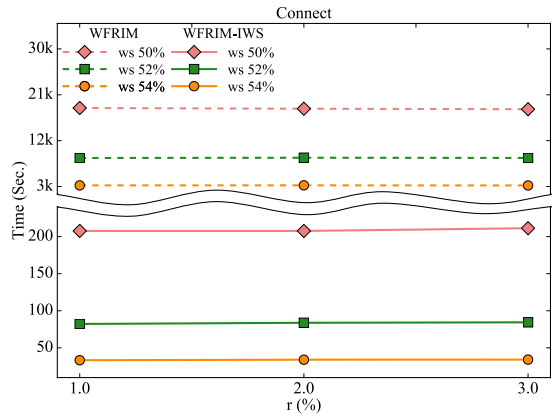
สำหรับเวลาที่ใช้ในการประมวลผลของขั้นตอนวิธี WFRIM และ WFRIM-IWS ในฐานข้อมูลที่มีคุณลักษณะข้อมูลเบาบาง Kosarak Retail T10I4D100K และ T40I10D100K แสดงในภาพที่ 5-7 ถึงภาพที่ 5-10 จากภาพแสดงให้เห็นว่า 1) เมื่อค่าขีดแบ่งความสม่ำเสมอเพิ่มขึ้นแล้วไม่ส่งผลกับเวลาในการประมวลผลของทั้งสองขั้นตอนวิธีแสดงในฐานข้อมูล Kosarak และ T40I10D100K แต่ในฐานข้อมูล Retail และ T10I4D100K เวลาที่ใช้ในการประมวลผลจะเพิ่มขึ้นเล็กน้อยเมื่อค่าขีดแบ่งความสม่ำเสมอมีค่าเพิ่มขึ้น 2) เวลาที่ใช้ในการประมวลผลของทั้งสองขั้นตอนวิธีจะเพิ่มขึ้นเมื่อค่าขีดแบ่งน้ำหนักสนับสนุนน้อยลงและ 3) ขั้นตอนวิธี WFRIM-IWS มีเวลาในการประมวลผลที่ดีกว่าขั้นตอนวิธี WFRIM ในทุกฐานข้อมูลที่มีคุณลักษณะข้อมูลเบาบางและทุกค่าขีดแบ่งความสม่ำเสมอและค่าขีดแบ่งน้ำหนักสนับสนุน



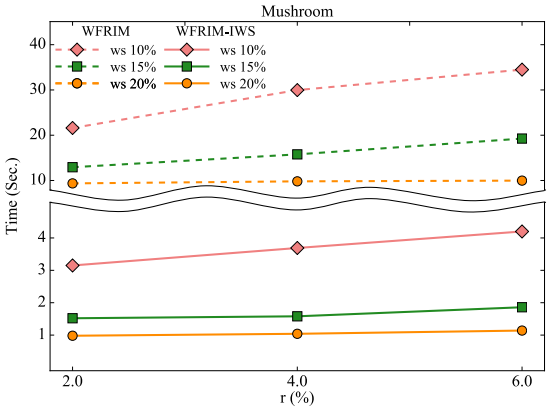
ภาพที่ 5-1 เวลาในการประมวลผลของ WFRIM และ WFRIM-IWS ในฐานข้อมูล Accident



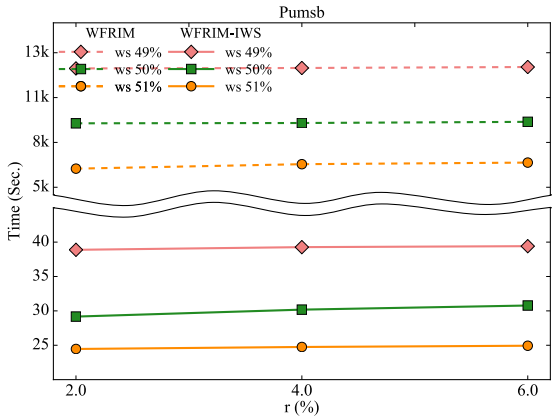
ภาพที่ 5-2 เวลาในการประมวลผลของ WFRIM และ WFRIM-IWS ในฐานข้อมูล Chess



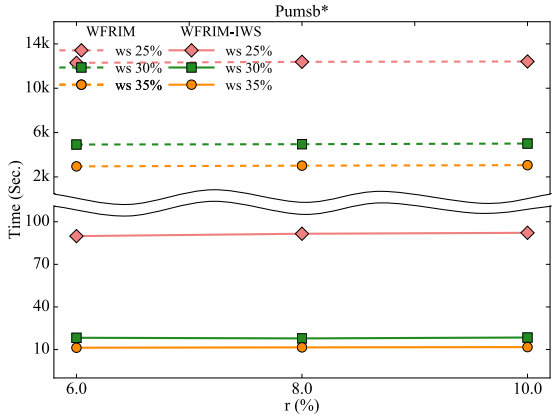
ภาพที่ 5-3 เวลาในการประมวลผลของ WFRIM และ WFRIM-IWS ในฐานข้อมูล Connect



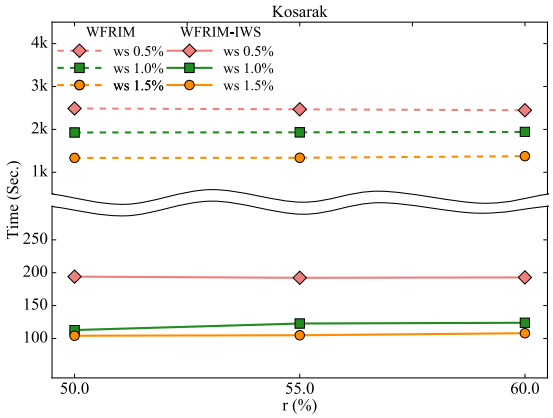
ภาพที่ 5-4 เวลาในการประมวลผลของ WFRIM และ WFRIM-IWS ในฐานข้อมูล Mushroom



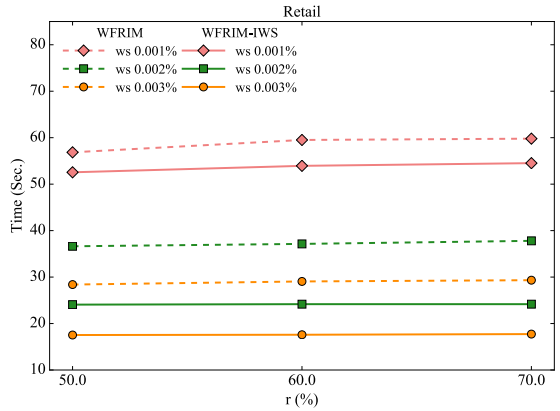
ภาพที่ 5-5 เวลาในการประมวลผลของ WFRIM และ WFRIM-IWS ในฐานข้อมูล Pumsb



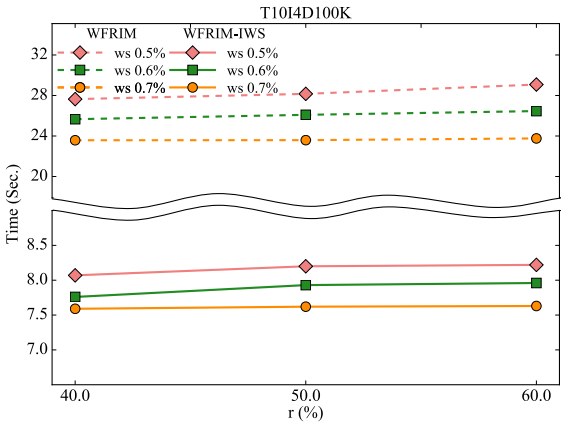
ภาพที่ 5-6 เวลาในการประมวลผลของ WFRIM และ WFRIM-IWS ในฐานข้อมูล Pumsb*



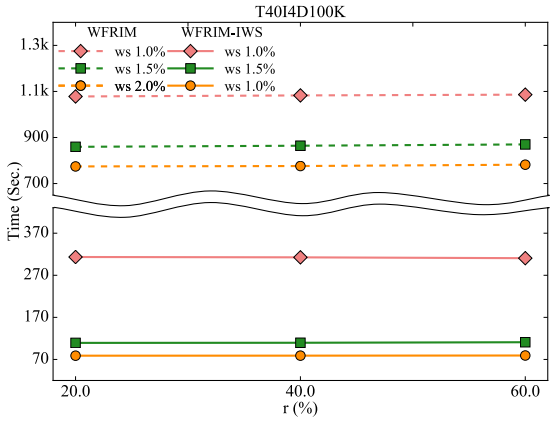
ภาพที่ 5-7 เวลาในการประมวลผลของ WFRIM และ WFRIM-IWS ในฐานข้อมูล Kosarak



ภาพที่ 5-8 เวลาในการประมวลผลของ WFRIM และ WFRIM-IWS ในฐานข้อมูล Retail



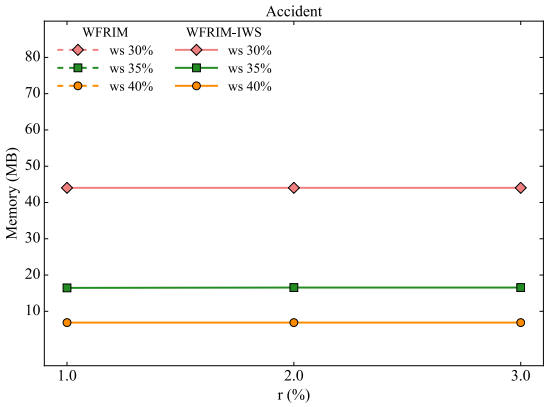
ภาพที่ 5-9 เวลาในการประมวลผลของ WFRIM และ WFRIM-IWS ในฐานข้อมูล T10I4D100K



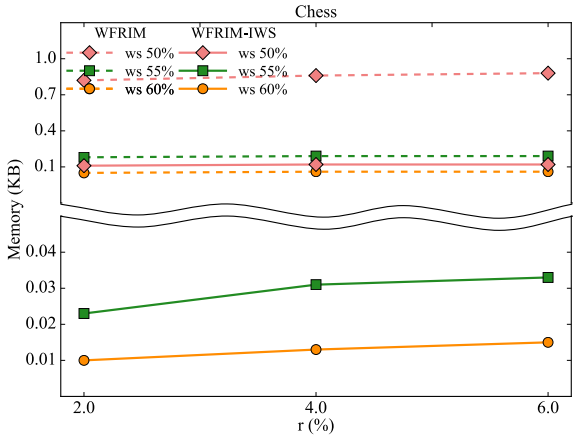
ภาพที่ 5-10 เวลาในการประมวลผลของ WFRIM และ WFRIM-IWS ในฐานข้อมูล T40I10D100K

5.3 หน่วยความจำที่ใช้ในการประมวลผล

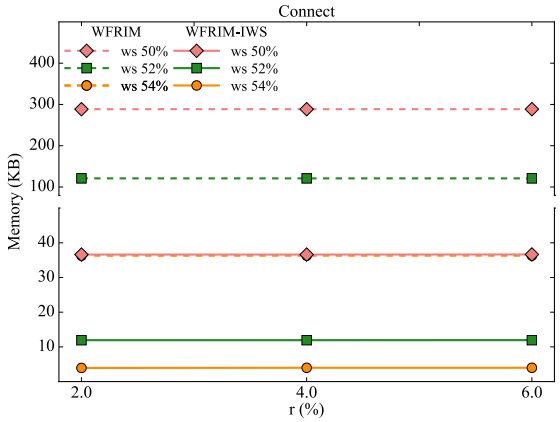
หน่วยความจำที่ใช้ในการประมวลผลของขั้นตอนวิธี WFRIM และ WFRIM-IWS แสดงในภาพที่ 5-11 ถึงภาพที่ 5-20 ซึ่งหน่วยความจำที่ใช้ในการประมวลผลของฐานข้อมูลที่มีคุณลักษณะข้อมูลหนาแน่นแสดงในภาพที่ 5-11 ถึง 5-16 ซึ่งภาพดังกล่าวแสดงให้เห็นว่าในฐานข้อมูล Accidents Connect Pumsb และ Pumsb* เมื่อมีการเพิ่มค่าขีดแบ่งความสม่ำเสมอในแต่ละค่าขีดแบ่งน้ำหนักสนับสนุนไม่ส่งผลต่อหน่วยความจำที่ใช้ในการประมวลผล แต่ในฐานข้อมูล Chess และ Mushroom จะมีการใช้หน่วยความจำเพิ่มขึ้นเมื่อมีการกำหนดค่าขีดแบ่งความสม่ำเสมอเพิ่มขึ้นในแต่ละค่าขีดแบ่งน้ำหนักสนับสนุน ในส่วนของฐานข้อมูลที่มีคุณลักษณะข้อมูลเบาบางนั้นหน่วยความจำที่ใช้ในการประมวลผลแสดงดังภาพที่ 5-17 ถึง 5-20 จากภาพฐานข้อมูล Kosarak T10I4D100K และ T40I10D100K เมื่อมีการเพิ่มค่าขีดแบ่งความสม่ำเสมอในแต่ละค่าขีดแบ่งน้ำหนักสนับสนุนไม่ส่งผลต่อหน่วยความจำที่ใช้ในการประมวลผล แต่ในฐานข้อมูล Retail หน่วยความจำที่ใช้ในการประมวลผลจะเพิ่มขึ้นเล็กน้อยเมื่อค่าขีดแบ่งความสม่ำเสมอมีค่าเพิ่มขึ้น ซึ่งในทุกฐานข้อมูลเมื่อกำหนดค่าขีดแบ่งน้ำหนักสนับสนุนน้อยลงหน่วยความจำที่ใช้ในการประมวลผลก็จะเพิ่มขึ้นเนื่องจากจำนวนรายการ/เซตรายการที่พิจารณาจะมีจำนวนมากขึ้นโดยขั้นตอนวิธี WFRIM จะมีการทำงานแบบวนซ้ำเพื่อสร้าง WFRI-tree และขั้นตอนวิธี WFRIM-IWS ก็มีการจัดเก็บโครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วงของเซตรายการที่เพิ่มขึ้นด้วยและขั้นตอนวิธี WFRIM-IWS ใช้หน่วยความจำในการประมวลผลที่น้อยกว่าขั้นตอนวิธี WFRIM ทั้งในฐานข้อมูลที่มีคุณลักษณะข้อมูลหนาแน่นและเบาบาง



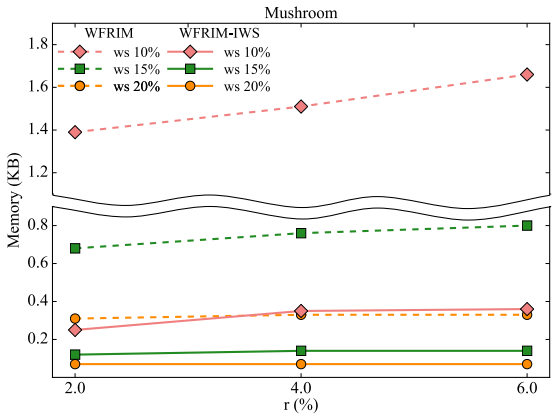
ภาพที่ 5-11 หน่วยความจำที่ใช้ของ WFRIM และ WFRIM-IWS ในฐานข้อมูล Accident



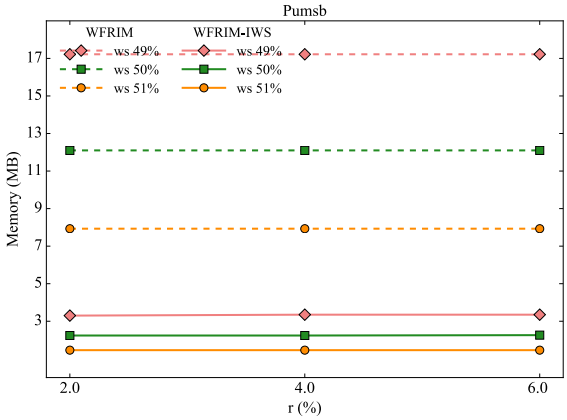
ภาพที่ 5-12 หน่วยความจำที่ใช้ของ WFRIM และ WFRIM-IWS ในฐานข้อมูล Chess



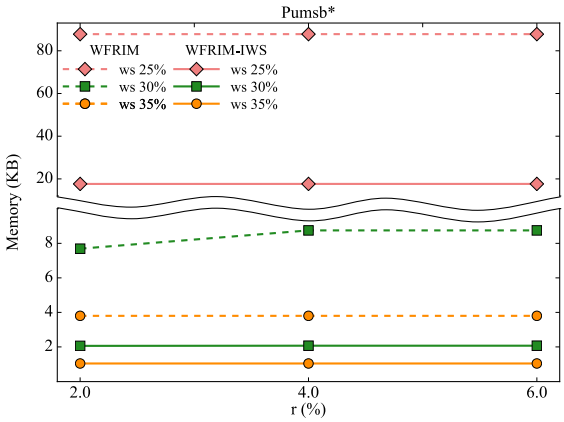
ภาพที่ 5-13 หน่วยความจำที่ใช้ของ WFRIM และ WFRIM-IWS ในฐานข้อมูล Connect



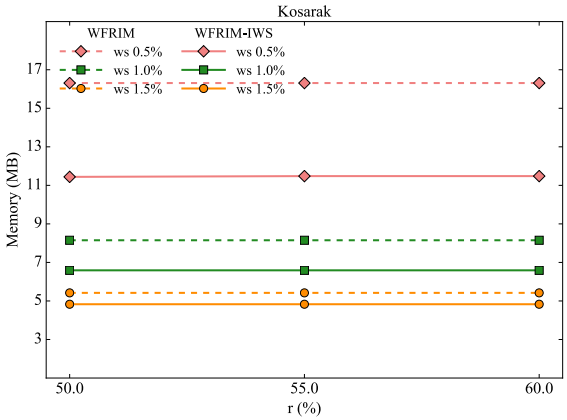
ภาพที่ 5-14 หน่วยความจำที่ใช้ของ WFRIM และ WFRIM-IWS ในฐานข้อมูล Mushroom



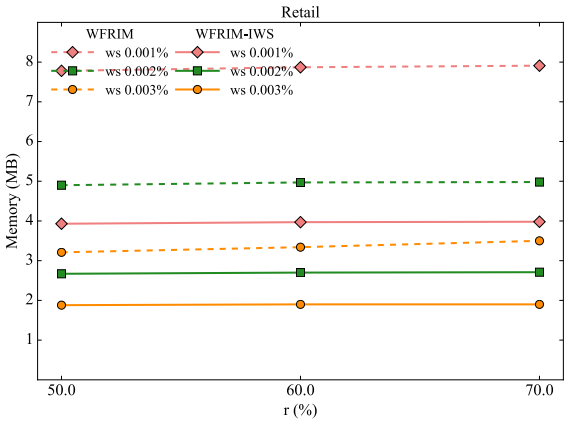
ภาพที่ 5-15 หน่วยความจำที่ใช้ของ WFRIM และ WFRIM-IWS ในฐานข้อมูล Pumsb



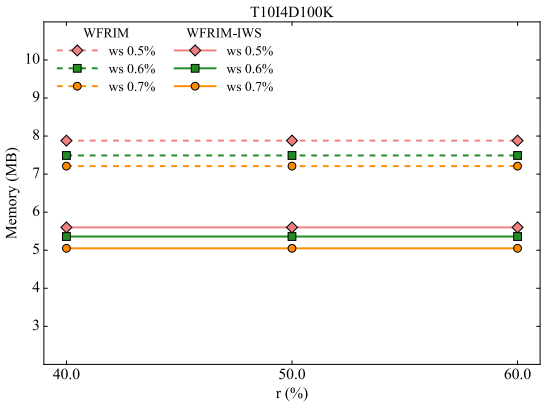
ภาพที่ 5-16 หน่วยความจำที่ใช้ของ WFRIM และ WFRIM-IWS ในฐานข้อมูล Pumsb*



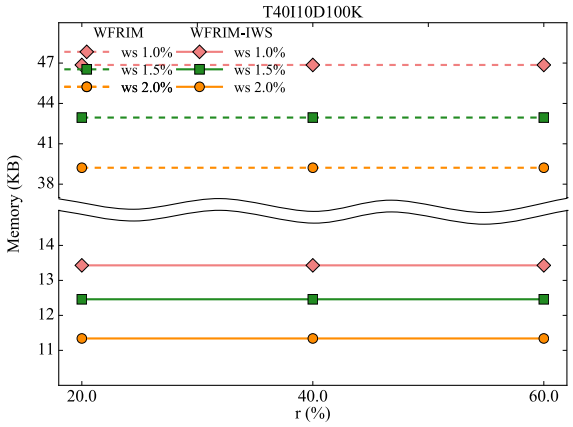
ภาพที่ 5-17 หน่วยความจำที่ใช้ของ WFRIM และ WFRIM-IWS ในฐานข้อมูล Kosarak



ภาพที่ 5-18 หน่วยความจำที่ใช้ของ WFRIM และ WFRIM-IWS ในฐานข้อมูล Retail



ภาพที่ 5-19 หน่วยความจำที่ใช้ของ WFRIM และ WFRIM-IWS ในฐานข้อมูล T10I4D100K

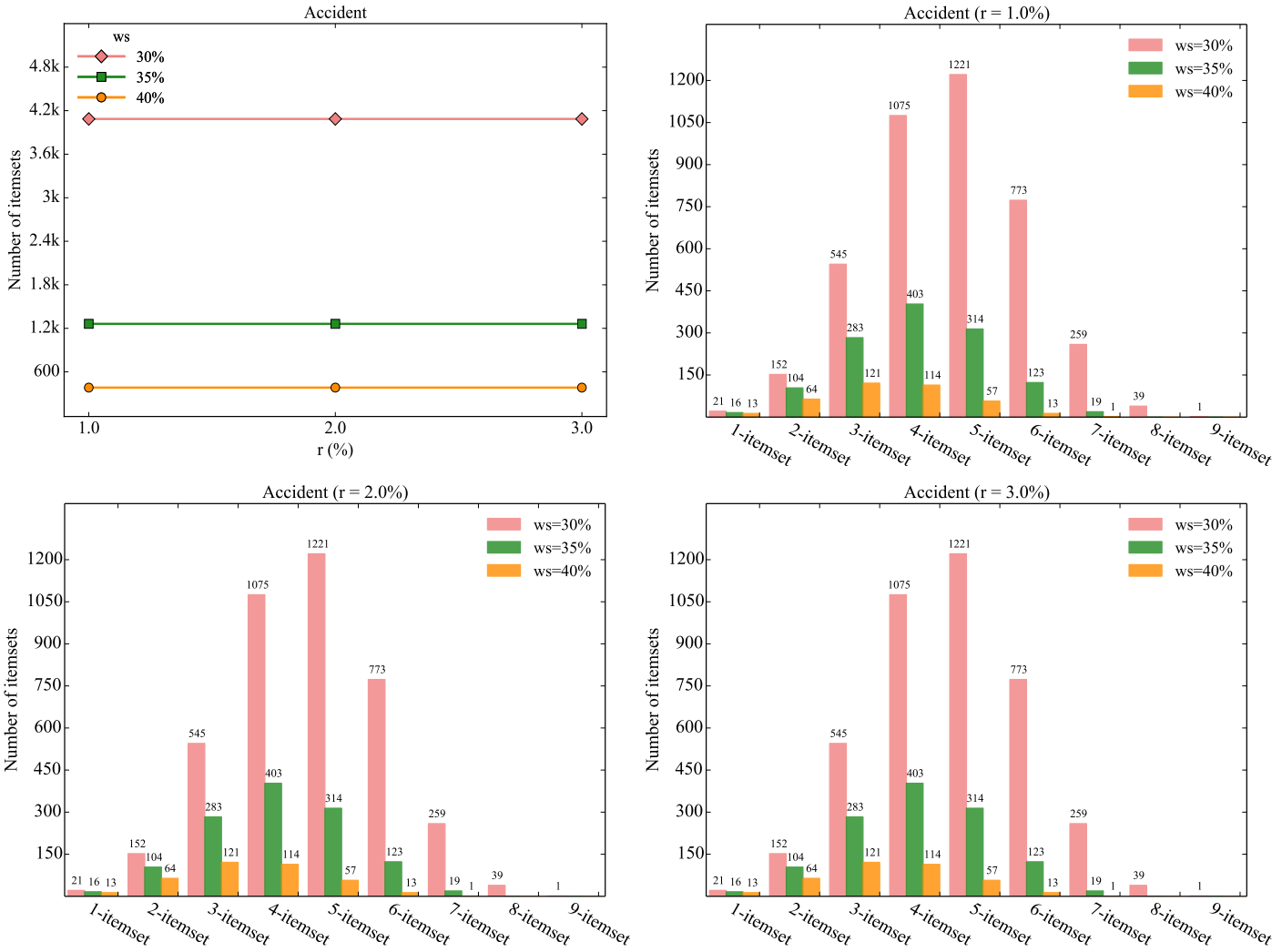


ภาพที่ 5-20 หน่วยความจำที่ใช้ของ WFRIM และ WFRIM-IWS ในฐานข้อมูล T40I10D100K

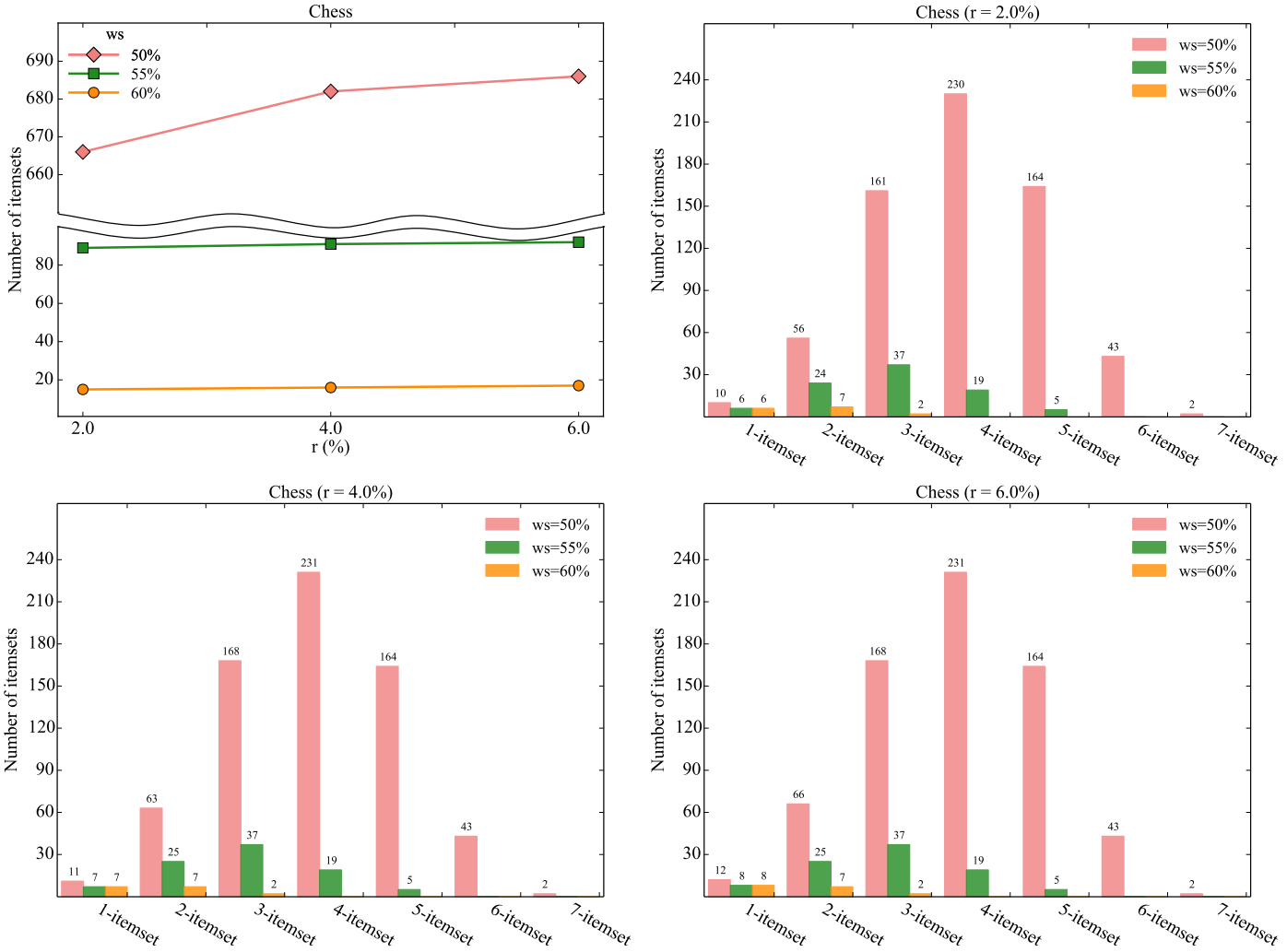
5.4 จำนวนของเซตรายการที่เป็นผลลัพธ์

จำนวนเซตรายการที่เป็นผลลัพธ์ของการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการจะแสดงในภาพที่ 5-21 ถึงภาพที่ 5-30 โดยที่ในแต่ละภาพแสดงจำนวนผลลัพธ์ทั้งหมดและผลลัพธ์ของเซตรายการขนาดต่าง ๆ ตามแต่ละค่าขีดแบ่งความสม่ำเสมอและค่าขีดแบ่งน้ำหนักสนับสนุน ซึ่งในฐานข้อมูลที่มีคุณลักษณะข้อมูลหนาแน่น Accidents Connect Pumsb และ Pumsb* เมื่อมีการเพิ่มค่าขีดแบ่งความสม่ำเสมอในแต่ละค่าขีดแบ่งน้ำหนักสนับสนุนขึ้นไม่ส่งผลให้จำนวนผลลัพธ์ของเซตรายการเพิ่มขึ้นด้วย แต่ในฐานข้อมูล Chess และ Mushroom เมื่อมีการเพิ่มค่าขีดแบ่งความสม่ำเสมอในแต่ละค่าขีดแบ่งน้ำหนักสนับสนุนขึ้นนั้นส่งผลให้ได้จำนวนผลลัพธ์ของเซตรายการเพิ่มขึ้น

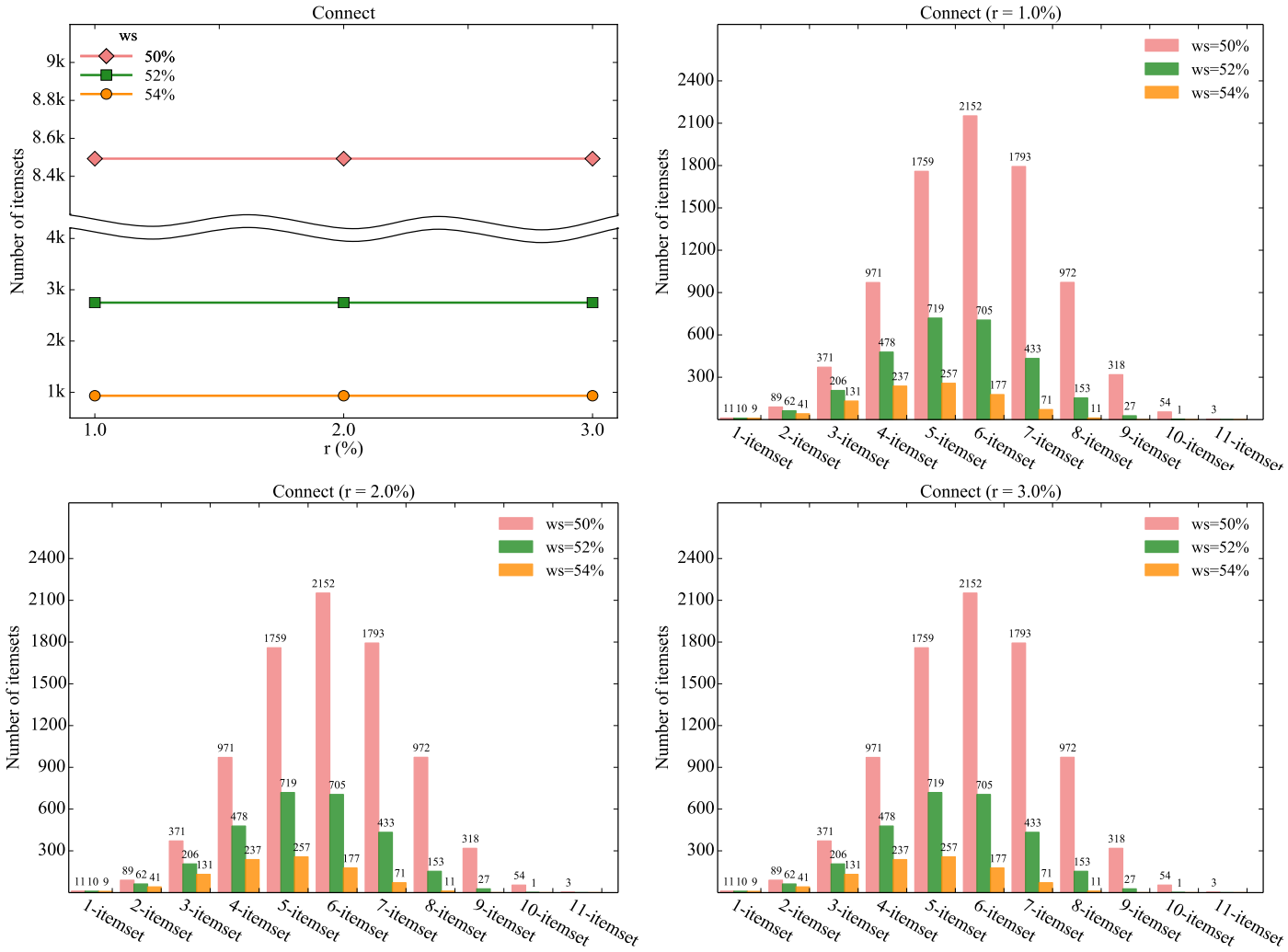
ในฐานข้อมูลที่มีคุณลักษณะข้อมูลเบาบางฐานข้อมูล Kosarak T10I4D100K และ T40I10D100K เมื่อมีการเพิ่มค่าขีดแบ่งความสม่ำเสมอในแต่ละค่าขีดแบ่งน้ำหนักสนับสนุนไม่ส่งผลให้จำนวนผลลัพธ์ของเซตรายการเพิ่มขึ้นด้วย แต่ในฐานข้อมูล Retail เมื่อมีการเพิ่มค่าขีดแบ่งความสม่ำเสมอในแต่ละค่าขีดแบ่งน้ำหนักสนับสนุนนั้นส่งผลให้ได้จำนวนผลลัพธ์ของเซตรายการเพิ่มขึ้น และจำนวนเซตรายการที่เป็นผลลัพธ์ของทุกฐานข้อมูลจะลดลงเมื่อมีการเพิ่มค่าขีดแบ่งน้ำหนักสนับสนุนอีกด้วย



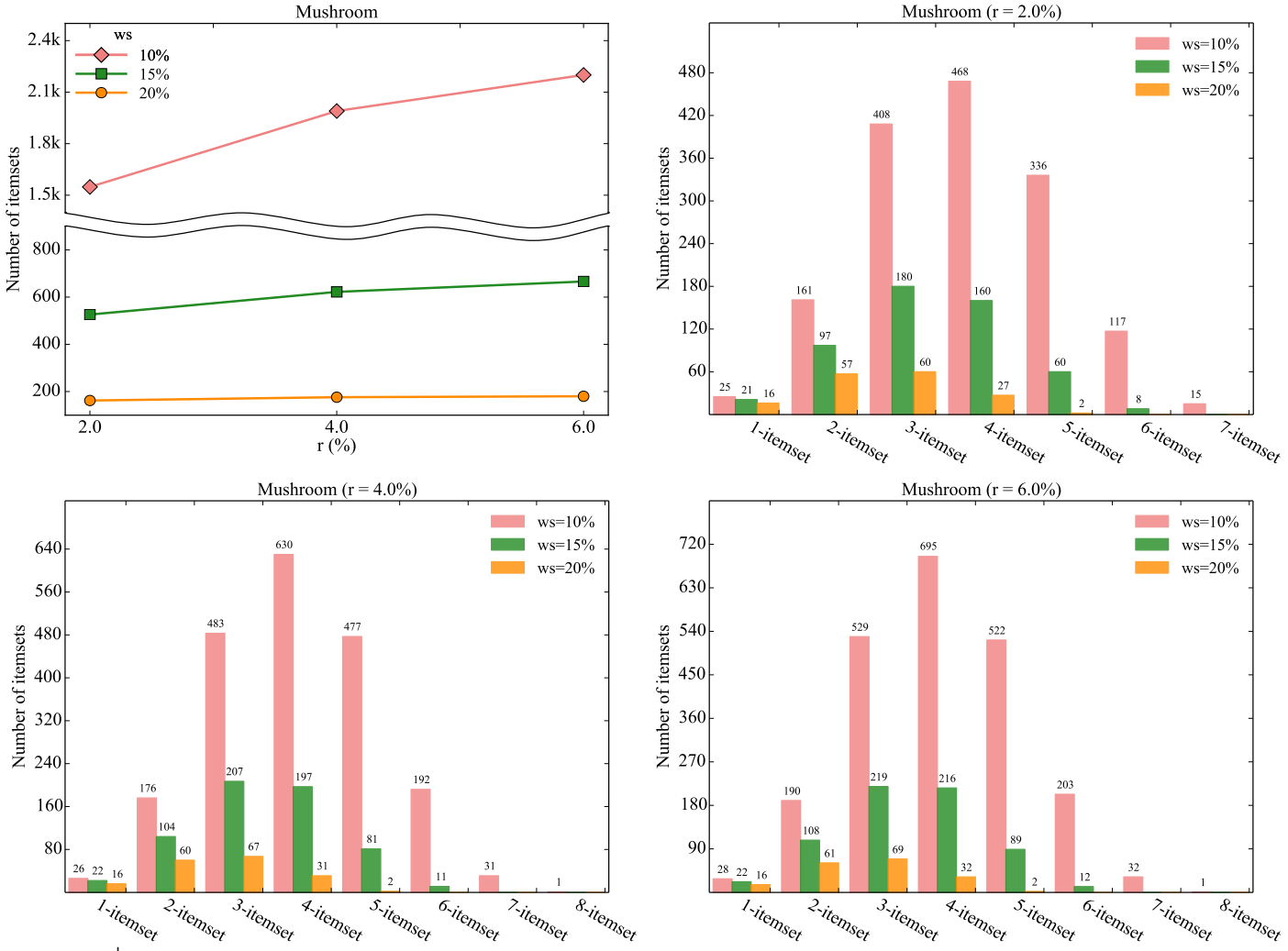
ภาพที่ 5-21 จำนวนเซตรายการที่เป็นผลลัพธ์ในฐานข้อมูล Accident



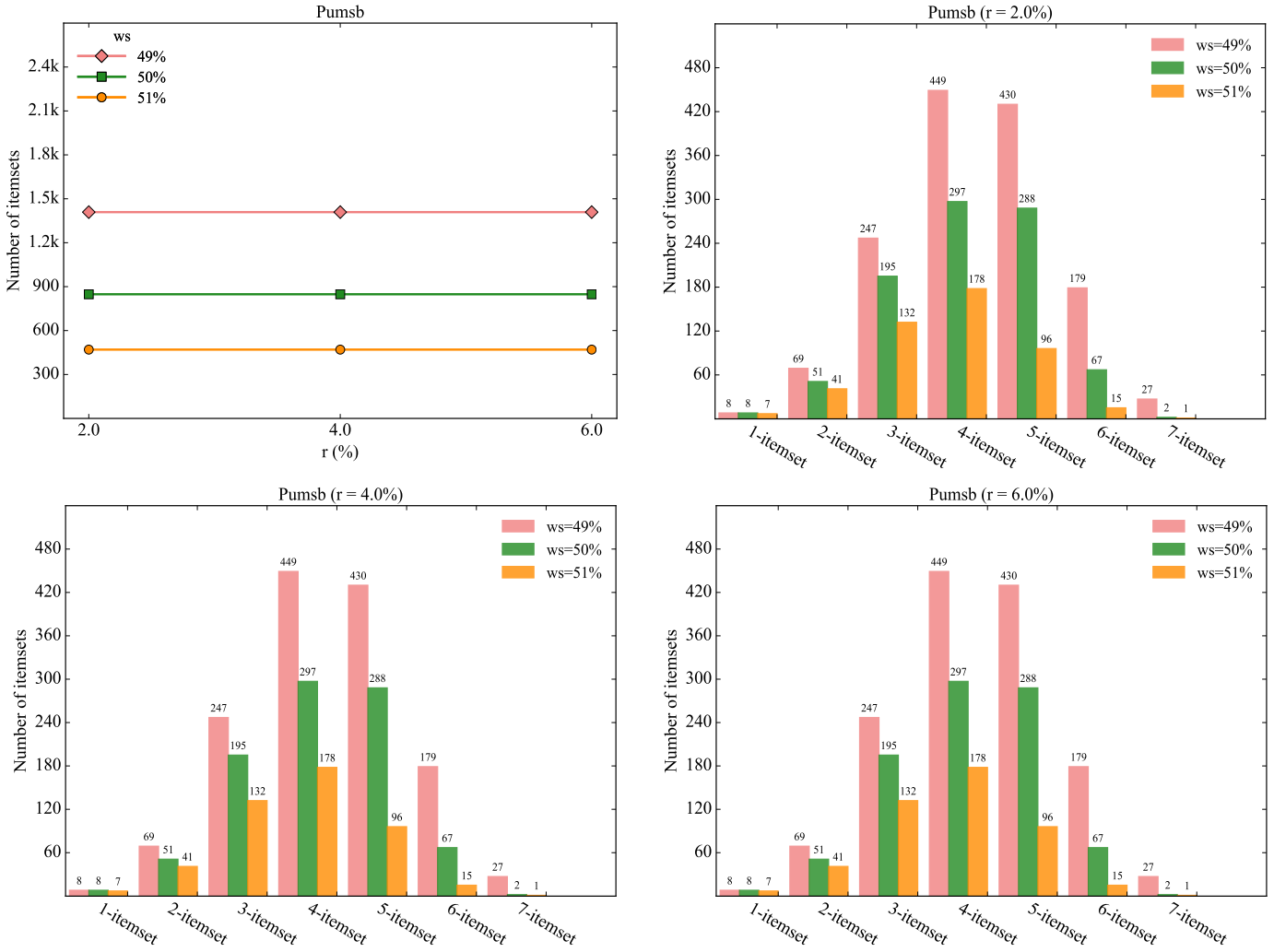
ภาพที่ 5-22 จำนวนเซตรายการที่เป็นผลลัพธ์ในฐานข้อมูล Chess



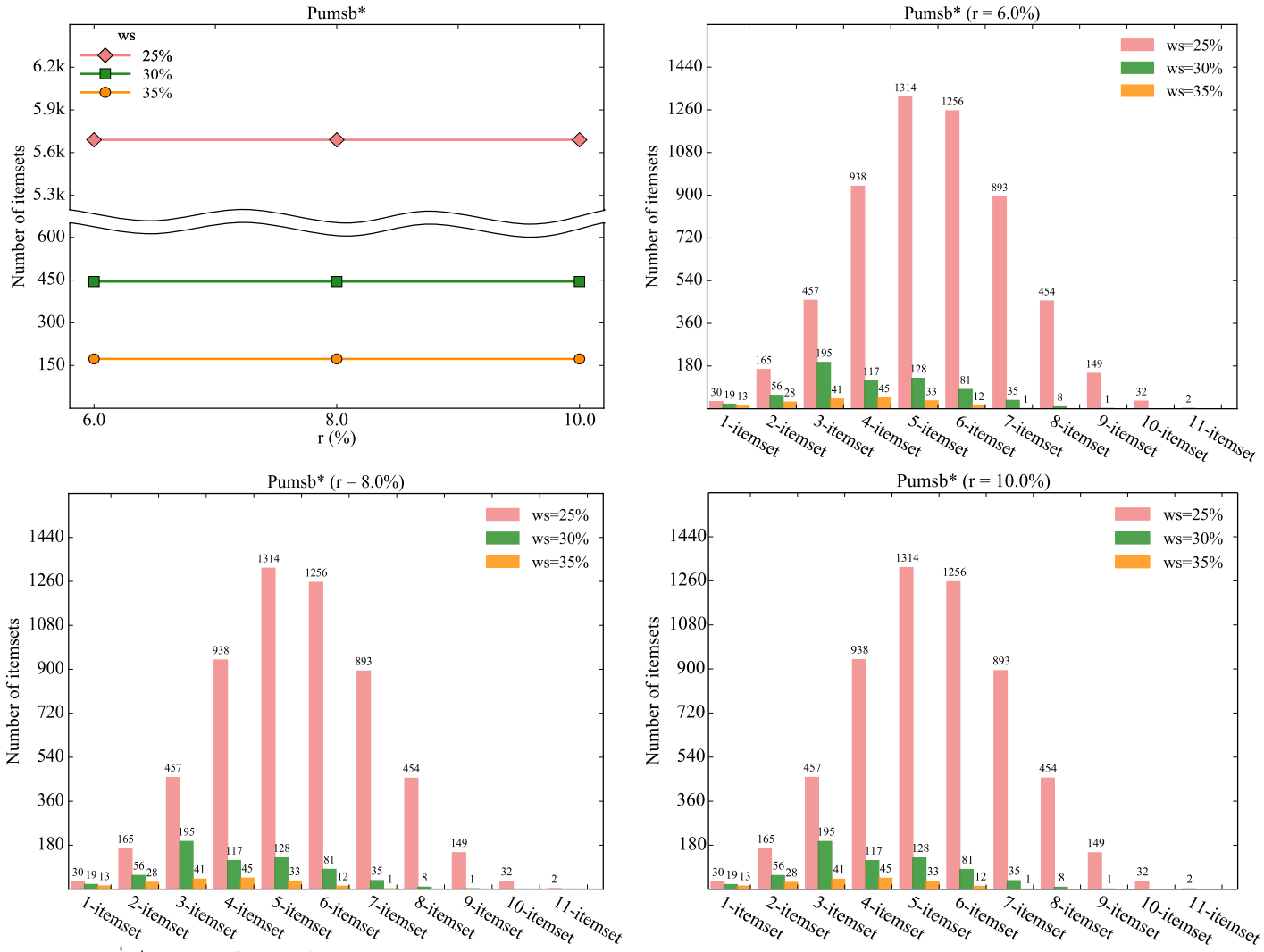
ภาพที่ 5-23 จำนวนเซตรายการที่เป็นผลลัพธ์ในฐานข้อมูล Connect



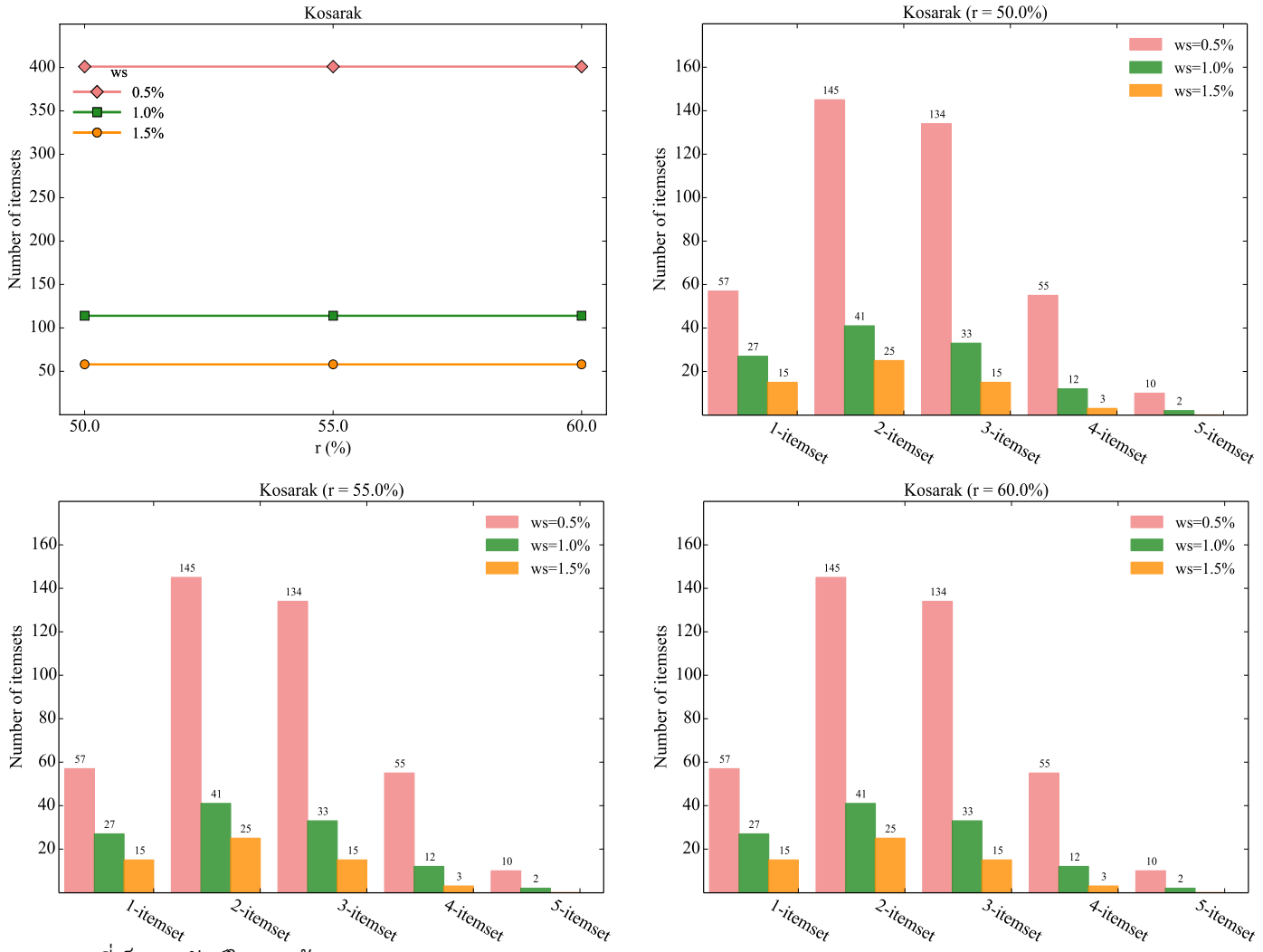
ภาพที่ 5-24 จำนวนเซตรายการที่เป็นผลลัพธ์ในฐานข้อมูล Mushroom



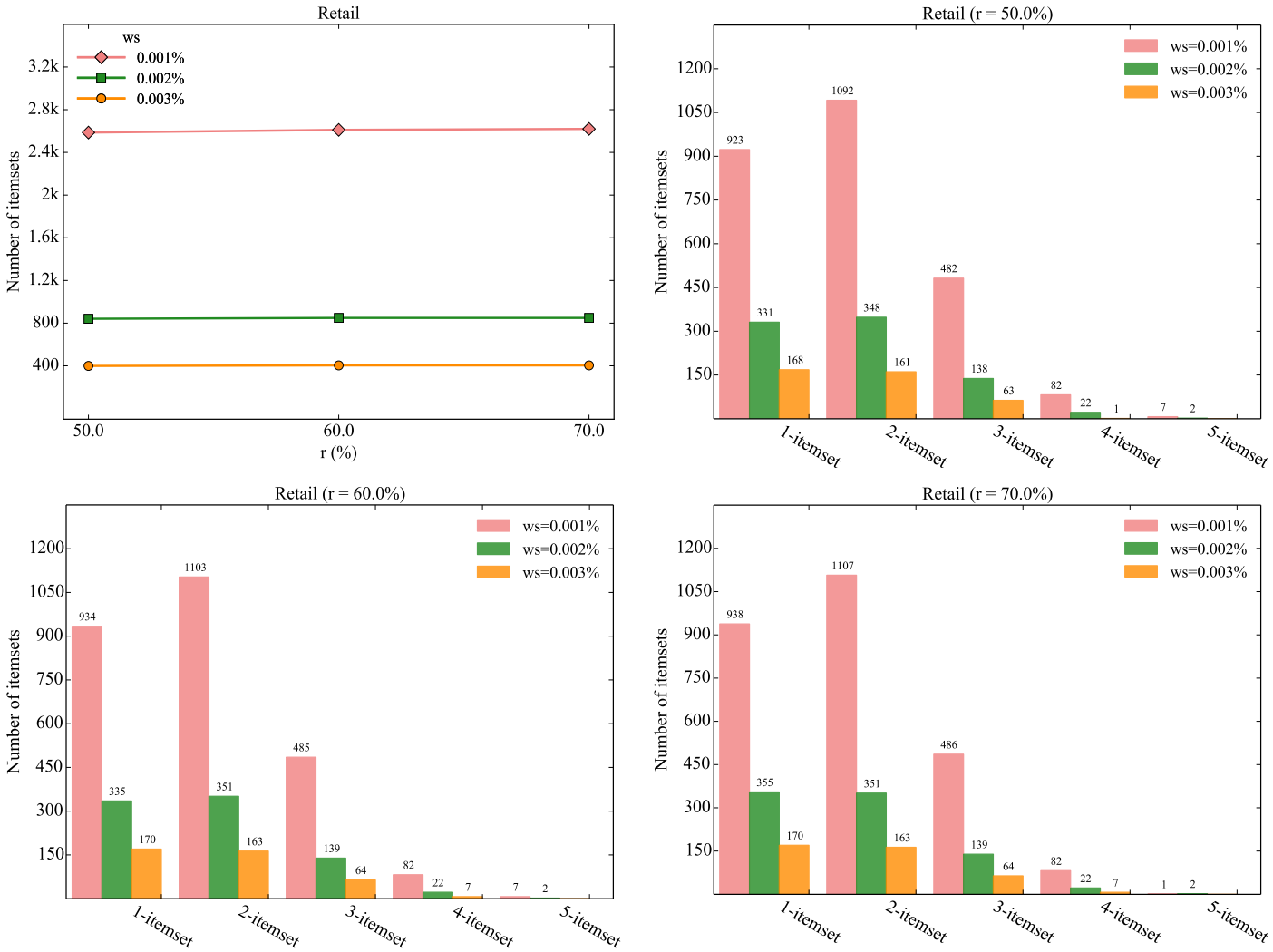
ภาพที่ 5-25 จำนวนเซตรายการที่เป็นผลลัพธ์ในฐานข้อมูล Pumsb



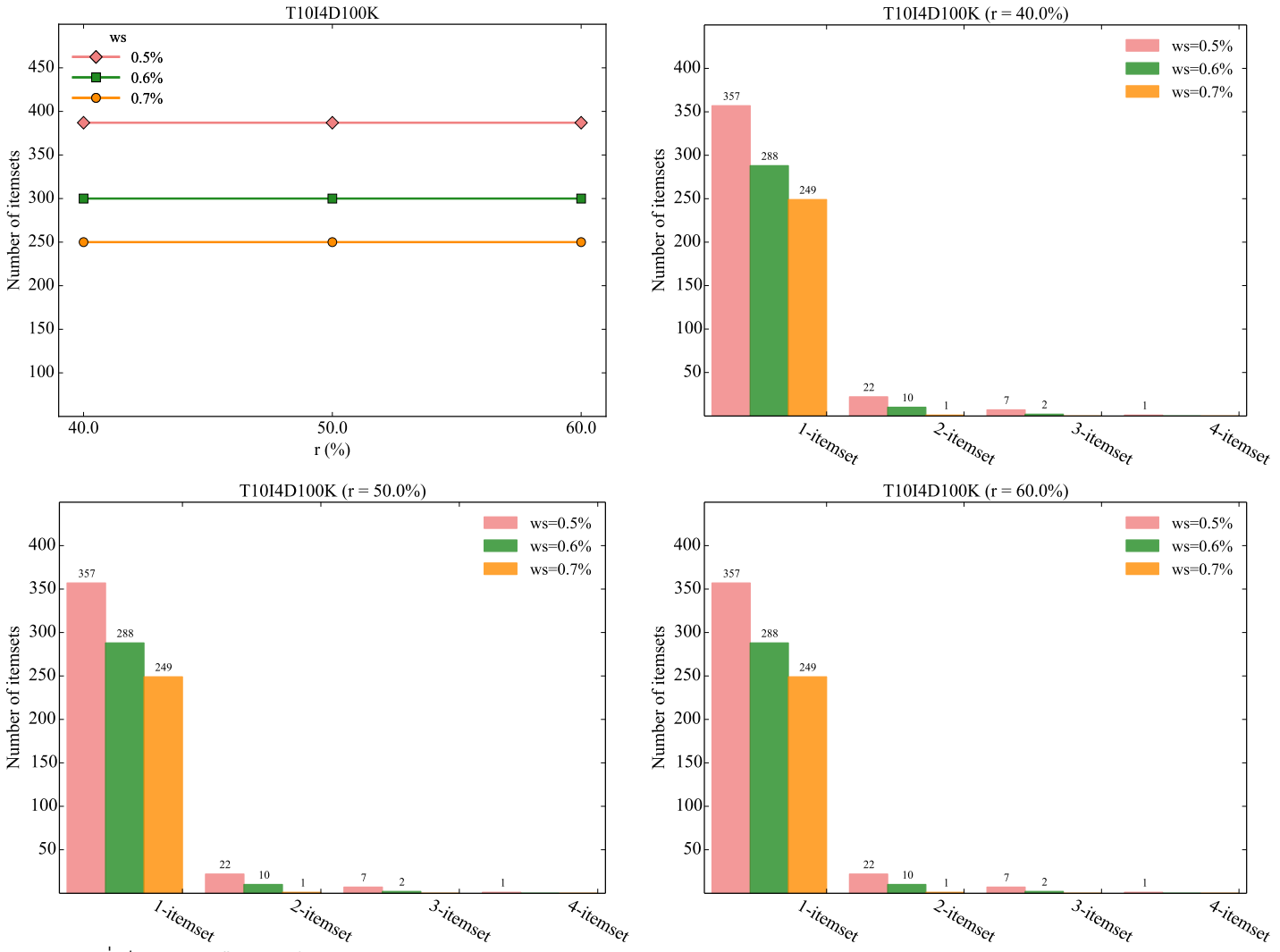
ภาพที่ 5-26 จำนวนเซตรายการที่เป็นผลลัพธ์ในฐานข้อมูล Pumsb*



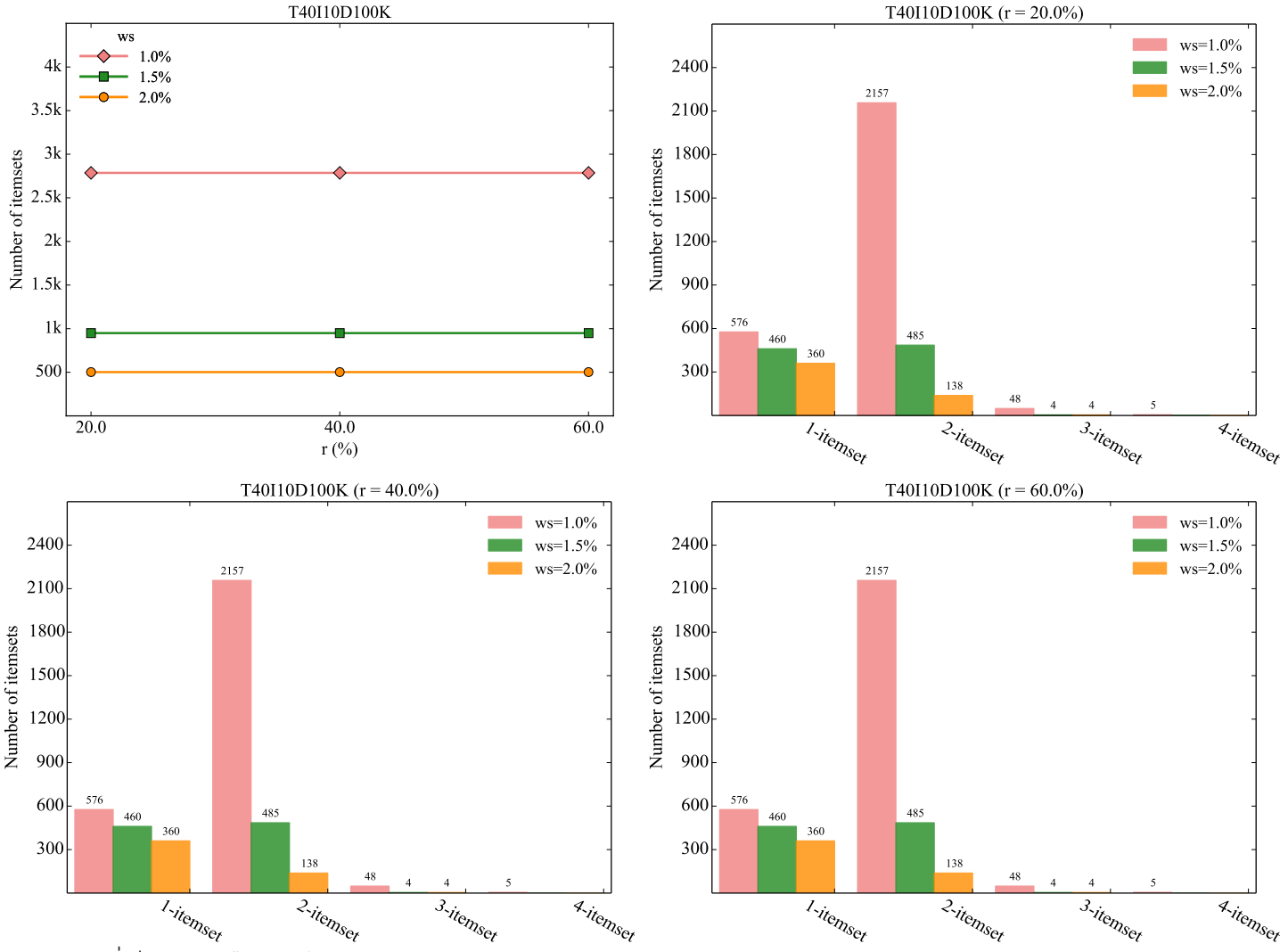
ภาพที่ 5-27 จำนวนเซตรายการที่เป็นผลลัพธ์ในฐานข้อมูล Kosarak



ภาพที่ 5-28 จำนวนเซตรายการที่เป็นผลลัพธ์ในฐานข้อมูล Retail



ภาพที่ 5-29 จำนวนเซตรายการที่เป็นผลลัพธ์ในฐานข้อมูล T10I4D100K



ภาพที่ 5-30 จำนวนเซตรายการที่เป็นผลลัพธ์ในฐานข้อมูล T40I10D100K

บทที่ 6
สรุปและอภิปรายผล

6.1 สรุปผลการดำเนินงาน

การค้นหาเซตรายการ/รูปแบบที่ปรากฏบ่อยและสม่ำเสมอ (Tanbeer et al., 2009) ได้ถูกนำเสนอขึ้นเพื่อค้นหาเซตรายการที่มีความน่าสนใจจากฐานข้อมูล ซึ่งวิธีการสำหรับการค้นหาเซตรายการดังกล่าวจะพิจารณาถึงพฤติกรรม/รูปแบบในการปรากฏของข้อมูล โดยพิจารณาความถี่หรือจำนวนครั้งและความสม่ำเสมอในการปรากฏขึ้นของข้อมูล จากแนวคิดพื้นฐานของการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอที่กล่าวมาข้างต้นนั้นจะทำการค้นหาเซตรายการโดยที่แต่ละรายการมีความสำคัญหรือความน่าสนใจเท่ากัน แต่สำหรับในการประยุกต์ใช้งานจริงหลาย ๆ แอป-พลิเคชันแต่ละรายการสามารถมีความสำคัญ/ความน่าสนใจที่แตกต่างกัน ด้วยเหตุนี้การค้นหาเซตรายการ/รูปแบบที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ (Cai, Fu, Cheng, & Kwong, 1998) (Vo & Coenen, 2013) จึงได้ถูกนำเสนอขึ้นในหลายงานวิจัย ซึ่งจะค้นหาเซตรายการโดยพิจารณาจำนวนครั้ง/ความถี่ในการปรากฏภายใต้เงื่อนไขที่แต่ละรายการมีความสำคัญ/ความน่าสนใจที่แตกต่างกัน

อย่างไรก็ตามการค้นหาเซตรายการ/รูปแบบที่ปรากฏบ่อยและสม่ำเสมอและการค้นหาเซตรายการ/รูปแบบที่ปรากฏบ่อยภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการที่มีอยู่นั้นอาจไม่เพียงพอหรือตอบสนองต่อความต้องการสำหรับการประยุกต์ใช้งานจริงในแง่มุมของการพิจารณาถึงลักษณะหรือพฤติกรรมในการปรากฏของเซตรายการที่น่าสนใจในฐานข้อมูลโดยที่แต่ละเซตรายการมีความสำคัญ/ความน่าสนใจที่แตกต่างกัน ดังนั้นจุดประสงค์หลักของวิทยานิพนธ์นี้จะมุ่งเน้นถึงปัญหาการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ ดังนี้ โดยนำเสนอปัญหาการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการจากฐานข้อมูล (Mining weighted-frequent-regular itemsets from transactional database, WFRIM) ซึ่งจะค้นหาเซตรายการที่มีลักษณะการปรากฏในฐานข้อมูลบ่อยและสม่ำเสมอโดยที่แต่ละรายการมีความสำคัญ/ความน่าสนใจที่แตกต่างกัน โดยขั้นตอนวิธีที่ใช้ในการค้นหาเซตรายการที่เป็นผลลัพธ์มีชื่อว่า Weighted-Frequent-Regular Itemsets Miner (WFRIM) ซึ่งใช้ WFRI-tree เป็นโครงสร้างในการจัดเก็บข้อมูลรวมถึงมีการประยุกต์ใช้แนวคิดการคำนวณหาค่าน้ำหนักที่มากสุด (Global maximum weight) และค่าน้ำหนักที่มากสุดของเซตรายการที่พิจารณา (Local maximum weight) เพื่อทำการลดทอนการพิจารณาเซตรายการที่ไม่สามารถเป็นผลลัพธ์ได้ อันนำมาซึ่งการลดทอนปริภูมิสถานะและ

เวลาในการประมวลผลได้ ต่อมานำเสนอขั้นตอนวิธีในการค้นหาเซตรายการปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการที่ได้พัฒนาประสิทธิภาพที่ดียิ่งขึ้นที่มีชื่อว่าวิธีการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการโดยใช้โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง (Weighted-Frequent-Regular Itemset Miner using Interval Word Segment structure, WFRIM-IWS) ซึ่งขั้นตอนวิธี WFRIM-IWS ประยุกต์ใช้โครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง (interval word segment) ในการจัดเก็บข้อมูลการปรากฏของรายการ/เซตรายการ และใช้ตารางค้นหา (Look-up table) เพื่อความรวดเร็วในการคำนวณค่าสนับสนุนและค่าความสม่ำเสมอจากโครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง โดยที่ใช้ WFRIM-Tree เป็นโครงสร้างในการจัดเก็บข้อมูลและการค้นหาแนวลึก (depth-first search strategy) สำหรับการค้นหาเซตรายการทั้งหมดที่เป็นผลลัพธ์ รวมถึงมีการประยุกต์ใช้แนวคิดการคำนวณหาค่าน้ำหนักที่มากสุด (Global maximum weight) และค่าน้ำหนักที่มากสุดของเซตรายการที่พิจารณา (Local maximum weight) เพื่อทำการลดทอนการพิจารณาเซตรายการที่ไม่สามารถเป็นผลลัพธ์ได้ อันนำมาซึ่งการลดทอนปริภูมิสถานะและเวลาในการประมวลผลได้

ในการทดสอบประสิทธิภาพของขั้นตอนวิธี WFRIM และ WFRIM-IWS ที่วิทยานิพนธ์นี้นำเสนอ ซึ่งได้ทำการทดสอบประสิทธิภาพของขั้นตอนวิธีที่นำเสนอใน 10 ข้อมูลโดยที่ฐานข้อมูลมีคุณลักษณะข้อมูลทั้งหนาแน่นและเบาบาง โดยมีการกำหนดค่าขีดแบ่งความสม่ำเสมอ (σ_r) และค่าขีดแบ่งน้ำหนักสนับสนุน (σ_{ws}) ซึ่งค่าขีดแบ่งทั้งสองกำหนดจากการพิจารณาคุณลักษณะของข้อมูลในแต่ละฐานข้อมูล ซึ่งจะแสดงผลการทดสอบประสิทธิภาพใน 3 แง่มุม 1) เวลาที่ใช้ในการประมวลผล 2) หน่วยความจำที่ใช้ในการประมวลผล 3) จำนวนเซตรายการที่เป็นผลลัพธ์

สำหรับผลการทดสอบประสิทธิภาพเชิงเวลา สามารถสรุปได้ว่าในฐานข้อมูลที่ข้อมูลมีคุณลักษณะหนาแน่น (Accidents Chess Connect Mushroom Pumsb และ Pumsb*) ขั้นตอนวิธี WFRIM-IWS ใช้เวลาในการประมวลผลที่ดีกว่าขั้นตอนวิธี WFRIM อยู่ 85.66%-99.69% และในฐานข้อมูลที่ข้อมูลมีคุณลักษณะเบาบาง (Kosarak Retail T10I4D100K และ T40I10D100K) ขั้นตอนวิธี WFRIM-IWS ใช้เวลาในการประมวลผลที่ดีกว่าขั้นตอนวิธี WFRIM อยู่ 7.56%-89.85%

สำหรับผลการทดสอบประสิทธิภาพเชิงหน่วยความจำ ฐานข้อมูลที่มีลักษณะหนาแน่น Chess และ Mushroom เมื่อมีการกำหนดค่าขีดแบ่งความสม่ำเสมอเพิ่มขึ้นจะส่งผลให้ใช้หน่วยความจำเพิ่มมากขึ้น เนื่องจากมีผลลัพธ์ของเซตรายการเพิ่มขึ้น และฐานข้อมูล Accidents Connect Pumsb และ Pumsb* จากผลการทดลองมีการใช้หน่วยความจำเท่าเดิมเมื่อมีการกำหนดค่าขีดแบ่งความสม่ำเสมอเพิ่มขึ้น เนื่องจากการเพิ่มขึ้นของการกำหนดค่าขีดแบ่งความสม่ำเสมอไม่ส่งผลต่อผลลัพธ์ของเซตรายการ สำหรับฐานข้อมูลที่มีลักษณะข้อมูลเบาบางฐานข้อมูล

Retail จะมีการใช้หน่วยความจำเพิ่มขึ้นเล็กน้อยเมื่อมีค่าขีดแบ่งความสม่ำเสมอเพิ่มขึ้นและฐานข้อมูล Kosarak T10I4D100K และ T40I10D100K การกำหนดค่าขีดแบ่งความสม่ำเสมอเพิ่มจะไม่ส่งผลต่อการใช้หน่วยความจำ สำหรับการกำหนดค่าขีดแบ่งน้ำหนักสนับสนุนนั้นเมื่อค่าขีดแบ่งน้ำหนักสนับสนุนน้อยลงจะส่งผลต่อการใช้หน่วยความจำที่เพิ่มมากขึ้น เนื่องจากผลลัพธ์ของเซตรายการจะมีจำนวนที่เพิ่มมากขึ้น

จำนวนเซตรายการที่เป็นผลลัพธ์ ในฐานข้อมูลที่มีคุณลักษณะข้อมูลหนาแน่น Chess และ Mushroom จะมีจำนวนผลลัพธ์ที่เพิ่มมากขึ้นเมื่อค่าขีดแบ่งความสม่ำเสมอเพิ่มขึ้น และฐานข้อมูล Accidents Connect Pumsb และ Pumsb* จะมีจำนวนผลลัพธ์ที่ไม่เปลี่ยนแปลงเมื่อค่าขีดแบ่งความสม่ำเสมอเพิ่มขึ้น สำหรับฐานข้อมูลที่มีลักษณะเบาบาง ดังเช่นฐานข้อมูล Retail จะมีจำนวนผลลัพธ์ที่เพิ่มมากขึ้นเมื่อค่าขีดแบ่งความสม่ำเสมอเพิ่มขึ้น แต่สำหรับฐานข้อมูล Kosarak T10I4D100K และ T40I10D100K มีจำนวนผลลัพธ์ที่ไม่เปลี่ยนแปลงเมื่อค่าขีดแบ่งความสม่ำเสมอเพิ่มขึ้น

6.2 อภิปรายผลการดำเนินงาน

ในการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการนั้นจากการทดสอบประสิทธิภาพจะเห็นได้ว่าขั้นตอนวิธี WFRIM-IWS มีเวลาที่ใช้ในการประมวลผลเร็วกว่าขั้นตอนวิธี WFRIM อย่างเห็นได้ชัดเจนในทุกฐานเพราะขั้นตอนวิธี WFRIM จะใช้เวลาในการท่อง WFRI-Tree เพื่อให้ได้ซึ่งเซตของหมายเลขทรานแซกชันที่มีการเรียงลำดับโดยที่เซตของหมายเลขทรานแซกชันแสดงถึงทรานแซกชันที่เซตรายการนั้น ๆ ปรากฏ โดยเฉพาะในฐานข้อมูลที่มีคุณลักษณะข้อมูลหนาแน่นโหนดหนึ่ง ๆ จะจัดเก็บหมายเลขทรานแซกชันจำนวนมากทำส่งผลให้การเรียงลำดับหมายเลขทรานแซกชันใช้เวลานาน และขั้นตอนวิธี WFRIM-IWS มีจำนวนของกลุ่มเวิร์ดน้อยเวลาที่ใช้ในอินเตอร์เซกชันจึงรวดเร็ว แต่ฐานข้อมูล Retail ทั้งสองขั้นตอนมีเวลาที่ใช้ในการประมวลผลที่ใกล้เคียงกันอันเนื่องมาจากข้อมูลมีคุณลักษณะข้อมูลเบาบางส่งผลให้แต่ละโหนดจัดเก็บหมายเลขทรานแซกชันน้อยทำให้การเรียงลำดับหมายเลขทรานแซกชันใช้เวลาเร็ว แต่ในทางกลับกันขั้นตอนวิธี WFRIM-IWS ในการประมวลผลสำหรับฐานข้อมูล Retail จะมีการสร้างโครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วงที่ประกอบไปด้วยหลายกลุ่มเวิร์ดทำให้ขั้นตอนในการอินเตอร์เซกชันใช้เวลาเยอะ ในส่วนของหน่วยความจำที่ใช้ในการประมวลผลที่ขั้นตอนวิธี WFRIM-IWS ใช้หน่วยความจำที่น้อยกว่าขั้นตอนวิธี WFRIM เพราะมีการบีบอัดหมายเลขทรานแซกชันให้อยู่ในรูปแบบเวิร์ดโดยที่ 1 เวิร์ดมีขนาด 16 บิตซึ่งแสดงถึง 16 ทรานแซกชันดังนั้นในการจัดเก็บหมายเลขทรานแซกชัน 16 ทรานแซกชันจึงใช้ 2 ไบต์

6.3 ข้อเสนอแนะ

1. ในขั้นตอนการประมาณค่าน้ำหนักสนับสนุนโดยการใช้เทคนิคการคำนวณหาค่าน้ำหนักที่มากสุด (Global maximum weight) และค่าน้ำหนักที่มากสุดของเซตรายการที่พิจารณา (Local maximum weight) นั้นบางครั้งค่าน้ำหนักสนับสนุนที่ได้รับจากการประมาณค่าอาจจะมีค่าน้ำหนักสนับสนุนมากเกินไปจึงมีสร้าง *WFRI-tree* ในขั้นตอน WFRIM และอินเตอร์เซกชันในขั้นตอน WFRIM-IWS เซตรายการที่ไม่เป็นผลลัพธ์ส่งผลให้ใช้เวลาในการประมวลผลที่มากขึ้น
2. ขั้นตอนวิธี WFRIM นั้นได้ทำการค้นหาผลลัพธ์เซตรายการขนาดต่าง ๆ บนโครงสร้างต้นไม้ที่เรียกว่า *WFRI-tree* ซึ่งจำเป็นต้องทำการเรียงลำดับทรานแซกชันในแต่ละโหนดเพื่อทำการค้นหาเซตรายการผลลัพธ์ที่มีขนาดใหญ่ขึ้น ถ้าทรานแซกชันมีจำนวนมากจะส่งผลต่อการใช้เวลาในการประมวลผลที่เพิ่มขึ้นด้วย
3. ขั้นตอนวิธี WFRIM-IWS ได้ใช้เทคนิคการอินเตอร์เซกชันของโครงสร้างแบ่งกลุ่มเวิร์ดเป็นช่วง (IWS) ในการคำนวณค่าความสม่ำเสมอและค่าสนับสนุนของเซตรายการที่ปรากฏร่วมกัน ในกรณีที่ฐานข้อมูลมีลักษณะเบาบางการสร้าง IWS ของแต่ละรายการ/เซตรายการจะมีจำนวนของกลุ่มเวิร์ดที่มากส่งผลต่อเวลาที่ใช้เพิ่มมากขึ้นในเทคนิคการอินเตอร์เซกชัน

บรรณานุกรม

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *ACM SIGMOD Conference*, 22(2), 207-206.

Agrawal, R., & Srikant, R. (1994) Fast algorithms for mining associationrules in large databases. *VLDB*, 487–499.

Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. *In: Proceedings of the IEEE International Conference on Data Engineering*, 3–14.

Ahmed, C. F., Tanbeer, S. K., Jeong, B.-S., Lee, Y.-K., Ho, T.-B., & Zhou, Z.-H., (2008). Mining Weighted Frequent Patterns in Incremental. *Databases, PRICAI 2008: Trends in Artificial Intelligence, Springer Berlin Heidelberg*, 933-938.

Amphawan, K., Lenca, P., & Surarerks, A. (2009). Mining top-k periodic-frequent patterns without support threshold. *In Proceedings of the 3rd international conference on advances in information technology*, *Communications in computer and information science*, 55, 18–29.

Amphawan, K., Lenca, P., & Surarerks, A. (2012). Mining top-k regular-frequent itemsets using database partitioning and support estimation. *Expert Systems with Applications*, 39(2), 1924–1936.

Cai, C. H., Fu, A. W. C., Cheng, C. H., & Kwong, W. W. (1998). Mining association rules with weighted items. *In Proceedings of Intl. Database Engineering and Applications Symposium (IDEAS 1998)*, 68–77.

Cong, G., Tung, A. K. H., Xu, X., Pan, F., & Yang J. (2004). FARMER: finding interesting rule groups in microarray datasets. *In Proceedings of the 2004 ACM SIGMOD international conference on Management of data (SIGMOD '04*), 143-154.

Fournier-Viger, P., Lin, J. C. W., Duong, Q. H., & Dam, T. L. (2016). PHM: Mining Periodic High-Utility Itemsets, *Springer International Publishing*, 64–79.

Giannella, C., Han, J., Pei, J., Yan, X., & Yu, P. S. (2003) Mining Frequent Patterns in Data Streams at Multiple Time Granularities, *In H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha (eds.), Next Generation Data Mining*.

Grahne, G. & Zhu, J. (2005). Fast algorithms for frequent itemset mining using FP-trees.

In IEEE Transactions on Knowledge and Data Engineering, 17, 10, 1347-1362.

Kiran, R. U., & Kitsuregawa, M. (2014). Novel techniques to reduce search space in periodic-frequent pattern mining. *In Proceedings of the 19th international conference on database systems for advanced applications*, 8422, 377–391.

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. *In Proceedings of the 2000 ACM SIGMOD international conference on Management of data (SIGMOD '00)*. ACM, 1-12.

Lan, G. C., Hong, T. P., & Lee, H. Y., (2014). An efficient approach for finding weighted sequential patterns from sequence databases. *Appl. Intell.* 41 (2), 439–452.

Lin, C.-H., Chiu, D.-Y., Wu, Y.-H., & Chen, A.L.P. (2005). Mining frequent itemsets from data streams with a time-sensitive sliding window, *In Proc. SIAM International Conference on Data Mining*.

Li, H.-F., Huang, H.-Y., Chen, Y.-C., Liu, Y.-J., & Lee, S.-Y. (2008). Fast and Memory Efficient Mining of High Utility Itemsets in Data Streams. *In Proc. of the 8th IEEE Int'l Conf. on Data Mining,* 881-886.

Lin, M.-Y., Lee, P.-Y., & Hsueh, S.-C. (2012). Apriori-based frequent itemset mining algorithms on MapReduce. *In Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication (ICUIMC '12)*, 76 , 8.

Malerba, D., Esposito, F., & Lisi, F.A. (2001). Mining spatial association rules in census data. *In Proceedings of Joint Conf. on New Techniques and Technologies for Statistcs and Exchange of Technology and Know-how*.

Nguyen, H., Vo, B., Nguyen, M., & Pedrycz, W. (2016). An efficient algorithm for mining frequent weighted itemsets using interval word segments, *Applied Intelligence*, 45, 4, 1008–1020.

Park, J. S., Chen, M. S., & Yu, P. S. (1995). An effective hash-based algorithm for mining association rules. *SIGMOD Rec.*, 24(2), 175–186.

Savasere, A., Omiecinski, E., & Navathe, S. (1995). An efficient algorithm for mining association rules in large databases. *In Proceeding of the 1995 international conference on very large data bases (VLDB'95),* 432–443.

Serban, G., Czibula, I. G., & Campan, A. (2006). A Programming Interface For Medical diagnosis Prediction, *Studia Universitatis*, *Babes-Bolyai*, *Informatica, LI(1)*,21-30.

Shenoy, P., Haritsa, J. R., Sudarshan, S., Bhalotia, G., Bawa, M., & Shah, D. (2000). Turbo-charging vertical mining of large databases. *In Proceedings of the 2000 ACM SIGMOD international conference on Management of data (SIGMOD '00)*, 22-33.

Sun, K., & Bai, B.a.i., (2008). Mining weighted association rules without preassigned weights. *IEEE Trans*. *Knowl. Data Eng*. 20 (4), 489–495.

Tanbeer, S. K., Ahmed, C. F., Jeong, B.-S., & Lee, Y. K. (2009). Discovering periodic-frequent patterns in transactional databases. *In Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, 242–253.

Tanbeer, S. K., Ahmed, C. F., & Jeong, B.-S. (2010a). Mining regular patterns in incremental transactional databases. In *Proceedings of the 12th International Asia-Pacific Web Conference, Buscan, Korea*. 375–377.

Tanbeer, S. K., Ahmed, C. F., & Jeong, B.-S. (2010b). Mining regular patterns in data streams. *In Proceedings of the 15th International Conference on Database Systems for Advanced Applications, Tsukuba, Japan, ser. Lecture Notes in Computer Science*. 399–413.

Tao, F. (2003). Weighted association rule mining using weighted support and significant framework. *9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 661–666.

Rashid, M., Karim, M., Jeong, B. -S., Choi, H .-J. (2012). Efficient mining regularly frequent patterns in transactional databases. *In Proceedings of the 17th international conference on database systems for advanced applications*, 7238, 258–271.

Vo, B., & Coenen, F., Le, B., (2013). A new method for mining frequent weighted itemsets based on wit-trees. *Expert Syst*. 40 (4), 1256–1264.

Wang, W., Yang, J., & Yu, P. (2004). War: Weighted association rules for item intensities. *Knowledge and Information Systems*, 6(2), 203–229.

Xia, D., Zhou, Y., Rong, Z., & Zhang, Z. (2013). Ipfp: An improved parallel fp-growth algorithm for frequent itemsetsmining, *In Proc. 59th ISI World Statistics Congress*, 4034-4039.

Yan, X., Han, J., & Afshar, R. (2003). CloSpan: Mining Closed Sequential Patterns in Large Datasets, *Proc. 2003 SIAM Int'l Conf. Data Mining (SDM '03)*, 166-177.

Yun, U., & Leggett, J. J. (2005). Wfim: Weighted frequent itemset mining with a weight range and a minimum weight. *in Proceedings of the 2005 SIAM International Conference on Data Mining*, 637– 640.

Yun, U., & Leggett, J., (2006). WSpan: weighted sequential pattern mining in large sequential database. *In Proceedings of IEEE International Conference on Intelligent Systems*, 512–517.

Zaki, M. J. (2000). Scalable algorithms for association mining. *In IEEE Transactions on Knowledge and Data Engineering*, 12, 3, 372-390, 2000.

Zaki, M. J. & Gouda, K. (2003). Fast vertical mining using diffsets. *In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '03)*, 326-335.

ภาคผนวก

ภาคผนวก ก

เอกสารรับรองผลการพิจารณาจริยธรรมการวิจัยในมนุษย์

ที่ ๕๐/๒๕๖๐

เอกสารรับรองผลการพิจารณาจริยธรรมการวิจัยในมนุษย์
มหาวิทยาลัยบูรพา

คณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์ มหาวิทยาลัยบูรพา ได้พิจารณาโครงการวิจัย

รหัสโครงการวิจัย Sci 014/2560
โครงการวิจัยเรื่อง การค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอภายใต้การกำหนดค่าน้ำหนักความสำคัญของแต่ละรายการ
หัวหน้าโครงการวิจัย นางสาวกิตติพา คลังวิสาร
หน่วยงานที่สังกัด นิสิตระดับบัณฑิตศึกษา คณะวิทยาการสารสนเทศ

คณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์ มหาวิทยาลัยบูรพา ได้พิจารณาแล้วเห็นว่าโครงการวิจัยดังกล่าวเป็นไปตามหลักการของจริยธรรมการวิจัยในมนุษย์ โดยที่ผู้วิจัยเคารพสิทธิและศักดิ์ศรีในความเป็นมนุษย์ ไม่มีการล่วงละเมิดสิทธิ สวัสดิภาพ และไม่ก่อให้เกิดภยันตรายแก่ตัวอย่างการวิจัยและผู้เข้าร่วมโครงการวิจัย

จึงเห็นสมควรให้ดำเนินการวิจัยในขอบข่ายของโครงการวิจัยที่เสนอได้ (ดูตามเอกสารตรวจสอบ)

๑. เอกสารโครงการวิจัยฉบับภาษาไทย ฉบับที่ ๑ วันที่ ๒๒ เดือน มีนาคม พ.ศ. ๒๕๖๐
๒. เอกสารชี้แจงผู้เข้าร่วมโครงการวิจัย ฉบับที่ - วันที่ - เดือน - พ.ศ. -
๓. เอกสารแบบแสดงความยินยอมของผู้เข้าร่วมโครงการวิจัย
ฉบับที่ - วันที่ - เดือน - พ.ศ. -
๔. เอกสารแสดงรายละเอียดเครื่องมือที่ใช้ในการวิจัยซึ่งผ่านการพิจารณาจากผู้ทรงคุณวุฒิแล้ว หรือชุดที่ใช้เก็บข้อมูลจริงจากผู้เข้าร่วมโครงการวิจัย ฉบับที่ - วันที่ - เดือน - พ.ศ. -

การรับรองผลการพิจารณาจริยธรรมการวิจัยในมนุษย์ฉบับนี้ มีผลถึงวันที่ ๑ เดือน เมษายน พ.ศ. ๒๕๖๑

ออกให้ ณ วันที่ ๒๒ เดือน มีนาคม พ.ศ. ๒๕๖๐

ลงนาม

(ผู้ช่วยศาสตราจารย์ ดร.วิทวัส แจ้งเอี่ยม)
ประธานคณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์
มหาวิทยาลัยบูรพา

ภาคผนวก ข

เอกสารเผยแพร่งานวิจัย

KST Research Center
Knowledge Smart Technologies
Since 2008

BURAPHA UNIVERSITY

February 1-4, 2017

@Amari Ocean Pattaya,
Chon Buri, Thailand

The 2017-9th International Conference on Knowledge and Smart Technology

"Crunching Information of Everything"

Organized by Faculty of Informatics,
Burapha University, Chon Buri, Thailand

ISBN 978-1-4673-9077-4

IEEE THAILAND SECTION
IEEE
IEEE Xplore® Digital Library
Technical Co-Sponsor
ECTI Association
SIPA
AIAT

Mining weighted-frequent-regular itemsets from transactional database

Kittipa Klangwisan*, Komate Amphawan†
Computational Innovation Laboratory, Faculty of Informatics, Burapha University, Chonburi, 20131, Thailand
Email: *kklangwisan@gmail.com, †komate@gmail.com

***Abstract*—Frequent-regular itemsets mining has been explored and proposed to find interesting itemsets based on their own occurrence behavior. Traditionally, an itemset is identified as interesting, if it occurs frequently and regularly in a database. However, this task only considers items without defining difference or significance of each item which may affect the missing of important/interesting knowledge in real-world applications. To address this issue, we introduce an approach on mining weighted-frequent-regular itemsets, (also called mining *WFRI*s). To mine *WFRI*s, a tree-and-pattern growth based algorithm called *WFRIM* (*Weighted-Frequent-Regular Itemsets Miner*) is proposed. An FP-tree like structure named *WFRI-tree* is designed to efficiently maintain candidate itemsets during mining process. The concept of *overestimated-weighted-frequency* of items/itemsets under global/local maximum weight is also applied to early prune search space. Experimental results on synthetic and real datasets show efficiency of *WFRIM* in the terms of computational time, memory consumption and capability to find valuable itemsets.**

***Keywords*—data mining; association rules; itemsets mining; frequent-regular itemsets; weight of importance**

I. Introduction

Mining interesting itemsets/patterns from databases plays an important role in many real-life applications (such as retail business, DNA analysis, mobile commerce, elder behavior analysis, financial analysis, telecommunication industry, etc). These itemsets can help to aid managers and/or decision makers to create efficient strategic plans and strategies. The first proposal on mining interesting itemsets is frequent itemsets mining (*FIM*) [1] which focuses on discovering itemsets with high frequency of occurrence. From [1], *FIM* is extended into several aspects such as *FIM* on incremental database/data streams, frequent-closed itemsets mining, maximal frequent itemset mining, top-k *FIM*, and so on. Recently, frequent-regular itemsets mining (*FRIM*) is one of the most interesting approaches on mining interesting itemsets (patterns). The main objective of *FRIM* is to discover itemsets that frequently and regularly occur in database (meet user-given frequency and regularity thresholds). Since the proposed in [2], *FRIM* is extended and improved into several aspects such as *FRIM* on incremental database or data stream [3], [4], top-k *FRIM*/top-k closed *FRIM* [5], [6], high utility-regular itemsets mining [7], [8], and so on.

Traditional *FIM* and *FRIM* assume that items have the same significance without taking into account of their weight of importance. However, in real-world applications, items can have different degree of importance (also called weight of importance). For example, the itemset "wine, salmon" generated from a retail business might be more important than the itemset "bread, milk", if they have the same support or the former has a lower support. The reason is that the first itemset usually give more profit and/or cost than the latter one, but the standard *FIM* and *FRIM* simply ignore this difference. To address this issue on *FIM*, Wei Wang et al. [9] proposed an approach for mining of weighted association rules (*WAR*) by allowing a weight to be associated with each item to react interest/intensity of each item. Since 2004, there are several extensions of *WAR* such as mining *WAR* using weighted support, weight range, length-decreasing support and over data streams ([10], [11], [12], [13], [14], [15], [16]). However, the existent approaches only focus on the term frequency which may not sufficient to find interesting itemsets under various aspects of occurrence behavior.

From the above issue, we propose to discover frequent-regular itemsets with user-given weights of importance on items. A problem of weighted-frequent-regular itemsets mining (also called *WFRI*s mining) is thus defined. To mine *WFRI*s, an efficient tree-based algorithm named *WFRIM* (*Weigthed-Frequent-Regular Itemsets Miner*) and a FP-tree like structure called *WFRI-tree* are designed and introduced. In addition, the concepts of global/local maximum weight and an overestimated-weighted frequency of an item/itemsets [10] are applied to prune search space. Experiments on synthetic and real datasets were conducted to investigate performance of our proposed *WFRIM* in the terms of computational time, memory usage and number of discovered itemsets.

II. Problem statement

In this section, we first describe basic notations on regularity of itemsets as in [2], [5]. Then, a weighted-frequency of an itemset is defined in the same manner as [14]. Last, we introduce definitions and the problem of mining frequent-regular itemsets based on inequality of items' importance.

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a unique set of items in which each item $i_p \in I$ has a non-negative real number w_{i_p} expressing its weight of importance. A set $X = \{i_p, \ldots, i_q\}$, where $1 \leq p \leq q \leq n$, is called an itemset, a pattern or a k-itemset if X contains k items. A transaction database $TDB = \{t_1, t_2, \ldots, t_m\}$ over I is a set of m transactions (*i.e.* $|TDB| = m$) where each transaction t_j has j as its unique transaction-identifier or time-stamp (called *tid* for short) and contains a set of items Y. If $X \subseteq Y$, it can be said that X occurs in transaction t_j or t_j contains X. Last, an ordered set of transactions containing X can be defined as

978-1-4673-9077-4/17/$31.00 ©2017 IEEE

$T^X = \{j, k, \ldots, l\}$ where j, k and $l \in [1, m]$ expressing a *tid* of transaction containing X.

A. Regularity of an itemset and a frequent-regular itemset

The regularity of an itemset X (denoted as r^X) can be defined as the maximum gap of consecutive transactions that X appears at least once. For example, if $r^X = 4$, it can be concluded that X occurs at least once in every four consecutive transactions.

The regularity of X on the transaction t_k containing X ($r^X_{t_k}$) can be calculated in the following three cases: (*i*) if t_k is the first transaction containing X, $r^X_{t_k}$ is set equal to k which is the first gap of consecutive transactions, start from 0 to k, containing X, (*ii*) if t_k is ordered after another transaction t_j containing X and there is no transaction t_y where $j < y < k$, $r^X_{t_k} = k - j$ expresses the gap of consecutive transactions between t_j and t_k containing X, and (*iii*) if t_k is the last transaction containing X, $r^X_{t_k} = m - k$ (where m is the total number of transaction in database) indicates the gap of consecutive transactions that X absences from database since its last occurrence in transaction t_k, respectively.

From the three cases above, the regularity of X in transaction database *TDB* can be computed as $r^X = max(r^X_{t_j}, r^X_{t_k}, \ldots, r^X_{t_l}, r^X_{t_l})$, where $j, k, l \in [1, m]$ are in the ascending order (Notice there are two times of $r^X_{t_l}$ following case (*ii*) and (*iii*)).

An itemset X is called a frequent-regular itemset if (*i*) its support $s^X = |T^X|$ is no less than the user-given support threshold σ_s and (*ii*) its regularity r^X calculated as above is no greater than the user-specified regularity threshold σ_r, respectively.

B. Weighted-frequency of a frequent-weighted itemset

With the set of items $I = \{i_1, i_2, \ldots, i_n\}$ and a corresponding set of weights $W = \{w_1, w_2, \ldots, w_n\}$, a weighted-frequency of an itemset $X \subseteq I$ is the production of the average weight of all items in X (denoted as $w^X = \frac{\sum_{i_p \in X} w^{i_p}}{|X|}$) and the support s^X of X which can be defined as $wf^X = w^X \times s^X$. Based on the concept of weighted-frequency, an itemset X is called a weighted-frequent itemset, if its weighted-frequency wf^X is no less than the user-specified weighted-support threshold σ_{ws}.

To early prune search space, Tao et al. [10] proposed the concepts of global maximual weight (denoted as *GMAXW* $= max(w^{i_1}, \ldots, w^{i_n})$), local maximum weight (defined as *LMAXW* $= \frac{\sum_{i_y \in Y} w^{i_y} + max(w^{i_q}, \ldots, w^{i_p})}{|Y| + 1}$ where Y is an itemset from previous considered iterations and $w^{i_p} \ldots, w^{i_q} \notin X$) and an overestimated-weight-frequency of an itemset (calculated as $owf^X =$ *GMAXW* $\times s^X$ or $owf^X =$ *LMAXW* $\times s^X$), respectively. With these concepts, an itemset X is identified as a *weighted-infrequent itemset* if its overestimated-weight-frequency owf^X is less than the user-given weighted-frequency threshold. Then, the itemset X and all of its supersets can be eliminated from consideration, since they cannot give high weighted-frequency and be interesting itemsets.

tid	item
1	a, b, c, d
2	c, e, f
3	a, b, e, f, g
4	a, b, c, f, g
5	d, e, g
6	a, b, c, e, g
7	a, b, c, e
8	a, b, d, e
9	b, c, e
10	a, e, g

A transactional database

item	weight
a	0.6
b	0.5
c	0.35
d	0.45
e	0.45
f	0.3
g	0.4

Weight Table

Fig. 1: An example of retail database

C. Mining frequent-regular itemsets under inequality of items' importance

From the concept of regularity and weighted-frequency, we can define the problem of mining frequent-regular itemsets based on user-given items' weights of importance as the task of mining a complete set of itemsets that having weighted-frequency no less than the user-given weight-frequency threshold, and having regularity no greater than the user-specified regularity threshold.

III. Proposed method

In this section, we here introduce an efficient algorithm named *weighted-frequent-regular itemsets miner* (*WFRIM*) used for mining frequent-regular itemsets with user-assigned weights of importance on items. A FP-tree-like structure, called *WFRI-tree* is utilized for maintaining candidate itemsets during mining process. The concept of global and local maximum weights is applied to hold *downward closure property* [1] which can help to early prune search space. *WFRIM* has two main steps *i.e.* 1) *WFRIM-initialization*—scanning database to create a header table and *WFRI-tree*, and 2) *WFRIM-growth*—finding a complete set of weighted-frequent-regular itemsets based on pattern growth concept.

A. WFRI-tree and header table

WFRI-tree is a FP-tree-like structure linked with a simple list of single items also called *header table*. As shown in Fig. 2, each entry of a header table is for an item which contains 5 information: *i*) item-name, i , *ii*) support of i, s^i, *iii*) regularity of i, r^i, *iv*) important weight of i, w^i, and *v*) a horizontal link to nodes in *WFRI-tree* with item i, l^i, respectively. Meanwhile, *WFRI-tree* contains paths in which each path indicates an itemset occurring in transactions. In each path, there are two types of nodes *i.e. i*) *internal node* containing item-name, a link to its parent and links to its children, and *ii*) *leaf node* storing the same information as *internal node* with a list of *tid*s that the itemset occurs. For example, *WFRI-tree* in Fig. 3 contains only one path of an itemset occurring in transaction t_1 of Fig. 1. Nodes in the path are sorted by ascending order of item's weight which causes the nodes of items 'c' and 'b' are internal nodes and the node of item 'a' is a leaf node with *tid* 1 of t_1. Each node in *WFRI-tree* is also linked with an entry (with the same item name) in the header table.

i	s^i	r^i	w^i
a	1	1	0.6
b	1	1	0.5
c	1	1	0.35
d	1	1	0.45
e	0	0	0.45
f	0	0	0.3
g	0	0	0.4

Header table H after scanning t_1

i	s^i	r^i	w^i
a	1	1	0.6
b	1	1	0.5
c	2	1	0.35
d	1	1	0.45
e	1	2	0.45
f	1	2	0.3
g	0	0	0.4

Header table H after scanning t_2

i	s^i	r^i	w^i
~~f~~	~~3~~	~~6~~	~~0.3~~
c	6	2	0.35
g	5	4	0.4
~~d~~	~~3~~	~~4~~	~~0.45~~
e	8	2	0.45
b	7	2	0.5
a	7	2	0.6

Header table H after scanning t_{10}

— eliminated by σ_r - - eliminated by σ_{ws}

Fig. 2: *WFRIM-initialization*

B. WFRIM algorithm

1) WFRIM-initialization (see Algo. 1): Given a transactional database with weights of importance on items (as in Fig. 1), a weight support threshold σ_{ws} be 2.0 and a regularity threshold σ_r be 4, respectively. *WFRIM-initialization* is to discover wighted-frequent-regular items and to create *WFRI-tree* for further mining. A header table H is firstly created and initialized to capture essential information of single items. The input database is scanned twice. From the first scanning (line 2-5), each transaction t_j is sequentially read and each item $i_p \in t_j$ is considered to update its support s^{i_p} and regularity r^{i_p}. For example of Fig. 2, the scanning of transaction $t_1 = \{a, b, c, d\}$ results in updating of supports and regularities of items 'a', 'b', 'c' and 'd' to be 1. For the transaction $t_2 = \{c, e, f\}$, the support s^c of item 'c' is updated to be 2, the supports of items 'e' and 'f' are initialized to be 1 and the regularities of items 'e' and 'f' are set to be 2 (Noted 'e' and 'f' firstly occur in t_2, then the current maximum regularity is 2). The reading process is repeated for all transactions in order to update supports and regularities of items. After scanning all transactions, regularity of each item is considered and item 'f' is observed that it has regularity greater than the regularity threshold. Then, item 'f' is removed from H (*i.e.* item 'f' is identified as an irregular item which cannot be or be a part of *WFRI*s). Next, all items in H are ordered by ascending order of their weights and the global maximum weight is then calculated as $GMAXW = max(w^{i_1}, w^{i_2}, \ldots, w^{i_{|H|}}) = max(0.35, 0.4, 0.45, 0.45, 0.5, 0.6) = 0.6$. Then, an overestimated weight frequency of each item $i_p \in H$, $owf^{i_p} = GMAXW \times s^{i_p}$, is computed as $owf^c = 3.6, owf^g = 3.0, owf^d = 1.8, owf^e = 4.8, owf^b = 4.2$, and $owf^a = 4.2$, respectively. Then, item 'd' is eliminated from the header table with since its $owf^d = 1.8$ is less than the weight-support threshold. *WFRIM-initialization* then recursively calucates global maximum weight and a weight frequency of each item (line 10-16), if all items having weights equal to *GMAXW* are removed out of H (Notice this recursive process can help to quickly prune search space which can help to save computational time). Last, items 'c','g','e','b', and 'a' are then identified as a weighted-frequent-regular item and collected as a result in *WFRI*s, since their weight-frequencies (actual) are not less than the weight-frequency threshold.

For the second scanning (line 20-31), the root node R of *WFRI-tree* is firstly created and initialized. Each transaction t_j is sequentially scanned and each item $i_p \in t_j$ is then considered and eliminated from t_j, if there is no existence of i_p's entry in the header table H. Next, all remaining items in t_j are sorted by the order of H. Last, a path of items in

Algorithm 1: WFRIM-initialization

Input: TDB, σ_r, σ_{ws}
Output: *WFRI-Tree*, *WFRIs*

1: create a header-table H with an entry for each item $i_p \in I$
2: **for** each transaction t_j in TDB **do**
3: **for** each item i_p in transaction t_j **do**
4: add support s^{i_p} in the entry of i_p of H by 1
5: calculate regularity r^{i_p} in the entry of i_p of H by t_j
6: **for** each item i_p in H **do**
7: **if** $r^i > \sigma_r$ **then**
8: remove the entry of i_p out of H
9: sort all items in H by ascending order of their weights
10: **repeat**
11: calculate global maximum weight of all items in H, $GMAXW = max(w^{i_1}, w^{i_2}, \ldots, w^{i_{|H|}})$
12: **for** each item i_p in H **do**
13: calculate overestimated weighted-frequency of i_p, $owf^{i_p} = GMAXW \times s^{i_p}$
14: **if** $owf^{i_p} < \sigma_{ws}$ **then**
15: remove the entry of i_p out of H
16: **until** all items with weight equal to $GMAXW$ are not removed from H
17: **for** each item i_p in H **do**
18: calculate weigthed-frequency $wf^{i_p} = w^{i_p} \times s^{i_p}$
19: $WFRIs \leftarrow WFRIs \cup i_p$ **if** $wf^{i_p} \geq \sigma_{ws}$
20: create and initial *WFRI-tree* with a root node R
21: **for** each transaction t_j in TDB **do**
22: remove item $i_p \in t_j$ such that $i_p \notin H$
23: sort t_j as the order of H
24: $temp \leftarrow R$
25: **for** each item i_p in transaction t_j **do**
26: **if** *temp* does not have a child node with i_p **then**
27: create a new node Z for i_p, set Z as a child node of *temp*, and link Z with *node-link* of i_p in a header-table
28: $temp \leftarrow Z$
29: **else**
30: $temp \leftarrow$ the child node of *temp* with i_p
31: collect *tid* j in T^{i_p} of *temp* Z, $T^{i_p} \leftarrow T^{i_p} \cup j$

t_j is created with a *tid* j at the leaf node (Notice if there exists a path of the itemset similar with t_j, a new path is not created but a list of *tid*s at the leaf node of the existence path is updated by j). For example, item 'd' is firstly eliminated from transaction t_1 and the remaining items are ordered as 'c', 'b' and then 'a', respectively. Then, a path of 'c, b, a' is created in *WFRI-tree* with tid 1 at the node of 'a' (as shown in Fig. 3). For transaction $t_2 = \{c, e, f\}$, item 'f' is removed and its order is 'c', and then 'e', respectively. A path of 'c, e' is thus created in *WFRI-tree* where the node of 'e' contains tid 2. After scanning all of database, we gain *WFRI-tree* as shown at the bottom of Fig. 3.

2) WFRIM-growth (see Algo. 2): To mine a complete set of *WFRI*s from *WFRIM-growth*, a pattern growth is recursively applied on *WFRI-tree*. First, itemset X is defined as a set of previous considered items which is set to be $\emptyset$ at the beginning. *WFRI-tree* is investigated that whether it contains only one path (also called single path). If the tree contains only single path of itemset P, each subset of P is considered and its weighted frequency is then calculated to identify *WFRI*s. Otherwise, each item in the header table H (starting from the bottom one, as in the example is 'a') is firstly considered and a new header table H^a is created to maintain single items occurring with 'a'. The link from H to nodes of 'a' in the *WFRI-tree* is then regarded. For each node of 'a' in the link, its ancestors are visited in order to collect their occurrence information (used for calculating its regularity) and to compute its support. For example, as shown in Fig. 4, the node of 'a' (the leftmost one with *tid* 1) occurring with items 'c' and 'b' is considered. Then, *tid* 1 is then collected in tidlist (*i.e.* list of tids) of entries of 'c' and 'b' of H^a and supports s^c and s^b are then initialized

Algorithm 2: WFRIM-growth

Input: *WFRI-Tree*,σ_r,σ_{ws}
Output: *WFRIs*

1: $X \leftarrow$ set of items considered from previous iterations (at beginning $X \leftarrow \emptyset$)
2: call *WFRI-growth* (*WFRI-tree*, $X = \emptyset$, σr, σ_{ws})
3: **Procedure** *WFRI-growth* (*WFRI-tree with* H, X, σ_r, σ_{ws})
4: **if** *WFRI-tree* contains only one path P **then**
5: **for** each sub-itemset Y of path P **do**
6: calculate weighted-frequency $wf^Y = \frac{\sum_{i_k \in Y} w^{i_k}}{|Y|} \times s^Y$ of Y
7: $WFRIs \leftarrow WFRIs \cup i_k$ **if** $wf^Y \geq \sigma_{ws}$
8: **else**
9: **for** each item i_p in the header-table H (in bottom-up manner) **do**
10: create a new header-table g to store all items in H except item i_p
11: **for** each node n^{i_p} linked with *node-link* of item i_p in the header-table H **do**
12: $Y \leftarrow$ the set of items in the same path with n^{i_p}
13: **for** each item $i_q \in Y$ **do**
14: update T^{i_q} of entry i_q of H^{i_p} by T^{i_p} of n_{i_p}, $T^{i_q} \leftarrow T^{i_q} \cup T^{i_p}$
15: **for** each item i_q in H^{i_p} **do**
16: compute r^{i_q} from T^{i_q} of entry i_q of H^{i_p}
17: remove the entry of i_q out of H^{i_p} **if** $r^{i_q} > \sigma_r$
18: **repeat**
19: calculate local maximum weighted as $LMAXW = \frac{\sum_{i_j \in X} w^{i_j} + w^{i_p} + max(w^{i_y}, \ldots, w^{i_z})}{|X| + 2}$ where $i_y, \ldots, i_z \in H^{i_p}$
20: **for** each item i_q in H^{i_p} **do**
21: calculate overestimated weighted-frequency $owf^{i_q} = LMAXW \times s^{i_q}$
22: remove entry i_q out of H^{i_p} **if** $owf^{i_q} < \sigma_{ws}$
23: **until** all items with weight of importance equal to $LMAXW$ are not removed from H^{i_p}
24: **for** each item i_q in H^{i_p} **do**
25: calculate weight-frequency $wf^{i_q} = \frac{\sum_{i_j \in X} w^{i_j} + w^{i_p} + w^{i_q}}{|X| + 2} \times s^{i_q}$
26: $WFRIs \leftarrow WFRIs \cup (X \cup i_p \cup i_q)$ **if** $wf^{i_q} \geq \sigma_{ws}$
27: create and initial a new *WFRI-tree* with Z as root
28: **for** each node n^{i_p} linked with *node-link* of item i_p in the header-table H **do**
29: $temp \leftarrow Z$
30: $Y \leftarrow$ the set of items in the same path with node n^{i_p} in which each item $i_q \in Y$ has an entry in H^{i_p}
31: **for** each item $i_q \in Y$ **do**
32: **if** *temp* does not have a child node with i_q **then**
33: create a new node X for i_q, set X to be a child node of *temp*, and link X with *node-link* of i_q of H^{i_p}
34: $temp \leftarrow X$
35: **else**
36: $temp \leftarrow$ the child node of X with item i_q
37: update T^{i_q} of *temp* by T^{i_p} of n^{i_p}, $T^{i_q} \leftarrow T^{i_q} \cup T^{i_p}$
38: $U \leftarrow$ the parent node of n^{i_p}
39: update T^U of U by T^{i_p} of n^{i_p}, $T^U \leftarrow T^U \cup T^{i_p}$
40: unlink n^{i_p} from *node-link* and remove node n^{i_p} out of *WFRI-tree*
41: call *WFRI-growth* (*new WFRI-tree with* H^{i_p}, $X \cup i_p$, σ_r, σ_{ws})
42: remove entry of i_p out of H

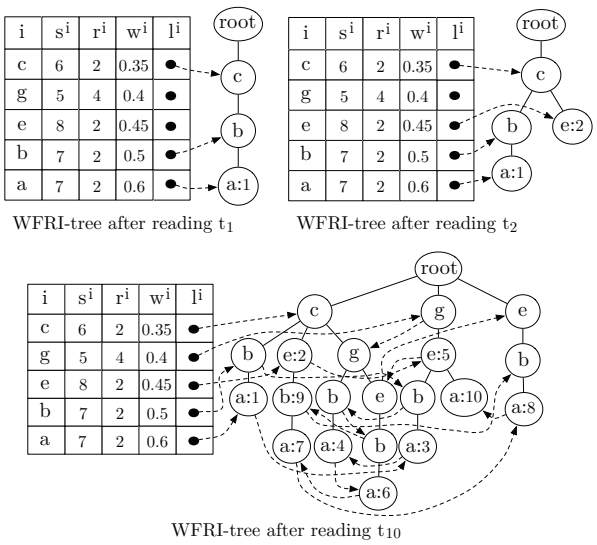


Fig. 3: *WFRI-tree* created from a transactional database

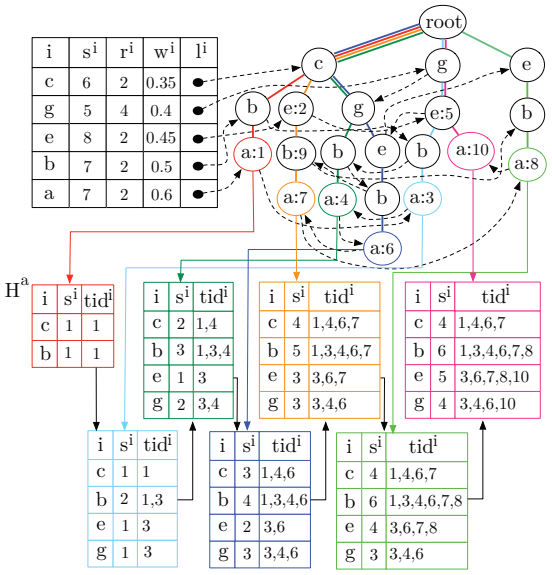


Fig. 4: The updated H^a

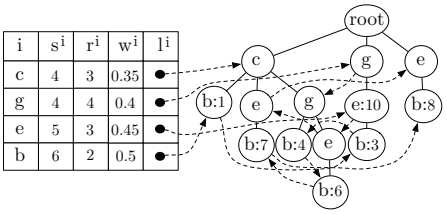


Fig. 5: *WFRI-tree*^{a} having 'a' as prefix

by 1. Next, the node of 'a' with *tid* 3 occurring with 'g', 'e' and 'b' is then considered. The tidlists of 'g', 'e' and 'b' in H^a are thus updated as $T^g = \{3\}$, $T^e = \{3\}$ and $T^b = \{1, 3\}$. Meanwhile, their supports are updated as $s^g = 1$, $s^e = 1$ and $s^b = 2$, respectively. After investigating all nodes of 'a', we gain the up-to-date header table H^a.

To prune search space, items with regularity greater than the user-defined regularity threshold σ_r are removed from H^a (from downward closure property in [2], these items and their supersets are irregular itemsets). The local maximum weight is thus computed as in line 20. Then, the overestimated-weight frequency of each item in H^a is thus calculated. For each item $i_j \in H^a$, if its overestimated-weight frequency $owf^{i_j} = LMAXW \times |T^{i_j}|$ is less than σ_{ws}, the entry of i_j is then removed from H^a (based on downward closure defined in [1], the itemset $X \cup a \cup i_j$ and its superset cannot have high weighted-frequency). The calculation of *LMAXW* and owf^{i_j} of each item i_j are then repeated, if all items with maximum weight of importance are removed from H^a. Next, each item $i_j \in H^a$ is considered again. Its weighted-frequency is then calculated as in line 26 . If wf^{i_j} is not less than σ_{ws}, the itemset $X \cup i_p \cup i_j$ is then identified and collected in set *WFRI*s.

Next, a root node of a new *WFRI-tree* related with items ocurring with 'a' is then initialized (denoted as *WFRI-tree*^{a}).

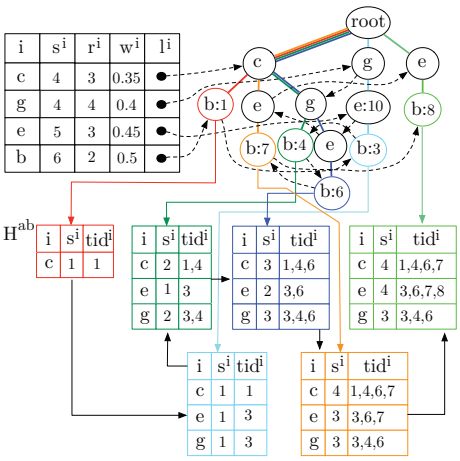


Fig. 6: The updated H^{ab}

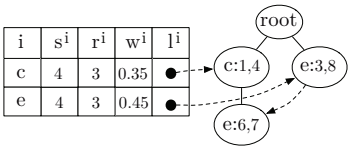


Fig. 7: *WFRI-tree*^{ab} having 'ab' as prefix

Each node of 'a' (with its ancestors) in the *WFRI-tree* is traversed again. Ancestor items do not having entries in H^a are removed from consideration and then a path for remaining ancestor items is created in the *WFRI-tree*^{a} in which a tidlist of node 'a' is added to the tidlist of the leaf node. A tidlist of node 'a' in the *WFRI-tree* is moved and merged with its parent and the node 'a' is removed from *WFRI-tree*. After completing the second traversal of all nodes of 'a' in *WFRI-tree*, we gain the *WFRI-tree*^{a} containing itemsets occurring with 'a' (as shown in Fig. 5). As in Fig. 6, *WFRIM-growth* is then recursively consider a new *WFRI-tree*^{a} with a prefix itemset $X = X \cup a$. It then creates the header table H^{ab} and *WFRI-tree*^{ab} to consider longer *WFRIs* (as shown in Fig. 7). This process is repeated until the current considered *WFRI-tree*^{X} contains only single path or all items in the current considered header table H^{i_p} are considered. At the end of *WFRIM-growth*, the set *WFRI*s contains a complete set of weighted-frequent-regular itemsets.

IV. Experimental Results

In this section, we investigate performance of *WFRIM* by using 3 bench mark datasets as detailed in Table I. *T10I4D100K* is a synthetic sparse dataset generated by IBM almaden generator. Meanwhile, mushroom and chess are real dense datasets downloaded from http://fimi.ua.ac.be/data/. Since weights of importance of items are not provided in these datasets, we have random them in the same manner as [9], [10], [12], [17] ranging from 0.1 to 0.9. Three experiments were designed and conducted to observe computational time, memory usage and number of discovered itemsets under the same parameters setting as [2], [10]. The proposed *WFRIM* is implemented on Python and run on a macbook pro with OS X El-Capitan, CPU speed at 2.4 GHz, RAM 8 GB. Based on the best of our knowledge, there is no train of taught to mine frequent-regular itemset based on consideration of items' weights of importance. Then, there is no comparative study in this paper. However, we then show the experiment in two aspects *i.e.* with and without assigning weights of importance on the datasets (labeled with weighted and unweighted, respectively).

TABLE I: Characteristics of datasets

Dataset	No. of items	Avg. transactions size	No. of transactions	category
Mushroom	119	23	8,124	dense
Chess	75	37	3,196	dense
T10I4D100K	1,000	10	100,000	sparse

As shown in Fig. 8, runtimes of *WFRIM* on the three datasets are shown. With the variation of regularity threshold (but a weighted-frequency threshold is fixed), *WFRIM* uses more computational time as the threshold increase. The reason is that with high regularity threshold, items/itemsets have more chance to be regular itemsets, then *WFRIM* have to take more time to consider items/itemsets with high regularity. Meanwhile, the variation on weighted-frequency threshold drastically causes increasing/decreasing on computational time. With the high weighted-frequency threshold, there is a large amount of itemsets prunned by the threshold. Then, computational time were reduced (thanks to global and local maximum weight and overestimated-weight-frequency of itemsets). In Fig. 9, the memory usage of *WFRIM* on the three datasets are illustrated. *WFRIM* consumes memory as similar as it computational time. With the increasing of regularity threshold, the number itemsets *WFRIM* have to maintain also increase. This causes the increasing of memory usage. Similarly, with the decreasing of weighted-frequency threshold, *WFRIM* consumes more memory to maintain more itemsets during mining process. Last, as shown in Fig. 10, the number of discovered itemsets is also observed. With the range of thresholds in this experiments, *WFRIM* can discover appropriate set of interesting itemsets which can let users doing further analysis.

V. Conclusion

In this paper, we have introduced an approach on mining frequent-regular itemsets under user-given items' weights of importance. The problem of weighted-frequent-regular itemsets mining (also called *WFRIs mining*) is thus defined. To mine *WFRIs*, an efficient algorithm named *WFRIM* (Weighted-Frequent-Regular Itemsets Miner) and a FP-like structure called *WFRI-tree* are then designed. The concepts of global/local maximum weight and overestimated-weight frequency of itemsets are applied to early prune search space. Experiments on synthetic and real datasets show that *WFIM* is efficient in the terms of computational time and memory usage and can discover valuable itemsets for further analysis.

References

[1] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *VLDB*, 1994, pp. 487–499.

[2] S. K. Tanbeer, C. F. Ahmed, B.-S. Jeong, and Y.-K. Lee, "Discovering periodic-frequent patterns in transactional databases," in *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, 2009, pp. 242–253.

[3] S. K. Tanbeer, C. F. Ahmed, and B.-S. Jeong, "Mining regular patterns in incremental transactional databases," in *Proceedings of the 12th International Asia-Pacific Web Conference, April 6-8, Buscan, Korea*, 2010, pp. 375–377.

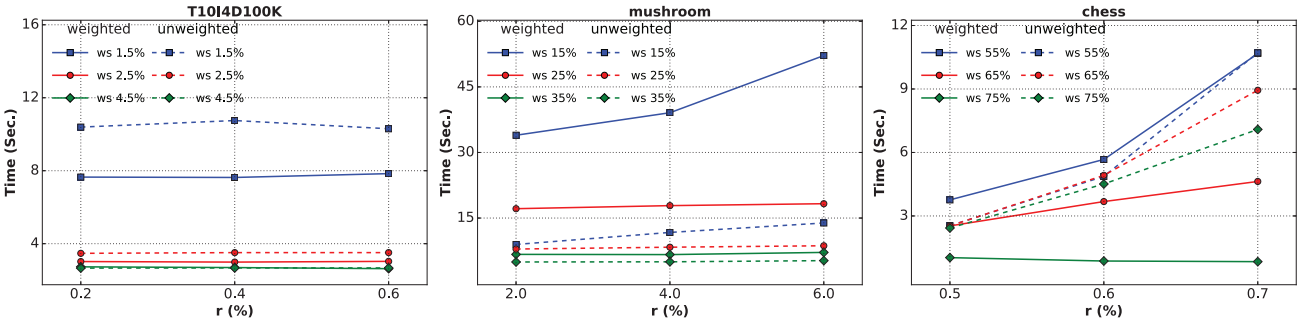


Fig. 8: Computational time of *WFRIM*.

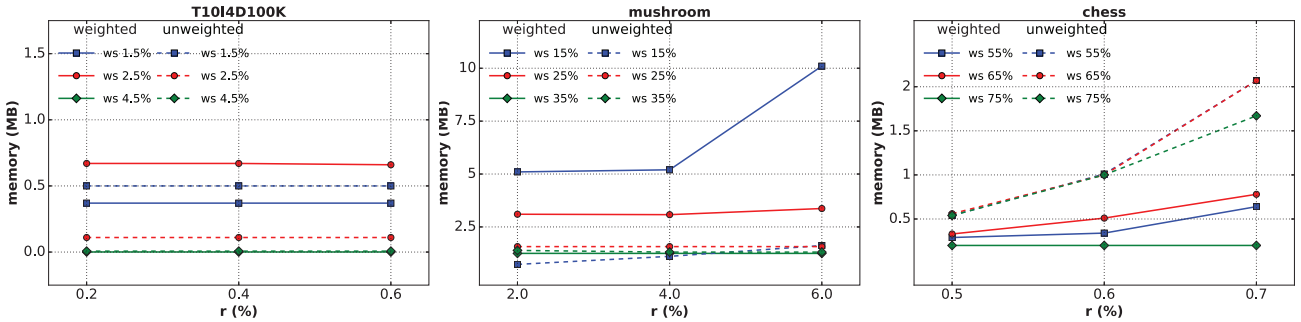


Fig. 9: Memory usage of *WFRIM*.

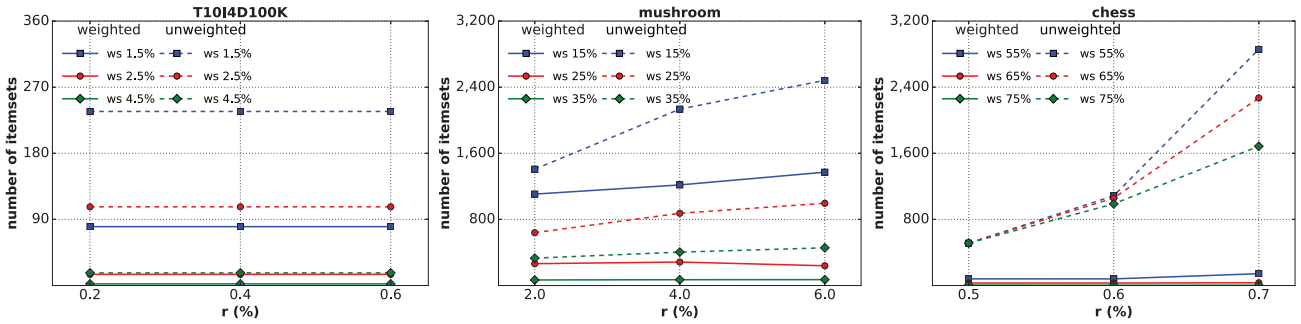


Fig. 10: Number of itemsets discovered from of *WFRIM*.

[4] ——, "Mining regular patterns in data streams," in *Proceedings of the 15th International Conference on Database Systems for Advanced Applications, April 1-4, Tsukuba, Japan*, ser. Lecture Notes in Computer Science, vol. 5981, 2010, pp. 399–413.

[5] K. Amphawan, P. Lenca, and A. Surarerks, "Mining top-k periodic-frequent patterns without support threshold," in *Proceedings of the 3rd International Conference on Advances in Information Technology*, vol. 55, 2009, pp. 18–29.

[6] K. Amphawan and P. Lenca, "Mining top-k frequent-regular closed patterns," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7882 – 7894, 2015.

[7] P. Fournier-Viger, J. C.-W. Lin, Q.-H. Duong, and T.-L. Dam, *PHM: Mining Periodic High-Utility Itemsets*, 2016, pp. 64–79.

[8] K. Amphawan, P. Lenca, A. Jitpattanakul, and A. Surarerks, "Mining high utility itemsets with regular occurrence," *Journal of ICT Research and Applications*, vol. 10, no. 2, pp. 153 – 176, 2016.

[9] W. Wang, J. Yang, and P. Yu, "War: Weighted association rules for item intensities," *Knowledge and Information Systems*, vol. 6, no. 2, pp. 203–229, 2004.

[10] F. Tao, F. Murtagh, and M. Farid, "Weighted association rule mining using weighted support and significance framework," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '03, 2003, pp. 661–666.

[11] U. Yun and J. J. Leggett, *WLPMiner: Weighted Frequent Pattern Mining with Length-Decreasing Support Constraints*, 2005, pp. 555–567.

[12] C. F. Ahmed, S. K. Tanbeer, B.-S. Jeong, and Y.-K. Lee, *Mining Weighted Frequent Patterns in Incremental Databases*, 2008, pp. 933–938.

[13] C. F. Ahmed, S. K. Tanbeer, and B. S. Jeong, "Efficient mining of weighted frequent patterns over data streams," in *High Performance Computing and Communications, 2009. HPCC '09. 11th IEEE International Conference on*, June 2009, pp. 400–406.

[14] C. F. Ahmed, S. K. Tanbeer, B.-S. Jeong, Y.-K. Lee, and H.-J. Choi, "Single-pass incremental and interactive mining for weighted frequent patterns," *Expert Systems with Applications*, vol. 39, no. 9, pp. 7976–7994, 2012.

[15] B. Vo, F. Coenen, and B. Le, "A new method for mining frequent weighted itemsets based on wit-trees," *Expert Systems with Applications*, vol. 40, no. 4, pp. 1256 – 1264, 2013.

[16] L. Cagliero and P. Garza, "Infrequent weighted itemset mining using frequent pattern growth," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 4, pp. 903–915, April 2014.

[17] U. Yun and J. J. Leggett, "Wfim: Weighted frequent itemset mining with a weight range and a minimum weight," in *Proceedings of the 2005 SIAM International Conference on Data Mining*, 2005, pp. 637–640.

Conference Program

the 2017 - 9th International Conference on Knowledge and Smart Technology (KST)

Amari Pattaya Hotel, Chon Buri Province, Thailand
Organized by KST Research Lab, Faculty of Informatics, Burapha University, Chonburi, Thailand
February 1-4, 2017

February 1, 2017	**15.00-17.00**	Registration					
	17.00-18.00	Steering & Technical Program Committee Meeting					
	18:00-20:30	Welcome Reception for Steering & Technical Program Committee and Keynote Speaker					
February 2, 2017	**08.00**	Registration					
	08.45	**Opening Ceremony / Aranda meeting room**					
	09.00	**Keynote-I**	Professor Dr. Katsumi Watanabe (Faculty of Science and Engineering, Waseda University, Tokyo, Japan)				
			Explicit and Implicit Aspects of Human Cognition and Behavior				
	09.50	**Keynote-II**	Professor Dr. Thomas Mandl (Information Science, University of Hildesheim, Germany)				
			Text Mining, Patent Retrieval and its Evaluation: Finding paths in the Labyrinth between Legal and Technical Challenges				
	10.40-11.00	Break					
		PaperID	**Catleya meeting room Regular: Computational Intelligence Chair: Dr.Sunisa Rimcharoen (BUU, Thailand)**	**PaperID**	**Rimsuan 2 meeting room Regular: Emerging Intelligent Technologies & Intelligent applications Chair: Dr. Paramate Horkaew (SUT, Thailand)**	**PaperID**	**Mokara meeting room Regular: Intelligent Computer Networks and Systems Chair: Dr.Nutthanon Leelatrakul (BUU, Thailand)**
	11.00	3	Distinguishing ACL Patients from Healthy Individuals using Multilayer Perceptron on Motion Patterns; By Worapan Kusakunniran, Nantawat Prachasri, Nattaporn Dirakbussarakom and Duangkamol Yangchaem	40	Towards an open monitoring platform for improving energy efficiency and thermal comfort in public buildings; By Antony Harfield and Waraporn Rattanongphisat	18	User Preferences Profiling Based on User Behaviors on Facebook Page Categories; By Rachsuda Jiamthapthaksin and Than Htike Aung
	11.20	4	Merging Artificial Immune System and Ordinal Optimization for Solving the Optimal Buffer Resource Allocation of Production Line; By Shih-Cheng Horng and Shieh-Shing Lin	95	An integrated smart technology system: A case study of logistic management of Electricity Generation Authority of Thailand (EGAT); By Oran Chieochan, Aukit SaoKaew and Ekkarat Boonchieng	71	Improvement of Container Scheduling for Docker using Ant Colony Optimization; By Chanwit Kaewkasi and Kornrathak Chuenmuneewong
	11.40	5	Research of Mining Algorithms for Uncertain Spatio-temporal Co-occurrence Pattern; By Zhanquan Wang, Bowen Lu, Fangli Ying, Man Kong and Minwei Tang	15	Human Fall Detection by Using the Body Vector; By Siriporn Pattamaset, Theekapun Charoenpong, Patsama Charoenpong and Chamaporn Chianrabutra	77	Impacts of Low Power Listening on Network Monitoring in Wireless Sensor Networks; By Krita Pattamasiriwat and Chaiporn Jaikaeo
	12.00-13.00	Lunch					

		PaperID	Catleya meeting room Regular: Computational Intelligence Chair: Dr.Sunisa Rimcharoen (BUU, Thailand)	PaperID	Rimsaun 2 meeting room Regular: Intelligent applications Chair: Dr. Antony Harfield (NU, Thailand)	PaperID	Mokara meeting room Human-Machine Interaction Workshop and Applications by ECTI Thailand Association Chair: Dr.Montri Phothisonothai (KMITL, Thailand)
February 2, 2017	**13.00**	6	Integrating Labeled Latent Dirichlet Allocation into Sentiment Analysis of Movie and General Domains; By Ryan Coughlin, Jean-Charles Coetsier and Rachsuda Jiamthapthaksin	21	An Intelligent System Architecture for Meal Assistant Robotic Arm; By Adna Sento, Pannawit Srisuk and Yuttana Kitjaidure		
	13.20	7	Adaptive Artificial Bee Colony Algorithm for solving the Capacitated Vehicle Routing Problem; By Sununta Mingprasert and Ruedee Masuchan	27	Low Cost Weather Station for Climate-Smart Agriculture; By Sonam Tenzin, Satetha Siyang, Theerapat Pobkrut and Teerakiat Kerdcharoen		
	13.40	8	Polysemy Detection in Distributed Representation of Word Sense; By Kana Oomoto, Haruka Oikawa, Eiko Yamamoto, Mitsuo Yoshida, Masayuki Okabe and Kyoji Umemura	28	Volatile Urine Biomarkers Detection in Type 2 Diabetes towards Use as Smart Healthcare Application; By Phuntsho Choden, Thara Seesaard, Tanthip Eamsa-Ard, Chutintorn Sriphrapradang and Teerakiat Kerdcharoen		
	14.00	17	3D Body Shape Clustering based on PSO by multi-fitness function; By Pornthep Sarakon, Supiya Charoensiriwath, Bunyarit Uyyanonvara and Hirohiko Kaneko	29	Electronic Nose Based Wireless Sensor Network for Soil Monitoring in Precision Farming System; By Ugyen Dorji, Theerapat Pobkrut and Teerakiat Kerdcharoen		
	14.20	19	Vehicle Detection on a Pint-Sized Computer; By Thanida Tangkocharoen and Ananta Srisuphab	42	Vietnamese-Thai Machine Translation using Rule-based Model; By Tung Nguyen Ba Son and Pusadee Seresangtakul		
	14.40-15.10	Break					

		PaperID	Catleya meeting room Regular: Computational Intelligence Chair: Dr.Nuthanon Leelatrakul (BUU, Thailand)	PaperID	Rimsaun 2 meeting room Regular: Intelligent applications Chair: Dr.Chalermpan Fongsamut (BUU, Thailand)	PaperID	Mokara meeting room Human-Machine Interaction Workshop and Applications by ECTI Thailand Association Chair: Dr.Montri Phothisonothai (KMITL, Thailand)
February 2, 2017	**15.10**	20	Thai Local Product Recommendation Using Ontological Content based Filtering; By Natedao Thotharat	43	Fault Detection for Circulating Water Pump Using Time Series Forecasting and Outlier Detection; By Manassakan Sanayha and Peerapon Vateekul		
	15.30	24	Study of Discretization Methods in Classification; By Kittichai Lavangnananda and Supharoek Chattanachot	45	A Gated Recurrent Unit Neural Network for Chinese Word Segmentation; By Xuelian Li, Hong Duan and Mu Xu		
	15.50	25	Majority Voting based on Q-Gaussian Activation Function Circular Extreme Learning Machine; By Sarutte Atsawaraungsuk	53	Time-Frequency Ratio Hashing for Content-Based Audio Retrieval; By Petcharat Panyapanuwat, Suwatchai Kamonsantiroj and Luepol Pipanmaekaporn		
	16.10	26	Inverse Kinematics Solution using Neural Networks from Forward Kinematics Equations; By Pannawit Srisuk, Adna Sento and Yutthana Kitjaidure	64	Isarn Dharma Alphabets Lexicon For Natural Language Processing; By Nongnud Phaiboon and Pusadee Seresangtakul		
	16.00-17.00	**Steering Committee Meeting / Mokara meeting room**					
	17.00-18.00	**The 3rd Meeting of International Consortium in Informatics (ICI Meeting) / Mokara meeting room**					
	18.00-20:30	**Welcome Party / Dinner at Aranda meeting room**					

February 3, 2017	08.00	Registration					
		PaperID	**Catleya meeting room Regular: Computational Intelligence Chair: Dr.Sartra Wongthanavasu (KKU, Thailand)**	**PaperID**	**Rimsaun 2 meeting room Joint Workshop on the 3rd Patent Mining and its Applications & Digital Science-based Issue Resolution Chair: Prof.Thomas Mandl, (Information Science University of Hildesheim, Germany)**	**PaperID**	**Mokara meeting room Special Session on Cognitive Science and Human Perception Chair: Prof. Katsumi Watanabe (Waseda University / the University of Tokyo, Japan) and Prof. Roberto Caldara (University of Fribourg, Switzerland)**
	09.00	37	Mining weighted-frequent-regular itemsets from transactional database; By Kittipa Klangwisan and Komate Amphawan	IPAMIN-01	Building Issue-Data Maps to Support the Resolution of Socio-National Issues; By Mi-Nyeong Hwang, Seungwoo Lee and Heeyoung Oh	55	Visual Attention is Captured by Task-Irrelevant Faces, but Not by Pareidolia Faces; By Atsunori Ariga and Katsuhiko Arihara
	09.20	39	Applying One-Versus-One SVMs to Classify Multi-Label Data with Large Labels Using Spark; By Suthipong Daengduang and Peerapon Vateekul	IPAMIN-02	Study on Extracting Implicit Patterns of Patent Data based on Timeline; By Athita Onuean, Jangwon Gim, Yunji Jang and Hanmin Jung	97	Impacts of cue reliability and explicit instruction on visual attention; By Kanji Tanaka and Katsumi Watanabe
	09.40	41	Sentiment Analysis For Short Chinese Text Based On Character-level Methods; By Yanxin An, Xinhuai Tang and Bin Xie	IPAMIN-03	An Improved Author-Topic over Time Model; By Guochao Sun, Shuo Xu, Xiaodong Qiao and Hongqi Han	98	A left eye bias for female faces; By Nayla Sokhn, Francesca Bertoli and Roberto Caldara
	10.00	44	Multiple ARIMA Subsequences Aggregate Time Series Model to Forecast Cash in ATM; By Paisit Khanarsa and Krung Sinapiromsaran	IPAMIN-04	The Clustering Analysis Method of Technology Competitor Based on Patent Text Terms; By Yongsheng Yu, Hongqi Han, Changqing Yao and Zhong Li	99	Time-Frequency Based Coherence Analysis of Red and Green Flickering Visual Stimuli for EEG-Controlled Applications; By Suchada Tantisatirapong, Panisa Dechwechprasit, Wongwit Senavongse, and Montri Phothisonothai
	10.20-10.50	Break					

		PaperID	**Catleya meeting room Regular: Computational Intelligence Chair: Dr.Sartra Wongthanavasu (KKU, Thailand)**	**PaperID**	**Rimsaun 2 meeting room Joint Workshop on the 3rd Patent Mining and its Applications & Digital Science-based Issue Resolution Chair: Prof.Thomas Mandl, (Information Science University of Hildesheim, Germany)**	**PaperID**	**Mokara meeting room Special Session on Cognitive Science and Human Perception & Special Session on Image Processing, Computer Vision, Human Perception, Knowledge-based System and its Applications Chair: Prof. Katsumi Watanabe (Waseda University / the University of Tokyo, Japan), Prof. Roberto Caldara (University of Fribourg, Switzerland) and Dr.Montri Phothisonothai, (KMITL, Thailand)**
February 3, 2017	**10.50**	49	3DDIR: The Distance Interior Ratio of Volumetric Models for Object Recognition; By Chinnawat Wongwises, Natsuda Kaothanthong and Wasit Limprasert	IPAMIN-05	Experimental Study of Time Series-based Dataset Selection for Effective Text Classification; By Yeonghun Chae, Do-Heon Jeong and Taehong Kim	100	Relations between personality traits and empathy for social pain and physical pain; By Aiko Murata and Katsumi Watanabe
	11.10	51	A Character-level Convolutional Neural Network with Dynamic Input Length for Thai Text Categorization; By Thanabhat Koomsubha and Peerapon Vateekul	IPAMIN-06	SVM-based Web Content Mining with Leaf Classification Unit from DOM-tree; By Yeongsu Kim and Seungwoo Lee	23	Automatic Text Imprint Analysis from Pill Images; By Siroratt Suntronsuk and Sukanya Ratanotayanon
	11.30	70	Thai Finger-Spelling Sign Language Recognition Using Global and Local Features with SVM; By Thongpan Pariwat and Pusadee Seresangtakul	IPAMIN-07	Ambiguity in Patent Vocabulary; By Jens Bertram and Thomas Mandl	33	Hand Posture Estimation from 2D Image Sequence by Hand Landmark Identification; By Pargorn Puttapirat and Theekapun Charoenpong
	11.50-13.00	Lunch					

		PaperID	Catleya meeting room Regular: Computational Intelligence Chair: Dr.Komate Amphawan (BUU, Thailand)	PaperID	Rimsaun 2 meeting room Regular: Intelligent applications Chair: Dr. Paramate Horkaew (SUT, Thailand)	PaperID	Mokara meeting room Special Session on Cognitive Science and Human Perception & Special Session on Image Processing, Computer Vision, Human Perception, Knowledge-based System and its Applications Chair: Prof. Katsumi Watanabe (Waseda University / the University of Tokyo, Japan), Prof. Roberto Caldara (University of Fribourg, Switzerland) and Dr.Montri Phothisonothai, (KMITL, Thailand)
February 3, 2017	**13.00**	47	Mining High-Utility Itemsets with Irregular Occurrence; By Supachai Laoviboon and Komate Amphawan	68	Reducing Waiting Time in Automatic Captioned Relay Service using Short Pause in Voice Activity Detection; By Kiettiphong Manovisut, Nattanun Thatphithakkul and Pokpong Songmuang	36	DCSeg: Decoupled CNN for Classification and Semantic Segmentation; By Robail Yasrab, Naijie Gu, Xiaoci Zhang and Asad Khan
	13.20	54	Rule Extraction from Electroencephalogram Signals Using Support Vector Machine; By Anuchin Chatchinarat, Kok Wai Wong and Chun Che Fung	69	Automated segmentation of media-adventitia and lumen from intravascular ultrasound images using non-parametric thresholding; By Anusorn Wong-Od, Annupan Rodtook, Suwanna Rasmequan and Krisana Chinnasarn	46	A Search for Geometric-Shape Objects in a Vector Image: Scalable Vector Graphics (SVG) File Format; By Chanon Seel-audom, Wassana Naiyapo and Varin Chouvatut
	13.40	59	Multiclass Support Vector Machine for Classification Spatial Data form Satellite Image; By Kanita Tangthaikwan and Narongdech Keeratipranon	75	Improving Thai-English Word Alignment for Interrogative Sentences in SMT by Grammatical Knowledge; By Kanyalag Phodong and Rachada Kongkachandra	50	CodeMage : Educational Programming Environment For Beginners; By S.J. Whittall, W.A.C. Prashandi, G.L.S. Himasha, D.I. De Silva and T. K. Suriyawansa
	14.00	79	Dog Cough Sound Classification Using Artificial Neural Network and the Selected Relevant Features from Discrete Wavelet Transform; By Panchana Kakabutr, Kullanan Sae Chen, Viewpaka Wangvisavawit, Praisan Padungweang and Olarn Rojanapornpun	83	iShop – Shopping application for visually challenged; By Dilshan De Silva, M.R. Aaquibah Nashry, Saraniya Varathalingam, Rubika Murugathas and T. K. Suriyawansa	58	Fine Tuning for Green Screen Matting; By Thanathorn Phoka, Warayu Jariyawattanarat and Attawith Sudsang
	14.20	80	University Ranking Prediction System by Analyzing Influential Global Performance Indicators; By Anika Tabassum, Mahamudul Hasan, Shibbir Ahmed, Rahnuma Tasmin, Deen Md. Abdullah and Tasnim Musharrat	88	Automatic Discovering Success Factor Relationship Entities in Articles using Named Entity Recognition; By Supattra Niboonkit, Worarat Krathu and Praisan Padungweang	73	Computational Analysis of Blood Parameters Separate by Centrifuge Technique; By Anon Wangboon, Pattarapong Phasukkit and Mongkol Keawbumrung
	14.40-15.10	Break					

February 3, 2017		PaperID	Catleya meeting room Regular: Computational Intelligence Chair: Dr.Komate Amphawan (BUU, Thailand)			PaperID	Mokara meeting room Special Session on Cognitive Science and Human Perception & Special Session on Image Processing, Computer Vision, Human Perception, Knowledge-based System and its Applications Chair: Prof. Katsumi Watanabe (Waseda University / the University of Tokyo, Japan), Prof. Roberto Caldara (University of Fribourg, Switzerland) and Dr.Montri Phothisonothai, (KMITL, Thailand)
	15.10	82	Social Network User Identification; By Laikhram Jamjuntra, Pantakan Chartsuwan, Peerapong Wonglimsamut, Kriengkrai Porkaew and Umaporn Supasitthimethee			74	Applied Integral Intensity Projection To Find The Numbers Of The Parking Spots; By Bunyawath Thepsathit, Vongsathorn Kaosaiyananda, Akara Charoensuk and Pramaul Choorat
	15.30	86	Discovering interesting itemsets based on change in regularity of occurrence; By Sumalee Eisariyodom and Komate Amphawan			76	Obstacle detection algorithm for unmanned aerial vehicles using binocular stereoscopic vision; By Pakorn Ueareeworakul and Saiyan Saiyod
	15.50	89	Partition-based Overlapping Clustering Using Clusters' Parameters and Relations; By Tanawat Limungkura and Peerapon Vateekul			91	Pattern Extraction from Northern Thai Fabrics Using Flexibly Matching Segments: Sarong Teenjok and Lanna Textiles; By Nattha Vasantapan and Varin Chouvatut
	16.10	96	Automated Cyberbullying Detection using Clustering Appearance Patterns; By Walisa Romsaiyud, Kodchakorn Na Nakornphanom, Pimpaka Prasertsilp, Piyaporn Nurarak and Pirom Konglerd				
	16.30	**Closing the Conference Ceremony**					

February 4, 2017	**08.30-09.00**	Registration
	09.00-12.00	Thailand Research Consortium for Informatics Field Trips (Burapha University)
	12.00	**Closing the Consortium Ceremony**

Certificate of Contributions

Kittipa Klangwisan and Komate Amphawan

Entitled

Mining weighted-frequent-regular itemsets from transactional database

Has Contributed To

The 2017- 9th International Conference on Knowledge and Smart Technology (KST)

February 1 - 4, 2017

Amari Pattaya, Chon Buri, Thailand

Organized by

Faculty of Informatics, Burapha University, Thailand

Chidchanok Lursinsap, Ph.D.
Faculty of Science, Chulalongkorn University
General Chair

Krisana Chinnasarn, Ph.D.
Faculty of Informatics, Burapha University
Dean

Technical Co-Sponsor

The 2018-10th International Conference on Knowledge and Smart Technology

"Cybernetics in the Next Decades"

KST Research Center
Knowledge & Smart Technologies
since 2008

BURAPHA UNIVERSITY
SINCE 1955

KST2018

Jan 31-Feb 3

@Kantary Hills Hotel
Chiangmai, Thailand

Organized by
Knowledge and Smart Technology Research Center
Faculty of Informatics, Burapha University

ISBN 978-1-5386-4014-2

Sponsored by IEEE IEEE Xplore Digital Library IEEE THAILAND SECTION

Patronage by ECTI depa AIAT BIA

Efficient weighted-frequent-regular itemsets mining using interval word segments structure

Kittipa Klangwisan*, Komate Amphawan†
Computational Innovation Laboratory, Faculty of Informatics, Burapha University, Chonburi, 20131, Thailand
Email: *kklangwisan@gmail.com, †komate@gmail.com

***Abstract*—Recently, weighted-frequent regular itemset mining has been introduced to discover interesting itemsets based on observation of their occurrence behavior and difference of items' importance. The weighted-support and regularity of occurrence are applied to measure interestingness of itemsets. However, with a large amount of items to be considered, the mining process consumes high computational time and memory. Thus, to efficiently mine such itemsets, this paper introduces a single-pass algorithm, called *WFRIM-IWS*. The interval word segments structure (*IWS*) is utilized for maintaining occurrence information of each itemset. A new looked up table is designed to quickly calculate frequency (weighted-support) and regularity of an itemset from *IWS*. Moreover, to early prune search space, the concept of *overestimated weighted-frequency* and *global/local maximum weights* are then applied. Experiments on synthetic and real datasets show that the proposed *WFRIM-IWS* outperforms the previous algorithm in the terms of computational time and memory consumption.**

***Keywords*—data mining; association rules; itemsets mining; frequent-regular itemsets; weight of importance**

I. Introduction

Frequent itemsets mining (*FIM* [1]) is a fundamental technique in data mining domain. *FIM* aims to discover interesting itemsets that have high frequency of occurrence in database. *FIM* currently plays an important role in several applications such as retail business, web log analytics, mobile commerce, biological data, financial analysis, etc. Since there are several applications apply *FIM* to gain hidden knowledge from database, *FIM* has attracted researchers and extended to various aspects, for example, *FIM* on data streams [2], *FIM* with big data [3], *FIM* with uncertain data [4], sequential itemset mining [5], top-k frequent itemset mining [6], frequent-regular itemset mining [7], [8] and so on.

From above, traditional *FIM* and all of its extension assume that all items have the same significant and consider items equally. However, in many real-world applications, items usually have different degree of importance such as weight of importance, profit, cost, risk, etc. For example, in a supermarket, "yogurt, granola" are bought more frequent than "caviar, vodka" but the latter one can give more profit than the first one. To address this issue, the problem of weighted-frequent itemset mining (*WFIM*) is proposed. An itemset with high weighted-support (*i.e.* its frequency of occurrence multiplied by its weight of importance) is identified as an interesting itemset. As *FIM*, *WFIM* also has several extensions such as *WFIM* on incremental databases [9], weighted sequential pattern mining [10], *WFIM* over data streams [11] and so on. In addition, the task of weighted-frequent-regular itemset mining (*WFRIM*) [12] is recently proposed to discover interesting itemsets based on their weight of importance and their occurrence behavior (in the terms of frequency and regularity of occurrence). However, with a large amount of items to be considered, the task of *WFRIM* consumes high computational time and memory. Thus, there is a room for improving efficiency of *WFRIM*'s algorithm.

Thus, in this paper, we here propose an efficient single-pass algorithm named *WFRIM-IWS* (Weighted-Frequent-Regular Itemset Miner using Inverval Word Segment structure) in order to discover a complete set of weighted-frequent-regular itemsets from transactional database. *WFRIM-IWS* applies several concepts for increasing its efficiency *i.e.* (*i*) the global/local maximum weights and the overestimated weighted support to prune search space, (*ii*) interval word segment structure (*IWS*) to maintain occurrence information of each item/itemsets, and (*iii*) a new look-up table is designed and utilized to quickly calculate support and regularity from *IWS* structure. Experiments were conducted on synthetic and real datasets to show efficiency of the proposed algorithm in the terms of computational time, memory usage and the number of discovered itemsets, respectively.

II. Problem Statement

In this section, we first introduce basic notations on items, weights of items, and transactional database. Then, definitions related to a regularity, a weight-support and a weighted-frequent regular itemset are described as in [12].

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of distinct items. A set $W = \{w_1, w_2, \ldots, w_n\}$ is a set of weights of importance of items in which each $w_j \in W$ (a non-negative real number) is the weight of item $i_j \in I$. A set $X = \{i_j, \ldots, i_k\} \subseteq I$, where $1 \leq j \leq k \leq n$ is called an itemset, a pattern or a *k*-itemset if X has k items. A transaction database $TDB = \{t_1, t_2, \ldots, t_m\}$ is a finite set of m transactions in which each transaction t_p is a tuple of (p, Y) where p is a unique transaction identifier (called *tid* for short) and $Y \subseteq I$ is an itemset appearing in transaction t_j. For an itemset X, if $X \subseteq Y$ (where $Y \in t_j$), it can be said that X occurs in t_j or t_j contains X, denoted as j^X (*i.e.* the j^{th} transaction in *TDB* contains X). Then, all occurrences of X can be defined as a set $TID^X = \{p^X, \ldots, q^X\}$ of ordered tids of transactions containing X. Last, the frequency of occurrence of X (also called support) can be defined as $s^X = |TID^X|$ which is the number of (*tid* of) transactions containing X.

978-1-5386-4015-9/18/$31.00 @2018 IEEE

Definition 1 (A Regularity of an itemset X): The regularity of an itemset X can be defined as the maximum number of consecutive transactions that X does not occur in database, denoted as $r^X = max(r_1^X, r_2^X, \ldots, r_{|TID^X|+1}^X)$, where (*i*) r_1^X indicates the first group of transactions that X does not occur in the database before its first occurrence, *i.e.* $r_1^X = p^X$ where p^X is the tid of the first transaction containing X, (*ii*) r_2^X to $r_{|TID^X|}^X$ are the groups of transaction between two consecutive occurrences of X, for example, if X occurs in transaction t_u and t_v (where $u < v$) and there is no other transactions between t_u and t_v containing X, then the regularity r_v^X can be calculated as $r_v^X = v^X - u^X$, and (*iii*) $r_{|TID^X|+1}^X$ is the last group of transactions that X does not occur in database after its last occurrence to the end of database, *i.e.* $r_{|TID^X|+1}^X = m - q^X$ where q is tid of the last transaction of *TDB* and m is $|TDB|$, respectively. With regularity r^X, it can be said that X occurs at least once in every consecutive r^X transactions.

Definition 2 (A frequent-regular itemset): As defined in [7], [8], an itemset X is called a *frequent-regular itemset*, if its support s^X is no less than a user-given minimum support threshold σ_s and its regularity r^X is no greater than a user-given maximum regularity threshold σ_r.

Definition 3 (A weighted-support of an itemset X): For an itemset $X = \{i_j, \ldots i_k\}$, there is a set of weights $W^X = \{w_{i_j}, \ldots, w_{i_k}\}$ associated with X and the weight of X can be calculated as $w^X = \frac{\sum_{i_j \in X} w_{i_j}}{|X|}$ (Noted that w^X is the average weight of all items in X). Then, the weighted-support of X can be defined as $ws^X = s^X \times w^X$ which expresses X's frequency of occurrence based on (multiplied by) its weight of importance.

Definition 4 (A weighted-frequent itemset): An itemset X is called a *weighted-frequent itemset* if its weighted support ws^X is no less than a user-given weighted support threshold σ_{ws}.

Definition 5 (A weighted-frequent regular itemset): An itemset X is called a *weighted-frequent regular itemset* if (*i*) its weighted support ws^X is no less than a user-given weighted support threshold σ_{ws} and (*ii*), its regularity r^X is no greater than a user-given regularity threshold σ_r.

Problem statement: Given a transactional database *TDB* with a set of weight $W = \{w_1, w_2, \ldots, w_n\}$ of items, a weighted-support threshold σ_{ws} and a regularity threshold σ_r, the problem of mining weighted-frequent regular itemsets is to discover a complete set of itemsets in which each itemset has weighted-support not less that σ_{ws} and regularity not greater than σ_r.

III. PROPOSED METHOD

In this section, we here present an efficient single-pass algorithm named *WFRIM-IWS* (Weighted-Frequent-Regular Itemsets Miner using Interval Word Segment structure). The concepts of the global/local maximum weights [13] and the overestimated weighted support are applied to hold *downward closure property* which can help to prune search space. The interval word segment structure (*IWS*) is utilized to efficiently maintain occurrence information of each item/itemset. A new look-up table is designed to quickly retrieve support and regularity from *IWS* and a tree structure called *WFRIM-tree* is employed to maintain items/itemsets, respectively.

Weight Table

item	a	b	c	d	e	f	g
weight	0.75	0.65	0.65	0.4	0.7	0.5	0.6

A transactional database

tid	Set of items	tid	Set of items	tid	Set of items
1	a, c, e	...	...	55	a, b, d, e
...	...	35	b, d, g	...	...
15	b, c, d, f, g	...	...	70	a, b, c, d, e, g
16	a, b, e, f, g	49	a, b, c, e, g	71	a, b, d, e
...	...	50	a, b, c, e, g	...	...
32	a, b, c, e, f, g	...	...		

Fig. 1. An example of transactional database

A. Global/Local maximum weights

To early prune weighted-infrequent items/itemsets, the concepts of the global maximum weight (*i.e. GMAXW* $= max(w_1, w_2, \ldots, w_n)$) and the local maximum weight (*i.e. LMAXW* $= \frac{\sum_{i_k \in K} w_{i_k} + max(w_{i_m}, \ldots, w_{i_n})}{|K|+1}$ where K is an itemset previously considered in the mining process and $w_{i_m}, \ldots, w_{i_n}$ is the set of weights of items $i_m, \ldots, i_n$ that have not yet considered) are applied. Then, the overestimate weighted support of an item i_j can be calculated by $ows^{i_j} = GMAXW \times s^{i_j}$. Meanwhile, the overestimate weighted support of an itemset $X = K \cdot i_m$ can be calculated as $ows^X = LMAXW \times s^X$. With the overestimate weighted support, the itemset X is identified as a weighted-infrequent itemset if its overestimate weight support ows^X is less than a user-given weighted-support threshold σ_{ws}. Therefore, the itemset X and all of its supersets can be pruned.

Example 1: Let's consider the transaction database of Fig. 1 and the weighted support threshold is set to be $\sigma_{ws} = 4$. The overestimate weighted support of the items 'f' can be calculated as $ows^f = GMAXW \times s^f = max(0.75, 0.65, 0.65, 0.4, 0.7, 0.5, 0.6) \times 3 = 0.75 \times 3 = 2.15$ and the weighted support of the itemset 'f' can be computed as $ws^f = w^f \times s^f = 0.5 \times 3 = 1.5$ (as shown in Fig.5(b)). Noted the overestimate weighted support ows^f is higher than ws^f. Then, the itemset 'f' is pruned due to its ows^f is less than σ_{ws}.

B. Interval word segment structure

An interval word segment structure *(IWS)* [14] is a dynamic bit-vector structure used for storing occurrence information of an item/itemset. For an itemset X, its interval word segment structure can be represented as a sequence of segments which can be defined as $IWS^X = \{sm_1^X, sm_2^X, \ldots, sm_y^X\}$. Each segment $sm_j^X \in IWS^X$ consists of two elements *i.e.* (*i*) a non-negative integer refers to the index of the first non-zero word (called *first-index* and denoted as $fi^{sm_j^X}$), and (*ii*) a sequence of non-zero words expressing occurrence behavior of X (defined as $WO^{sm_j^X} = (wo_1^{sm_j^X}, wo_2^{sm_j^X}, \ldots, wo_{|WO^{sm_j^X}|}^{sm_j^X})$. Noted that each word $wo_k^{sm_j^X} \in WO^{sm_j^X}$ contains 16 bits started from the least significant bit to the most significant bit. Each p^{th} bit can be 0 if X does not occur in transaction t_p. Otherwise, the p^{th} bit is set to be 1, respectively.

Therefore, the IWS^X can also be represented as $IWS^X = \left\{ \left(fi^{sm_1^X} \langle wo_1^{sm_1^X}, wo_2^{sm_1^X}, \ldots, wo_{|WO^{sm_1^X}|}^{sm_1^X} \rangle \right), \ldots, \left(fi^{sm_y^X} \langle wo_1^{sm_y^X}, wo_2^{sm_y^X}, \ldots, wo_{|WO^{sm_y^X}|}^{sm_y^X} \rangle \right) \right\}$.

For each occurrence of the itemset X in a transaction t_p, the tid p of t_p is collected in IWS^X as described below (see Algo 1). As in line 1-2, the word index (wi_p) and the value of word based on tid p (wo_p) are first calculated. Then, if t_p is the first transaction that contains X, a new segment is created with the word index wi_p and the word wo_p and then added to IWS^X (line 3-5). Otherwise, the last occurrence lo^X of X is considered in order to find its word index wi_{lo^X} and then the storing of p into IWS^X can be performed within the 3 cases as follows:

1) if the word index of lo^X and p are the same, the last word of the last segment of IWS^X is updated by the word wo_p (line 8-10),
2) if the word index of p is next to that of lo^X, the word wo_p is added at the tail of the last segment of IWS^X (line 11-12),
3) if the word index of p is not next to that of lo^X, a new segment is created with the word index wi_p and the word wo_p and then added to IWS^X (line 13-15), respectively.

Example 2: Let's consider the transactional database as in Fig. 1 and the item 'a' occurring in $t_1, t_{16}, t_{32}, t_{49}, t_{50}, t_{55}, t_{70}$ and t_{71}, respectively. Then, with the first occurrence of 'a' in t_1, the word index and value of word of the tid $p = 1$ can be calculated as $wi_1 = \lfloor \frac{1-1}{16} \rfloor + 1 = 0 + 1 = 1$ and $wo_1 = 2^{((1-1) \bmod 16)} = 2^0 = 1$. Then, a new segment $sm_1^a = (1, \langle 1 \rangle)$ is created and added to IWS^a. Next, storing the occurrence of 'a' in the transaction t_{16} belongs to case 1 as above, since the word index of t_1 and t_{16} are the same. Then, the last word of the last segment of IWS^a is updated by $wo_{16} = 32768$, i.e. $IWS^a = (1\langle 32769 \rangle)$. For the occurrence of 'a' in the transaction t_{32}, the word index and value of word of $p = 32$ can be calculated as $wi_{32} = \lfloor \frac{32-1}{16} \rfloor + 1 = 1 + 1 = 2$ and $wo_{32} = 2^{((32-1) \bmod 16)} = 2^{15} = 32768$. Then, the latest occurrence of 'a' (in t_{16}) is considered and its word index ($wo_{16} = 1$) is calculated. Since the word index of tid $p = 32$ is next to that of the previous occurrence $lo^a = 16$ (case 2 of the above), then the word $wo_{32} = 32768$ is added at the tail of the last segment of IWS^a ($IWS^a = (1\langle 32769, 32768 \rangle)$). For other occurrences, Algo. 1 repeats consideration on the 3 cases above and then IWS^a contains two segments as shown in Fig. 2.

To calculate support and regularity of an itemset X from its IWS^X, a look-up table is created as in [15] (see Fig 3). With the look-up table containing 2^{16} records (*i.e.* 16 is the size of each word used for storing occurrence information of each itemset), we can compute support and regularity of each word contained in a segment of IWS^X. Each record of the look-up table contains four information *i.e.* (*i*) s : the support (frequency of occurrence) of the considered word, (*ii*) pf : the position of the first appearance of bit 1 in the considered word started from the least significant bit (indicates the gap

Algorithm 1: ***collect a tid into IWS***

Input: IWS^X, p, lo^X
Output: IWS^X

1: $wi_p \leftarrow \lfloor \frac{p-1}{wordsize} \rfloor + 1$
2: $wo_p \leftarrow 2^{((p-1) mod\ wordsize)}$
3: **if** $lo^X = 0$ **then**
4: create a new segment $sm \leftarrow (wi_p, \langle wo_p \rangle)$
5: $IWS^X \leftarrow IWS^X \cup sm$
6: **else**
7: $wi_{lo^X} \leftarrow \lfloor \frac{lo^X - 1}{wordsize} \rfloor + 1$
8: **if** $(wi_p - wi_{lo^X}) = 0$ **then**
9: $wo \leftarrow$ the last word of the last segment of IWS^X
10: $wo \leftarrow wo + wo_p$
11: **else if** $(wi_p - wi_{lo^X}) = 1$ **then**
12: add wo_p at the tail of the last segment of IWS^X
13: **else**
14: create a new segment $sm \leftarrow (wi_p, \langle wo_p \rangle)$
15: $IWS^X \leftarrow IWS^X \cup sm$

item	set of transaction containing item a
a	1, 16, 32, 49, 50, 55, 70, 71

bit value	tid	word index	word value
1000000000000001	16-1	1	32769 (32768+1)
1000000000000000	32-17	2	32768
0000000000000000	48-33	3	0
0000000010000011	64-49	4	67 (64+2+1)
0000000011000000	80-65	5	96 (64+32)
IWS^a	{<1(32769, 32768)>, <4(67, 96)>}		

Fig. 2. An example of conversion transaction to *IWS* structure

of missing before the first occurrence of X in the considered word), (*iii*) nl : the number of consecutive bits of 0 from the most significant bit (expresses the group of transactions that X does not occur in the database after the last occurrence of X in the considered word until the end of the word), and (*iv*) mg : the maximum number of sequence of 0 between two bits of 1 (demonstrates the maximum group of transactions between its two occurrence in the considered word), respectively.

The support of each word in each segment of IWS^X can directly retrieved from the element s of the look-up table, for example, the first word in the first segment of IWS^a is 32769 and its support is 2. Mean while, the second word in the first segment of IWS^a is 32768 in which its support is 1, respectively.

In addition, to calculate the regularity from IWS^X, there are four cases to consider each word $wo^{sm_y^x}$ of a segment sm_y^x and then calculate the regularity $r^{wo^{sm_y^x}}$ as described as follows:

1) if $wo^{sm_y^X}$ is $wo_1^{sm_1^X}$ (*i.e.* $wo_1^{sm_1^X}$ is the first word of the first segment sm_1^X of IWS^X) : $r^{wo^{sm_y^X}} \leftarrow max((((fi^{sm_1^X} - 1) \times 16) + pf(wo^{sm_y^X})), mg(wo^{sm_y^X}))$,
2) if $wo^{sm_y^X}$ is $wo_1^{sm_y^X}$ (*i.e.* $wo_1^{sm_y^X}$ is the first word of the y^{th} segment sm_y^X of IWS^X where $y \in [2, |TWS^X|]$) : $r^{wo^{sm_y^X}} \leftarrow max((nl(lwo^{sm_{y-1}^X}) + ((fi^{sm_y^X} - 1 - li^{sm_{y-1}^X}) \times 16) + pf(wo^{sm_y^X})), mg(wo^{sm_y^X}))$,
3) if $wo^{sm_y^X}$ is $wo_z^{sm_y^X}$ (*i.e.* $wo_z^{sm_y^X}$ is the z^{th} word (not

the first word) of the y^{th} segment) : $r^{wo^{sm_y^X}} \leftarrow max((nl(wo_{z-1}^{sm_y^X}) + pf(wo^{sm_y^X})), mg(wo^{sm_y^X}))$

4) if $wo^{sm_y^X}$ is $wo^{sm^X_{|IWS^X|}}_{|WO^{sm^X_{|IWS^X|}}|}$ (*i.e.* $wo^{sm^X_{|IWS^X|}}_{|WO^{sm^X_{|IWS^X|}}|}$ is the last word of the last segment $sm^X_{|IWS^X|}$ of IWS^X) : $r^{wo^{sm_y^X}} \leftarrow (nl(wo^{sm_y^X}) + ((lwo^{TDB} - li^{sm^X_{|IWS^X|}} \times 16))$,

where (*i*) $fi^{sm_1^X}$ is the start word index of the first segment; (*ii*) $pf(wo^{sm_y^X})$ is the number of consecutive bit of 0 before the first bit of 1 (start to consider from the least significant bit); (*iii*) $mg(wo^{sm_y^X})$ is the maximum number of sequence of 0 between two bits of 1 retrieved from the look-up table; (*iv*) $nl(lwo^{sm^X_{y-1}})$ is the number of bits from the last bit 1 of the last word $lwo^{sm^X_{y-1}}$ of the $(y-1)^{th}$ segment retrieved from the look-up table; (*v*) $fi^{sm_y^X}$ is the first word index of the y^{th} segment; (*vi*) $li^{sm^X_{y-1}}$ is the last word index of the $(y-1)^{th}$ segment; (*vii*) $nl(wo_{z-1}^{sm_y^X})$ is the number of bits from the last bit 1 of the $(z-1)^{th}$ word in the y^{th} segment retrieved from the look-up table; (*viii*) lwo^{TDB} is the last word index used to keep occurrence information of the last 16 transactions of *TDB*; (*ix*) $li^{sm^X_{|IWS^X|}}$ is the word index of the last word in the last segment of IWS^X, respectively.

Example 3: From the transactional database of Fig. 1, the interval word segment structure of item 'a' can be represented as $IWS^a = \{\langle 1, (32769, 32768)\rangle, \langle 4, (67, 96)\rangle\}$. *Then, the regularity* r^a *of the item 'a' can be calculated from the first word of the first segment, the second word of the first segment, the first word of the second segment and the second word of the second segment (in case 3 and case 4) which can be denoted as* $r^a = max(r^{wo_1^{sm_1^a}}, r^{wo_2^{sm_1^a}}, r^{wo_1^{sm_2^a}}, r^{wo_2^{sm_2^a}}$ *(case 3)* $, r^{wo_2^{sm_2^a}}$ *(case 4))* $= max(15, 16, 17, 15, 9) = 17$ *where (i)* $r^{wo_1^{sm_1^a}} = max((((fi^{sm_1^a} - 1) \times 16) + pf(wo^{sm_1^a})), mg(wo^{sm_1^a})) = max((1-1) \times 16 + 1, 15) = 15$, *(ii)* $r^{wo_2^{sm_1^a}} = max((nl(wo_1^{sm_1^a}) + pf(wo_2^{sm_1^a})), mg(wo_2^{sm_2^a})) = max(0 + 16, 0) = 16$, *(iii)* $r^{wo_1^{sm_2^a}} = max((nl(lwo^{sm_1^a}) + ((fi^{sm_2^a} - 1 - li^{sm_1^a}) \times 16) + pf(wo_1^{sm_2^a})), mg(wo_1^{sm_2^a})) = max(0 + ((4-1-2) \times 16) + 1, 4) = 17$, *(iv)* $r^{wo_2^{sm_2^a}}$ *(case 3)* $= max((nl(wo_1^{sm_2^a}) + pf(wo_2^{sm_2^a})), mg(wo_2^{sm_2^a})) = max(9 + 6, 0) = 15$, *and (v)* $r^{wo_2^{sm_2^a}}$ *(case 4)* $= nl(wo_2^{sm_2^a}) + ((lwo^{TDB} - li^{sm_2^a}) \times 16)) = 9 + ((5-5) \times 16) = 9$, *respectively.*

Last, to generate a longer size itemset $Z = X \cup Y$ by considering a pair of itemset X and Y having the same prefix (*i.e.* $X = \{i_j, ..., i_k, i_p\}$, $Y = \{i_j, ..., i_k, i_q\}$ and the prefix $= \{i_j, ..., i_k\}$), IWS^X and IWS^Y are intersected together in order to find transactions that both X and Y occur together and collected in IWS^Z. As detailed in Algo. 2, each segment sm_p^X of IWS^X and sm_q^Y of IWS^Y are considered. If sm_p^X and sm_q^Y are overlap, the first index of *fi* and the last index *li* of overlapping are first calculated to know the scope of consideration (line 5-6). Then, each word $wo_u^{sm_p^X}$ and $wo_u^{sm_q^Y}$ under the scope *fi* and *li* are then intersected and collected in the word *wo*. Then, the word wo is then considered in two cases :

1) if the value of *wo* is 0 (*i.e* the itemset X and Y do not occur together during the word index u^{th}), the set of word *WO* is then investigated. If *WO* is empty, the *fi* is increased by 1 due to the scope of occurrence of Z might start from the $(u+1)^{th}$ word. Otherwise, a new segment $sm = (fi\langle WO\rangle)$ is created and added at the tail of IWS^Z.
2) if the value of *wo* is greater than 0 (the itemset X and Y occur together during the word index u^{th}), the word *wo* is thus added to the tail of the set *WO* and the consideration will move to the next word.

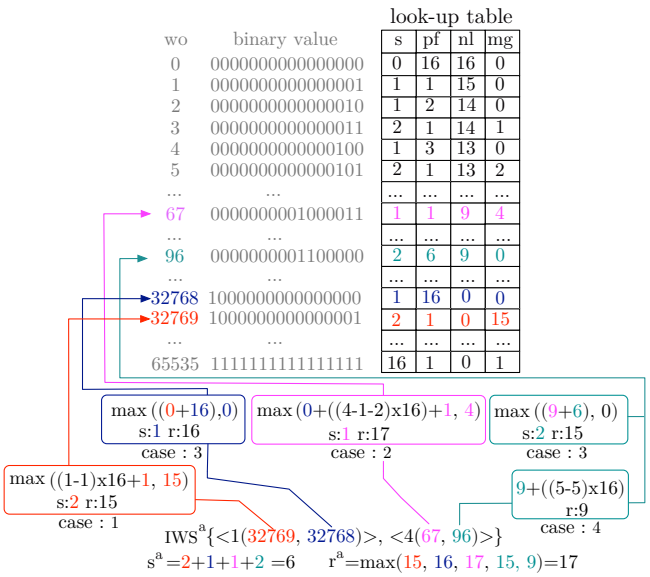


Fig. 3. An example of detail of look-up table and method for calculating regularity of IWS^a

At the end of consideration of word in segment sm_p^X and sm_q^Y, a new segment $sm = \langle fi, WO\rangle$ is created and added at the tail of IWS^Z, if the set of word *WO* is not empty. Besides, whenever the sm_p^X and sm_q^Y are not overlap, the index p will be increased by 1 if the last word index of segment sm_p^X is less than the last word index of segment sm_q^Y. Otherwise, q will be increased by 1.

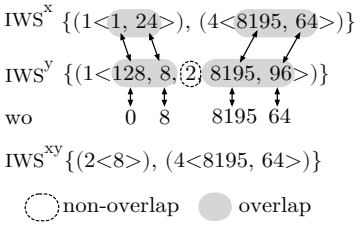


Fig. 4. An example of intersection method for IWS^x and IWS^y

Example 4: Let's consider the $IWS^x = \{(1\langle 1, 24\rangle), \langle 4, (8195, 64)\rangle\}$ *and* $IWS^y = \{(1\langle 128, 8, 2, 8195, 96\rangle)\}$ *as in Fig. 4. To generate itemset 'x,y', the* IWS^x *and* IWS^y *are intersected to calculate weighted-frequency, regularity and to collect occurrence information of the itemset 'x,y'. First, the first segment* $sm_1^x = (1\langle 1, 24\rangle)$ *and* $sm_1^y = (1\langle 128, 8, 2, 8195, 96\rangle)$ *are considered (due to it is overlap). The scope of consideration based on the first word index and the last word index are then calculated as* $fi = max(fi_{sm_1^x}, fi_{sm_1^y}) = max(1, 1) = 1$ *and*

$li = min(li_{sm_1^x}, li_{sm_1^y}) = min(fi_{sm_1^x} + |WO^{sm_1^x}| - 1, fi_{sm_1^y} + |WO^{sm_1^y}| - 1) = min(1+2-1, 1+5-1) = min(2,5) = 2$. *Each word* sm_1^x *and* sm_1^y *of between the word index* $fi = 1$ *and* $li = 2$ *is sequentially intersected. For the word* $wo_1^{sm_1^x} = 1$ *and* $wo_1^{sm_1^y} = 128$*, its intersection is* $wo = 1 \cap 128 = 0$*. Then, the value of fi is increased by* 1 *(Noted that right now* $fi = 2$*). Meanwhile, for the word* $wo_2^{sm_1^x} = 24$ *and* $wo_2^{sm_1^y} = 8$*, its intersection is* $wo = 24 \cap 8 = 8$*. Thus, the word wo is collected into the set WO. At then end of consideration on the word based on the scope fi and li, a new segment* $sm = (2\langle 8 \rangle)$ *is created and add into* $IWS^{x,y}$*. Next, the consideration is moved to the segment* sm_2^x *and* sm_1^y *since they are overlap and the intersection process is repeated in the same manner as above. Last, after considering all overlapping segments,* $IWS^{x,y}$ *is as shown in Fig. 4.*

Algorithm 2: *IWS's intersection on a pair of IWS*

Input: IWS^X, IWS^Y
Output: IWS^Z

```
1:  IWS^Z ← ∅
2:  p ← 1 and q ← 1
3:  while p < |IWS^X| and q < |IWS^Y| do
4:    if sm_p^X and sm_q^Y are overlap then
5:      fi ← max(fi_{sm_p^X}, fi_{sm_q^Y})
6:      li ← min(fi_{sm_p^X} + |WO_{sm_p^X}| - 1, fi_{sm_q^Y} + |WO_{sm_q^Y}| - 1)
7:      u ← fi
8:      WO ← ∅
9:      while u < li do
10:       wo ← wo_u^{sm_p^X} ∩ wo_u^{sm_q^Y}
11:       if wo = 0 then
12:         if WO = ∅ then
13:           fi ← fi + 1
14:         else
15:           IWS^Z ← IWS^Z ∪ ⟨fi, WO⟩
16:           fi ← u + 1
17:           WO ← ∅
18:       else
19:         WO ← WO ∪ wo
20:         u ← u + 1
21:     if WO ≠ ∅ then
22:       IWS^Z ← IWS^Z ∪ ⟨fi, WO⟩
23:   else
24:     if li_{sm_p^X} < li_{sm_q^Y} then
25:       p ← p + 1
26:     else
27:       q ← q + 1
```

C. WFRIM-IWS algorithm

WFRIM-IWS consists of 2 main steps *i.e.* (*i*) DBscanning and (*ii*) WFRI-Mining, respectively. In DBscanning step, the *WFRIM-tree* is firstly created with a root R and a node for each item i_k is created and set to be a child node of R. Then, each transaction $t_p \in DB$ is scanned and each item $i_k \in t_p$ is considered and the information in the node of i_k is thus updated (*i.e.* the support s^{i_k} is increased by 1 and the IWS^{i_k}, the regularity r^{i_k} and the last occurrence lo^{i_k} are updated tid p of t_p). After scanning all transactions, the regularity of each item is lastly updated and the item with regularity greater than the user-given regularity threshold (σ_r) is removed from the *WFRIM-tree*. Next, the global maximum weight (*GMAXW*) is calculated and the over-estimated weighted-support is calculated. For an item i_k, if its over-estimated weighted-support ows^{i_k} is less than the user-given weighted-support threshold (σ_{ws}), the i_k is then removed from the *WFRIM-tree*. Otherwise, its actual weighted-support ws^{i_k} is computed and the item is identified as a weighted-frequent regular item if its ws^{i_k} is not less than σ_{ws}. The process of *GMAXW*'s calculation and pruning low weighted-support is thus repeated if all the items with maximum weights are pruned (*i.e.* if all the items with maximum weights are pruned, the value of *GMAXW* needs to be recalculated). Last, at the end of DBscanning step, we gain *WFRIM-tree* with nodes of candidate single items (*i.e.* items with the over-estimated weighted-support $\geq \sigma_{ws}$).

Example 5: From the transactional database of Fig. 1 with items a, b, c, d, e*, and* f*, let the regularity threshold* σ_r *be* 20 *and the weighted-support* σ_{ws} *be* 4.0*. First, WFRIM-tree is created with a root* R *and a node for each item* a, b, c, d, e *and* f *is created and linked to be a child node of* R*. Next, the transaction* $t_1 = \{a, c, e\}$ *is read and the* IWS^a*,* IWS^c *and* IWS^e *are then updated by* 1 *(see Algo. 2 for details of collecting a tid in an IWS). The support, regularity and last occurrence of item* a, c *and* e *are then updated as shown in Fig. 5(a). The reading and the updating process are repeated for all transactions in which at the end of reading we gain WFRIM-tree in Fig. 5(b).*

Each item in the WFRIM-tree is considered. Its regularity is calculated (based on case 4 of looking up regularity from the look up table). From the Fig. 5(b), we can observe that the regularity $r^f = 48$ *is greater than* $\sigma_r = 20$*. Then, the item* f *is removed from the WFRIM-tree.*

The global maximum weight is calculated from the maximum weight of items (remaining items after pruning by the regularity threshold) that is GMAXW $= max(w^a, w^b, w^c, w^d, w^e, w^g) = max(0.75, 0.65, 0.65, 0.4, 0.7, 0.6) = 0.75$*. Then, each item is repeatedly considered and its overestimated weighted support is calculated. From the Fig. 5(b), it can be seen that item* d *has* $ows^d = 0.75 \times 5 = 3.75$ *which is less than* σ_{ws}*. Then, item* d *is then removed from WFRIM-tree. Finally, all items in the WFRIM-tree are ordered by descending order of their weights as the order is 'a, e, b, c, g', respectively.*

To mine the completed set of weighted-frequent regular itemsets, the mining process is repeatedly performed on the *WFRIM-tree*. First, a child of root R with an item i_j is considered. The next child of i_j with the item i_k (*i.e.* i_k is located next to the item i_j) is thus also regarded and merged with i_j to be $Z = i_j \cdot i_k$. Then, the local maximum weight and the weight of Z are calculated as $LMAXW = w^Z = \frac{w_{i_j} + w_{i_k}}{2}$. The IWS^{i_j} and IWS^{i_k} are then intersected to collect the occurrence information of Z into IWS^Z and to calculate the support s^Z and the regularity r^Z of Z (by looking up from the header table). The weighted-support ws^Z is calculated by $ws^Z = w^Z \times s^Z$. If the regularity r^Z is not greater than σ_r and the weighted-support ws^Z is not less than σ_{ws}, a node of Z is created with its IWS^Z and linked to be a child node of i_j. Also, the itemset Z is identified and collected as a weighted-frequent-regular itemset. However, if the regularity r^Z is greater than σ_r or he weighted-support ws^Z is less than σ_{ws}, the item i_l (the item next to the item i_k) is considered. The

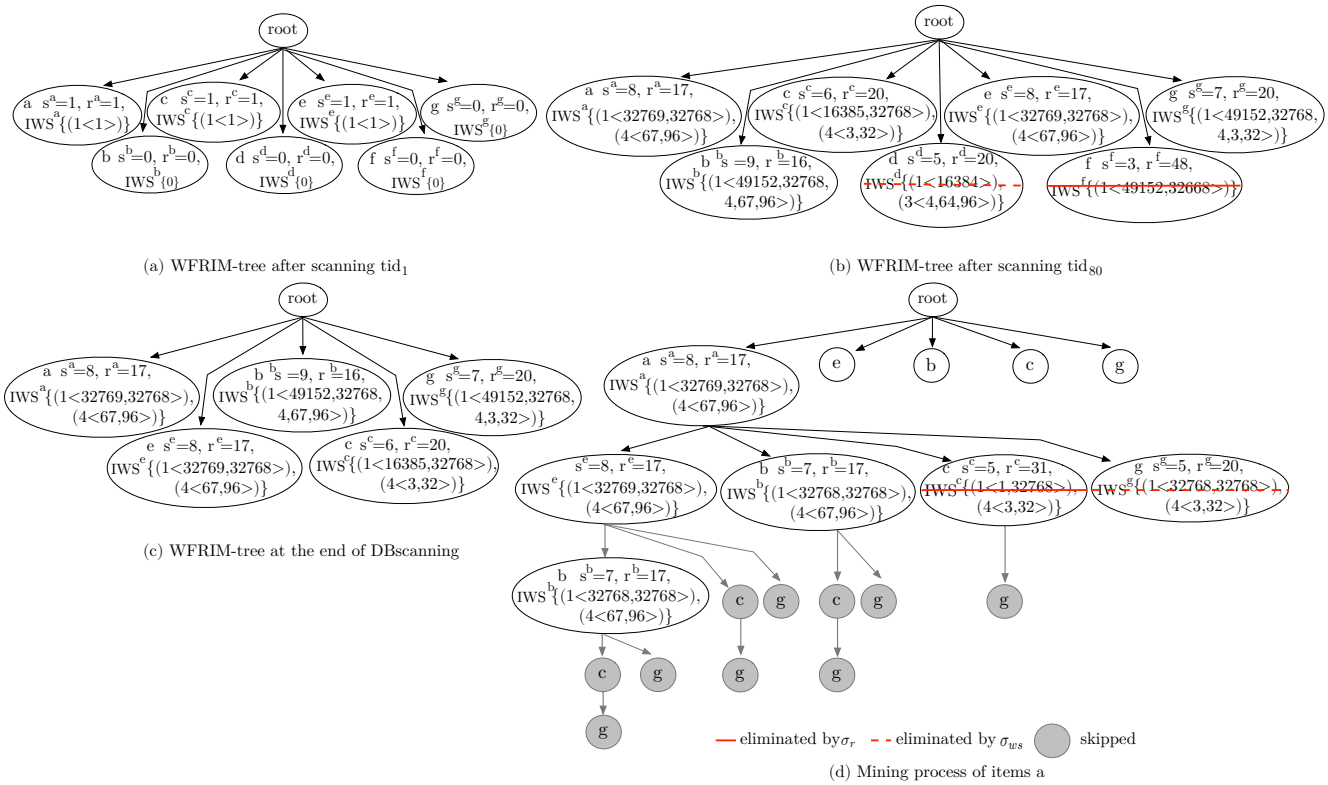


Fig. 5. An example of *WFRIM-IWS* algorithm

merging, calculation on *LMAXW*, intersection and calculation on weighted support are then repeated.

After the itemset Z is generated, the merging of i_j with another item i_k to generate the itemset $Z = i_j \cdot i_k$ is repeated. The IWS^{i_j} and IWS^{i_k} are intersected. The support s^Z and the regularity r^Z are also looked-up. The overestimate weighted support is thus calculated by *LMAXW* (from the previous calculation). If the regularity r^Z is greater than the regularity threshold or the overestimate weighted support ows^Z is less than the weighted support threshold, the itemset Z is then removed from the consideration. Otherwise, a node of Z is created with its IWS^Z and linked to be a child node of i_j. The actual weighted support ws^Z is thus calcualted. Last, the itemset Z is identified and collected as a weighted-frequent-regular itemset if its weighted support is not less than the weighted support threshold.

The merging of i_j with another item is repeated until all child node of H are considered. Then, if i_j have more than one child, the mining process is recursively performed on the node of i_j and all child of i_j will be merged to each other. Otherwise, i_j and a child of i_j are then removed from *WFRIM-tree*.

Example 6: Let the WFRIM-tree from DBscanning contains five items as shown in Fig. 5(c). To mine a complete set of WFRIs, mining step as detailed in Algo. 4 is performed. First, the item 'a' with highest weight and the item 'e' with the second highest weight are merged to be the itemset 'ae'. Then, the local maximum weight and the weight of ae are then calculated as $LMAXW = w^{ae} = \frac{w^a + w^e}{2} = \frac{0.75+0.7}{2} = 0.725$. *The* $IWS^a = \{(1\langle 32769, 32768\rangle), (4\langle 67, 96\rangle)\}$ *and the* $IWS^e = \{(1\langle 32769, 32768\rangle), (4\langle 67, 96\rangle)\}$ *are then intersected and collected into* $IWS^{ae} = \{(1\langle 32769, 32768\rangle), (4\langle 67, 96\rangle)\}$. *Then, the regularity* $r^{ae} = 17$ *and the* $s^{ae} = 8$ *are looked up from the look-up table and the weighted support* ws^{ae} *is calculated as* $ws^{ae} = w^{ae} \times s^{ae} = 0.725 \times 8 = 5.8$, *respectively. Since the regularity* $r^{ae} = 17$ *is less than* $\sigma_r = 20$ *and the weighted support* $ws^{ae} = 5.8$ *is greater than* $\sigma_{ws} = 4.0$, *'ae' is identified as a weighted-frequent-regular itemset. Then, the node of* ae *is created and link as a child node of 'a'.*

Next, the item 'a' is merged with the item 'b' to be 'ab'. The IWS^a *and the* IWS^b *are then intersect to be The* $IWS^{ab} = \{(1\langle 32768, 32768\rangle), (4\langle 67, 96\rangle)\}$. *The regularity* $r^{ab} = 17$ *and the* $s^{ab} = 7$ *are looked up from the look-up table and the overestimate weighted support* ows^{ab} *is calculated as* $ows^{ab} = LMAXW \times s^{ab} = 0.725 \times 7 = 5.075$. *Since the regularity* $r^{ab} = 17$ *is less than* $\sigma_r = 20$ *and the over estimate weighted support* $ows^{ab} = 5.075$ *is greater than* $\sigma_{ws} = 4.0$, *the node of* ab *is created and link as a child node of 'a'. Then, the* $ws^{ab} = w^{ab} \times s^{ae} = \frac{0.75+.065}{2} \times 7 = 4.9$ *is calculated and then* ab *is identified as a weighted-frequent-regular itemset due to its weight support is greater than* σ_{ws}.

The mering of the item 'a' with another item located after the item 'b' is repeated in the same manner as above. Then, if 'a' has more than one child, the process of merging is moved to merge pair of child of 'a'. Otherwise, 'a' and child of 'a' are removed from WFRIM-tree and the merging process is move to merge item 'e' with other items after 'e' as shown in Fig. 5(d).

After finish merging all pairs of items/itemsets in the WFRIM-tree, we then gain a complete set of weighted-frequent-regular itemsets contained in the WFRIM-tree.

Algorithm 3: *DBscanning*

Input: *TDB*, σ_r, σ_{ws}
Output: *WFRIM-Tree*, *WFRIs*

create a *WFRIM-tree* with root R.
create a node of item $i_j \in I$ and set to be a child of R
for each transaction t_p in *TDB* **do**
  for each item i_k in transaction t_p **do**
    collect(IWS^{i_k}, p, lo^{i_k})
    $s^{i_k} \leftarrow s^{i_k} + 1$
    $r^{i_k} \leftarrow max(r^{i_k}, p - lo^{i_k})$
    (if t_p is the first transaction containing i_k, $r^{i_k} \leftarrow p$)
    $lo^{i_k} \leftarrow p$
for each node of item i_j in *WFRIM-tree* **do**
  $r^{i_k} \leftarrow max(r^{i_k}, m - lo^{i_k})$
  (where m is the tid of the last transaction of TDB)
  if $r^{i_k} > \sigma_r$ **then**
    remove node of i_j out of *WFRIM-tree*
repeat
  $GMAXW \leftarrow max(w_{i_1}, w_{i_2}, \ldots, w_{i_{|I|}})$
  for each node of item i_j in *WFRIM-tree* **do**
    $ows^{i_k} \leftarrow GMAXW \times s^{i_k}$
    if $ows^{i_k} < \sigma_{ws}$ **then**
      remove node of i_j out of *WFRIM-tree*
    else
      $ws^{i_j} \leftarrow w_{i_j} \times s^{i_j}$
      $WFRIs \leftarrow WFRIs \cup i_k$ **if** $ws^{i_k} \geq \sigma_{ws}$
until R does not have a child node with $w_{i_j} = GMAXW$
reorder child node of R by weight descending order

TABLE I. CHARACTERISTICS OF DATASETS

Dataset	No. of items	Avg. transactions size	No. of transactions	category
Mushroom	119	23	8,124	dense
Chess	75	37	3,196	dense
T10I4D100K	1,000	10	100,000	sparse
Retail	16,469	10.3	88,162	sparse

IV. EXPERIMENTAL RESULTS

To evaluate the performance of the *WFRIM-IWS* algorithm, experiments on four benchmark datasets (downloaded from http://fimi.ua.ac.be/data and detailed as in table I) were conducted. Three issues *i.e.* computational time, memory usage and the number of discovered results are investigated and compared with *WFRIM* algrithm [12] (*i.e. WFRIM* is the first algorithm on weighted-frequent-regular itemsets mining). The weighted support and the regularity thresholds are set in the same manner as the previous approaches [13], [16], [17], [15], [12] which are in the range of $0.001 - 35\%$ and $0.2 - 6\%$. respectively. Noted that the thresholds used in each dataset is based on density of occurrence of items/itemsets in the dataset. *WFRIM-IWS* and *WFRIM* are implemented in Python and run on a PC with Windows 10, CPU speed at 3.4 GHz, RAM 8GB.

The computational time of both algorithms is shown in Fig. 6 in which each line indicates a computational time based on a fixed value of weighted-support and a variation of regularity threshold. From the figure, it shows that (*i*) the increasing of regularity threshold does not affect computational time of both algorithms on sparse dataset but the computational time of both algorithms on dense datasets increases as the regularity threshold increase, (*ii*) the computational time of both algorithms increases as the weighted support threshold increase

Algorithm 4: *WFRI-mine*

Input: *WFRIM-Tree* with root R, σ_r, σ_{ws}
Output: *WFRIs*

$X \leftarrow \emptyset$ and $w^X \leftarrow 0$
mining (node of R, X, w^X, σ_r, σ_{ws})

Procedure *mining* (node of H, X, w^X, σ_r, σ_{ws})
for each child of H with itemset $U = X \cdot i_p$ **do**
  repeat
    $Y = X \cdot i_q$ is the itemset of another child node of H
    $Z \leftarrow X \cdot i_p \cdot i_q$
    $LMAXW \leftarrow w^Z \leftarrow \frac{(w^X \times |X|) + w_{i_p} + w_{i_q}}{|X|+2}$
    $IWS^Z \leftarrow intersect(IWS^U, IWS^Y)$
    $r^Z \leftarrow lookup\text{-}r(IWS^Z)$
    $s^Z \leftarrow lookup\text{-}f(IWS^Z)$
    $ws^Z \leftarrow w^Z \times s^Z$
  until $r^Z < \sigma_r$ and $ws^Z \geq \sigma_{ws}$
  create a node for itemset Z with its information and then set to be a child node of U
  $WFRIs \leftarrow WFRIs \cup Z$
  for each child node of H with itemset $V = X \cdot i_q$ **do**
    $Z \leftarrow X \cdot i_p \cdot i_q$
    $IWS^Z \leftarrow intersect(IWS^U, IWS^V)$
    $r^Z \leftarrow lookup\text{-}r(IWS^Z)$
    $s^Z \leftarrow lookup\text{-}f(IWS^Z)$
    $ows^Z \leftarrow LMAXW \times s^Z$
    if $r^Z < \sigma_r$ and $ows^Z \geq \sigma_{ws}$ **then**
      create a node for itemset Z with its information and then set to be a child node of U
      $ws^Z \leftarrow s^Z \times \frac{((w^X \times |X|) + w_{i_p} + w_{i_q})}{|X|+2}$
      if $ws^Z \geq \sigma_{ws}$ **then**
        $WFRIs \leftarrow WFRIs \cup Z$
  if U has more than one child **then**
    mining (node of U, U, $\frac{w^X \times |X| + w_{i_p}}{|X|+1}$, σ_r, σ_{ws})
  else
    remove U and the child of U out of *WFRIM-tree*

(Noted that with the increasing of weight support threshold, there are more and more items/itemsets meets the threshold and then both algorithms have to take more time to consider these items/itemsets), (*iii*) *WFRIM-IWS* significantly outperforms on computational time than *WFRIM* up to $60 - 70\%$ on sparse datasets and $40 - 98\%$ on dense datasets, respectively.

For the memory aspect, Fig. 7 shows the memory usage of both algorithms in which the *WFRIM-IWS* algorithm consumes less memory than *WFRIM* (thanks to the benefit of *IWS* structure). However, on T10I4D100K, *WFRIM-IWS* consumes more memory than *WFRIM*. It is because based on the setting of threshold, there are only weighted-freuqent-regular items generated. Then, *WFRIM* does not need to recursively create *WFRI-tree* (each *WFRI-tree* consumes high memory). Thus, the memory usage of *WFRIM* is less than *WFRIM-IWS*.

Last, the number of discovered weighted-frequent-regular itemsets is shown in Fig. 8 which can conclude that (*i*) the increasing of weighted-support threshold results in the decreasing of results and (*ii*) the increasing of regularity threshold causes the increasing of results, respectively.

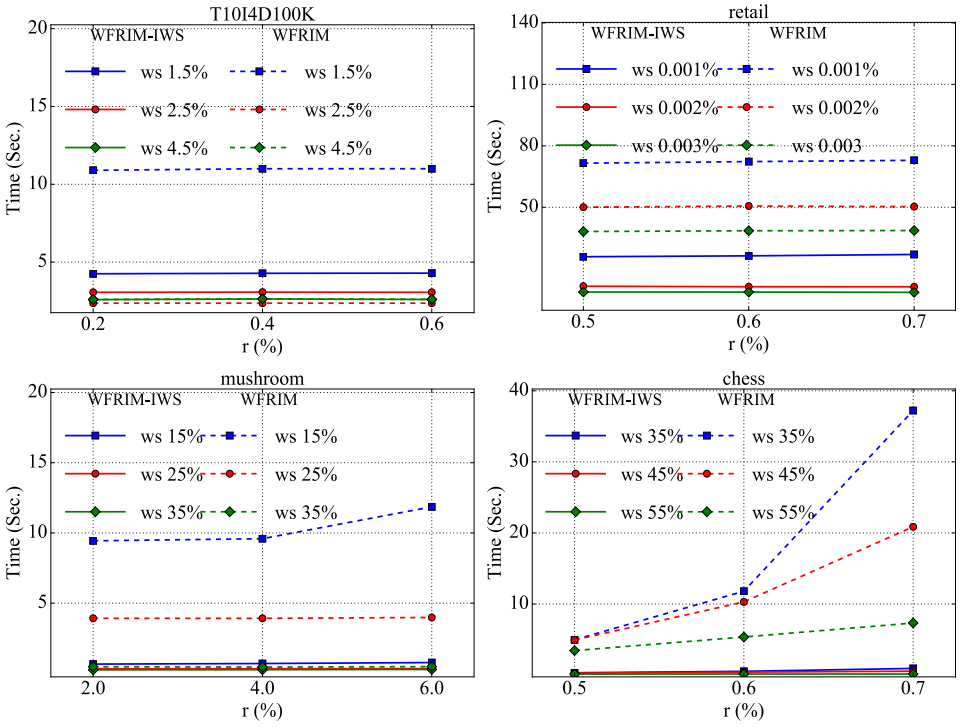


Fig. 6. Computational time of the both algorithms for *WFRIM* under different σ_r and σ_{ws}.

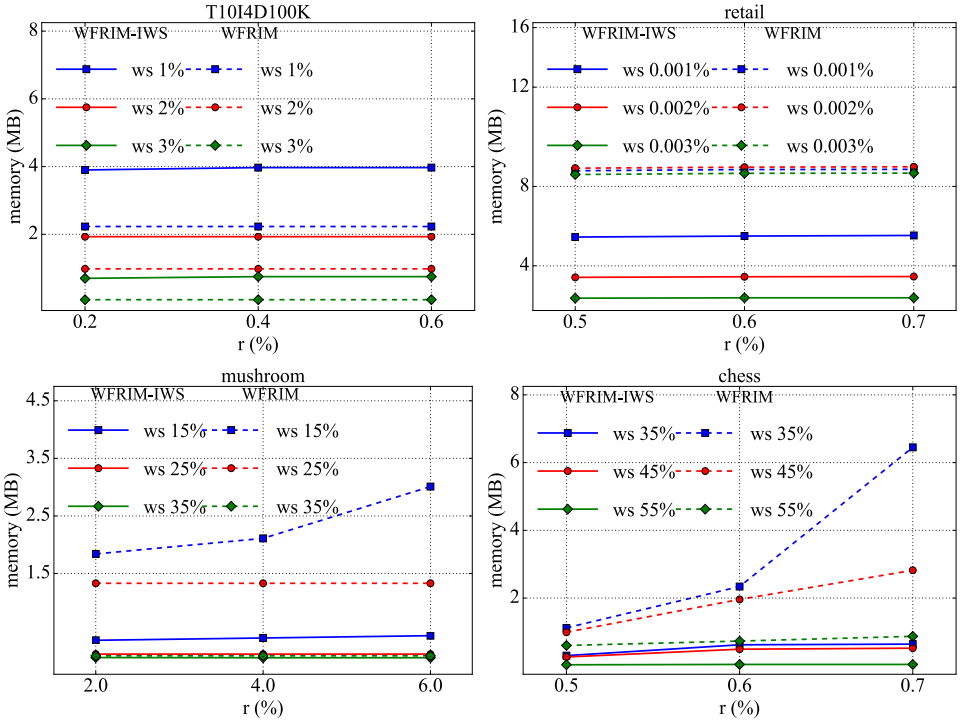


Fig. 7. Memory usage of the both algorithms for *WFRIM* under different σ_r and σ_{ws}.

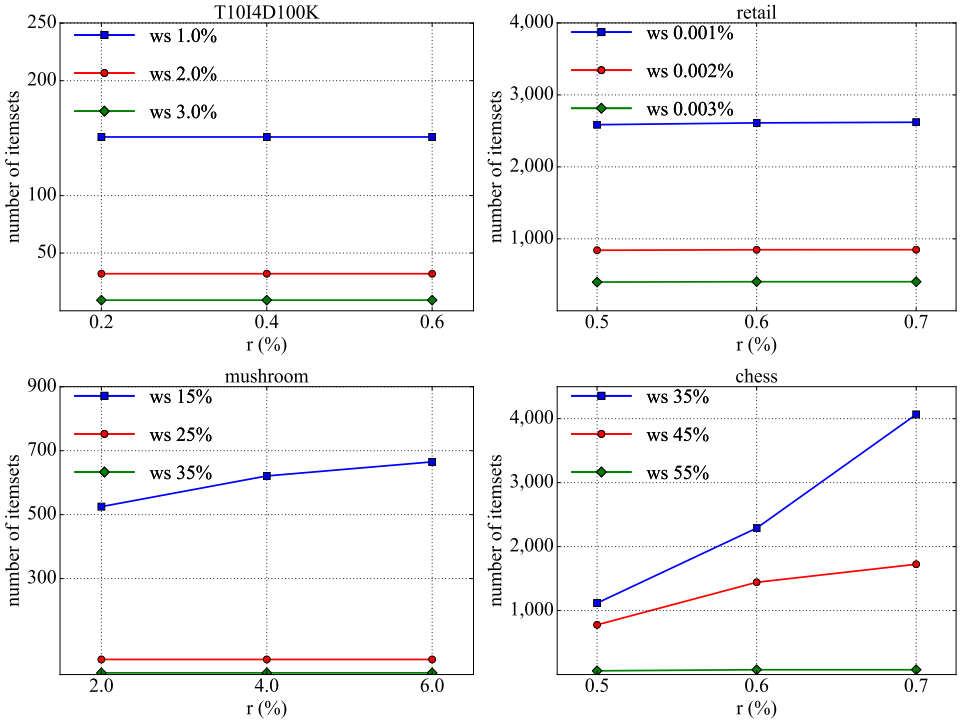


Fig. 8. The number of itemsets discovered from *WFRIM*.

V. Conclusion

In this paper, we have proposed a new efficient single-pass algorithm named *WFRIM-IWS* for mining weighted-frequent-regular itemsets. *WFRIM-IWS* applied the interval word segment structure to maintain occurrence information of itemsets. A new look-up table on an *IWS* is designed to quickly look-up on support and regularity of the itemset. A tree structure called WFRIM-tree and the depth-first search strategy are adopt to maintain and generate itemsets during mining process. Moreover, the concepts of overestimated weighted-frequency and global/local maximum weights are applied to prune search space. Experiments on both real and synthetic datasets showed that the proposed *WFRIM-IWS* algorithm outperforms the previous algorithm on mining frequent-weighted-regular itemsets and it is efficient in the terms of computational time and memory usage.

References

[1] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *SIGMOD Rec.*, vol. 22, no. 2, pp. 207–216, 1993.

[2] H.-F. Li and S.-Y. Lee, "Mining frequent itemsets over data streams using efficient window sliding techniques," *Expert Systems with Applications*, vol. 36, no. 2, Part 1, pp. 1466 – 1477, 2009.

[3] S. Moens, E. Aksehirli, and B. Goethals, "Frequent itemset mining for big data," in *2013 IEEE International Conference on Big Data*, 2013, pp. 111–118.

[4] C. C. Aggarwal, Y. Li, J. Wang, and J. Wang, "Frequent pattern mining with uncertain data," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09, 2009, pp. 29–38.

[5] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the Eleventh International Conference on Data Engineering*, 1995, pp. 3–14.

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

[7] S. K. Tanbeer, C. F. Ahmed, B.-S. Jeong, and Y.-K. Lee, "Discovering periodic-frequent patterns in transactional databases," in *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, 2009, pp. 242–253.

[8] K. Amphawan, P. Lenca, and A. Surarerks, "Mining top-k periodic-frequent patterns without support threshold," in *Proceedings of the 3rd International Conference on Advances in Information Technology*, vol. 55, 2009, pp. 18–29.

[9] C. F. Ahmed, S. K. Tanbeer, B.-S. Jeong, Y.-K. Lee, and H.-J. Choi, "Single-pass incremental and interactive mining for weighted frequent patterns," *Expert Systems with Applications*, vol. 39, no. 9, pp. 7976 – 7994, 2012.

[10] U. Yun and J. J. Leggett, "Wspan: Weighted sequential pattern mining in large sequence databases," in *2006 3rd International IEEE Conference Intelligent Systems*, 2006, pp. 512–517.

[11] G. Lee, U. Yun, and K. H. Ryu, "Sliding window based weighted maximal frequent pattern mining over data streams," *Expert Systems with Applications*, vol. 41, no. 2, pp. 694 – 708, 2014.

[12] K. Klangwisan and K. Amphawan, "Mining weighted-frequent-regular itemsets from transactional database," in *2017 9th International Conference on Knowledge and Smart Technology (KST)*, Feb 2017, pp. 66–71.

[13] F. Tao, F. Murtagh, and M. Farid, "Weighted association rule mining using weighted support and significance framework," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '03, 2003, pp. 661–666.

[14] H. Nguyen, B. Vo, M. Nguyen, and W. Pedrycz, "An efficient algorithm for mining frequent weighted itemsets using interval word segments," *Applied Intelligence*, vol. 45, no. 4, pp. 1008–1020, 2016.

[15] K. Amphawan and P. Lenca, "Mining top-k frequent-regular closed patterns," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7882 – 7894, 2015.

[16] U. Yun and J. J. Leggett, "Wfim: Weighted frequent itemset mining with a weight range and a minimum weight," in *Proceedings of the 2005 SIAM International Conference on Data Mining*, 2005, pp. 637–640.

[17] C. F. Ahmed, S. K. Tanbeer, B.-S. Jeong, and Y.-K. Lee, *Mining Weighted Frequent Patterns in Incremental Databases*, 2008, pp. 933–938.

Conference Program

the 2018 - 10th International Conference on Knowledge and Smart Technology (KST)

Kantary Hills Hotel, Chiangmai, Thailand
Organized by KST Research Lab, Faculty of Informatics, Burapha University, Chonburi, Thailand
January 31 - February 3, 2018

January 31, 2018	**15.00-16.30**	Registration			
	16.30-17.30	Local Organizing & Technical Program Committee Meeting			
	17:30-20:30	Welcome Reception for Steering & Technical Program Committee and Keynote Speakers @Goodview Village Chiangmai			
February 1, 2018	**08.00-08.40**	Registration			
	08.40-09.00	**Opening Ceremony**			
	09.00-09.45	**Keynote-I**	Dr. Sun-Hwa Hahn (Korea Institute of Science and Technology Information, Republic of Korea)		
			Big data-driven Biomedical research		
	09.45-10.30	**Keynote-II**	Professor Dr. Pascal Bouvry (University of Luxembourg, Luxembourg)		
			Resource allocation in the Cyberworld		
	10.30-11.00	Break			
		PaperID	**Doi Luang meeting room** **Regular: Computational Intelligence** **Chair: Komate Amphawan and Ureerat Suksawatchon (BU, Thailand)**	**PaperID**	**Doi Nang meeting room** **Regular: Emerging Intelligent Technologies** **Chair: Paramate Horkaew (SUT, Thailand)**
	11.00-11.20	1570400478	Integrating Bat Algorithm to Ordinal Optimization for Solving the Facility-Sizing Problem; By Shih-Cheng Horng; Shieh-Shing Lin	1570404035	A Framework for Recommender System Based on Game Theory in Social Networks; By Lu Yang; Tao Hong; Anilkumar Kothalil Gopalakrishnan
	11.20-11.40	1570403905	Sampled-data control for continuous-time Markovian jump fuzzy systems; By Junmin Park; PooGyeon Park	1570404049	Short-Term Sales Forecast of Perishable Goods for Franchise Business; By Chao-Lung Yang; Hendri Sutrisno
	11.40-12.00	1570404766	Measure of the Ability Dominance of Tournament Systems and Subjective Judgment; By Tomoyuki Maekawa; Hidehito Honda; Kazuhiro Ueda	1570412878	Automatically Identifying Themes and Trends in Software Engineering Research; By Kenneth Cosh; Sakgasit Ramingwong; Narissara Eiamkanitchat; Lachana Ramingwong
	12.00-13.00	Lunch Steering Committee Meeting & International Consortium in Informatics Meeting / Free Discussion @Doisaket Ballroom			

		PaperID	**Doi Luang meeting room** **Regular: Computational Intelligence** **Chair: Khamron Sunat (KKU, Thailand)**	**PaperID**	**Doi Nang meeting room** **Regular: Emerging Intelligent Technologies** **Chair: Antony Harfield (NU, Thailand)**
February 1, 2018	**13.00-13.20**	1570409242	q-Sine Circular Extreme Learning Machine for High Dimensional Data; By Sarutte Atsawaraungsuk; Narin Thipayang	1570413025	Perceived Usability of Teleworking Options in Creative Knowledge Work; By Aaro Hazak
	13.20-13.40	1570409870	Optimizing Location-Routing Problem using Iterative Combination of GA and VNS; By Cheng Jiang; Worapan Kusakunniran	1570413401	A Novel Approach to Extract Important Keywords from Documents Applying Latent Semantic Analysis; By H. M. Mahedi Hasan; Falguni Sanyal; Dipankar Chaki
	13.40-14.00	1570412146	Feature Selection for Composer Classification Method using Quantity of Information; By Ayaka Takamoto; Mitsuo Yoshida; Kyoji Umemura; Yuko Ichikawa	1570413571	Development of Digital Media using Augmented Reality for HM King Prajadhipok's Interests in Arts and Culture; By Porawat Visutsak; Fuangfar Pensiri
	14.00-14.20	1570412670	A Differential Evolution-based Rule Ordering of Cellular Automata for Classification; By Pattapon Wanna; Sartra Wongthanavasu; Jetsada Ponkaew		
	14.20-14.50	Break			

		PaperID	**Doi Luang meeting room** **Regular: Computational Intelligence** **Chair: Chakchai So-In and Sirapat Chiewchanwattana (KKU, Thailand)**	**PaperID**	**Doi Nang meeting room** **Regular: Intelligent applications** **Chair: Pusadee Seresangtakul and Khamron Sunat (KKU, Thailand)**
February 1, 2018	**14.50-15.10**	1570412727	Time Series based Gastropod Classification; By Janya Onpans; Nutthanon Leelathakul; Sunisa Rimcharoen	1570398057	3D Reconstruction of Long Bone using Kinect; By Thanandon Imaromkul; Wiphada Dendee; Sarocha Chokewiwattana; Worapan Kusakunniran
	15.10-15.30	1570412755	Curriculum Analysis Based on Cerebral Hemisphere Functions Using Association Rule; By Xangyang; Narissara Eiamkanitchat; Sakgasit Ramingwong; Lachana Ramingwong	1570400875	NavTU: Android Navigation App for Thai People with Visual Impairments; By Nawin Somyat; Teepakorn Wongsansukjaroen; Wuttinan Longjaroen; Songyot Nakariyakul
	15.30-15.50	1570412782	3D Model Compression over ASCII Encoded Using Rotational and Reflective Symmetry; By Thanyathon Tayangkanon; Pavadee Sompagdee; Xin Li	1570403939	Thai Lottery Number Reader App for Blind Lottery Ticket Sellers; By Nawin Somyat; Songyot Nakariyakul
	15.50-16.10	1570412850	Efficient weighted-frequent-regular itemsets mining using interval word segments structure; By Kittipa Klangwisan; Komate Amphawan	1570405076	Multi Sensor based Approach for Road Region Extraction for Autonomous Vehicle; By Ranjith Rochan; Aarthi Alagammai; Sujatha J
	16.10-17.00	**Move to Night Safari Chiangmai**			
	17.00-20:30	**Welcome Reception / Banquet @Night Safari Chiangmai**			

February 2, 2018		Registration			
		PaperID	Doi Luang meeting room Regular: Computational Intelligence Chair: Kittichai Lavangnananda (KMUTT, Thailand)	PaperID	Doi Nang meeting room Regular: Intelligent applications Chair: Porawat Visutsak (KMUT-NB, Thailand) and Fuangfar Pensiri (SDU, Thailand)
	08.50-09.10	1570412928	Discovery and Visualization of Expertise Evolution and Tendency; By Kallaya Songklang; Akara Prayote	1570407645	Armband Gesture Recognition on Electromyography Signal for Virtual Control; By Tanasanee Phienthrakul
	09.10-09.30	1570413067	Mining Acceleration Data for Smartphone-based Fall Detection; By Luepol Pipanmaekaporn; Puritud Wichinawakul; Suwatchai Kamolsantiroj	1570409200	DWT/DCT-based Invisible Digital Watermarking Scheme for Video Stream; By Jantana Panyavaraporn; Paramate Horkaew
	09.30-09.50	1570413225	Longest Matching and Rule-based Techniques for Khmer Word Segmentation; By Pakrigna Long; Veera Boonjing	1570410379	Lowercase Letters in Text-Based CAPTCHA: A Visual Perception Analysis; By Chatpong Tangmanee
	09.50-10.10	1570413383	A Lossless Image Compression Algorithm using Differential Subtraction Chain; By Chiratheep Chianphatthanakit; Anuparp Boonsongsrikul; Somjet Suppharangsan	1570412665	Polar Space Contour Detection for Automated Optic Cup Segmentation; By Wuttichai Luangruangrong; Krisana Chinnasarn
	10.10-10.30	1570413588	Creative Knowledge Employees' Assesment of Flexitime Utilisability; By Aaro Hazak	1570412973	Automatic Crochet Pattern Generation from 2D Sketching; By Pikanate Nakjan; Sukanya Ratanotayanon; Natchayar Porwongsawang
	10.30-11.00	Break			

		PaperID	Doi Luang meeting room Regular: Intelligent Computer Networks and Systems Chair: Paramate Horkeaw (SUT, Thailand)	PaperID	Doi Nang meeting room Regular: Intelligent Computer Networks and Systems Chair: Anuparp Boonsongsrikul (BUU, Thailand)
February 2, 2018	11.00-11.20	1570393535	An Intelligent Renewables-based Power Scheduling System for Internet of Energy; By Chenn-Jung Huang; Jui-Ting Hsiao; Chao-Yang Deng; Kai-Wen Hu	1570412876	A System for Ultraviolet Monitoring, Alert, and Prediction; By Chanon Purananunak; Saruda Yanavanich; Chanawong Tongpoon; Tanasanee Phienthrakul
	11.20-11.40	1570397331	Real Time Air Quality Monitoring; By Sumanth Reddy Enigala; Hamid Shahnasser	1570413083	Activity Recognition of Multiple Subjects for Homecare; By Oscal T.-C. Chen; Hung Manh Ha; Wei-Chih Lai
	11.40-12.00	1570402962	Multi-step-ahead Host Load Prediction with GRU Based Encoder-Decoder in Cloud Computing; By Chenglei Peng; Yang Li; Yao Yu; Yu Zhou; Sidan Du	1570413552	A Security Architecture Framework for Critical Infrastructure with Ring-based Nested Network Zones; By Sarayut Chaisuriya; Somnuk Keretho; Surasak Sanguanpong; Prasong Praneetpolgrang
	12.00-13.00	Lunch			

		PaperID	Doi Luang meeting room Regular: Intelligent Computer Networks and Systems Chair: Prajaks Jitngernmadan and Pusit Kulkasem (BUU, Thailand)	PaperID	Doi Nang meeting room Special Session: Cognitive Science and Human Perception Chair: Prof. Katsumi Watanabe (Waseda University / the University of Tokyo, Japan) and Prof. Roberto Caldara (University of Fribourg, Switzerland)
February 2, 2018	13.00-13.20	1570403758	Smart Detection and Reporting of Potholes via Image-Processing using Raspberry-Pi Microcontroller; By Mae Garcillanosa; Jian Mikee Pacheco; Rowie Reyes; Junelle Joy San Juan	1570403816	Audiovisual semantic congruency effect with onomatopoeia; By Antonio Fidalgo; Kohske Takahashi; Aiko Murata; Katsumi Watanabe
	13.20-13.40	1570403965	Application for outdoor dust monitoring using RF wireless power transmission; By Hyun-Sik Choi	1570405404	Prosopagnosia or Prosopdysgnosia - Facing up to a change of concepts; By Thomas Alrik Sørensen; Morten Storm Overgaard
	13.40-14.00	1570407546	Wireless Sensor Networks for Microclimate Monitoring in Edamame Farm; By Jigme Norbu; Theerapat Pobkrut; Satetha Siyang; Chayanin Khunarak; Thinley Namgyel; Teerakiat Kerdcharoen	1570412139	Is Choice Overload Replicable?; By Atsunori Ariga
	14.00-14.20	1570408127	A Study of Air Pollution Smart Sensors LPWAN via NB-IoT for Thailand Smart Cities 4.0; By Sarun Duangsuwan; Aekarong Takarn; Rachan Nujankaew; Punyawi Jamjareegulgarn	1570412326	Link between color-space association, left-right confusion, mirror image copy, and autistic traits; By Hanako Ikeda; Makoto Wada; Katsumi Watanabe
	14.20-14.50	Break			

February 2, 2018		PaperID	Doi Luang meeting room Regular: Intelligent Computer Networks and Systems Chair: Kunnate Anupharwan and Pusit Kulkasem (BUU, Thailand)		Doi Nang meeting room Special Session: Image Processing, Computer Vision, Human Perception, Knowledge-based System and its Applications Chair: Montri Phothisonothai (KMITL, Thailand)
	14.50-15.10	1570408749	Prediction of Acidity Levels of Fresh Roasted Coffees Using E-nose and Artificial Neural Network; By Yu Thazin; Theerapat Pobkrut; Teerakiat Kerdcharoen	1570412711	Readiness of Local Government Websites for Eastern Economic Corridor (EEC); By Prajaks Jitngernmadan
	15.10-15.30	1570408755	Effects of supplementary LED light on the growth of lettuce in a smart hydroponic system; By Thinley Namgyel; Chayanin Khunarak; Satetha Siyang; Theerapat Pobkrut; Jigme Norbu; Teerakiat Kerdcharoen	1570413316	Building Minimal Classification Rules for Breast Cancer Diagnosis; By Phonethep Douangnoulack; Veera Boonjing
	15.30-15.50	1570410138	Bi-slotted Fast Query Tree-based Anti-Collision Algorithm for Large Scale of RFID Systems; By Yoschanin Sasiwat	1570413593	Detection and Recognition of the Myanmar Characters from the Dissimilar Images; By Ohnmar Khin; Montri Phothisonothai; Somsak Choomchuay
	15.50-16.10	1570410208	A smart photovoltaic system with Internet of Thing: A case study of the smart agricultural greenhouse; By Anukit Saokaew; Oran Chieochan; Ekkarat Boonchieng	1570417238	Distributed Scheduling of Electric Vehicles in a Residential Area in Thailand; By Tumisang K Nguvauva; Somsak Kittipiyakul
	16.10-16.30	1570412827	Smart Hydroponic Lettuce Farm using Internet of Things; By Tanabut Changmai; Sethavidh Gertphol; Pariyanuj Chulaka	1570413573	Classification of in vitro blood stages of Plasmodium falciparum Based on Fuzzy Inference System; By Suchada Tantisatirapong; Montri Phothisonothai
	16.30	**Closing Ceremony**			

February 3, 2018	**08.30-09.00**	Registration
	09.00-12.00	Thailand Research Consortium for Informatics Field Trips (Chiangmai University)
	12.00	**Conclude Consortium**

Certificate of Contributions

Kittipa Klangwisan and Komate Amphawan

Entitled

Efficient weighted-frequent-regular itemsets mining using interval word segments structure

Has Contributed To

The 2018 - 10th International Conference on Knowledge and Smart Technology (KST)

January 31 - February 3, 2018

Kantary Hills, Chiang Mai, Thailand

Organized by

Faculty of Informatics, Burapha University, Thailand

Chidchanok Lursinsap, Ph.D.
Faculty of Science, Chulalongkorn University
General Chair

Krisana Chinnasarn, Ph.D.
Faculty of Informatics, Burapha University
Dean

Technical Co-Sponsor