การวิเคราะห์พฤติกรรมผู้บริโภคด้วยการวิเคราะห์ความเปลี่ยนแปลงของพฤติกรรมการซื้อสินค้า

สุมาลี อิสริโยดม

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา
กรกฎาคม 2561

CONSUMERS' BEHAVIOR ANALYSIS BASED ON ANALYSIS OF CHANGING OF BUYING BEHAVIOR

SUMALEE EISARIYODOM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE MASTER DEGREE OF SCIENCE
IN COMPUTER SCIENCE
FUCULTY OF INFORMATICS
BURAPHA UNIVERSITY
JULY 2018

คณะกรรมการควบคุมวิทยานิพนธ์และคณะกรรมการสอบวิทยานิพนธ์ ได้พิจารณาวิทยานิพนธ์ของ สุมาลี อิสริโยดม 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)

คณะวิทยาการสารสนเทศอนุมัติให้รับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ของมหาวิทยาลัยบูรพา

.....คณบดีคณะวิทยาการสารสนเทศ
(ผู้ช่วยศาสตราจารย์ ดร. กฤษณะ ชินสาร)

วันที่ 23 เดือน กรกฎาคม พ.ศ. 2561

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยความกรุณาจาก ผู้ช่วยศาสตราจารย์ ดร.โกเมศ อัมพวัน อาจารย์ที่ปรึกษาหลัก ที่กรุณาให้คำปรึกษา แนะนำแนวทางที่ถูกต้อง มอบประสบการณ์ที่ดี คอยผลักดันและให้กำลังใจ ตลอดจนแก้ไขข้อบกพร่องต่าง ๆ ด้วยความละเอียดถี่ถ้วนและเอาใจใส่ด้วยดีเสมอมา ผู้วิจัยรู้สึกซาบซึ้งเป็นอย่างยิ่ง จึงขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอกราบขอบพระคุณ ครอบครัวอิสริโยดม ญาติ พี่ เพื่อน น้อง ทุก ๆ คน ที่คอยให้แง่คิดที่ดี ให้กำลังใจ ให้คำปรึกษา คอยอยู่เคียงข้างในยามที่ท้อและเหนื่อย คอยผลักดันให้สู้ ให้ก้าวเดินต่อไป และคอยสนับสนุนผู้วิจัยเสมอมา

ขอขอบพระคุณอาจารย์และพี่ ๆ บุคลากร เพื่อน และน้อง ทุก ๆ คน ในคณะวิทยาการสารสนเทศ ที่มอบมิตรภาพที่ดี คอยช่วยเหลือ คอยให้คำปรึกษาให้กับผู้วิจัยเสมอมา

ขอบคุณตนเองที่เข้มแข็ง อดทน พยายาม ต่ออุปสรรคต่าง ๆ ที่เข้ามาและรับผิดชอบในสิ่งที่ตนเองเลือกได้สำเร็จ

คุณค่าและประโยชน์ของวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบเป็นกตัญญูกตเวทิตาแด่บุพการี บูรพาจารย์ และผู้มีพระคุณทุกท่านทั้งในอดีตและปัจจุบัน ที่ทำให้ข้าพเจ้าเป็นผู้ที่มีการศึกษาและประสบความสำเร็จมาจนตราบเท่าทุกวันนี้

สุมาลี อิสริโยดม

56910116: สาขาวิชา: วิทยาการคอมพิวเตอร์; วท.ม. (วิทยาการคอมพิวเตอร์)
คำสำคัญ: การวิเคราะห์พฤติกรรมผู้บริโภค การวิเคราะห์ข้อมูลการซื้อสินค้า การวิเคราะห์ข้อมูลเพื่อสนับสนุนการตัดสินใจ การทำเหมืองข้อมูล การค้นหาเซตรายการ (รูปแบบ) ความเปลี่ยนแปลงของพฤติกรรม

สุมาลี อิสริโยดม: การวิเคราะห์พฤติกรรมผู้บริโภคด้วยการวิเคราะห์ความเปลี่ยนแปลงของพฤติกรรมการซื้อสินค้า (CONSUMERS' BEHAVIOR ANALYSIS BASED ON ANALYSIS OF CHANGING OF BUYING BEHAVIOR)
คณะกรรมการควบคุมงานวิทยานิพนธ์: ผู้ช่วยศาสตราจารย์ ดร. โกเมศ อัมพวัน, Ph.D. 96 หน้า. ปี พ.ศ. 2561.

ในปัจจุบันเป็นยุคที่ธุรกิจอยู่ในภาวะที่มีการแข่งขันสูง ทำให้การวิเคราะห์พฤติกรรมผู้บริโภคจึงเป็นประเด็นปัญหาที่สำคัญ โดยองค์กรและธุรกิจต่าง ๆ จำเป็นต้องทราบถึงพฤติกรรมของผู้บริโภครวมถึงการเปลี่ยนแปลงพฤติกรรมของผู้บริโภค ด้วยเหตุนี้ จึงนำไปสู่การค้นหาเซตรายการที่น่าสนใจด้วยการวิเคราะห์ความเปลี่ยนแปลงของพฤติกรรมผู้บริโภค ที่ซึ่งมีงานวิจัยหนึ่งที่น่าสนใจเกี่ยวกับความเปลี่ยนแปลงของการปรากฏขึ้นในแง่มุมของความถี่ที่เพิ่มขึ้นเมื่อเวลาผ่านไป แต่อย่างไรก็ตามการค้นหาเซตรายการภายใต้ความเปลี่ยนแปลงลักษณะของการปรากฏในด้านของการปรากฏอย่างสม่ำเสมอก็เป็นแง่มุมที่น่าสนใจเช่นกัน ที่ซึ่งจะช่วยให้นำไปใช้ประโยชน์ได้อย่างกว้างขวางในหลาย ๆ ด้าน อาทิเช่น การติดตามการเปลี่ยนแปลงพฤติกรรมการซื้อสินค้าของผู้บริโภคในร้านค้าปลีก การสังเกตการเปลี่ยนแปลงความสม่ำเสมอของผลกระทบหลังจากที่ผู้ป่วยมีการใช้ยา การสังเกตการเปลี่ยนแปลงความสม่ำเสมอของเกณฑ์สำหรับการจองโรงแรมของนักท่องเที่ยวและอื่น ๆ ดังนั้นในงานวิทยานิพนธ์นี้จะนำเสนอ 1) การค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น ด้วยการพิจารณาความเปลี่ยนแปลงของพฤติกรรมการปรากฏขึ้นในแง่มุมความสม่ำเสมอที่เพิ่มขึ้นเมื่อเวลาผ่านไปภายใต้ค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนดด้วยขั้นตอนวิธีไมโคร ที่ซึ่งประยุกต์ใช้โครงสร้างต้นไม้ที่เรียกว่า อิคโคร-ทรี ที่ช่วยให้อ่านข้อมูลจากฐานข้อมูลรายการเพียงครั้งเดียวเท่านั้น และยังมีการลดทอนจากสมบัติปิดการลดลง เพื่อลดเวลาในการประมวลผลข้อมูลและลดพื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลได้อย่างมีประสิทธิภาพ อย่างไรก็ตามขั้นตอนวิธีนี้ ได้มีสร้างผลลัพธ์เป็นจำนวนมาก (Overwhelming) ทำให้ผู้ใช้หรือผู้ที่สนใจไม่สามารถนำผลลัพธ์ดังกล่าวไปใช้งานได้หรือวิเคราะห์ได้ และผลลัพธ์ที่ได้ก็อาจจะไม่น่าสนใจหรือบ่งบอกถึงข้อมูลที่สำคัญได้ ด้วยเหตุนี้จึงได้นำเสนอ 2) การค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้น ด้วยการพิจารณาเซตรายการที่ปรากฏสม่ำเสมอภายใต้ค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนดและแนวโน้มความเปลี่ยนแปลงของพฤติกรรมการปรากฏขึ้นอย่างสม่ำเสมอที่เพิ่มขึ้นเมื่อเวลาผ่านไปภายใต้ค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนดด้วยขั้นตอนวิธีริคครอม และประยุกต์ใช้โครงสร้างข้อมูลที่เรียกว่า *NIWS* สำหรับจัดเก็บข้อมูลที่ปรากฏขึ้นของแต่ละเซตรายการ โดยจะทำการทดลองในฐานข้อมูลรายการจริงและฐานข้อมูลรายการที่ถูกสังเคราะห์ขึ้น ที่ซึ่งจะแสดงประสิทธิภาพในด้านของเวลาที่ใช้ในการประมวลผลข้อมูล พื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูล และจำนวนผลลัพธ์ของเซตรายการที่ค้นพบ

56910116: MAJOR: COMPUTER SCIENCE; M.S. (COMPUTER SCIENCE)
KETWORDS: CUSTOMERS' BEHAVIOR ANALYSIS/ BUYING BEHAVIOR ANALYSIS/ DECISION SUPPORT DATA ANALYSIS/ DATA MINING/ ITEMSETS (PATTERN) MINING/ CHANGE ON BEHAVIOR

SUMALEE EISARIYODOM: CONSUMERS' BEHAVIOR ANALYSIS BASED ON ANALYSIS OF CHANGING OF BUYING BEHAVIOR. ADVISORY COMMITTEE: KOMATE AMPHAWAN, Ph.D. 96 P. 2018.

Nowadays, consumers' behavior analysis is a crucial issue in competitive business. There is a need to know consumers' behavior including changes of consumers' behavior. This leads to an emergence of mining interesting itemset. Since the first proposed to discover emerging patterns (*EPs*) which can help to know trends and differences on occurrences of itemsets in the term of frequency. However, mining *EPs* only considers changing on frequency of occurrence of itemsets which may not sufficient to express change on regularity or irregularity of itemsets in several real-life applications such as tracking changes of buying behavior, monitoring changes of effects on patients after using medicines, observe change in travelers preferences of hotel business and so on. To solve the above limitation, we propose to (*i*) Discovering interesting itemsets based on change in regularity of occurrence. An efficient single-pass algorithm based on pattern-growth concept named *MICRO*. A tree-based structure called *ICRO-tree* is also designed to efficiently maintain candidate itemsets with their essential information. A property used for pruning search space is also introduced in order to reduce resource usage during mining process. However, this approach overwhelming of generated results and difficulties to the users. Hence, it is helpful to avoid this which can help users to be more efficient to look for interesting information and/or knowledge from these itemsets. Therefore, to address this issue, we propose to (*ii*) Mining regular itemsets with interesting changes in regularity of occurrence in order to generate a compact set of results based on the user-given regularity and change thresholds. An efficient single-pass algorithm named *RICROM* and a new interval word segment structure called *NIWS* are designed to efficiently mine such itemsets and maintain occurrence information of each itemset. Experiments were done in real and synthetic datasets. The results illustrate the efficiency of runtime, memory usage and the number results of discovered.

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญภาพ

สารบัญภาพ (ต่อ)

สารบัญภาพ (ต่อ)

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ณ ปัจจุบันเป็นยุคที่เศรษฐกิจอยู่ในภาวะฝืดเคือง การค้าขายสินค้า/บริการสามารถทำได้ยากมากขึ้น เนื่องจากบริษัท/องค์กรต่าง ๆ มีนโยบายในการประหยัดค่าใช้จ่ายมากยิ่งขึ้น และไม่เว้นแม้แต่ผู้บริโภครายย่อยก็เช่นกัน นอกจากนั้นในการทำธุรกิจหนึ่ง ๆ ที่ไม่ใช่ธุรกิจที่มีลักษณะเป็นแบบผูกขาดไม่ว่าจะเป็นธุรกิจขนาดเล็ก ธุรกิจ SME ธุรกิจขนาดกลางหรือธุรกิจขนาดใหญ่จะมีบริษัทที่พยายามช่วงชิงส่วนแบ่งทางการตลาดเป็นจำนวนมาก ด้วยเหตุนี้จึงเป็นเหตุให้บริษัท/องค์กรต่าง ๆ ได้มีการประยุกต์ใช้ข้อมูล ข่าวสาร และสารสนเทศเพื่อช่วยในการดำเนินธุรกิจให้มีความสะดวก ถูกต้อง แม่นยำมากขึ้น โดยในการประยุกต์ใช้ข้อมูล ข่าวสาร และสารสนเทศมักใช้เป็นข้อมูลส่วนเสริมในการประกอบการตัดสินใจที่จะกำหนดและวางแผนการดำเนินธุรกิจเพื่อที่จะเพิ่มยอดขาย เพิ่มผลกำไร หรือลดต้นทุนของการดำเนินธุรกิจ

ในการดำเนินธุรกิจโดยส่วนใหญ่มักจะมุ่งเน้นที่การสร้างความพึงพอใจให้แก่ลูกค้าและการรักษาฐานลูกค้า ที่ซึ่งจะเป็นการรักษาส่วนแบ่งทางการตลาดแก่บริษัท/องค์กร ดังนั้นการวิเคราะห์พฤติกรรมผู้บริโภคในแง่มุมต่าง ๆ จะสามารถทราบถึงพฤติกรรมของผู้บริโภคและช่วยให้สามารถวิเคราะห์แนวโน้มการซื้อสินค้าของผู้บริโภค และยังสามารถมีส่วนร่วมในการดำเนินธุรกิจได้ การวิเคราะห์พฤติกรรมผู้บริโภคสามารถดำเนินการได้ในหลายแง่มุม แต่อย่างไรก็ตามการวิเคราะห์พฤติกรรมผู้บริโภคด้วยการค้นหาเซตรายการ (รูปแบบ) ที่น่าสนใจจากข้อมูลการซื้อสินค้าของลูกค้าเป็นวิธีการหนึ่งที่ได้รับความนิยมอย่างแพร่หลายในธุรกิจต่าง ๆ โดยเริ่มแรกได้มีการคิดค้นการวิเคราะห์พฤติกรรมผู้บริโภคด้วยการค้นหาเซตรายการที่ปรากฏบ่อย (Frequent Itemsets Mining, FIM) ที่สามารถบ่งบอกได้ถึงสิ่งของหรือเหตุการณ์ที่ปรากฏขึ้นพร้อมกันบ่อย ๆ ตัวอย่างเช่น ในธุรกิจห้างสรรพสินค้าหรือธุรกิจค้าปลีกจะทำการหารายการสินค้าที่ถูกซื้อร่วมกันบ่อย ๆ เพื่อช่วยในการจัดทำโปรโมชันสินค้า ช่วยในการจัดชั้นวางสินค้าให้สินค้าที่ถูกซื้อร่วมกันบ่อย ๆ ให้อยู่ในพื้นที่ใกล้ ๆ กัน เพื่ออำนวยความสะดวกให้แก่ลูกค้าและช่วยกระตุ้นการจับจ่ายใช้สอยของลูกค้า นอกจากนั้นยังช่วยในการจัดทำแค็ตตาล็อกสินค้าให้สินค้าที่ถูกซื้อร่วมกันบ่อย ๆ ได้อยู่ใกล้ ๆ กัน ที่ซึ่งการดำเนินการทั้งหมดนี้จะช่วยให้ห้างสรรพสินค้าสามารถกระตุ้นการซื้อสินค้าของลูกค้าและสามารถอำนวยความสะดวกให้กับลูกค้า อันนำมาซึ่งการรักษาฐานลูกค้าให้ยังคงซื้อสินค้ากับห้างสรรพสินค้าต่อไป

แนวความคิดเบื้องต้นของการค้นหาเซตรายการที่ปรากฏบ่อยจะประยุกต์ใช้ค่าความถี่หรือจำนวนครั้งในการปรากฏขึ้นของเซตรายการ ที่ซึ่งเป็นตัวชี้วัดความสำคัญหรือความน่าสนใจของเซตรายการ แต่อย่างไรก็ตาม การพิจารณาเพียงค่าความถี่ของการปรากฏอาจจะไม่เพียงพอต่อการวิเคราะห์ข้อมูลในหลาย ๆ แง่มุม ด้วยเหตุนี้จึงเป็นเหตุให้แนวความคิดเกี่ยวกับการค้นหาเซตรายการที่ปรากฏบ่อยถูกพัฒนาอย่างต่อเนื่องในหลาย ๆ แง่มุม อาทิเช่น การค้นหาเซตรายการที่ปรากฏบ่อยแบบเรียงลำดับ (Frequent sequential itemsets mining) การค้นหาเซตรายการที่ปรากฏบ่อย

ภายใต้ค่าน้ำหนักของแต่ละรายการ (Frequent weighted itemsets mining) การค้นหาเซตรายการภายใต้ความเปลี่ยนแปลงลักษณะของการปรากฏ (Emerging itemsets mining) การค้นหาเซตรายการที่มีค่าคุณประโยชน์สูง (High utility itemsets mining) การค้นหาเซตรายการที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ (Frequent-regular itemsets mining) และอื่น ๆ

จากงานวิจัยข้างต้นโดยส่วนใหญ่มักจะมุ่งเน้นที่การค้นหาเซตรายการภายใต้การวัดความน่าสนใจในแง่มุมต่าง ๆ โดยมีงานวิจัยหนึ่งที่น่าสนใจเกี่ยวกับความเปลี่ยนแปลงของการปรากฏขึ้น ในแง่มุมของความถี่ที่เพิ่มขึ้นเมื่อเวลาผ่านไป แต่อย่างไรก็ตามการค้นหาเซตรายการภายใต้ความเปลี่ยนแปลงลักษณะของการปรากฏในด้านของการปรากฏอย่างสม่ำเสมอก็เป็นแง่มุมที่น่าสนใจเช่นกัน ที่ซึ่งช่วยให้ทราบถึงแนวโน้มพฤติกรรมการซื้อสินค้าที่ปรากฏขึ้นอย่างสม่ำเสมอ ที่บริษัทจะได้รับจากลูกค้าเพื่อใช้ประกอบการตัดสินใจ จากการทราบข้อมูลดังกล่าวจะทำให้ผู้บริหารสามารถคิดกลยุทธ์ เพื่อสามารถกระตุ้นการซื้อสินค้าของผู้บริโภคอันนำมาซึ่งผลประกอบการของบริษัทที่เพิ่มขึ้นได้ โดยในตอนเริ่มต้นการวิเคราะห์แนวโน้มความเปลี่ยนแปลงของพฤติกรรมการซื้อสินค้าของลูกค้ามักจะใช้กับข้อมูลที่เป็นการซื้อสินค้าในห้างสรรพสินค้า ที่ซึ่งในปัจจุบันห้างสรรพสินค้าหลายแห่งได้มีการจัดเก็บข้อมูลของลูกค้าเพื่อใช้ในการวิเคราะห์พฤติกรรมการซื้อสินค้าเมื่อเวลาผ่านไป ที่ซึ่งทำให้ทราบถึงช่วงการปรากฏขึ้นของความสม่ำเสมอในการซื้อสินค้า เมื่อทราบข้อมูลดังกล่าวจึงทำให้ในห้างสรรพสินค้าสามารถทราบถึงแนวโน้มในการซื้อสินค้าของลูกค้าในแต่ละช่วงเวลา (หมายเหตุในรอบวัน สัปดาห์ เดือน ปี หรือ ช่วงเทศกาล และอื่น ๆ) ที่ซึ่งทำให้ห้างสรรพสินค้าสามารถทราบจำนวนในการสต็อกสินค้าไว้ได้ และจะช่วยลดปัญหาในการสต็อกสินค้าที่มากเกินไปได้ด้วย (Liu, Hsu, Han, & Xia, 2000) นอกจากนี้ในทางการแพทย์ ได้มีการวิเคราะห์แนวโน้มความเปลี่ยนแปลงของการเกิดโรคมะเร็งปอด โดยวิเคราะห์จากความสม่ำเสมอของพฤติกรรมผู้ป่วยที่สูบบุหรี่และไม่สูบบุหรี่ ที่ซึ่งจากการวิเคราะห์ทำให้แพทย์สามารถทราบแนวโน้มของการเกิดโรคมะเร็งปอดในแต่ละระยะของโรค เพื่อหาแนวทางในการรักษาให้กับผู้ป่วยต่อไป (Dong & Li, 2005) อีกทั้งในปัจจุบันยังพบว่าโรคหลายชนิดมีสาเหตุมาจากความผิดปกติของยีน อาทิเช่น การวินิจฉัยโรคเนื้องอกในลำไส้ใหญ่ โดยพิจารณาแนวโน้มความผิดปกติของเซลล์เนื้องอกจากการเปรียบเทียบความผิดปกติของยีนที่อยู่ภายในเซลล์ ที่ซึ่งจากการวิเคราะห์ทำให้แพทย์ทราบถึงแนวโน้มที่เซลล์เนื้องอกนั้นจะพัฒนาเป็นก้อนเนื้อมะเร็งในระยะต่าง ๆ แพทย์จึงสามารถวางแผนในการรักษาและจ่ายยาให้เหมาะสม (Li & Wong, 2002) และอื่น ๆ (Tanbeer, Hassan, Alrubaian, & Jeong, 2015)

ดังนั้นในงานวิทยานิพนธ์นี้จึงมุ่งเน้นที่การวิเคราะห์ความเปลี่ยนแปลงของพฤติกรรมผู้บริโภคภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอที่เพิ่มขึ้นเมื่อเวลาผ่านไป โดยการวิเคราะห์พฤติกรรมการบริโภคที่นำเสนอจะเหมาะกับทุกธุรกิจการค้าที่มีรายการสินค้าที่หลากหลายและมีการขายสินค้าอย่างต่อเนื่อง อาทิเช่น ธุรกิจค้าปลีก ธุรกิจค้าส่ง ธุรกิจร้านอาหาร ธุรกิจเกี่ยวกับเฟอร์นิเจอร์ ธุรกิจเกี่ยวกับอุปกรณ์ไฟฟ้า ห้างสรรพสินค้า ธุรกิจขายตรง ธุรกิจขายอะไหล่รถจักรยานยนต์หรือรถยนต์ ธุรกิจเกี่ยวกับอาหารทะเล ธุรกิจเกี่ยวกับวัสดุอุปกรณ์การเกษตร ธุรกิจเกี่ยวกับวัสดุอุปกรณ์ก่อสร้าง ธุรกิจเกี่ยวกับผลิตภัณฑ์ทารกและเด็ก ธุรกิจเกี่ยวกับอุปกรณ์เครื่องเขียนและสิ่งพิมพ์ ธุรกิจเกี่ยวกับอุปกรณ์คอมพิวเตอร์และโทรศัพท์เคลื่อนที่ ธุรกิจเกี่ยวกับดอกไม้และ

ผลิตภัณฑ์ทางการเกษตร ธุรกิจเกี่ยวกับเบเกอรี่ ธุรกิจเกี่ยวกับเวชกรรมและอุปกรณ์ทางการแพทย์ ธุรกิจเกี่ยวกับเครื่องประดับ ธุรกิจสิ่งทอและเครื่องแต่งกาย ธุรกิจเกี่ยวกับเว็บไซต์ต่าง ๆ ที่ถูกเข้าถึงอย่างต่อเนื่อง ที่ซึ่งจะทำให้เจ้าของธุรกิจทราบถึงความนิยมของเว็บไซต์ที่ทางบริษัทจัดทำขึ้น และในวงการแพทย์ที่จะช่วยให้แพทย์ทราบถึงข้อมูลการเปลี่ยนแปลง เพื่อใช้เป็นแนวทางในการหาวิธีบำบัดหรือรักษาผู้ป่วยได้ และอื่น ๆ

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อสร้างแนวคิดใหม่ในการตรวจสอบพฤติกรรมของการบริโภคที่สามารถนำไปประยุกต์ใช้ในธุรกิจต่าง ๆ ได้จริง โดยแนวคิดดังกล่าวสามารถประยุกต์ใช้ได้กับ ธุรกิจค้าปลีก ธุรกิจค้าส่ง ธุรกิจร้านอาหาร ธุรกิจเกี่ยวกับเฟอร์นิเจอร์ ธุรกิจเกี่ยวกับอุปกรณ์ไฟฟ้า ห้างสรรพสินค้า ธุรกิจขายตรง ธุรกิจขายอะไหล่รถจักรยานยนต์หรือรถยนต์ ธุรกิจเกี่ยวกับอาหารทะเล ธุรกิจเกี่ยวกับวัสดุอุปกรณ์การเกษตร ธุรกิจเกี่ยวกับวัสดุอุปกรณ์ก่อสร้าง ธุรกิจเกี่ยวกับผลิตภัณฑ์ทารกและเด็ก ธุรกิจเกี่ยวกับอุปกรณ์เครื่องเขียนและสิ่งพิมพ์ ธุรกิจเกี่ยวกับอุปกรณ์คอมพิวเตอร์และโทรศัพท์เคลื่อนที่ ธุรกิจเกี่ยวกับดอกไม้และผลิตภัณฑ์ทางการเกษตร ธุรกิจเกี่ยวกับเบเกอรี่ ธุรกิจเกี่ยวกับเวชกรรมและอุปกรณ์ทางการแพทย์ ธุรกิจเกี่ยวกับเครื่องประดับ ธุรกิจสิ่งทอและเครื่องแต่งกาย ธุรกิจเกี่ยวกับเว็บไซต์ต่าง ๆ ที่ถูกเข้าถึงอย่างต่อเนื่องที่ซึ่งจะทำให้เจ้าของธุรกิจทราบถึงความนิยมของเว็บไซต์ที่ทางบริษัทจัดทำขึ้น อีกทั้งยังสามารถนำมาประยุกต์กับทางการแพทย์เพื่อใช้เป็นแนวทางในการป้องกัน/รักษาโรคต่าง ๆ ได้อีกด้วยและอื่น ๆ

2. เพื่อศึกษาความเปลี่ยนแปลงของพฤติกรรมผู้บริโภคโดยพิจารณาจากข้อมูลการซื้อสินค้าของลูกค้า โดยจะทำการพิจารณาถึงความแตกต่างระหว่างผลกำไร/ขาดทุนระหว่างช่วงเวลาต่าง ๆ ของการซื้อสินค้า

3. เพื่อทราบถึงกลุ่มของรายการสินค้าที่มีความเปลี่ยนแปลงจากการซื้อสินค้าของผู้บริโภค ที่ซึ่งจากข้อมูลดังกล่าวจะนำไปสู่การค้นหาสาเหตุของการเกิดขึ้นของพฤติกรรม/รูปแบบการบริโภค และสามารถนำข้อมูลดังกล่าวไปประกอบการตัดสินใจเพื่อที่จะสามารถทำการพัฒนาผลิตภัณฑ์และ/หรือขั้นตอนการดำเนินธุรกิจต่อได้

4. เพื่อให้ผู้ที่สนใจสามารถนำแนวคิดที่นำเสนอ ไปศึกษาเพื่อทำการพัฒนาหรือประยุกต์ใช้ในงานวิจัย/งานวิทยานิพนธ์หรือประยุกต์ใช้ในการดำเนินธุรกิจของตนเองต่อไป

1.3 ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย

1. ได้ทราบถึงข้อมูลที่ได้จากการวิเคราะห์พฤติกรรมการบริโภคของลูกค้า ที่ซึ่งเป็นข้อมูลสำหรับสนับสนุนการตัดสินใจที่จะดำเนินการกระตุ้นพฤติกรรมการจับจ่ายใช้สอยของผู้บริโภคได้ อาทิเช่น การจัดทำโปรโมชัน การนำเสนอรายการสินค้าใหม่ ๆ และอื่น ๆ โดยการวิเคราะห์พฤติกรรมการบริโภคที่นำเสนอจะเหมาะกับทุกธุรกิจการค้าที่มีรายการสินค้าที่หลากหลายและมีการขายสินค้าต่อเนื่อง อาทิเช่น ธุรกิจค้าปลีก ธุรกิจค้าส่ง ธุรกิจร้านอาหาร ธุรกิจเกี่ยวกับเฟอร์นิเจอร์ ธุรกิจเกี่ยวกับอุปกรณ์ไฟฟ้า ห้างสรรพสินค้า ธุรกิจขายตรง ธุรกิจขายอะไหล่รถจักรยานยนต์หรือรถยนต์ ธุรกิจเกี่ยวกับอาหารทะเล ธุรกิจเกี่ยวกับวัสดุอุปกรณ์การเกษตร ธุรกิจเกี่ยวกับวัสดุอุปกรณ์ก่อสร้าง ธุรกิจเกี่ยวกับผลิตภัณฑ์ทารกและเด็ก ธุรกิจเกี่ยวกับอุปกรณ์เครื่องเขียนและสิ่งพิมพ์

ธุรกิจเกี่ยวกับอุปกรณ์คอมพิวเตอร์และโทรศัพท์เคลื่อนที่ ธุรกิจเกี่ยวกับดอกไม้และผลิตภัณฑ์ทางการเกษตร ธุรกิจเกี่ยวกับเบเกอรี่ ธุรกิจเกี่ยวกับเวชกรรมและอุปกรณ์ทางการแพทย์ ธุรกิจเกี่ยวกับเครื่องประดับ ธุรกิจสิ่งทอและเครื่องแต่งกาย ธุรกิจเกี่ยวกับเว็บไซต์ต่าง ๆ ที่ถูกเข้าถึงอย่างต่อเนื่องที่ซึ่งจะทำให้เจ้าของธุรกิจทราบถึงความนิยมของเว็บไซต์ที่ทางบริษัทจัดทำขึ้น อีกทั้งทำให้ทางการแพทย์ได้แนวทางในการป้องกัน/รักษาโรคต่าง ๆ ได้อีกด้วยและอื่น ๆ

2. ได้ขั้นตอนวิธีสำหรับการวิเคราะห์พฤติกรรมการบริโภคที่มีประสิทธิภาพในแง่มุมของเวลาที่ใช้ในการประมวลผลข้อมูล พื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูล และจำนวนผลลัพธ์เซตรายการที่ค้นพบ

3. ได้ผลงานวิจัยที่ตีพิมพ์ในงานประชุมวิชาการ ดังนี้

- 2017, 9th International Conference on knowledge and Smart Technology
- 2018, 3rd International Conference on Digital Arts, Media and Technology

4. สามารถนำแนวคิดการวิเคราะห์พฤติกรรมการบริโภคและขั้นตอนวิธีที่นำเสนอไปต่อยอดเพื่อการดำเนินการวิจัยขั้นสูงสืบไป

1.4 ขอบเขตของการวิจัย

1. ข้อมูลการซื้อสินค้าที่จะทำการพิจารณาจะต้องมีลักษณะเป็นแบบทรานแซกชันที่ประกอบไปด้วยรายการสินค้าและจำนวนชิ้นของแต่ละสินค้าที่ถูกซื้อในแต่ละทรานแซกชัน (ข้อมูลที่จะนำมาวิเคราะห์จะสามารถค้นหาได้จากธุรกิจที่มีรายการสินค้าที่หลายหลากและมีการสั่งซื้อสินค้าอย่างต่อเนื่อง)

2. การวิเคราะห์พฤติกรรมผู้บริโภคจะเป็นการวิเคราะห์ความเปลี่ยนแปลงของพฤติกรรมผู้บริโภคโดยพิจารณาจากข้อมูลการซื้อสินค้าของลูกค้า โดยผู้ที่ต้องการผลลัพธ์จะต้องทำการกำหนดค่าขีดแบ่ง (Threshold) เพื่อใช้เป็นเกณฑ์สำหรับวัดความน่าสนใจของเซตรายการที่จะทำการค้นหาจากข้อมูลที่ต้องการวิเคราะห์ โดยในงานวิจัยนี้จะใช้ค่าขีดแบ่งการเปลี่ยนแปลง (Change value threshold) และค่าขีดแบ่งความสม่ำเสมอ (Regularity threshold)

3. การวัดผลของการวิเคราะห์จะสามารถดำเนินการได้ใน 3 แง่มุมคือ เวลาที่ใช้ในการประมวลผลข้อมูล พื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูล และจำนวนผลลัพธ์เซตรายการที่ค้นพบ ตามลำดับ โดยผลลัพธ์ที่ได้จากการวิเคราะห์สามารถนำไปประกอบการตัดสินใจได้ แต่การตัดสินใจและการติดตามผลการตัดสินใจไม่สามารถดำเนินการได้

1.5 แผนการดำเนินการวิจัย

ในการดำเนินการวิจัยนี้ ดำเนินงานภายใต้ระยะเวลาที่ระบุไว้ดังตารางที่ 1.1

ตารางที่ 1.1 ระยะเวลาในการดำเนินงานวิจัย

ปี	แผนการดำเนินงาน	เดือน											
		1	2	3	4	5	6	7	8	9	10	11	12
2557	จัดเตรียมข้อมูลที่จะใช้สำหรับการวิเคราะห์ข้อมูลผู้บริโภค								←	—	—	—	→
2558	ศึกษางานวิจัยที่เกี่ยวข้องโดยมุ่งเน้นที่การวิเคราะห์การเปลี่ยนแปลงของพฤติกรรมการซื้อสินค้าและปัจจัยที่เอื้อต่อการซื้อสินค้า	←	—	—	—	—	—	—	—	—	—	—	→
2559	กำหนดขอบเขตของการวิเคราะห์พฤติกรรมผู้บริโภค และกำหนดนิยามพื้นฐานที่เกี่ยวข้องกับการวิเคราะห์พฤติกรรมผู้บริโภคด้วยการวิเคราะห์ความเปลี่ยนแปลงของพฤติกรรมการซื้อสินค้า	←	—	—	→								
2559	ออกแบบขั้นตอนการวิเคราะห์ข้อมูลผู้บริโภค พัฒนาโปรแกรม ทดสอบประสิทธิภาพของการวิเคราะห์ความเปลี่ยนแปลงของพฤติกรรมการซื้อสินค้าในแง่มุมต่าง ๆ และเขียนบทความตีพิมพ์ในงานประชุมวิชาการ					←	—	—	—	—	→		
2559	จัดทำเอกสารและสอบโครงร่างวิทยานิพนธ์										←	—	→

ตารางที่ 1.1 (ต่อ)

ปี	แผนการดำเนินงาน	เดือน											
		1	2	3	4	5	6	7	8	9	10	11	12
2560	ตรวจสอบข้อบกพร่องในการวิเคราะห์ความเปลี่ยนแปลงของพฤติกรรมการซื้อสินค้ารวมถึงแนวทางในการเพิ่มประสิทธิภาพขั้นตอนวิธีการวิเคราะห์ข้อมูลผู้บริโภค	←	—	—	→								
2560	คิดค้นวิธีการและขั้นตอนวิธีที่ขจัดข้อบกพร่อง เพื่อเพิ่มความสามารถในการวิเคราะห์ข้อมูลได้อย่างมีประสิทธิภาพมากขึ้น				←	—	—	→					
2560	พัฒนาโปรแกรม ทดสอบประสิทธิภาพของการวิเคราะห์ความเปลี่ยนแปลงของพฤติกรรมการซื้อสินค้าในแง่มุมต่าง ๆ และเขียนบทความตีพิมพ์ในงานประชุมวิชาการ								←	—	—	—	→
2561	จัดทำเอกสารฉบับสมบูรณ์และสอบวิทยานิพนธ์	←	—	—	—	—	—	→					

บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีพื้นฐาน งานวิจัยที่เกี่ยวข้อง และคุณลักษณะของฐานข้อมูลรายการที่ใช้ในการทดลอง โดยเริ่มต้นจากนิยามและทฤษฎีพื้นฐานในการค้นหาเซตรายการที่น่าสนใจภายใต้การวัดความน่าสนใจในแง่มุมต่าง ๆ ดังนี้

2.1 ทฤษฎีพื้นฐาน

2.1.1 การค้นหาเซตรายการที่ปรากฏบ่อย (Mining frequent itemsets)

การค้นหาเซตรายการที่ปรากฏบ่อยเป็นการค้นหาเซตรายการที่น่าสนใจภายใต้การพิจารณาจำนวนครั้ง/ความบ่อย/ความถี่ในการปรากฏขึ้นของเซตรายการเหล่านั้น โดยปัญหาการค้นหาเซตรายการที่ปรากฏบ่อยจะมุ่งเน้นที่การค้นหาเซตของรายการสินค้าที่ถูกซื้อร่วมกันบ่อย ๆ ที่ซึ่งจะทำให้บริษัท/ห้างร้าน/สถานประกอบการสามารถทราบถึงปริมาณการซื้อสินค้า แล้วสามารถนำข้อมูลดังกล่าวไปเป็นส่วนประกอบในการจัดทำโปรโมชัน การจัดการคลังสินค้า การทำแค็ตตาล็อกสินค้า การจัดวางชั้นสินค้า หรือในทางการแพทย์สามารถที่จะค้นหาเซตรายการของโรคที่เกิดร่วมกันบ่อย ๆ อาทิเช่น โรคอ้วน[1] เมื่อเป็นแล้วจะเป็นสาเหตุให้เกิดโรคอื่น ๆ ตามมา โรคเบาหวานก็เป็นโรคหนึ่งที่ส่งผลมาจากผู้ที่มีน้ำหนักตัวเกินหรือมีดัชนีมวลกายมากกว่า 35 กก./ตร.ม. จึงทำให้ผู้ที่เป็นโรคอ้วนมีโอกาสเกิดโรคเบาหวานมากกว่าคนทั่วไปถึง 20 เท่า ด้วยเหตุนี้จึงทำให้แพทย์สามารถหาวิธีการรักษาหรือจ่ายยาให้เหมาะสมกับผู้ป่วยได้ โดยปัญหาการค้นหาเซตรายการที่ปรากฏบ่อย (Agrawal, Imielinski, & Swami, 1993) สามารถนิยามได้ ดังนี้

นิยามที่ 2.1 *เซต* $I = \{ i_1, i_2, \dots , i_m \}$ *เป็นเซตของรายการ (Items) ที่อาจหมายถึงสิ่งของหรือเหตุการณ์ที่ต้องการหาความสัมพันธ์*

นิยามที่ 2.2 *เซต* $X = \{ i_p, i_{p+1}, \dots , i_q \} \subseteq I$ *เรียกว่า เซตรายการ (Set of items, an itemset หรือ a pattern) ที่ประกอบด้วยหลายรายการ*

นิยามที่ 2.3 $TDB = \{ t_1, t_2, \dots , t_n \}$ *คือ ฐานข้อมูลรายการหรือฐานข้อมูลแบบทรานแซกชัน (Transactional database) ที่ซึ่งแต่ละทรานแซกชัน* $t_j \in TDB$ *ประกอบด้วย 1) หมายเลขกำกับทรานแซกชัน (Unique transaction identifier, tid) tid=j และ 2) เซตของรายการ* $Y \leq I$ *ที่ถูกบรรจุอยู่ในทรานแซกชันนั้น ๆ (แสดงดังภาพที่ 2.1)*

[1] https://m.bangkokpattayahospital.com/th

A transactional database

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

tid	Set of items
11	b, e
12	b, e
13	b, e
14	b, e
15	b, e
16	d, e
17	b, e
18	d
19	d, e, f
20	d, f

tid	Set of items
21	a, b, c, d
22	a, b, c, d
23	a, b, d, e
24	a, b, d
25	a, b, e
26	d, e
27	d, e
28	d
29	d, e, f
30	d, f

tid	Set of items
31	d, f
32	d, f
33	d
34	d, f
35	d
36	b, e
37	d
38	d
39	d, e, f
40	d, f

ภาพที่ 2.1 ตัวอย่างฐานข้อมูลรายการที่ประกอบไปด้วยหมายเลขทรานแซกชัน (tid) และเซตของรายการที่ปรากฏในทรานแซกชัน (Set of items)

ถ้าเซตรายการ $X \subseteq Y$ สามารถสรุปได้ว่าเซตรายการ X ปรากฏขึ้นในทรานแซกชัน t_j หรือทรานแซกชัน t_j มี X บรรจุอยู่ สามารถแสดงในรูปแบบของสัญลักษณ์ได้เป็น j^X ดังนั้นเมื่อทำการตรวจสอบเซตรายการ X ว่าปรากฏขึ้นในทรานแซกชันใดบ้างในฐานข้อมูลรายการ TDB จะทำให้ทราบถึง เซตของหมายเลขทรานแซกชันที่มีเซตรายการ X ปรากฏขึ้น สามารถนิยามได้ดังนี้

นิยามที่ 2.4 *$T^X = \{ j^X, (j+1)^X, \dots, k^X \}$ เมื่อ $1 \le j < k \le |TDB|$ คือ เซตของหมายเลขทรานแซกชัน (tid) ที่ถูกเรียงลำดับจากน้อยไปมาก เพื่อประสิทธิภาพในการประมวลผล (สามารถเรียกโดยย่อได้เป็น tidset)*

นิยามที่ 2.5 *s^X คือ อัตรา (ร้อยละ) ค่าสนับสนุนของเซตรายการ X ที่ปรากฏขึ้นในฐานข้อมูลรายการ (หมายเหตุ ที่ซึ่งบ่งบอกถึงจำนวนครั้ง/ความบ่อย/ความถี่ในการปรากฏขึ้นของเซตรายการ X ฐานข้อมูลรายการ) สามารถคำนวณได้เป็น*

$$s^X = \frac{|T^X|}{|TDB|} \times 100\% \qquad (2.1)$$

นิยามที่ 2.6 *เซตรายการ X จะเป็นเซตรายการที่ปรากฏบ่อยก็ต่อเมื่อ s^X มีค่ามากกว่าหรือเท่ากับค่าขีดแบ่งสนับสนุน (Support threshold, σ_s) ที่ผู้ใช้กำหนด*

ปัญหาการค้นหาเซตรายการที่ปรากฏบ่อยจะเป็นการค้นหาเซตรายการที่มีอัตรา (ร้อยละ) ค่าสนับสนุนของเซตรายการที่ปรากฏขึ้นมากกว่าหรือเท่ากับค่าขีดแบ่งสนับสนุนที่ผู้ใช้กำหนด

ตัวอย่างที่ 2.1 *กำหนดให้ ฐานข้อมูลรายการประกอบด้วย 40 ทรานแซกชัน โดยมีรายการทั้งหมด 6 รายการ ได้แก่ รายการ ‘a’, ‘b’, ‘c’, ‘d’, ‘e’ และ ‘f’ แสดงดังภาพที่ 2.1 เมื่อทำการพิจารณารายการ ‘a’ สามารถระบุได้ถึงเซตของหมายเลขทรานแซกชันที่มีรายการ ‘a’ ปรากฏอยู่ ดังนี้ $T^a = \{ 1^a, 2^a, 3^a, 4^a, 5^a, 6^a, 21^a, 22^a, 23^a, 24^a, 25^a \}$ นอกจากนั้นยังสามารถคำนวณอัตรา*

(ร้อยละ) ค่าสนับสนุนของรายการ 'α' ได้เป็น $s^{\alpha} = \frac{|T^{\alpha}|}{|TDB|} \times 100\% = \frac{11}{40} \times 100\% = 27.5\%$ *(หมายเหตุ ค่าสนับสนุนของการปรากฏขึ้นของรายการ 'α' ปรากฏทั้งสิ้น 11 ทรานแซกชันในฐานข้อมูลรายการ) ถ้ากำหนดให้ค่าขีดแบ่งสนับสนุนมีค่าเท่ากับ 15% สามารถสรุปได้ว่ารายการ 'α' เป็นรายการที่ปรากฏบ่อย เนื่องจากมีอัตรา (ร้อยละ) ค่าสนับสนุนของรายการ 'α' ที่ซึ่งมากกว่าค่าขีดแบ่งสนับสนุน*

2.1.2 การค้นหาเซตรายการที่ปรากฏบ่อยและปรากฏสม่ำเสมอ (Mining frequent-regular itemsets)

การค้นหาเซตรายการที่ปรากฏบ่อยและปรากฏสม่ำเสมอจะเป็นการค้นหาเซตรายการที่น่าสนใจภายใต้การพิจารณาจำนวนครั้ง/ความบ่อย/ความถี่ร่วมกับความสม่ำเสมอในการปรากฏขึ้นของเซตรายการนั้น ๆ โดยในแง่มุมของความสม่ำเสมอในการปรากฏจะพิจารณาจากช่วงหรือระยะห่างที่มากที่สุดที่มีเซตรายการปรากฏขึ้นอย่างน้อยหนึ่งครั้งในฐานข้อมูลรายการ โดยปัญหาการค้นหาเซตรายการที่ปรากฏบ่อยและปรากฏสม่ำเสมอจะมุ่งเน้นที่การค้นหาเซตของรายการสินค้าที่ถูกซื้อร่วมกันบ่อย ๆ และมีการซื้ออย่างสม่ำเสมอร่วมด้วย ที่ซึ่งจะทำให้บริษัท/ห้างร้าน/สถานประกอบการสามารถทราบถึงปริมาณการซื้อสินค้าและช่วงของการซื้อสินค้านั้น แล้วจึงสามารถนำข้อมูลดังกล่าวไปเป็นส่วนประกอบในการการสต็อกสินค้า เพื่อให้สินค้าเพียงพอสำหรับลูกค้าหรือลดปัญหาการสต๊อกสินค้าที่มากเกินไป อีกทั้งยังสามารถประยุกต์ใช้ในทางการแพทย์ อาทิเช่น การตรวจสอบการเต้นของหัวใจเพื่อใช้ในการป้องกันและรักษาภาวะหัวใจเต้นผิดจังหวะ[2] โดยการวัดอัตราความถี่และความสม่ำเสมอของการเต้นของหัวใจ จะช่วยให้แพทย์สามารถนำไปวินิจฉัยโรค ตรวจหาความรุนแรง ติดตามผลการรักษาต่อไปได้ ที่ซึ่งจะทำให้สามารถนิยามได้ดังนี้

นิยามที่ 2.7 *ค่าความสม่ำเสมอของเซตรายการ X ภายใต้การปรากฏครั้งหนึ่ง ๆ ของเซตรายการ X ในทรานแซกชัน* $t_k \in T^X$ *กล่าวคือ* r_k^X *ที่ซึ่งสามารถคำนวณได้จาก 3 กรณี ดังนี้*

1. ถ้าทรานแซกชัน t_k *เป็นทรานแซกชันที่มีเซตรายการ X ปรากฏขึ้นครั้งแรก ดังนั้นค่าความสม่ำเสมอที่สืบเนื่องจากการปรากฏขึ้นของเซตรายการ X ในทรานแซกชัน* t_k *สามารถคำนวณและแทนสัญลักษณ์ได้เป็น* $fr_k^X = k$ *(หมายเหตุ* fr_k^X *จะบ่งบอกถึงช่วงในการปรากฏขึ้นของเซตรายการ X ที่ปรากฏขึ้นครั้งแรก)*

2. ถ้าทรานแซกชัน t_k *เป็นทรานแซกชันที่มีเซตรายการ X ปรากฏขึ้น และทรานแซกชัน* t_j *เป็นทรานแซกชันที่ปรากฏขึ้นก่อนหน้าทรานแซกชัน* t_k *ที่มีเซตรายการ X ปรากฏขึ้นเช่นกัน (หมายเหตุ* $T^X = \{\dots, t_j, t_k, \dots\}$*) ดังนั้นค่าความสม่ำเสมอที่สืบเนื่องจากการปรากฏขึ้นของเซตรายการ X ในทรานแซกชัน* t_k *สามารถคำนวณได้จาก* $r_k^X = k - j$ *(หมายเหตุ* r_k^X *จะบ่งบอกถึงช่วงของการปรากฏขึ้นของเซตรายการ X ระหว่างทรานแซกชัน* t_k *และทรานแซกชัน* t_j*)*

[2] http://www.piyavate.com/article/frontend/article_detail/id/419

3. ถ้าทรานแซกชัน t_k เป็นทรานแซกชันของเซตรายการ X ที่ปรากฏขึ้นครั้งสุดท้ายในฐานข้อมูลรายการ ดังนั้นค่าความสม่ำเสมอที่สืบเนื่องจากการปรากฏขึ้นของเซตรายการ X ในทรานแซกชัน t_k สามารถคำนวณและแทนสัญลักษณ์ได้เป็น $lr_k^X = |TDB| - k$ (หมายเหตุ lr_k^X จะบ่งบอกถึงช่วงของทรานแซกชัน t_k ที่มีเซตรายการ X ปรากฏขึ้นครั้งสุดท้ายกับ $|TDB|$)

จากนิยามที่ 2.7 การปรากฏขึ้นครั้งหนึ่ง ๆ จะสามารถคำนวณค่าความสม่ำเสมอของการปรากฏขึ้นครั้งนั้น ๆ ได้ แต่อย่างไรก็ตาม ในการที่จะทราบถึงพฤติกรรมการปรากฏขึ้นของเซตรายการ X ว่ามีการปรากฏขึ้นอย่างสม่ำเสมอหรือไม่ สามารถพิจารณาได้จากช่วงหรือระยะห่างที่มากที่สุดของทรานแซกชันที่มีเซตรายการ X ปรากฏขึ้นอย่างน้อยหนึ่งทรานแซกชันในฐานข้อมูลรายการ สามารถนิยามได้ดังนี้

นิยามที่ 2.8 *r^X คือ อัตรา (ร้อยละ) ค่าความสม่ำเสมอของเซตรายการ X (หมายเหตุ ที่ซึ่งบ่งบอกถึงช่วงหรือระยะห่างที่มากที่สุดของเซตรายการ X ที่ปรากฏขึ้นอย่างน้อยหนึ่งทรานแซกชันในฐานข้อมูลรายการ) สามารถคำนวณได้เป็น*

$$r^X = \frac{max\,(fr_j^X, r_{j+1}^X, \ldots, r_k^X, lr_k^X)}{|TDB|} \times 100\% \qquad (2.2)$$

นิยามที่ 2.9 *เซตรายการ X จะเป็นเซตรายการที่ปรากฏสม่ำเสมอก็ต่อเมื่อ r^X มีค่าน้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอ (Regularity threshold, σ_r) ที่ผู้ใช้กำหนด*

ปัญหาการค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอ เป็นการค้นหาเซตรายการที่มีอัตรา (ร้อยละ) ค่าสนับสนุนมากกว่าหรือเท่ากับค่าขีดแบ่งสนับสนุนที่ผู้ใช้กำหนด และมีอัตรา (ร้อยละ) ค่าความสม่ำเสมอน้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด

ตัวอย่างที่ 2.2 *จากฐานข้อมูลรายการ แสดงดังภาพที่ 2.1 สามารถระบุได้ถึงเซตของหมายเลขทรานแซกชันที่มีเซตรายการ 'ab' ปรากฏอยู่ ได้ดังนี้ $T^{ab} = \{ 1^{ab}, 2^{ab}, 3^{ab}, 4^{ab}, 5^{ab}, 21^{ab}, 22^{ab}, 23^{ab}, 24^{ab}, 25^{ab} \}$ โดยอัตรา (ร้อยละ) ค่าสนับสนุนของเซตรายการ 'ab' จะมีค่าเท่ากับ 25 % ของทรานแซกชันทั้งหมด (หมายเหตุ ค่าสนับสนุนของการปรากฏขึ้นของเซตรายการ 'ab' ปรากฏทั้งสิ้น 10 ทรานแซกชันในฐานข้อมูลรายการ) และเมื่อคำนวณอัตรา (ร้อยละ) ค่าความสม่ำเสมอของเซตรายการ 'ab' จะได้ $r^{ab} = max\,(fr_1^{ab}, r_2^{ab}, r_3^{ab}, r_4^{ab}, r_5^{ab}, r_{21}^{ab}, r_{22}^{ab}, r_{23}^{ab}, r_{24}^{ab}, r_{25}^{ab}, lr_{25}^{ab}) = max\,(1, 2\text{-}1, 3\text{-}2, 4\text{-}3, 5\text{-}4, 21\text{-}5, 22\text{-}21, 23\text{-}22, 24\text{-}23, 25\text{-}24, 40\text{-}25) = max\,(1, 1, 1, 1, 1, 16, 1, 1, 1, 1, 15) = \frac{16}{40} \times 100\% = 40\%$ (หมายเหตุ ค่าความสม่ำเสมอของการปรากฏขึ้นของเซตรายการ 'ab' จะปรากฏขึ้นอย่างน้อยหนึ่งครั้งในทุก ๆ 16 ทรานแซกชัน) ถ้ากำหนดให้ค่าขีดแบ่งสนับสนุนมีค่าเท่ากับ 15% และค่าขีดแบ่งความสม่ำเสมอมีค่าเท่ากับ 50% ดังนั้น สามารถสรุปได้ว่าเซตรายการ 'ab' เป็นเซตรายการที่ปรากฏบ่อยและปรากฏสม่ำเสมอ เนื่องจากมีอัตรา (ร้อยละ) ค่าสนับสนุนของเซตรายการ 'ab' ที่ซึ่งมากกว่าค่าขีดแบ่งสนับสนุน และมีอัตรา (ร้อยละ) ค่าความสม่ำเสมอ ที่ซึ่งน้อยกว่าค่าขีดแบ่งความสม่ำเสมอ*

2.1.3 การค้นหาเซตรายการภายใต้ความเปลี่ยนแปลงลักษณะของการปรากฏ (Mining emerging itemsets)

การค้นหาเซตรายการภายใต้ความเปลี่ยนแปลงลักษณะของการปรากฏขึ้นของเซตรายการในเชิงความถี่ เริ่มต้นจากการกำหนดฐานข้อมูลรายการใน 2 ช่วงเวลา (TDB_1 และ TDB_2) และการค้นหาเซตรายการภายใต้ความเปลี่ยนแปลงลักษณะของการปรากฏ สามารถนิยามได้ ดังนี้

A transactional database TDB_1

tid	Set of items	tid	Set of items	tid	Set of items	tid	Set of items
1	a, b, c, d	11	b, e	21	a, b, c, d	31	d, f
2	a, b, c, d	12	b, e	22	a, b, c, d	32	d, f
3	a, b, d, e	13	b, e	23	a, b, d, e	33	d
4	a, b, d	14	b, e	24	a, b, d	34	d, f
5	a, b, e	15	b, e	25	a, b, e	35	d
6	a, e	16	d, e	26	d, e	36	b, e
7	b, e	17	b, e	27	d, e	37	d
8	d	18	d	28	d	38	d
9	d, e, f	19	d, e, f	29	d, e, f	39	d, e, f
10	d, f	20	d, f	30	d, f	40	d, f

A transactional database TDB_2

tid	Set of items	tid	Set of items	tid	Set of items	tid	Set of items
1	a, b, c	15	a, b	29	b, e, h	43	b, e, h
2	a, b, c	16	a, b, e	30	b, e, h	44	a, e
3	a, b	17	b, e	31	b, e, h	45	a, e
4	a, b, e	18	b, e	32	b, e	46	a, b
5	a, b, e, g	19	b, e, h	33	b, e	47	a, e, g
6	a, b, e, g	20	b, e	34	b, e	48	a, b, e, g, h
7	a, b, e, g	21	b, e, h	35	b, e, h	49	a, b, c
8	a, e	22	b, e	36	b, e, h	50	a, b, c
9	a, e	23	b, e, h	37	b	51	a, b
10	b, e	24	b, e, h	38	b	52	a, b, e
11	a, e, g	25	b, e	39	b	53	a, b, e, g
12	a, b, e, g, h	26	b, e	40	b	54	a, b, e, g
13	a, b, c	27	b, e, h	41	b, e, h	55	a, b, e, g
14	a, b, c	28	b, e	42	b, e, h	56	a, e

ภาพที่ 2.2 ตัวอย่างฐานข้อมูลรายการใน 2 ช่วงเวลา (TDB_1 และ TDB_2) ที่ประกอบไปด้วยหมายเลขทรานแซกชัน (tid) และเซตรายการที่ปรากฏในทรานแซกชัน (Set of items)

กำหนดให้ฐานข้อมูลรายการ TDB_1 เป็นฐานข้อมูลรายการของการซื้อสินค้าจากลูกค้าในช่วงเวลาที่ 1 ประกอบด้วย 40 ทรานแซกชัน โดยมีรายการทั้งหมด 6 รายการ ได้แก่ รายการ *'a', 'b', 'c', 'd', 'e'* และ *'f'* ส่วนฐานข้อมูลรายการ TDB_2 เป็นฐานข้อมูลรายการของการซื้อสินค้าจากลูกค้าในช่วงเวลาที่ 2 ประกอบด้วย 56 ทรานแซกชัน โดยมีรายการทั้งหมด 6 รายการ ได้แก่ รายการ *'a', 'b', 'c', 'e', 'g'* และ *'h'* แสดงดังภาพที่ 2.2 โดยจะพิจารณาการปรากฏขึ้นของรายการต่าง ๆ ณ ฐานข้อมูลรายการ TDB_1 และ TDB_2

นิยามที่ 2.10 *GR^X คือ อัตราการเติบโตหรือการเพิ่มขึ้นของค่าสนับสนุนของเซตรายการ X จากฐานข้อมูลรายการ TDB_1 ไปยัง TDB_2 (หมายเหตุ บ่งบอกถึง ค่าสนับสนุนเพิ่มขึ้นจากฐานข้อมูลรายการหนึ่งเทียบกับอีกฐานข้อมูลรายการหนึ่ง) สามารถคำนวณได้ดังนี้*

$$GR^X = \begin{cases} \text{ไม่สามารถระบุได้} & \text{, ถ้ารายการ X ไม่ปรากฏขึ้นในฐานข้อมูลรายการ } TDB_1 \text{ และ } TDB_2 \\ \frac{s^X_{TDB_2}}{s^X_{TDB_1}} & \text{, ถ้ารายการ X ปรากฏขึ้นทั้งสองฐานข้อมูลรายการ } TDB_1 \text{ และ } TDB_2 \\ \infty & \text{, ถ้ารายการ X ไม่ปรากฏขึ้นในฐานข้อมูลรายการ } TDB_1 \\ 0 & \text{, ถ้ารายการ X ไม่ปรากฏขึ้นในฐานข้อมูลรายการ } TDB_2 \end{cases} \quad (2.3)$$

สำหรับการค้นหาเซตรายการภายใต้ความเปลี่ยนแปลงลักษณะการปรากฏขึ้นของเซตรายการในเชิงความถี่สามารถทำการพิจารณาเซตรายการในลักษณะต่าง ๆ ดังต่อไปนี้

1. เซตรายการภายใต้ความเปลี่ยนแปลงลักษณะของการปรากฏที่เกิดขึ้นใหม่ (Emerging itemsets) จะเป็นเซตรายการที่มีค่าสนับสนุนเพิ่มขึ้นจากฐานข้อมูลรายการหนึ่งเทียบกับอีกฐานข้อมูลรายการหนึ่ง อาทิเช่น ในการซื้อสินค้าในช่วงเทศกาลคริสมาสต์ และปีใหม่[3] ผู้ประกอบการธุรกิจ/ร้านค้า/ห้างสรรพสินค้าจะทราบถึงแนวโน้มพฤติกรรมการซื้อสินค้าของลูกค้าจากการจัดเก็บข้อมูลการซื้อสินค้าของแต่ละเทศกาลในแต่ละปีที่ผ่านมา ที่ซึ่งจะทำให้ผู้ประกอบการธุรกิจ/ร้านค้า/ห้างสรรพสินค้ามีการวางกลยุทธ์และปรับรูปแบบสินค้าให้เข้ากับเทศกาล อาทิเช่น ในห้างสรรพสินค้าจะมีการดัดแปลงสินค้ามาเป็นกระเช้าเครื่องดื่ม กระเช้าอาหารเพื่อสุขภาพ หรือธุรกิจที่ขายอุปกรณ์ IT ก็สามารถนำสินค้ามาจัดเป็นชุดของขวัญของจับฉลากได้ และอื่น ๆ ทั้งนี้เพื่อตอบโจทย์ความต้องการของลูกค้า และมีผลทำให้ผู้ประกอบการธุรกิจ/ร้านค้า/ห้างสรรพสินค้ามียอดขายสินค้าที่เพิ่มขึ้นจากช่วงวันธรรมดาที่ไม่ใช่เทศกาล

นิยามที่ 2.11 *เซตรายการ X จะเป็นเซตรายการที่น่าสนใจภายใต้ความเปลี่ยนแปลงลักษณะของการปรากฏที่เกิดขึ้นใหม่ ก็ต่อเมื่อเซตรายการปรากฏขึ้นทั้งสองฐานข้อมูลรายการและ GR^X มีค่ามากกว่าหรือเท่ากับค่าขีดแบ่งอัตราการเติบโต (Growth-rate threshold, σ_{GR}) ที่ผู้ใช้กำหนด*

ปัญหาการค้นหาเซตรายการภายใต้ความเปลี่ยนแปลงลักษณะของการปรากฏที่เกิดขึ้นใหม่เป็นการค้นหาเซตรายการที่มีเซตรายการปรากฏขึ้นทั้งสองฐานข้อมูลรายการและมีอัตราการเติบโตมากกว่าหรือเท่ากับค่าขีดแบ่งอัตราการเติบโต

ตัวอย่างที่ 2.3 *จากฐานข้อมูลรายการ แสดงดังภาพที่ 2.2 สามารถระบุได้ถึงเซตของหมายเลขทรานแซกชันที่มีรายการ ‘a’ ปรากฏอยู่ในฐานข้อมูลรายการ TDB_1 ได้ดังนี้ $T^a_{TDB_1}$ = { 1^a, 2^a, 3^a, 4^a, 5^a, 6^a, 21^a, 22^a, 23^a, 24^a, 25^a} โดยอัตรา (ร้อยละ) ค่าสนับสนุนของรายการ ‘a’ มีค่าเท่ากับ 27.5% ของทรานแซกชันทั้งหมด และสามารถระบุได้ถึงเซตของหมายเลขทรานแซกชันที่มี*

[3] http://www.thaismeresearch.com/buying-gift-in-holiday-insight-survey/

รายการ 'a' ปรากฏอยู่ในฐานข้อมูลรายการ TDB_2 ได้ดังนี้ $T^{a}_{TDB_2}$ = { 1^a, 2^a, 3^a, 4^a, 5^a, 6^a, 7^a, 8^a, 9^a, 11^a, 12^a, 13^a, 14^a, 15^a, 16^a, 44^a, 45^a, 46^a, 47^a, 48^a, 49^a, 50^a, 51^a, 52^a, 53^a, 54^a, 55^a, 56^a } โดยอัตรา (ร้อยละ) ค่าสนับสนุนของรายการ 'a' มีค่าเท่ากับ 50% ของทรานแซกชันทั้งหมด ดังนั้นเมื่อทำการคำนวณถึงอัตราการเติบโต จะได้ $GR^a = \frac{s^a_{TDB_2}}{s^a_{TDB_1}} = \frac{50\%}{27.5\%} = 1.82$ จะเห็นว่าในฐานข้อมูลรายการ TDB_2 รายการ 'a' จะปรากฏเพิ่มขึ้นเป็น 1.82 เท่าจากที่ปรากฏในฐานข้อมูลรายการ TDB_1 ดังนั้นถ้าผู้ใช้กำหนดค่าขีดแบ่งอัตราการเติบโตไว้ที่ 1.5 เท่า สามารถบอกได้ว่ารายการ 'a' เป็นรายการที่น่าสนใจภายใต้ความเปลี่ยนแปลงลักษณะของการปรากฏที่เกิดขึ้นใหม่ เนื่องจากรายการ 'a' ปรากฏขึ้นทั้งสองฐานข้อมูลรายการและมีอัตราการเติบโตที่ซึ่งมากกว่าค่าขีดแบ่งอัตราการเติบโต

2. เซตรายการภายใต้ความเปลี่ยนแปลงลักษณะของการปรากฏที่ไม่คาดหวัง (Unexpected changes) จะเป็นเซตรายการภายใต้ความเปลี่ยนแปลงในส่วนของรายการต่าง ๆ ในเซตรายการ อาทิเช่น ในการรักษาโรคไซนัสอักเสบเฉียบพลัน[4] (หมายเหตุ พิจารณาถึงแนวโน้มการจ่ายยาให้กับผู้ป่วย) แพทย์จะให้คำแนะนำ โดยให้รับประทานน้ำที่เพียงพอ พยายามอย่าให้จมูกแห้ง และจะให้ยาลดน้ำมูกร่วมกับยาแก้อักเสบมารับประทานเป็นเวลา 10 วัน ถ้ายังไม่ได้ผลก็จะให้รับประทานยาลดน้ำมูกร่วมกับยาแก้อักเสบตัวนี้ต่ออีก 14 วัน จากนั้นทำการพบแพทย์อีกครั้ง ถ้าอาการยังไม่ดีขึ้นก็จะให้รับประทานยาลดน้ำมูกร่วมกับยาแก้อักเสบตัวใหม่ ที่ตัวยามีการควบคุมเชื้อโรคที่กว้างขึ้น

นิยามที่ 2.12 *เซตรายการ X จะเป็นเซตรายการภายใต้ความเปลี่ยนแปลงลักษณะการปรากฏที่ไม่คาดหวัง ก็ต่อเมื่อ $s^X_{TDB_2}$ มีค่ามากกว่า $s^Y_{TDB_1}$*

ปัญหาการค้นหาเซตรายการที่น่าสนใจภายใต้ความเปลี่ยนแปลงลักษณะการปรากฏที่ไม่คาดหวังเป็นการค้นหาเซตรายการที่มีอัตรา (ร้อยละ) ค่าสนับสนุนของเซตรายการ X ในฐานข้อมูลรายการ TDB_2 มากกว่าอัตรา (ร้อยละ) ค่าสนับสนุนของเซตรายการ Y ในฐานข้อมูลรายการ TDB_1

ตัวอย่างที่ 2.4 *จากฐานข้อมูลรายการ แสดงดังภาพที่ 2.2 สามารถระบุได้ถึงเซตของหมายเลขทรานแซกชันที่มีเซตรายการ 'ad' ปรากฏอยู่ในฐานข้อมูลรายการ TDB_1 ได้ดังนี้ $T^{ad}_{TDB_1}$ = { 1^{ad}, 2^{ad}, 3^{ad}, 4^{ad}, 21^{ad}, 22^{ad}, 23^{ad}, 24^{ad} } โดยอัตรา (ร้อยละ) ของการปรากฏของเซตรายการ 'ad' เท่ากับ 20% ของทรานแซกชันทั้งหมด แต่ในฐานข้อมูลรายการ TDB_2 สามารถระบุได้ถึงเซตของหมายเลขทรานแซกชันที่มีเซตรายการ 'ab' ปรากฏอยู่ ได้ดังนี้ $T^{ab}_{TDB_2}$ = { 1^{ab}, 2^{ab}, 3^{ab}, 4^{ab}, 5^{ab}, 6^{ab}, 7^{ab}, 12^{ab},13^{ab}, 14^{ab}, 15^{ab}, 16^{ab}, 46^{ab}, 48^{ab}, 49^{ab}, 50^{ab}, 51^{ab}, 52^{ab}, 53^{ab}, 54^{ab}, 55^{ab} } โดยอัตรา (ร้อยละ) ค่าสนับสนุนของเซตรายการ 'ab' เท่ากับ 37.5% ของทรานแซกชันทั้งหมด ดังนั้น เมื่อพิจารณาที่เซตรายการ 'ad' จะมีความเปลี่ยนแปลงเกิดขึ้นที่รายการในเซตรายการ โดยจะเปลี่ยนจากรายการ 'd' ที่ปรากฏขึ้นร่วมกับรายการ 'a' ไปเป็นรายการ 'b' ที่*

[4] https://www.doctor.or.th/article/detail/1753

ปรากฏขึ้นร่วมกับรายการ 'a' แทน ด้วยเหตุนี้ จึงสามารถสรุปได้ว่าเซตรายการ 'ab' เป็นเซตรายการภายใต้ความเปลี่ยนแปลงลักษณะการปรากฏที่ไม่คาดหวัง เนื่องจากอัตรา (ร้อยละ) ค่าสนับสนุนของเซตรายการ 'ab' ในฐานข้อมูลรายการ TDB_2 มีค่ามากกว่า อัตรา (ร้อยละ) ค่าสนับสนุนของเซตรายการ 'ad' ในฐานข้อมูลรายการ TDB_1

3. เซตรายการภายใต้ความเปลี่ยนแปลงลักษณะการปรากฏที่ปรากฏขึ้นในฐานข้อมูลรายการหนึ่งแต่ไม่ปรากฏขึ้นในอีกฐานข้อมูลรายการหนึ่ง (Added/Perished itemsets) จะเป็นเซตรายการที่ปรากฏบ่อย ๆ ในฐานข้อมูลรายการหนึ่ง ๆ แต่ไม่ปรากฏในอีกฐานข้อมูลรายการหนึ่ง ๆ

- เซตรายการเพิ่มเติม (Added itemsets) เป็นเซตรายการที่มีการปรากฏบ่อยในฐานข้อมูลรายการ TDB_2 แต่ไม่ปรากฏในฐานข้อมูลรายการ TDB_1 อาทิเช่น ในทางการแพทย์ได้มีการตรวจพบโรคที่ชื่อว่า โรคติดต่ออุบัติใหม่ (Emerging infectious disease)[5] อาทิเช่น โรคติดเชื้อไวรัสซิกกา โรคซาร์ส โรคไข้หวัดนกสายพันธุ์ H5N1 โรคมือ เท้า ปาก และอื่น ๆ โดยจะเกิดขึ้นใหม่ในทุก ๆ ปี และมีแนวโน้มที่จะพบมากขึ้นเรื่อย ๆ (หมายเหตุ จากเดิมที่ไม่มีโรคดังกล่าวปรากฏขึ้นเลย) เนื่องจากสภาพภูมิอากาศโลกที่เปลี่ยนแปลงไป การเดินทางติดต่อระหว่างผู้คนในโลกอย่างไร้พรมแดน หรือโรคติดต่อที่มาจากสัตว์มาสู่คน และอื่น ๆ จึงมีผลทำให้เกิดการเจริญเติบโตของเชื้อโรค และแพร่ลุกลามติดต่อกันได้อย่างรวดเร็ว ด้วยเหตุนี้จึงทำให้แพทย์สามารถเตรียมความพร้อมที่จะรับมือ หาวิธีการป้องกัน และรักษา สำหรับการเกิดโรคติดต่ออุบัติใหม่ได้

นิยามที่ 2.13 *เซตรายการ X จะเป็นเซตรายการเพิ่มเติมก็ต่อเมื่อ GR^X มีค่าเท่ากับ ∞*

ปัญหาการค้นหาเซตรายการเพิ่มเติม เป็นการค้นหาเซตรายการที่มีอัตราการเติบโตเท่ากับอินฟินิตี้ (Infinity)

ตัวอย่างที่ 2.5 *จากฐานข้อมูลรายการ แสดงดังภาพที่ 2.2 สามารถระบุได้ถึงเซตของหมายเลขทรานแซกชันที่มีเซตรายการ 'aeg' ปรากฏอยู่ในฐานข้อมูลรายการ TDB_2 ได้ดังนี้ $T_{TDB_2}^{aeg} = \{ 5^{aeg}, 6^{aeg}, 7^{aeg}, 11^{aeg}, 12^{aeg}, 47^{aeg}, 48^{aeg}, 53^{aeg}, 54^{aeg}, 55^{aeg} \}$ โดยอัตรา (ร้อยละ) ค่าสนับสนุนของเซตรายการ 'aeg' เท่ากับ 17.86% ของทรานแซกชันทั้งหมด แต่เซตรายการ 'aeg' ไม่ปรากฏในฐานข้อมูลรายการ TDB_1 จึงมีค่าอัตรา (ร้อยละ) ค่าสนับสนุนของเซตรายการ 'aeg' เท่ากับ 0 ดังนั้นเมื่อทำการคำนวณถึงอัตราการเติบโต จะได้ $GR^{aeg} = \frac{s_{TDB_2}^{aeg}}{s_{TDB_1}^{aeg}} = \frac{17.86\%}{0} = \infty$ ดังนั้น เซตรายการ 'aeg' เป็นเซตรายการเพิ่มเติม เนื่องจากมีอัตราการเติบโตเท่ากับอินฟินิตี้*

- เซตรายการที่ขาดหายไป (Perished itemsets) เป็นเซตรายการที่มีการปรากฏบ่อยในฐานข้อมูล TDB_1 แต่ไม่ปรากฏในฐานข้อมูลรายการ TDB_2 อาทิเช่น เทศกาลลอยกระทง (หมายเหตุ จะพิจารณาถึงเอกลักษณ์ในงานเทศกาลลอยกระทงของแต่ละจังหวัด) โดยเทศกาลลอยกระทงของ

[5] http://beid.ddc.moph.go.th/beid_2014/th/diseases

จังหวัดเชียงใหม่ จะมีประเพณียี่เป็ง[6] ร่วมด้วย ที่ซึ่งจัดขึ้นทุก ๆ ปี โดยจะปล่อยโคมลอยขึ้นเต็มท้องฟ้า และสามารถชมได้เฉพาะในจังหวัดเชียงใหม่เท่านั้น

นิยามที่ 2.14 *เซตรายการ X จะเป็นเซตรายการที่ขาดหายไปก็ต่อเมื่อ* GR^X *มีค่าเท่ากับ 0*

ปัญหาการค้นหาเซตรายการที่ขาดหายไป เป็นการค้นหาเซตรายการที่มีค่าอัตราการเติบโตเท่ากับศูนย์

ตัวอย่างที่ 2.6 *ฐานข้อมูลรายการ แสดงดังภาพที่ 2.2 สามารถระบุได้ถึงเซตของหมายเลขทรานแซกชันที่มีเซตรายการ 'abd' ปรากฏอยู่ในฐานข้อมูลรายการ* TDB_1 *ได้ดังนี้* $T^{abd} = \{ 1^{abd}, 2^{abd}, 3^{abd}, 4^{abd}, 21^{abd}, 22^{abd}, 23^{abd}, 24^{abd} \}$ *โดยอัตรา (ร้อยละ) ค่าสนับสนุนของเซตรายการ 'abd' เท่ากับ 20% ของทรานแซกชันทั้งหมด แต่เซตรายการ 'abd' ไม่ปรากฏในฐานข้อมูลรายการ* TDB_2 *จึงมีค่าอัตรา (ร้อยละ) ค่าสนับสนุนของเซตรายการ 'abd' เท่ากับ 0 ดังนั้นเมื่อทำการคำนวณถึงอัตราการเติบโต จะได้* $GR^{abd} = \frac{s^{abd}_{TDB_2}}{s^{abd}_{TDB_1}} = \frac{0}{20\%} = 0$ *ดังนั้นเซตรายการ 'abd' เป็นเซตรายการที่ขาดหายไป เนื่องจากมีอัตราการเติบโตเท่ากับศูนย์*

2.2 งานวิจัยที่เกี่ยวข้อง

การค้นหาเซตรายการที่ปรากฏบ่อยเป็นปัญหาที่พิจารณาถึงความน่าสนใจของเซตรายการในแง่มุมของจำนวนครั้ง/ความถี่ของการปรากฏขึ้นในฐานข้อมูลรายการภายใต้ค่าขีดแบ่งสนับสนุนที่ผู้ใช้กำหนด โดยทำให้ทราบถึงเซตรายการที่มีการปรากฏร่วมกันบ่อย ๆ ในการค้นหาเซตรายการที่ปรากฏบ่อยนั้น ได้มีนักวิจัยพัฒนาขั้นตอนวิธีที่ซึ่งแตกต่างกันไปตามวัตถุประสงค์ของการสร้างขั้นตอนวิธีนั้นขึ้นมา โดยขั้นตอนวิธีการค้นหาเซตรายการที่ปรากฏบ่อยที่ได้รับความนิยมและเป็นที่รู้จักโดยเริ่มจาก (Agrawal, & Srikant, 1994) ได้นำเสนอการค้นหาเซตรายการที่ปรากฏบ่อยด้วยขั้นตอนวิธีอะพริออริ (Apriori) โดยการอ่านข้อมูลจากฐานข้อมูลรายการเพื่อสร้างเซตรายการแคนดิเดต (Candidate itemsets) สำหรับพิจารณาเซตรายการที่คาดว่าจะเป็นเซตรายการที่น่าสนใจ นอกจากนี้ได้มีการนำเสนอสมบัติปิดการลดลง (Downward closure property) กล่าวคือ ถ้าเซตรายการใด ๆ ไม่เป็นเซตรายการที่น่าสนใจ และซูเปอร์เซตทั้งหมดของเซตรายการก็จะไม่เป็นเซตรายการที่น่าสนใจด้วย ที่ซึ่งจะช่วยลดทอนเซตรายการที่ไม่เป็นผลลัพธ์ออกจากการพิจารณา ทำให้ประหยัดเวลาในการประมวลผลและลดหน่วยความจำในการจัดเก็บ

แต่เนื่องจากขั้นตอนวิธีนี้ มีการอ่านข้อมูลหลายครั้งและยังต้องทำการหาเซตรายการแคนดิเดตจำนวนมาก (Han, Pei, & Yin, 2000) จึงได้มีการพัฒนาขั้นตอนวิธีในการค้นหาเซตรายการที่ปรากฏบ่อยขึ้นมาใหม่ ด้วยการค้นหาเซตรายการที่ปรากฏบ่อยด้วยขั้นตอนวิธีเอฟพี-โกรท (Frequent pattern growth, FP-growth) เพื่อลดเวลาในการคำนวณ โดยไม่มีการหาเซตรายการแคนดิเดตทุกขนาด และจะอ่านข้อมูลจากฐานข้อมูลรายการเพียง 2 ครั้งเท่านั้น โดยใช้โครงสร้างข้อมูลที่มีชื่อว่า เอฟพี-ทรี (Frequent pattern tree, FP-tree) ที่ซึ่งเป็นโครงสร้างต้นไม้ที่ใช้เก็บ

[6] https://travel.kapook.com/view68159.html

ข้อมูลไว้ในโหนดรายการ (Node) แต่เนื่องจากในบางครั้งโครงสร้างต้นไม้มีขนาดใหญ่ ทำให้ต้องสร้างโหนดรายการเป็นจำนวนมาก และใช้เวลามากในการท่องไปยังโหนดที่ต้องการ

เพื่อลดเวลาที่ใช้ในการประมวลผลข้อมูลให้สามารถค้นหาเซตรายการที่ปรากฏบ่อยได้เร็วขึ้น (Zaki, 2000) ได้พัฒนาขั้นตอนวิธีการค้นหาเซตรายการที่ปรากฏบ่อยด้วยขั้นตอนวิธีอีแคลท (Equivaleuce class transformation, Eclat) ด้วยการอ่านข้อมูลจากฐานข้อมูลเพียงครั้งเดียวเท่านั้น และไม่มีการหาเซตรายการแคนดิเดตทุกขนาด โดยใช้การอินเตอร์เซกชัน (Intersection) ของเซตหมายเลขทรานแซกชันระหว่างสองเซตรายการ แต่หากว่าฐานข้อมูลรายการมีขนาดใหญ่มาก จึงเป็นผลทำให้การจัดเก็บหมายเลขทรานแซกชันของแต่ละเซตรายการต้องใช้พื้นที่หน่วยความจำในการจัดเก็บข้อมูลเป็นจำนวนมาก จึงได้มีนักวิจัยได้มีการพัฒนาโครงสร้างข้อมูลที่ใช้ในการจัดเก็บการปรากฏขึ้นของเซตรายการโดยการบีบอัดข้อมูล สามารถช่วยลดเวลาในการประมวลผลข้อมูล ลดพื้นที่หน่วยความจำในการจัดเก็บข้อมูลได้ อาทิเช่น บิตเวกเตอร์ (Bit-vectors) (Dong & Han, 2007) โดยถ้าเซตรายการใดปรากฏในทรานแซกชัน j^{th} จะทำให้บิตในลำดับที่ j^{th} มีค่าเป็น 1 และทางกลับกัน ถ้าเซตรายการใดไม่ปรากฏในทรานแซกชัน j^{th} จะทำให้บิตในลำดับที่ j^{th} มีค่าเป็น 0 (หมายเหตุ 8 บิต มีค่าเท่ากับ 1 ไบต์ (Byte)) หากบิตเวกเตอร์มีไบต์เป็น 0 จำนวนมาก ก็จะทำให้สิ้นเปลืองพื้นที่ในการจัดเก็บ รวมถึงสิ้นเปลืองเวลาในการประมวลผล (Vo, Hong, & Le, 2012) จึงได้นำเสนอไดนามิคบิตเวกเตอร์ (Dynamic bit-vectors) โดยจะทำการลดทอนไบต์ที่มีค่าเท่ากับ 0 ในหัวและท้ายของสายบิตเวกเตอร์ ที่ซึ่งสามารถใช้เวลาในการคำนวณที่รวดเร็วขึ้นและลดพื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลได้เพียงบางส่วนเท่านั้นต่อมา (Nguyen, Vo, Nguyen, & Pedrycz, 2016) ได้นำเสนออินเทอร์วอลเวิร์ดเซกเมนต์ (Interval word segment) โดยการลดทอนไบต์ที่เป็น 0 ทั้งหมด ทำให้ประหยัดเวลาและลดหน่วยความจำในการจัดเก็บได้อย่างมีประสิทธิภาพ

ในการค้นหาเซตรายการที่ปรากฏบ่อยมีหลาย ๆ งานวิจัยที่ได้นำขั้นตอนวิธีที่กล่าวมาข้างต้นไปประยุกต์ใช้ อาทิเช่น การค้นหาเซตรายการที่ปรากฏบ่อยแบบปิดเคอันดับแรกโดยหลีกเลี่ยงค่าขีดแบ่งสนับสนุน (Han, Wang, Lu, & Tzvetkov, 2002) การค้นหาเซตรายการที่ปรากฏบ่อยและตรวจสอบจากหน้าต่างขนาดใหญ่บนข้อมูลกระแส (Data streams) (Mozafari, Thakkar, & Zaniolo, 2008) การค้นหาเซตรายการที่ปรากฏบ่อยอย่างเป็นลำดับจากฐานข้อมูลรายการตามความน่าจะเป็น (Muzammal & Raman, 2011) นอกจากนี้การค้นหาเซตรายการที่ปรากฏบ่อยถูกนำไปใช้ในด้านอื่น ๆ อย่างกว้างขวาง อาทิเช่น ทางด้านการแพทย์และการวิเคราะห์ข้อมูลทางชีวภาพ (Sallaberry, Pecheur, Bringay, roche, & Teisseire, 2011) ตลาดหุ้นและการวิเคราะห์เครือข่ายโปรตีน (Sim, Li, Gopalkrishnan, & Liu, 2009) การวิเคราะห์สภาพแวดล้อมเครือข่าย (Fang, Deng, & Ma, 2009) การวิเคราะห์ข้อมูลทางจราจร (Liu, Zheng, Chawla, Yuan, & Xing, 2011) และอื่น ๆ

จากงานวิจัยดังกล่าวได้พิจารณาการค้นหาเซตรายการที่น่าสนใจแค่ในแง่มุมของการปรากฏบ่อยเท่านั้น ด้วยเหตุนี้ (Tanbeer, Ahmed, Jeong, & Lee, 2009) จึงได้ทำการพิจารณาในแง่มุมของการปรากฏบ่อยร่วมกับปรากฏสม่ำเสมอ โดยการค้นหาเซตรายการที่ปรากฏบ่อยร่วมกับ

ปรากฏสม่ำเสมอนั้น ทำให้สามารถทราบถึงเซตรายการที่ปรากฏขึ้นร่วมกันบ่อย ๆ พร้อมกับเซตรายการที่ปรากฏขึ้นสม่ำเสมอภายใต้ค่าขีดแบ่งสนับสนุนและค่าความสม่ำเสมอที่ผู้ใช้กำหนด ที่ซึ่งบ่งบอกถึงระยะห่างหรือช่วงเวลาที่มากที่สุดในการปรากฏขึ้น/ไม่ปรากฏขึ้นของเซตรายการ ต่อมาได้มีนักวิจัยและผู้ที่สนใจได้ทำการพัฒนาต่อยอด อาทิเช่น การค้นหาเซตรายการที่ปรากฏบ่อยและปรากฏสม่ำเสมอเคอันดับแรกจากฐานข้อมูลรายการโดยหลีกเลี่ยงค่าขีดแบ่งสนับสนุน (Amphawan, Lenca, & Surarerks, 2009) การค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอในฐานข้อมูลรายการกระแส (Tanbeer, Ahmed, & Jeong, 2010a) การค้นหาเซตรายการที่ปรากฏบ่อยและสม่ำเสมอในฐานข้อมูลรายการที่เพิ่มขึ้น (Tanbeer, Ahmed, & Jeong, 2010b) การค้นหาเซตรายการที่ปรากฏบ่อยและปรากฏสม่ำเสมอเคอันดับแรกในเซตรายการแบบปิด (Amphawan & Lenca, 2015) นอกจากนี้ยังสามารถประยุกต์ใช้การค้นหาเซตรายการในแง่มุมของการปรากฏบ่อยร่วมกับปรากฏสม่ำเสมอกับงานหลาย ๆ ด้าน อาทิเช่น การค้นหาเซตรายการที่มีกลุ่มของดัชนีหุ้นที่เพิ่มขึ้นบ่อยและสม่ำเสมอสำหรับเพิ่มความสนใจให้กับนักลงทุน การวิเคราะห์เว็บไซต์ (Website) เชิงพาณิชย์ (Shah & Kaur, 2014) ด้วยการค้นหาเซตรายการที่มีการเข้าดูข้อมูลบนเว็บไซต์ที่มีความบ่อยและสม่ำเสมอเพื่อปรับปรุงเวลาในการเข้าถึงและพัฒนาเว็บไซต์ให้มีความน่าสนใจมากยิ่งขึ้น การวิเคราะห์เครือข่ายเซ็นเซอร์ร่างกาย (Tanbeer et al., 2015) ที่ช่วยให้แพทย์สามารถติดตามพฤติกรรม/กิจกรรมสำหรับการประเมินสุขภาพของผู้ป่วยได้สะดวกสบายมากยิ่งขึ้นและอื่น ๆ

นอกเหนือจากงานวิจัยข้างต้น ยังมีงานวิจัยที่ได้ทำการพิจารณาถึงแนวโน้มการเปลี่ยนแปลงของพฤติกรรมของการปรากฏขึ้น ซึ่งในการพิจารณาแนวโน้มที่ปรากฏขึ้นเมื่อเทียบกับเวลาที่ผ่านไปหรือการพิจารณาความแตกต่างระหว่างพฤติกรรมการปรากฏขึ้นของข้อมูล (Dong & Li, 1999) ได้คิดค้นการค้นหาเซตรายการภายใต้ความเปลี่ยนแปลงลักษณะของการปรากฏ โดยในเริ่มต้นได้ทำการพิจารณาเซตรายการที่มีความแตกต่างในด้านความถี่/จำนวนครั้งของการปรากฏของเวลาที่ผ่านไปโดยเซตรายการใดก็ตามที่มีความแตกต่างของจำนวนครั้งอย่างมีนัยสำคัญจะถูกเป็นเซตรายการที่น่าสนใจ การค้นหาเซตรายการภายใต้ความเปลี่ยนแปลงลักษณะของการปรากฏถูกประยุกต์ใช้ในหลาย ๆ แขนง เช่น 1) ธุรกิจท่องเที่ยวสามารถทำการประยุกต์ใช้การค้นหาเซตรายการภายใต้ความเปลี่ยนแปลงลักษณะของการปรากฏ ในการค้นหาความเปลี่ยนแปลงของปัจจัยที่ส่งผลต่อการเลือกจองโรงแรมเพื่อเข้าพักระหว่างท่องเที่ยว (Li, Law, Vu, Rong, & Zhao, 2015) ซึ่งจากการค้นหาความเปลี่ยนแปลงดังกล่าวจะทำให้โรงแรมต่าง ๆ สามารถปรับปรุงกลยุทธ์การดำเนินธุรกิจ ปรับปรุงกลยุทธ์ทางการตลาด และปรับปรุงคุณภาพของการบริการเพื่อที่จะทำให้ตรงต่อความต้องการของลูกค้าได้มากขึ้น 2) การวิเคราะห์ข้อมูลทางชีวสารสนเทศ ที่ซึ่งจะประยุกต์ใช้การค้นหาความเปลี่ยนแปลงเพื่อทำการค้นหากลุ่มของยีนที่น่าสนใจ (Li & Wong, 2002), (Wang, Zhao, Zhao, Wang, & Qiao, 2010) 3) การแพทย์ได้ประยุกต์ใช้การค้นหาความเปลี่ยนแปลงในการค้นหาความเปลี่ยนแปลงของผู้ป่วยโรคมะเร็งเมื่อได้รับยา (Huang, Gan, Lu, & Huan, 2013) และในด้านอื่น ๆ การตรวจสอบการเปลี่ยนแปลงพฤติกรรมของผู้บริโภค (Kim, Song, & Kim, 2005) การตรวจสอบการเปลี่ยนแปลงสำหรับเซตรายการอย่างเป็นลำดับ (Tsai & Shieh, 2009) การค้นหา

เซตรายการภายใต้การเปลี่ยนแปลงพฤติกรรมของผู้บริโภคในห้างสรรพสินค้าออนไลน์ (Song, Kim, & Kim, 2001) การค้นหาเซตรายการภายใต้การเปลี่ยนแปลงพฤติกรรมของผู้บริโภคในตลาดค้าปลีก (Chen, Chiu, & Chang, 2006) การค้นหาเซตรายการภายใต้การแข่งขันสูงโดยพิจารณาการเปลี่ยนแปลงในแนวโน้มของสิทธิบัตร (Shih, Liu, & Hsu, 2010) การค้นหาเซตรายการภายใต้ความเปลี่ยนแปลงลักษณะการปรากฏเพื่อช่วยในการค้นพบความรู้ทางพิษวิทยา (Coquin et al., 2015) การวิเคราะห์ดนตรีโฟล์ก (Folk music) สำหรับความแตกต่างของเซตรายการ (Neubarth & Conklin, 2016) การวิเคราะห์เทคโนโลยีโฟโตโวลตาอิค (Photovoltaics Technology) (García-Vico, Montes, Aguilera, Carmona, & Jesus, 2016) โดยจากประโยชน์ที่ค่อนข้างหลากหลายของการค้นหาเซตรายการภายใต้ความเปลี่ยนแปลงลักษณะของการปรากฏ ที่ซึ่งสามารถประยุกต์ใช้ได้กับแขนงต่าง ๆ จึงเป็นเหตุให้การค้นหาเซตรายการภายใต้ความเปลี่ยนแปลงลักษณะของการปรากฏยังคงได้รับความสนใจจากนักวิจัยต่าง ๆ ที่ซึ่งพยายามที่จะวิเคราะห์ลักษณะการเปลี่ยนแปลงในแอปพลิเคชันต่าง ๆ

การค้นหาเซตรายการที่น่าสนใจที่กล่าวมาข้างต้นทั้ง 3 แง่มุม ได้แก่ 1) การค้นหาเซตรายการที่ปรากฏบ่อย เป็นการค้นหาเซตรายการที่น่าสนใจภายใต้การพิจารณาจำนวนครั้ง/ความบ่อย/ความถี่ในการปรากฏขึ้น 2) การค้นหาเซตรายการที่ปรากฏบ่อยร่วมกับปรากฏสม่ำเสมอเป็นการค้นหาเซตรายการที่น่าสนใจภายใต้การพิจารณาจำนวนครั้ง/ความบ่อย/ความถี่ร่วมกับความสม่ำเสมอในการปรากฏขึ้นของเซตรายการนั้น ๆ และ 3) การค้นหาเซตรายการภายใต้ความเปลี่ยนแปลงลักษณะของการปรากฏ โดยทำการพิจารณาเซตรายการที่มีความแตกต่างในด้านความถี่/จำนวนครั้งของการปรากฏของเวลาที่ผ่านไป แต่อย่างไรก็ตาม การค้นหาเซตรายการภายใต้ความเปลี่ยนแปลงลักษณะของการปรากฏ โดยพิจารณาเซตรายการที่มีความแตกต่างในด้านของการปรากฏอย่างสม่ำเสมอของเวลาที่ผ่านไป ที่ซึ่งเป็นอีกแง่มุมที่น่าสนใจเช่นกัน โดยเซตรายการใดก็ตามที่มีความแตกต่างในช่วงหรือระยะห่างที่มากที่สุดของทรานแซกชันอย่างมีนัยสำคัญจะถูกเป็นเซตรายการที่น่าสนใจ

2.3 คุณลักษณะของฐานข้อมูลรายการที่ใช้ในการทดลอง

ในงานวิทยานิพนธ์นี้ได้ใช้ข้อมูลที่เผยแพร่จากองค์กรต่าง ๆ (Public data) และข้อมูลมาตรฐาน (Benchmark data) สำหรับการค้นหาเซตรายการที่น่าสนใจ ที่ซึ่งเป็นข้อมูลที่ได้รับความน่าเชื่อถือ สามารถดาวน์โหลด (Download) จากเว็บไซต์ fimi[7] โดยข้อมูลที่ใช้ประกอบไปด้วยข้อมูล 2 ประเภท ได้แก่ 1) ข้อมูลจริงมีทั้งหมด 8 ฐานข้อมูลรายการ ได้แก่ Accidents, Chess, Connect, Kosarak, Mushroom, Pumsb, Pumsb* และ Retail 2) ข้อมูลที่ถูกสังเคราะห์ขึ้นซึ่งจัดทำและเผยแพร่โดย IBM Almaden[8] มีทั้งหมด 2 ฐานข้อมูลรายการ ได้แก่ T10I4D100K และ T40I10D100K โดยในงานวิทยานิพนธ์นี้จะแบ่งครึ่งฐานข้อมูลรายการออกเป็นสองส่วนเท่า ๆ กัน (ฐานข้อมูลรายการ TDB_1 และ ฐานข้อมูลรายการ TDB_2) แสดงดังตารางที่ 2.1

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

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

ตารางที่ 2.1 คุณลักษณะของฐานข้อมูลรายการ

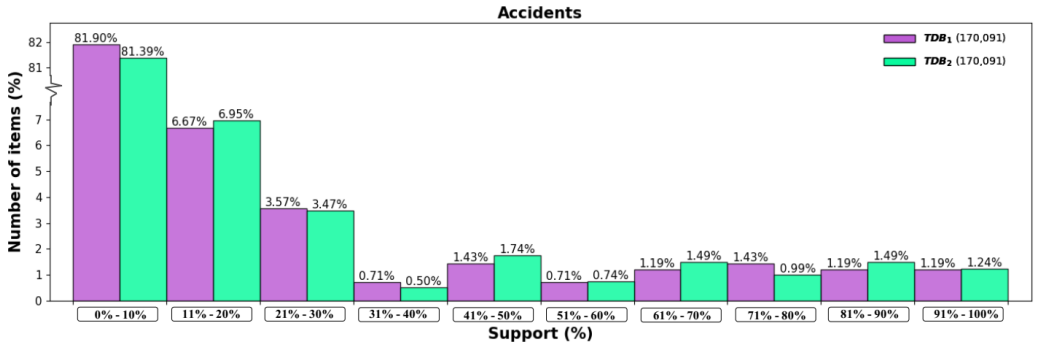
ฐานข้อมูลรายการ	จำนวนรายการ	จำนวนทรานแซกชัน	จำนวนความยาวเฉลี่ยทรานแซกชัน	ลักษณะข้อมูล
Accidents	468	340,182	33.8	หนาแน่น
Chess	75	3,196	37	หนาแน่น
Connect	129	67,556	43	หนาแน่น
Mushroom	119	8,124	23	หนาแน่น
Pumsb	7,117	49,046	74	หนาแน่น
Pumsb*	7,117	49,046	50.5	หนาแน่น
Kosarak	41,270	990,002	8.1	เบาบาง
Retail	16,470	88,162	10.3	เบาบาง
T10I4D100K	1,000	100,000	10	เบาบาง
T40I10D100K	1,000	100,000	40	เบาบาง

ตารางที่ 2.1 แสดงคุณลักษณะของฐานข้อมูลรายการทั้งข้อมูลจริงและข้อมูลที่ถูกสังเคราะห์ขึ้น โดยแสดงให้เห็นถึงจำนวนรายการ จำนวนทรานแซกชัน จำนวนความยาวเฉลี่ยของทรานแซกชัน และลักษณะของข้อมูลในแต่ละฐานข้อมูลรายการ

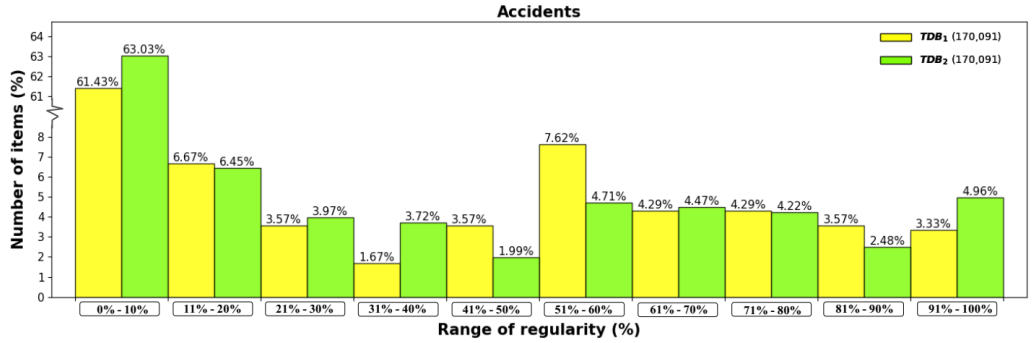
จากข้อมูลดังกล่าวข้างต้น การค้นหาเซตรายการที่น่าสนใจจะต้องพิจารณาถึงลักษณะของข้อมูลในแต่ละฐานข้อมูลรายการก่อน เพื่อสามารถตรวจสอบได้ว่าขั้นตอนวิธีที่นำเสนอสามารถทำงานได้อย่างมีประสิทธิภาพกับข้อมูลที่มีลักษณะใด โดยพิจารณาจากค่าสนับสนุนและค่าความสม่ำเสมอของรายการที่ปรากฏขึ้นในแต่ละฐานข้อมูลรายการ ที่ซึ่งในการพิจารณาจะแบ่งค่าสนับสนุนและค่าความสม่ำเสมอเป็นอัตรา (ร้อยละ) ของรายการที่ปรากฏขึ้นในแต่ละฐานข้อมูลรายการทั้งหมด 10 ช่วง ได้แก่ 0%-10%, 11%-20%, 21%-30%, 31%-40%, 41%-50%, 51%-60%, 61%-70%, 71%-80%, 81%-90% และ 91%-100%

ฐานข้อมูลรายการ Accidents เป็นฐานข้อมูลรายการที่รวบรวมข้อมูลการเกิดอุบัติเหตุทางจราจรบริเวณแถบพื้นที่ทางตอนเหนือของประเทศเบลเยี่ยม (Flanders) ในช่วงปีคริสต์ศักราช 1991-2000 ที่ซึ่งจัดเก็บโดยสถาบันสถิติแห่งชาติ (National Institute of Statistics, NIS) สังเกตได้ว่ามีรายการที่ปรากฏขึ้นของค่าสนับสนุนในช่วง 0%-10% เป็นส่วนมาก (หมายเหตุ 0%-10% ของ 170,091 ทรานแซกชัน คือ รายการที่มีจำนวนครั้งของการปรากฏระหว่าง 0-17,009 ครั้ง) โดยในฐานข้อมูลรายการ TDB_1 มีรายการที่ปรากฏขึ้น 81.90% (หมายเหตุ ฐานข้อมูลรายการ TDB_1 มีรายการที่ปรากฏขึ้น 383 รายการ) และฐานข้อมูลรายการ TDB_2 มีรายการที่ปรากฏขึ้น 81.39% (หมายเหตุ ฐานข้อมูลรายการ TDB_2 มีรายการที่ปรากฏขึ้น 381 รายการ) แสดงดังภาพที่ 2.3 และรายการส่วนมากมีความสม่ำเสมอในการปรากฏอยู่ในช่วง 0%-10% (หมายเหตุ 0%-10% ของ

170,091 ทรานแซกชัน คือ รายการมีการปรากฏขึ้นอย่างน้อยหนึ่งทรานแซกชัน มีระยะห่างกันในช่วงที่ไม่เกิน 17,009) โดยในฐานข้อมูลรายการ TDB_1 มีรายการที่ปรากฏขึ้น 61.43% (หมายเหตุ ฐานข้อมูลรายการ TDB_1 มีรายการที่มีความสม่ำเสมอ 287 รายการ) และฐานข้อมูลรายการ TDB_2 มีรายการที่ปรากฏขึ้น 63.03% (หมายเหตุ ฐานข้อมูลรายการ TDB_2 มีรายการที่มีความสม่ำเสมอ 295 รายการ) แสดงดังภาพที่ 2.4

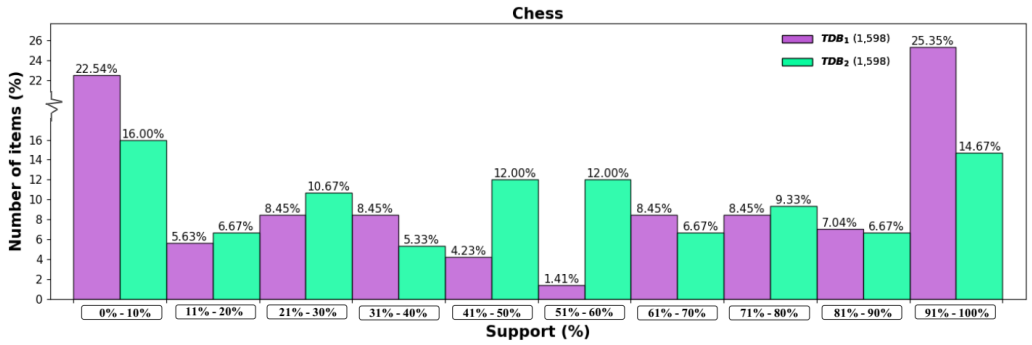


ภาพที่ 2.3 กราฟแสดงค่าสนับสนุนของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ Accidents

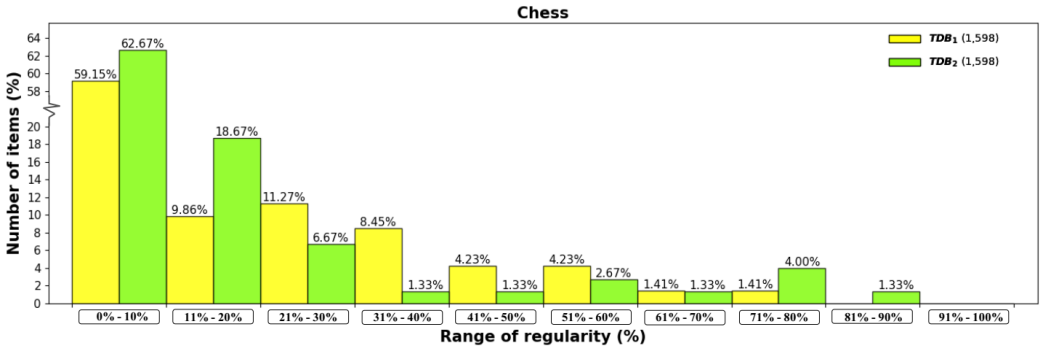


ภาพที่ 2.4 กราฟแสดงค่าความสม่ำเสมอของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ Accidents

ฐานข้อมูลรายการ Chess และ Connect ถูกรวบรวมจาก UCI Machine Learning Repository[9] ที่ซึ่งจัดเก็บวิธีการเดินหมากในระหว่างการแข่งขันในแต่ละทรานแซกชัน โดยในฐานข้อมูลรายการ Chess มีรายการที่ปรากฏขึ้นของค่าสนับสนุนใกล้เคียงกันในทุก ๆ ช่วง แสดงดังภาพที่ 2.5 และรายการส่วนมากมีความสม่ำเสมอในการปรากฏอยู่ในช่วง 0%-10% ของทรานแซกชันทั้งหมดในฐานข้อมูลรายการ แสดงดังภาพที่ 2.6



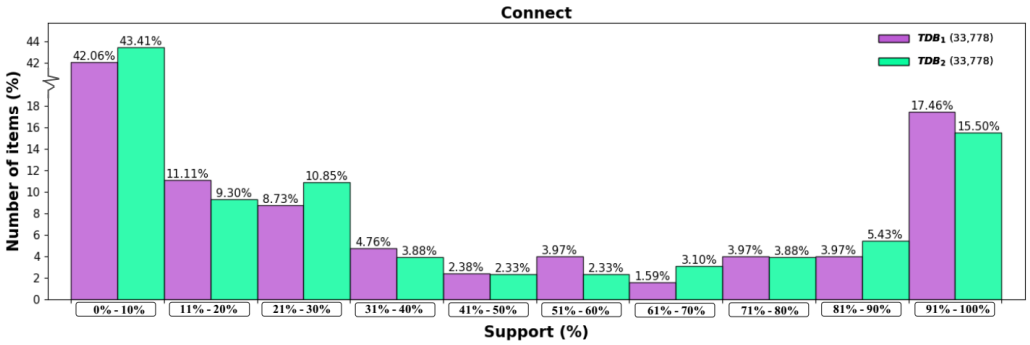
ภาพที่ 2.5 กราฟแสดงค่าสนับสนุนของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ Chess



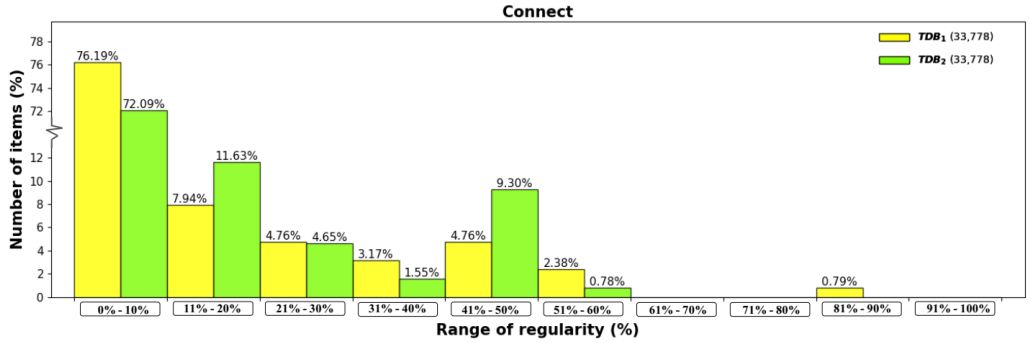
ภาพที่ 2.6 กราฟแสดงค่าความสม่ำเสมอของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ Chess

[9] http://archive.ics.uci.edu/ml/

ฐานข้อมูลรายการ Connect มีรายการที่ปรากฏขึ้นของค่าสนับสนุนในช่วง 0%-10% เป็นส่วนมาก โดยในฐานข้อมูลรายการ TDB_1 มีรายการปรากฏขึ้น 42.06% และฐานข้อมูลรายการ TDB_2 มีรายการปรากฏขึ้น 43.41% ส่วนรายการที่เหลือมีการปรากฏขึ้นในแต่ละช่วงมากน้อยต่อเนื่องกัน แสดงดังภาพที่ 2.7 และรายการส่วนมากมีความสม่ำเสมอในการปรากฏอยู่ในช่วง 0%-10% ของทรานแซกชันทั้งหมดในฐานข้อมูลรายการ แสดงดังภาพที่ 2.8

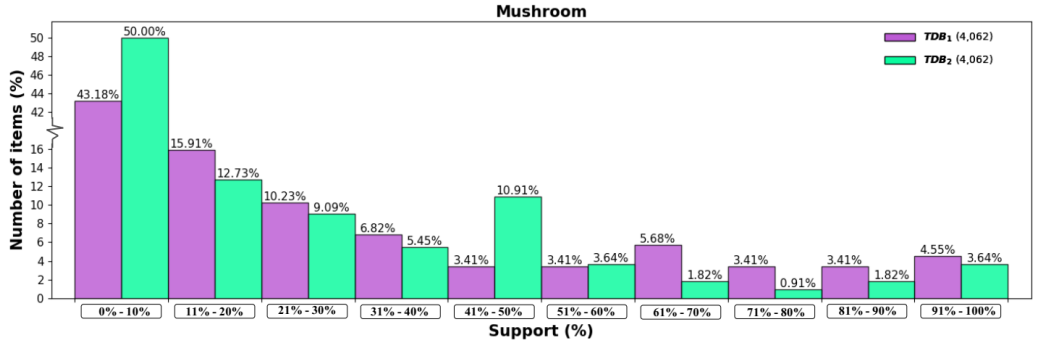


ภาพที่ 2.7 กราฟแสดงค่าสนับสนุนของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ Connect

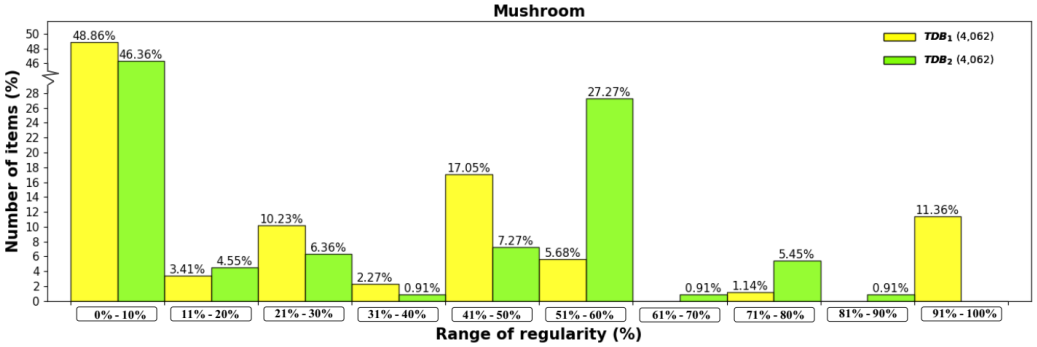


ภาพที่ 2.8 กราฟแสดงค่าความสม่ำเสมอของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ Connect

ฐานข้อมูลรายการ Mushroom เป็นฐานข้อมูลรายการที่จัดเก็บคุณลักษณะของสายพันธุ์ของเห็ดแต่ละชนิด โดยรายการส่วนมากในการปรากฏขึ้นของค่าสนับสนุนและความสม่ำเสมออยู่ในช่วง 0%-10% ของทรานแซกชันทั้งหมดในฐานข้อมูลรายการ แสดงดังภาพที่ 2.9 และภาพที่ 2.10

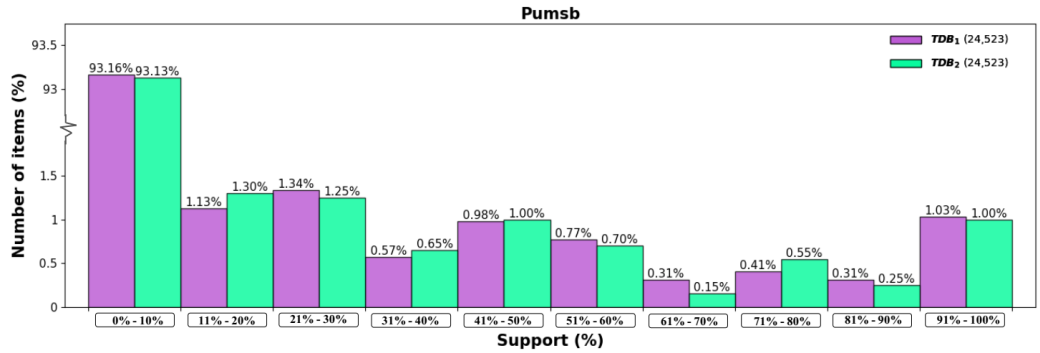


ภาพที่ 2.9 กราฟแสดงค่าสนับสนุนของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ Mushroom

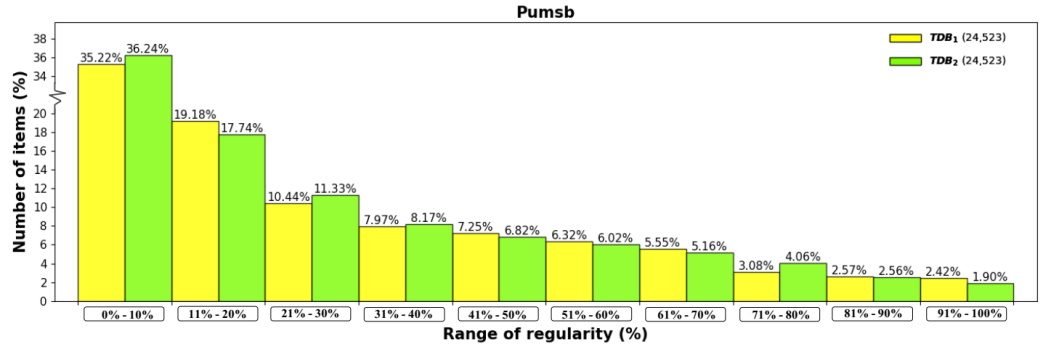


ภาพที่ 2.10 กราฟแสดงค่าความสม่ำเสมอของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ Mushroom

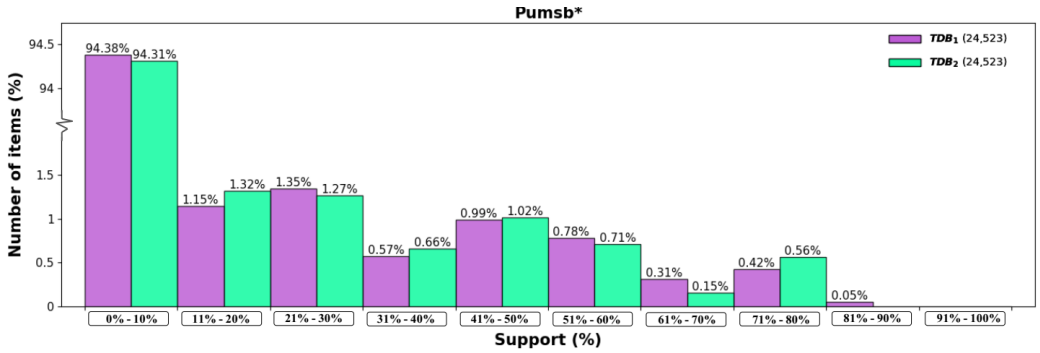
ฐานข้อมูลรายการ Pumsb และฐานข้อมูลรายการ Pumsb* เป็นฐานข้อมูลรายการที่จัดเก็บข้อมูลการสำรวจสำมะโนประชากรและที่อยู่อาศัย ที่ซึ่งทั้งสองฐานข้อมูลรายการนี้มีรายการในการปรากฏขึ้นของค่าสนับสนุนและความสม่ำเสมออยู่ในช่วง 0%-10% เป็นส่วนมาก แสดงดังภาพที่ 2.11 ภาพที่ 2.12 ภาพที่ 2.13 และ ภาพที่ 2.14



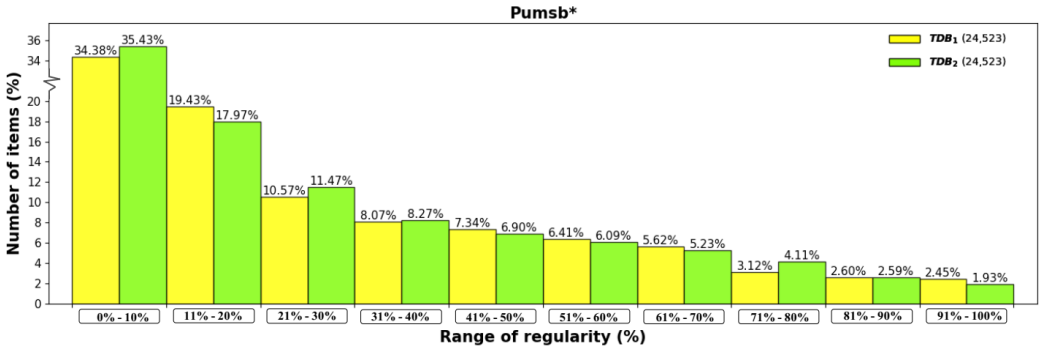
ภาพที่ 2.11 กราฟแสดงค่าสนับสนุนของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ Pumsb



ภาพที่ 2.12 กราฟแสดงค่าความสม่ำเสมอของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ Pumsb

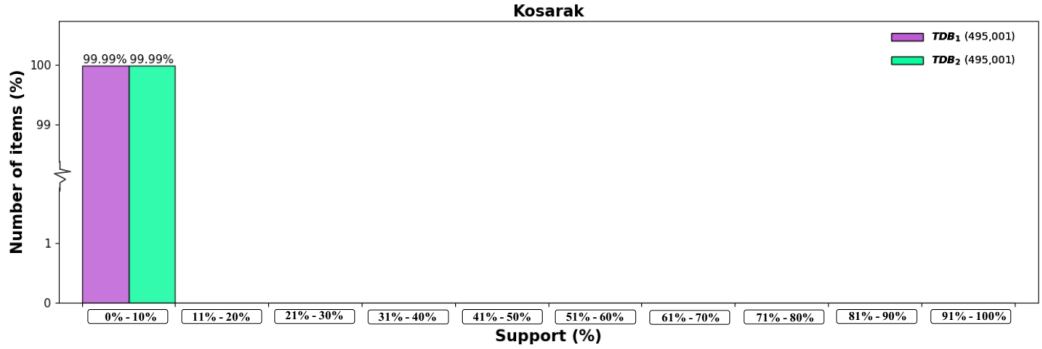


ภาพที่ 2.13 กราฟแสดงค่าสนับสนุนของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ Pumsb*

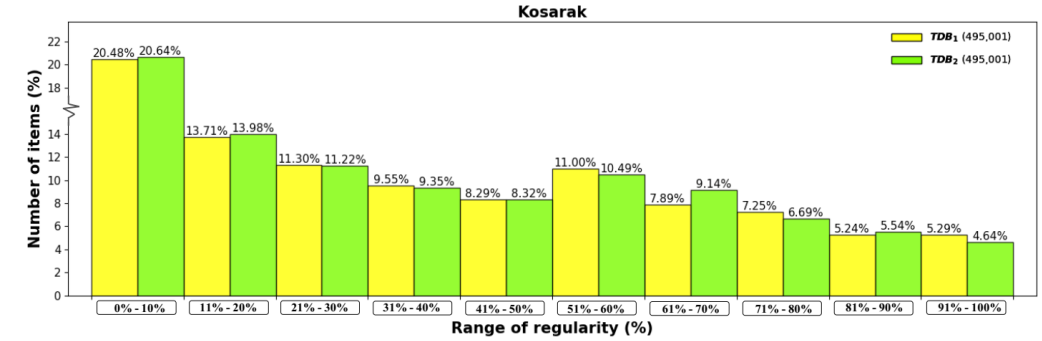


ภาพที่ 2.14 กราฟแสดงค่าความสม่ำเสมอของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ Pumsb*

ฐานข้อมูลรายการ Kosarak เป็นฐานข้อมูลรายการที่จัดเก็บข้อมูลการคลิกบนเว็บไซต์ข่าวออนไลน์ของประเทศฮังการี โดยรายการทั้งหมดมีการปรากฏขึ้นของค่าสนับสนุนอยู่ในช่วง 0%-10% แสดงดังภาพที่ 2.15 และรายการมีความสม่ำเสมอในการปรากฏอยู่ในทุก ๆ ช่วงของฐานข้อมูลรายการ แสดงดังภาพที่ 2.16

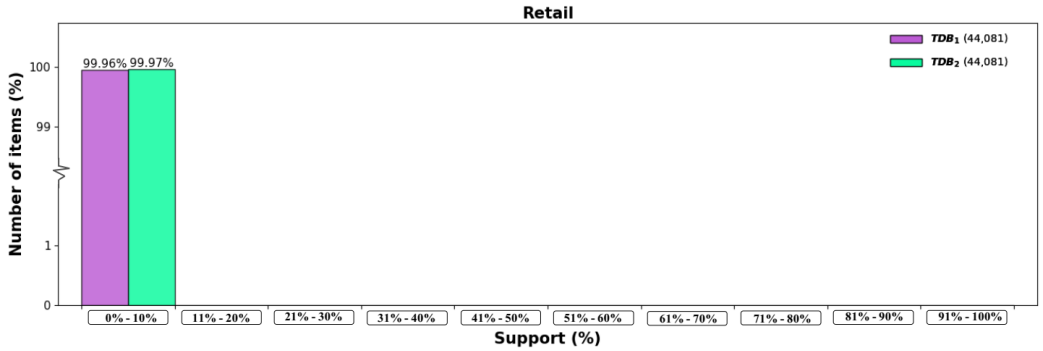


ภาพที่ 2.15 กราฟแสดงค่าสนับสนุนของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ Kosarak

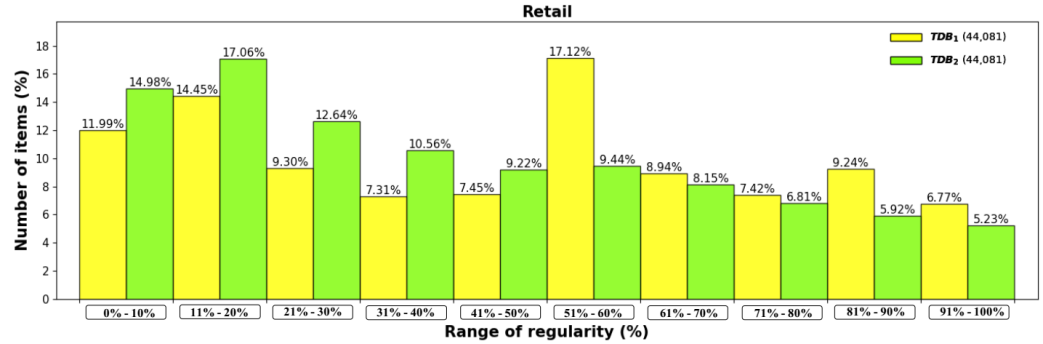


ภาพที่ 2.16 กราฟแสดงค่าความสม่ำเสมอของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ Kosarak

ฐานข้อมูลรายการ Retail ได้จัดเก็บข้อมูลทรานแซกชันในแต่ละตะกร้าสินค้า ณ ร้านค้าในประเทศเบลเยี่ยมตั้งแต่เดือนธันวาคมปีคริสต์ศักราช 1999 จนถึงเดือนพฤศจิกายนปีคริสต์ศักราช 2000 ที่ซึ่งรายการทั้งหมดมีการปรากฏขึ้นของค่าสนับสนุนอยู่ในช่วง 0%-10% แสดงดังภาพที่ 2.17 และรายการมีความสม่ำเสมอในการปรากฏอยู่ในทุก ๆ ช่วงของฐานข้อมูลรายการ แสดงดังภาพที่ 2.18

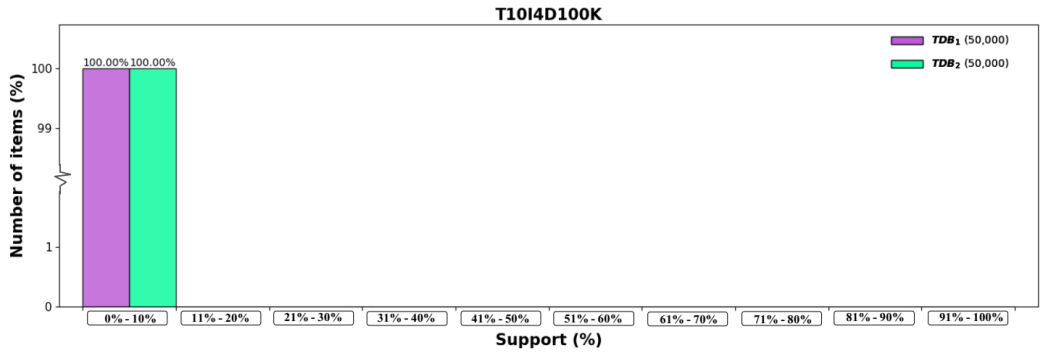


ภาพที่ 2.17 กราฟแสดงค่าสนับสนุนของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ Retail

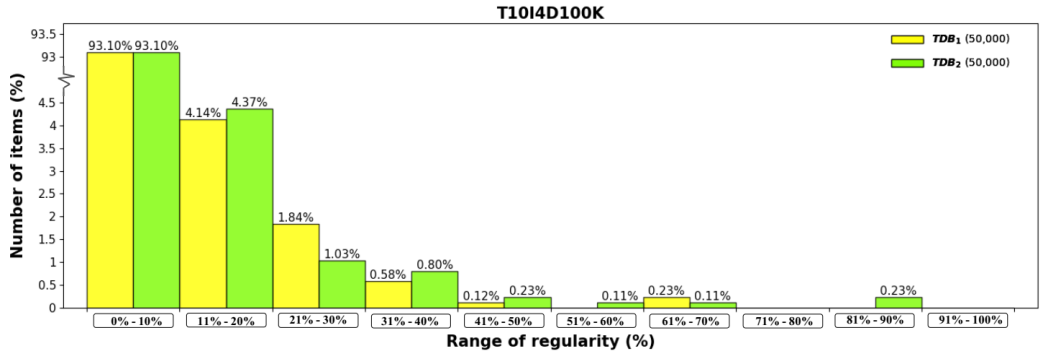


ภาพที่ 2.18 กราฟแสดงค่าความสม่ำเสมอของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ Retail

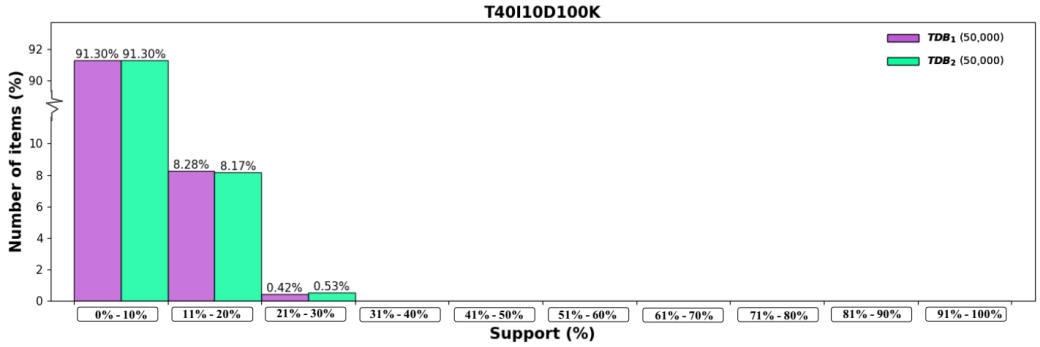
ฐานข้อมูลรายการ T10I4D100K และฐานข้อมูลรายการ T40I10D100K เป็นข้อมูลสังเคราะห์ที่จำลองทรานแซกชันของการซื้อสินค้าในธุรกิจค้าปลีก โดยทั้งสองฐานข้อมูลรายการนี้มีรายการในการปรากฏขึ้นของค่าสนับสนุนและความสม่ำเสมออยู่ในช่วง 0%-10% เป็นส่วนมาก แสดงดังภาพที่ 2.19 ภาพที่ 2.20 ภาพที่ 2.21 และ ภาพที่ 2.22



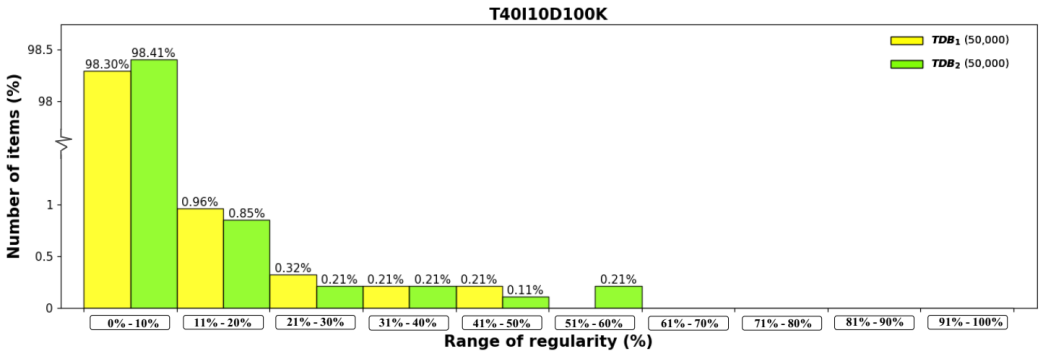
ภาพที่ 2.19 กราฟแสดงค่าสนับสนุนของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ T10I4D100K



ภาพที่ 2.20 กราฟแสดงค่าความสม่ำเสมอของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ T10I4D100K



ภาพที่ 2.21 กราฟแสดงค่าสนับสนุนของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ T40I10D100K



ภาพที่ 2.22 กราฟแสดงค่าความสม่ำเสมอของรายการที่ปรากฏขึ้นในฐานข้อมูลรายการ T40I10D100K

บทที่ 3
การค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น

จากบทที่กล่าวมาในข้างต้น ได้มีหลายงานวิจัยนำเสนอการค้นหาเซตรายการภายใต้การวัดความน่าสนใจในแง่มุมต่าง ๆ โดยมีงานวิจัยหนึ่งที่สนใจเกี่ยวกับความเปลี่ยนแปลงของการปรากฏขึ้นในแง่มุมของความถี่ที่เพิ่มขึ้นเมื่อเวลาผ่านไป แต่อย่างไรก็ตาม การค้นหาเซตรายการภายใต้การเปลี่ยนแปลงลักษณะของการปรากฏอย่างสม่ำเสมอก็เป็นอีกแง่มุมที่น่าสนใจเช่นกัน ที่ซึ่งสามารถช่วยให้ทราบถึงพฤติกรรมการซื้อสินค้าที่ปรากฏขึ้นอย่างสม่ำเสมอ ที่จะช่วยให้บริษัทจะได้รับข้อมูลเพื่อใช้ประกอบการตัดสินใจ ซึ่งจากการทราบถึงข้อมูลดังกล่าวจะทำให้ผู้บริหารสามารถคิดกลยุทธ์ เพื่อสามารถกระตุ้นการซื้อสินค้าของผู้บริโภคอันนำมาซึ่งผลประกอบการของบริษัทที่เพิ่มขึ้นได้

ด้วยเหตุนี้ ในบทนี้จึงนำเสนอการค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏ (Discovering interesting itemsets based on change in regularity of occurrence) ด้วยการพิจารณาความเปลี่ยนแปลงของพฤติกรรมการปรากฏขึ้นในแง่มุมความสม่ำเสมอที่เพิ่มขึ้นเมื่อเวลาผ่านไปภายใต้ค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด (Change value threshold, σ_c) นอกจากนั้นจะนำเสนอขั้นตอนวิธีการทำงานที่เรียกว่า ไมโคร (Mining interesting Itemsets based on their Change on Regularity of Occurrence, *MICRO*) ที่ซึ่งประยุกต์ใช้โครงสร้างต้นไม้ที่เรียกว่า อิคโคร-ทรี (ICRO-tree) ที่จะสามารถช่วยให้อ่านข้อมูลจากฐานข้อมูลรายการเพียงครั้งเดียวเท่านั้น นอกจากนี้ยังมีการลดทอนจากสมบัติปิดการลดลง เพื่อลดเวลาในการประมวลผลข้อมูลและลดพื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลได้อย่างมีประสิทธิภาพ โดยในการค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอที่ปรากฏขึ้นนั้น สามารถแบ่งออกเป็น 2 ขั้นตอนวิธี คือ ขั้นตอนวิธีการสร้างอิคโคร-ทรี (ICRO-tree construction) และการค้นหาเซตรายการภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอที่ปรากฏขึ้นด้วยขั้นตอนวิธีอิคโคร-โกรท (ICRO-growth) โดยนิยามและรายละเอียดขั้นตอนวิธีการทำงาน สามารถอธิบายได้ดังนี้

3.1 นิยาม

ในงานวิจัยนี้ได้ประยุกต์ใช้นิยามพื้นฐานดังที่กล่าวในบทที่ 2 บางส่วน อาทิเช่น การคำนวณอัตรา (ร้อยละ) ค่าความสม่ำเสมอ การคำนวณอัตราการเติบโตหรือการเพิ่มขึ้นของค่าสนับสนุน แต่สำหรับการพิจารณาความเปลี่ยนแปลงของพฤติกรรมการปรากฏขึ้นอย่างสม่ำเสมอที่เพิ่มขึ้นเมื่อเวลาผ่านไปภายใต้ค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด จะมีนิยามที่แตกต่างจากเดิมที่ซึ่งสามารถนิยามได้ ดังนี้

นิยามที่ 3.1 *c^X คือ อัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นในรายการที่มีค่าความสม่ำเสมอเพิ่มขึ้นจากฐานข้อมูลรายการ TDB_1 และ TDB_2 (หมายเหตุ โดยฐานข้อมูลรายการไม่จำเป็นต้องมีขนาดเท่ากัน) สามารถคำนวณได้ ดังนี้*

$$c^X = \begin{cases} \text{ไม่สามารถระบุได้} & \text{, ถ้ารายการ X ไม่ปรากฏขึ้นในฐานข้อมูลรายการ } TDB_1 \text{ และ } TDB_2 \\ \dfrac{r^X_{TDB_2}}{r^X_{TDB_1}} & \text{, ถ้ารายการ X ปรากฏขึ้นในทั้งสองฐานข้อมูลรายการ } TDB_1 \text{ และ } TDB_2 \\ \infty & \text{, ถ้ารายการ X ไม่ปรากฏขึ้นในฐานข้อมูลรายการ } TDB_1 \\ 0 & \text{, ถ้ารายการ X ไม่ปรากฏขึ้นในฐานข้อมูลรายการ } TDB_2 \end{cases} \quad (3.1)$$

นิยามที่ 3.2 *เซตรายการ X จะเป็นเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น ก็ต่อเมื่อเซตรายการ X ปรากฏขึ้นทั้งสองฐานข้อมูลรายการและมีค่า c^X มากกว่าหรือเท่ากับค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด*

ปัญหาการค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น จะเป็นการค้นหาเซตรายการที่ปรากฏขึ้นทั้งสองฐานข้อมูลรายการ และมีอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นมากกว่าหรือเท่ากับค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด

ตัวอย่างที่ 3.1 *จากฐานข้อมูลรายการ TDB_1 ประกอบด้วย 40 ทรานแซกชัน และฐานข้อมูลรายการ TDB_2 ประกอบด้วย 56 ทรานแซกชัน แสดงดังภาพที่ 2.2 เมื่อทำการพิจารณารายการ ‘a’ สามารถระบุได้ถึงเซตของหมายเลขทรานแซกชันที่มีรายการ ‘a’ ปรากฏอยู่ในฐานข้อมูลรายการ TDB_1 ดังนี้ $T^a = \{ 1^a, 2^a, 3^a, 4^a, 5^a, 6^a, 21^a, 22^a, 23^a, 24^a, 25^a \}$ โดยอัตรา (ร้อยละ) ค่าความสม่ำเสมอ $r^a_{TDB_1}$ มีค่าเท่ากับ 37.5% ของทรานแซกชันทั้งหมด และสามารถระบุได้ถึงเซตของหมายเลขทรานแซกชันที่มีรายการ ‘a’ ปรากฏอยู่ในฐานข้อมูลรายการ TDB_2 ได้ดังนี้ $T^a_{TDB_2} = \{ 1^a, 2^a, 3^a, 4^a, 5^a, 6^a, 7^a, 8^a, 9^a, 11^a, 12^a, 13^a, 14^a, 15^a, 16^a, 44^a, 45^a, 46^a, 47^a, 48^a, 49^a, 50^a, 51^a, 52^a, 53^a, 54^a, 55^a, 56^a \}$ โดยอัตรา (ร้อยละ) ค่าความสม่ำเสมอ $r^a_{TDB_2}$ มีค่าเท่ากับ 50% จากนั้นสามารถคำนวณอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นของรายการ ‘a’ ได้ดังนี้ $c^a = \dfrac{r^a_{TDB_2}}{r^a_{TDB_1}} = \dfrac{50\ \%}{37.5\ \%} = 1.33$ เท่า (หมายเหตุ ในฐานข้อมูลรายการ TDB_2 รายการ ‘a’ มีค่าความสม่ำเสมอเพิ่มขึ้น 1.33 เท่า จากฐานข้อมูลรายการ TDB_1) ถ้าผู้ใช้กำหนดค่าขีดแบ่งการเปลี่ยนแปลงไว้ 1.25 เท่า ดังนั้น สามารถกล่าวได้ว่ารายการ ‘a’ เป็นเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น เนื่องจากมีอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นมากกว่าค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด*

3.2 วิธีการดำเนินงานวิจัย

ในส่วนของวิธีการดำเนินงานวิจัยนี้ จะอธิบายถึงโครงสร้างข้อมูลอิคโคร-ทรี (ICRO-tree structure) ที่ใช้สำหรับจัดเก็บข้อมูล และขั้นตอนวิธีไมโคร (MICRO algorithm) ที่ใช้สำหรับการค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น ดังนี้

3.2.1 โครงสร้างข้อมูลอิคโคร-ทรี

โครงสร้างข้อมูลอิคโคร-ทรี เป็นโครงสร้างข้อมูลต้นไม้ที่แต่ละโหนด (Node) ในเส้นทาง (Path) หนึ่ง ๆ ของอิคโคร-ทรี จะประกอบไปด้วยชื่อรายการ (i_k) ตัวเชื่อมโยงโหนดพ่อ (Parent node) กับโหนดลูก (Child node) ในเส้นทางเดียวกัน และตัวเชื่อมโยงไปยังโหนดอื่น ๆ ที่มีชื่อรายการเดียวกัน (Node-link) นอกจากนี้โครงสร้างข้อมูลอิคโคร-ทรียังมีการจัดเก็บเซตของหมายเลขทรานแซกชันที่ปรากฏขึ้นในฐานข้อมูลรายการ TDB_1 ($T^{i_k}_{TDB_1}$) และ/หรือ TDB_2 ($T^{i_k}_{TDB_2}$) ไว้ที่โหนดใบ (Leaf node) หรือโหนดสุดท้ายของเส้นทางเท่านั้น และยังมีตารางรายการ (Header table) ที่ไว้สำหรับจัดเก็บข้อมูลที่ปรากฏขึ้น ที่ซึ่งประกอบไปด้วย 5 ข้อมูล ดังต่อไปนี้

1. ชื่อรายการ
2. อัตรา (ร้อยละ) ค่าความสม่ำเสมอของรายการ i_k ในฐานข้อมูลรายการ TDB_1 ($r^{i_k}_{TDB_1}$)
3. อัตรา (ร้อยละ) ค่าความสม่ำเสมอของรายการ i_k ในฐานข้อมูลรายการ TDB_2 ($r^{i_k}_{TDB_2}$)
4. อัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นของรายการ i_k (c^{i_k})
5. ตัวเชื่อมโยงไปยังทุกโหนดของรายการ i_k (l^{i_k})

3.2.2 ขั้นตอนวิธีไมโคร

สำหรับการค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น ที่เรียกว่า ไมโคร สามารถแบ่งลักษณะการทำงานออกเป็น 2 ขั้นตอนวิธี ดังนี้

1. ขั้นตอนวิธีการสร้างอิคโคร-ทรี จะเป็นการอ่านข้อมูลจากฐานข้อมูลรายการ TDB_1 และ TDB_2 เพียงครั้งเดียวเท่านั้น และจัดเก็บข้อมูลที่ปรากฏขึ้นของเซตรายการในทรานแซกชันหนึ่ง ๆ ไปยังอิคโคร-ทรี และตารางรายการ ที่ซึ่งในกระบวนการนี้จะทำให้ได้อิคโคร-ทรี ที่เป็นโครงสร้างต้นไม้ประกอบด้วยเซตของรายการที่มีปรากฏขึ้นในฐานข้อมูลรายการ TDB_1 และ TDB_2 และได้เซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นขนาด 1 รายการ โดยมีรายละเอียดขั้นตอนวิธีการทำงาน ดังต่อไปนี้

Algorithm 1: *ICRO-tree* construction

Input: TDB_1, TDB_2, σ_c

Output: *ICRO-tree*: a tree-structure contains sets of items occurring in transactions of TDB_1 and TDB_2, *ICROs*: a set of interesting items with significant change on regularity of occurrence

1: initial *ICRO-tree* with a root node as R, and initial a header table of all items
2: **for** each transaction t_j in database TDB_1 **do**
3: set $tempNode$ as R
4: **for** each item i_k in transaction t_j **do**
5: update $r^{i_k}_{TDB_1}$ at the entry of item i_k in the header table by considering *tid* j of t_j
6: **if** there is no a child node of $tempNode$ with item i_k **then**
7: create a new node for item i_k, set it to be a child node of $tempNode$ and link it with the header table
8: $tempNode \leftarrow$ the new node of item i_k
9: **else**
10: $tempNode \leftarrow$ a child node of $tempNode$ with item i_k
11: add *tid* j of t_j into $T^{i_l}_{TDB_1}$ of $tempNode$ *i.e.* $T^{i_l}_{TDB_1} \leftarrow T^{i_l}_{TDB_1} \cup j$
12: read each transaction of TDB_2 in the same manner as scanning of TDB_1
13: **for** each item i_k in the header table **do**
14: **if** i_k does not occur in TDB_1 or TDB_2 **then**
15: remove all nodes of item i_k out of *ICRO-tree*
16: **else**
17: compute c^{i_k} by $\frac{r^{i_k}_{TDB_2}}{r^{i_k}_{TDB_1}}$
18: **if** $c^{i_k} \geq \sigma_c$ **then**
19: collect i_k in *ICROs* as an interesting item with significant change on regularity of occurrence

ภาพที่ 3.1 ขั้นตอนวิธีการสร้างอิคโคร-ทรี

ขั้นตอนวิธีการสร้างอิคโคร-ทรี แสดงดังภาพที่ 3.1 เริ่มต้นจากการกำหนดอิคโคร-ทรีกับโหนดราก (*Root node, R*) และสร้างตารางรายการสำหรับทุกรายการ จากนั้นอ่านข้อมูลในฐานข้อมูลรายการ TDB_1 และพิจารณาแต่ละทรานแซกชัน $t_j = \{ i_k, \ldots, i_l \}$ (บรรทัดที่ 2-11) ด้วยการกำหนดโหนดปัจจุบัน (*tempNode*) ชี้ไปยังโหนดราก *R* (หมายเหตุ โหนดปัจจุบัน คือ ตัวชี้ข้อมูลเพื่อใช้ตรวจสอบว่ามีเส้นทางในอิคโคร-ทรีหรือไม่) จากนั้นพิจารณาแต่ละรายการ i_k ในทรานแซกชัน t_j แล้วคำนวณอัตรา (ร้อยละ) ค่าความสม่ำเสมอของรายการ i_k ในฐานข้อมูลรายการ TDB_1 ($r^{i_k}_{TDB_1}$) เพื่ออัปเดต (Update) ลงในตารางรายการจากนั้นพิจารณาดังนี้

● ถ้าโหนดปัจจุบันยังไม่มีโหนดลูกเป็นรายการ i_k ให้ทำการสร้างโหนดขึ้นมาใหม่สำหรับรายการ i_k (n_{i_k}) แล้วกำหนดให้โหนดรายการ n_{i_k} ดังกล่าวเป็นโหนดลูกของโหนดปัจจุบัน จากนั้นทำการเชื่อมโยงโหนดรายการ n_{i_k} กับรายการที่มีชื่อเหมือนกันจากตารางรายการ และกำหนดให้โหนดปัจจุบันชี้ไปยังโหนดรายการ n_{i_k} (บรรทัดที่ 6-8)

● ถ้าโหนดปัจจุบันมีโหนดลูกเป็นรายการ i_k อยู่แล้ว จะกำหนดให้โหนดปัจจุบันชี้ไปยังโหนดลูกของรายการ i_k ที่มีอยู่ (บรรทัดที่ 9-10)

เมื่อพิจารณาครบทุกรายการในทรานแซกชัน t_j แล้วจะทำให้โหนดปัจจุบันจะถูกชี้ไปยังโหนดใบรายการ i_l ในทรานแซกชัน t_j จากนั้นอัปเดตหมายเลขของทรานแซกชันที่ j ไปยังเซตของหมายเลขทรานแซกชันที่ปรากฏขึ้นในฐานข้อมูลรายการ TDB_1 ของโหนดรายการ i_l ($T^{i_l}_{TDB_1} \leftarrow T^{i_l}_{TDB_1} \cup j$) เมื่อพิจารณาครบทุกทรานแซกชัน t_j ในฐานข้อมูลรายการ TDB_1 แล้ว ต่อมาทำการอ่านข้อมูลจากฐานข้อมูลรายการ TDB_2 ที่ซึ่งมีขั้นตอนวิธีเหมือนกันกับการอ่านข้อมูลจากฐานข้อมูลรายการ TDB_1 ทุกประการ (บรรทัดที่ 11-12) แต่ในขั้นตอนวิธีการคำนวณอัตรา (ร้อยละ) และการจัดเก็บค่าความสม่ำเสมอของรายการ i_k ในฐานข้อมูลรายการจะเปลี่ยนจาก $r^{i_k}_{TDB_1}$ ไปเป็น $r^{i_k}_{TDB_2}$ (บรรทัดที่ 5) และในการจัดเก็บเซตของหมายเลขทรานแซกชันที่ปรากฏขึ้นในฐานข้อมูลรายการของโหนดใบรายการ i_l จะเปลี่ยนจาก $T^{i_l}_{TDB_1}$ ไปจัดเก็บใน $T^{i_l}_{TDB_2}$ (บรรทัดที่ 11)

ขั้นตอนวิธีถัดไปจะทำการตรวจสอบแต่ละรายการ i_k ในตารางรายการ (บรรทัดที่ 13) โดยพิจารณาดังนี้

● ถ้ารายการ i_k ใดไม่ปรากฏขึ้นในฐานข้อมูลรายการ TDB_1 และ/หรือ TDB_2 จะทำการลบโหนดรายการ i_k นั้นออกจากอิคโคร-ทรี จากนั้นรายการ i_k ก็จะถูกลบออกจากตารางรายการด้วย (บรรทัดที่ 14-15)

● ถ้ารายการ i_k ใดปรากฏขึ้นในทั้งฐานข้อมูล TDB_1 และ TDB_2 จะทำการคำนวณอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น แล้วนำไปเปรียบเทียบกับค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด โดยถ้าอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นมีค่ามากกว่าหรือเท่ากับค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด จะทำการจัดเก็บรายการ i_k ไว้ในเซตของอิคโคร (ICROs) ที่ซึ่งเป็นเซตของรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น (บรรทัดที่ 16-19)

2. ขั้นตอนวิธีอิคโคร-โกรท เป็นกระบวนการเรียกซ้ำ (Recursive) การทำงานของขั้นตอนวิธีอิคโคร-ทรี โดยที่จะพิจารณาแต่ละรายการ i_k ในตารางรายการและเซตรายการที่ปรากฏขึ้นร่วมกับรายการ i_k ที่ซึ่งขั้นตอนวิธีอิคโคร-โกรทสามารถแบ่งได้เป็น 2 กรณี แสดงดังภาพที่ 3.2 ดังต่อไปนี้

Algorithm 2: *ICROs*-growth

Input: *ICRO-tree*, σ_c

Output: *ICROs*: a complete set of interesting itemsets with significant change on regularity of occurrence

Procedure: *ICROs*-growth (*ICRO-tree* with a root node R, X, σ_c) \\Note that X is a set of considered items from previous iterations (X is $\emptyset$ at the first mining)
if *ICRO-tree* contains only single path P **then**
  compute c^Q of itemsets Q of path P
  if $c^Q \geq \sigma_c$ **then**
    for each combination of items in path P (abbreviated as β) **do**
      merge β with X and then compute $c^{\beta \cup X}$ from T_{TDB_1} and T_{TDB_2} of the leaf node of path P
      collect $\beta \cup X$ in *ICROs* as an interesting itemsets with significant change on regularity of occurrence
else
  for each item i_k in the header table of *ICRO-tree* (start from the last to the second one) **do**
    $X \leftarrow X \cup i_k$ \\collect i_k to be a member of already considered itemset
    create and initial an item-list called $iList_{i_k}$ for maintaining all items (with their occurrence information) occurring together with item i_k
    for each node n_{i_k} linked in *node-link* of item i_k **do**
      for each item i_{an} located in the same vertical path as node n_{i_k} **do**
        \\Note that $iList_{i_k}^{i_{an}}$ be an entry of item i_{an} in $iList_{i_k}$, merge $T_{TDB_1}^{i_{an}}$ in the node n_{i_k} with $T_{TDB_1}^{i_k}$ in $iList_{i_k}^{i_{an}}$ and merge $T_{TDB_2}^{i_{an}}$ in the node n_{i_k} with $T_{DB_2}^{i_k}$ in $iList_{i_k}^{i_{an}}$
    for each item i_{an} in $iList_{i_k}$ **do**
      if i_{an} does not occur with X in TDB_1 or TDB_2 **then**
        remove the entry of i_{an} from $iList_{i_k}$
      else
        compute $c^{i_{an} \cup X}$ from $T_{TDB_1}^{i_{an}}$ and $T_{TDB_2}^{i_{an}}$ of $iList_{i_k}^{i_{an}}$
        if $c^{i_{an} \cup X} \geq \sigma_c$ **then**
          collect $i_{an} \cup X$ in *ICROs* as an interesting itemsets with significant change on regularity of occurrence
    if $|iList_{i_k}| > 1$ **then**
      create and initial *ICRO-tree* with root node as Z and a header table for items in $iList_{i_k}$
      for each node n_{i_k} linked in *node-link* of item i_k **do**
        set Y as a set of items in the same vertical path as n_{i_k} where there is an entry of the item in $iList_{i_k}$
        update *ICRO-tree* with root Z by Y and $T_{TDB_1}^{i_k}$ (also $T_{TDB_2}^{i_k}$) in the node n_{i_k}
        \\Note that $n_{i_{k-1}}$ is the parent node of the node n_{i_k} with item i_{k-1}, merge $T_{TDB_1}^{i_k}$ with $T_{TDB_1}^{i_{k-1}}$ and $T_{TDB_2}^{i_k}$ with $T_{TDB_2}^{i_{k-1}}$ and then remove n_{i_k} with all of its information
      call *ICROs*-growth (*ICRO-tree* with Z as root, X, σ_c)
    $X \leftarrow X - i_k$ \\remove i_k out of the set of already considered items, since all itemsets consisting of i_k are already considered and remove $iList_{i_k}$

ภาพที่ 3.2 การค้นหาเซตรายการที่มีการเปลี่ยนแปลงของความสม่ำเสมอที่ปรากฏขึ้นด้วยขั้นตอนวิธีอิคโคร-โกรท

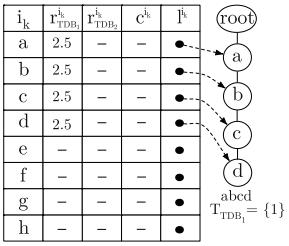
● ถ้าอิคโคร-ทรีมีเส้นทางแค่เพียงเส้นทางเดียว (Single path, *P*) (บรรทัดที่ 2-7) โดยกำหนดให้เซตรายการ *Q* เป็นเซตของรายการในเส้นทาง *P* จากนั้นคำนวณอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นของเซตรายการ *Q* จากเซตของหมายเลขทรานแซกชันในฐานข้อมูลรายการ TDB_1 และ TDB_2 ของโหนดใบในเส้นทาง *P* ถ้าอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นของเซตรายการ *Q* มีค่ามากกว่าหรือเท่ากับค่าขีดแบ่งการเปลี่ยนแปลง แล้วจะทำการหาสับเซตของเซตรายการ *Q* แล้วนำแต่ละรายการ/เซตรายการ (*β*) มารวมกับเซตรายการที่พิจารณาก่อนหน้าที่สร้างขึ้นจากกระบวนการทำซ้ำก่อนหน้า (*X*) *(β U X)* แล้วจะจัดเก็บเซตรายการ *β U X* ไว้ในเซตของอิคโคร ที่ซึ่งเป็นเซตของเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น

● ถ้าอิคโคร-ทรีมีเส้นทางมากกว่า 1 เส้นทาง (บรรทัดที่ 8-29) จะทำการพิจารณาแต่ละรายการ i_k ในตารางรายการ โดยเริ่มจากรายการที่อยู่ลำดับสุดท้ายจนถึงลำดับที่สอง ที่ซึ่งรายการ i_k ที่พิจารณาจะถูกจัดเก็บรวมกับเซตของเซตรายการ *X* ที่ถูกพิจารณาก่อนหน้า ($X \leftarrow X\ U\ i_k$) และสร้างลิสต์สำหรับจัดเก็บรายการและหมายเลขของทรานแซกชันที่ปรากฏขึ้นร่วมกับรายการ i_k ($iList_{i_k}$) (บรรทัดที่ 10-11) ต่อมาทำการท่องไปยังอิคโคร-ทรีตามตัวเชื่อมโยงของแต่ละโหนดรายการ n_{i_k} ที่ซึ่งแต่ละโหนดรายการ n_{i_k} จะมีแต่ละโหนดรายการที่อยู่ในระดับที่สูงกว่าโหนดรายการ n_{i_k} เรียกว่า โหนดรายการบรรพบุรุษ (Ancestor node, $n_{i_{an}}$) ที่ซึ่งอยู่ในเส้นทางเดียวกันกับโหนดรายการ n_{i_k} ในแนวตั้ง (Vertical) โดยจะนำแต่ละรายการบรรพบุรุษ i_{an} จัดเก็บเข้าไปยังลิสต์ $iList_{i_k}$ และทำการรวมเซตของหมายเลขทรานแซกชันที่ปรากฏขึ้น $T^{i_{an}}_{TDB_1}$ และ $T^{i_{an}}_{TDB_2}$ ของโหนดรายการ n_{i_k} ไปยังลิสต์ $iList^{i_{an}}_{i_k}$ (บรรทัดที่ 12-14) เมื่อพิจารณาครบทุกโหนดรายการ n_{i_k} ตามตัวเชื่อมโยงในอิคโคร-ทรีแล้ว จะทำการพิจารณาและตรวจสอบแต่ละรายการ i_{an} ในลิสต์ $iList^{i_{an}}_{i_k}$ ดังนี้ ถ้ารายการ i_{an} ปรากฏขึ้นร่วมกับเซตรายการ *X* ทั้งสองฐานข้อมูลรายการให้ทำการคำนวณอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นของเซตรายการ $i_{an}\ U\ X$ จากเซตของหมายเลขทรานแซกชัน $T^{i_{an}}_{TDB_1}$ และ $T^{i_{an}}_{TDB_2}$ ใน $iList^{i_{an}}_{i_k}$ ถ้าอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นของเซตรายการ $i_{an}\ U\ X$ มีค่ามากกว่าหรือเท่ากับค่าขีดแบ่งการเปลี่ยนแปลง แล้วจะจัดเก็บเซตรายการ $i_{an}\ U\ X$ ไว้ในเซตของอิคโคร แต่ถ้ารายการ i_{an} ปรากฏขึ้นร่วมกับเซตรายการ *X* แค่ฐานข้อมูลรายการ TDB_1 หรือ TDB_2 ก็จะลบรายการ i_{an} ออกจากการพิจารณาและออกจากลิสต์ $iList_{i_k}$ (บรรทัดที่ 15-21)

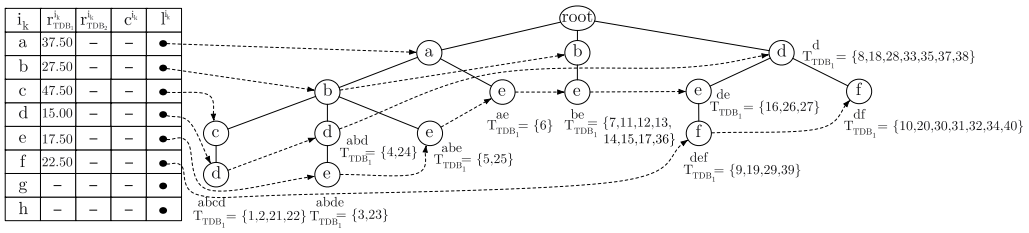
จากนั้นทำการตรวจสอบ ถ้าลิสต์ $iList_{i_k}$ มีรายการมากกว่า 1 รายการ จะทำการสร้างอิคโคร-ทรีต้นใหม่กับโหนดราก *Z* เรียกว่า คอนดิชันนอลอิคโคร-ทรี และสร้างตารางรายการสำหรับทุกรายการในลิสต์ $iList_{i_k}$ แล้วทำการท่องไปยังแต่ละโหนดรายการ n_{i_k} ตามตัวเชื่อมโยงในอิคโคร-ทรีอีกครั้งและกำหนดให้เซต *Y* เป็นเซตของรายการในเส้นทางเดียวกันในแนวตั้งของโหนดรายการ n_{i_k} จากนั้นนำเส้นทาง Y ไปอัปเดตในคอนดิชันนอลอิคโคร-ทรี และจัดเก็บเซตของหมายเลขทรานแซกชันที่ปรากฏขึ้นไว้ที่โหนดใบของเส้นทาง *Y* (บรรทัดที่ 22-26)

หลังจากสร้างคอนดิชันนอลอิคโคร-ทรีของเซตรายการ *Y* เรียบร้อยแล้ว จากนั้นจะลบโหนดรายการ n_{i_k} ที่ไม่จำเป็นทิ้ง โดยจะต้องทำการย้าย/รวมเซตของหมายเลขทรานแซกชันที่ปรากฏขึ้นของ $T^{i_k}_{TDB_1}$ และ $T^{i_k}_{TDB_2}$ ในโหนดรายการ n_{i_k} ให้กับโหนดพ่อของโหนดรายการ n_{i_k} ก่อน แล้วจึงสามารถลบโหนดรายการ n_{i_k} และเซตของหมายเลขทรานแซกชันออกจากอิคโคร-ทรี จากนั้นทำซ้ำขั้นตอนวิธีอิคโคร-โกรท (บรรทัดที่ 27-28) จนกระทั่งลิสต์ $iList_{i_k}$ มีรายการน้อยกว่าหรือเท่ากับ 1 รายการ จึงทำการลบรายการ i_k ออกจากเซตของรายการที่ถูกพิจารณาแล้ว และลบลิสต์ $iList_{i_k}$ ด้วย *(บรรทัดที่ 29)*

ตัวอย่างที่ 3.2 *กำหนดให้ค่าขีดแบ่งการเปลี่ยนแปลงมีค่าเท่ากับ 1.25 เท่า โดยจะพิจารณาแนวโน้มความเปลี่ยนแปลงของพฤติกรรมการซื้อสินค้าอย่างสม่ำเสมอที่เพิ่มขึ้นของฐานข้อมูลรายการ TDB_1 และ TDB_2 แสดงดังภาพที่ 2.2 โดยเริ่มต้นจากการสร้างอิคโคร-ทรีด้วยการกำหนดตารางรายการสำหรับทุกรายการและโหนดราก จากนั้นอ่านทรานแซกชัน t_1 = { a, b, c, d } จากฐานข้อมูลรายการแล้วนำรายการที่ปรากฏขึ้นในทรานแซกชัน t_1 ไปสร้างเส้นทางในอิคโคร-ทรี โดยจะมีโหนดรายการ 'd' เป็นโหนดใบ ที่ซึ่งจะถูกจัดเก็บเซตของหมายเลขทรานแซกชันเป็น $T^{abcd}_{TDB_1}$ = { 1 } และรายการ 'a', 'b', 'c' และ 'd' ที่ในตารางรายการจะถูกอัปเดตอัตรา (ร้อยละ) ค่าความสม่ำเสมอในฐานข้อมูลรายการ TDB_1 ($r^{i_k}_{TDB_1}$) แสดงดังภาพที่ 3.3 จากนั้นทำซ้ำขั้นตอนวิธีดังกล่าวของทรานแซกชันที่เหลือทั้งหมดในฐานข้อมูลรายการ TDB_1 แสดงดังภาพที่ 3.4*

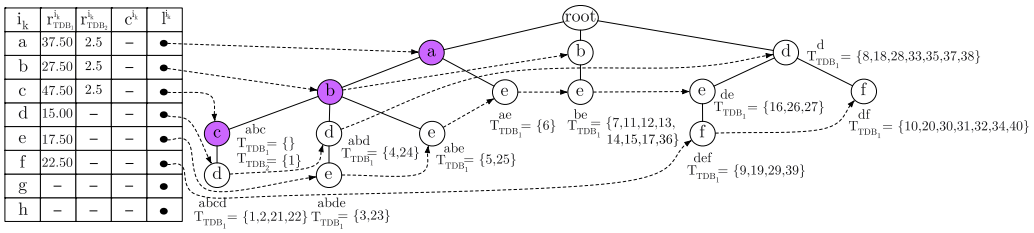


ภาพที่ 3.3 อิคโคร-ทรีหลังจากที่อ่านทรานแซกชัน t_1 ของฐานข้อมูลรายการ TDB_1

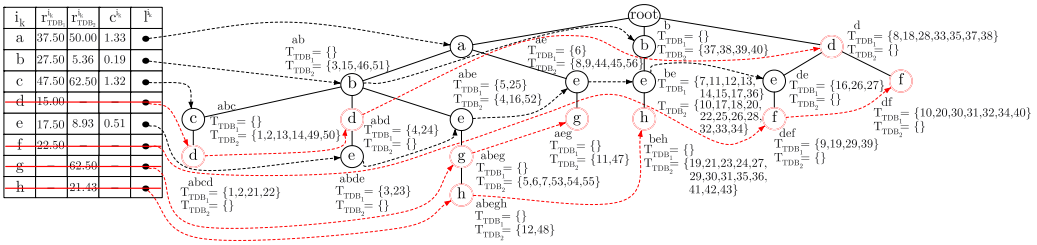


ภาพที่ 3.4 อิคโคร-ทรีหลังจากที่อ่านครบทุกทรานแซกชันของฐานข้อมูลรายการ TDB_1

ต่อมาทำการอ่านทรานแซกชันแรก t_1 = { a, b, c } ของฐานข้อมูลรายการ TDB_2 โดยแต่ละรายการที่ปรากฏในทรานแซกชัน t_1 จะถูกอัปเดตอัตรา (ร้อยละ) ค่าความสม่ำเสมอของฐานข้อมูลรายการ TDB_2 ($r^{i_k}_{TDB_2}$) ในตารางรายการ และเส้นทางของทรานแซกชัน t_1 ในฐานข้อมูลรายการ TDB_2 จะถูกสร้างและ/หรืออัปเดตในอิคโคร-ทรี โดยที่มีโหนดรายการ 'c' เป็นโหนดใบที่ซึ่งจะถูกจัดเก็บหมายเลขของทรานแซกชันเป็น $T^{abc}_{TDB_2}$ = { 1 } แสดงดังภาพที่ 3.5 จากนั้นทำซ้ำขั้นตอนวิธีดังกล่าวกับทรานแซกชันทั้งหมดในฐานข้อมูลรายการ TDB_2 แสดงดังภาพที่ 3.6

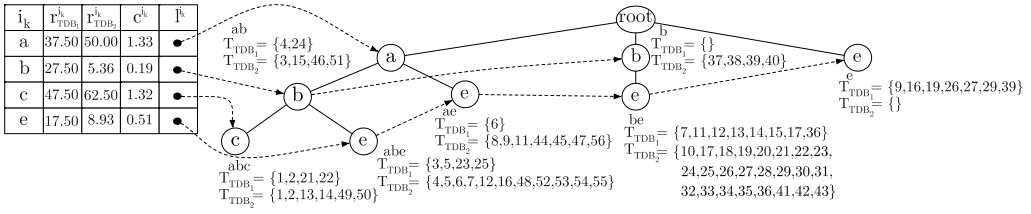


ภาพที่ 3.5 อิคโคร-ทรีหลังจากที่อ่านทรานแซกชัน t_1 ของฐานข้อมูลรายการ TDB_2



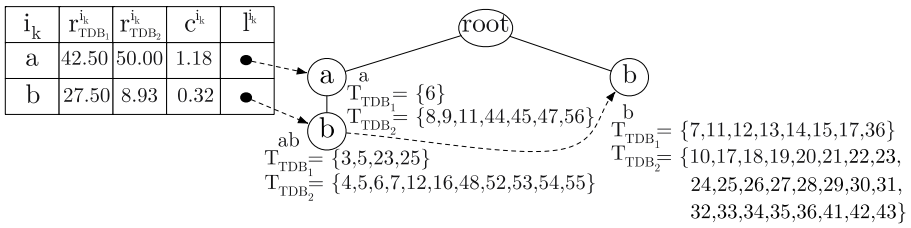
ภาพที่ 3.6 อิคโคร-ทรีหลังจากที่สร้างขึ้นจากฐานข้อมูลรายการ TDB_1 และ TDB_2

จากอิคโคร-ทรีที่สร้างขึ้นจากฐานข้อมูลรายการ TDB_1 และ TDB_2 สังเกตจากตารางรายการได้ว่า รายการ 'd', 'f', 'g' และ 'h' ไม่ปรากฏขึ้นในฐานข้อมูลรายการ TDB_1 หรือ TDB_2 แสดงดังภาพที่ 3.6 ดังนั้น จะลบรายการดังกล่าวออกจากตารางรายการและอิคโคร-ทรี (จากสมบัติปิดการลดลง) แสดงดังภาพที่ 3.7 และสำหรับรายการที่ผ่านการพิจารณาจะถูกนำไปคำนวณอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น ที่ซึ่งรายการ 'a' และ 'c' เป็นเซตรายการที่ปรากฏสม่ำเสมอภายใต้ค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด เนื่องจากรายการ 'a' และ 'c' มีอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นเป็น 1.33 เท่า และ 1.32 เท่าตามลำดับ ที่ซึ่งมีค่ามากกว่าค่าขีดแบ่งการเปลี่ยนแปลง ส่วนรายการ 'b' และ 'e' มีอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น ซึ่งมีค่าน้อยกว่าค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด จึงไม่นำมาจัดเก็บเป็นผลลัพธ์



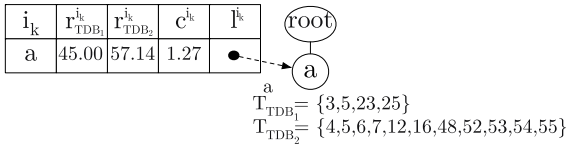
ภาพที่ 3.7 อิคโคร-ทรีหลังจากที่ลบรายการ *'d', 'f', 'g'* และ *'h'*

ในขั้นตอนวิธีการอิคโคร-โกรท จะเริ่มจากการพิจารณารายการที่อยู่ลำดับสุดท้ายในตารางรายการ ที่ซึ่งได้แก่รายการ 'e' โดยจะทำการท่องไปยังแต่ละโหนดของรายการ 'e' ในอิคโคร-ทรีตามตัวเชื่อมโยงในตารางรายการ ที่ซึ่งเส้นทางแรก คือ 'eba' โดยที่รายการ 'a' และ 'b' จะถูกพิจารณาที่ปรากฏขึ้นร่วมกับรายการ 'e' ในทรานแซกชันที่ 3, 5, 23 และ 25 ของฐานข้อมูลรายการ TDB_1 และทรานแซกชันที่ 4, 5, 6, 7, 12, 16, 48, 52, 53, 54 และ 55 ของฐานข้อมูลรายการ TDB_2 เส้นทางถัดไป ได้แก่ 'ea' โดยรายการ 'a' จะถูกพิจารณาที่ปรากฏขึ้นร่วมกับรายการ 'e' ในทรานแซกชันที่ 6 ของฐานข้อมูลรายการ TDB_1 และทรานแซกชันที่ 8, 9, 11, 44, 45, 47 และ 56 ของฐานข้อมูลรายการ TDB_2 เมื่อทำการรวมทรานแซกชันของรายการ 'a' ที่ถูกพิจารณาร่วมกับรายการ 'e' จะได้ $T^{ea}_{TDB_1}$ *= { 3, 5, 6, 23, 25 } และ* $T^{ea}_{TDB_2}$ *= { 4, 5, 6, 7, 8, 9, 11, 12, 16, 44, 45, 47, 48, 52, 53, 54, 55, 56 } เส้นทางถัดไป ได้แก่ 'eb' โดยรายการ 'b' จะถูกพิจารณาที่ปรากฏขึ้นร่วมกับรายการ 'e' ในทรานแซกชันที่ 7, 11, 12, 13, 14, 15, 17 และ 36 ของฐานข้อมูลรายการ TDB_1 เมื่อทำการรวมทรานแซกชันของรายการ 'b' ที่ถูกพิจารณาร่วมกับรายการ 'e' จะได้* $T^{eb}_{TDB_1}$ *= { 3, 5, 7, 11, 12, 13, 14, 15, 17, 23, 25, 36 } และ* $T^{eb}_{TDB_2}$ *= { 4, 5, 6, 7, 10, 12, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 41, 42, 43, 48, 52, 53, 54, 55 } เมื่อท่องทุกเส้นทางทั้งหมดของรายการ 'e' จะเห็นได้ว่ารายการ 'a' และ 'b' ปรากฏขึ้นร่วมกับรายการ 'e' ทั้งสองฐานข้อมูล จึงทำการคำนวณหาอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น จะได้* c^{ea} *= 1.18 เท่าและ* c^{eb} *= 0.32 เท่า ที่ซึ่งเซตรายการ 'ea' และเซตรายการ 'eb' มีอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นน้อยกว่าค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด ดังนั้นจึงไม่จัดเก็บเซตรายการ 'ea' และเซตรายการ 'eb' เป็นผลลัพธ์ จากนั้นทำการสร้างคอนดิชันนอลอิคโคร-ทรีของรายการ 'e' แสดงดังภาพที่ 3.8*

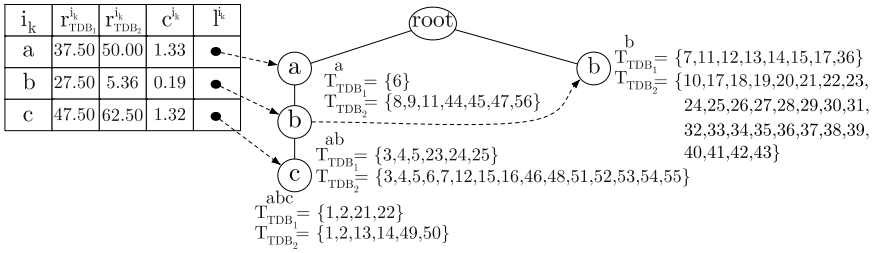


ภาพที่ 3.8 คอนดิชันนอลอิคโคร-ทรีของรายการ *'e'*

จากภาพที่ 3.8 สังเกตได้ว่ารายการ ‘b’ เป็นรายการสุดท้ายในตารางรายการและมีคอนดิชันนอลอิคโคร-ทรีของรายการ ‘e’ มากกว่า 1 เส้นทาง ดังนั้นจึงพิจารณาเซตรายการ ‘eb’ แล้วทำการหาเซตรายการของเส้นทางในคอนดิชันนอลอิคโคร-ทรีของรายการ ‘e’ ตามตัวเชื่อมโยงของโหนดรายการ ‘b’ ในตารางรายการ ได้เป็น รายการ ‘a’ (หมายเหตุ รายการ ‘a’ จะถูกพิจารณาที่ปรากฏขึ้นร่วมกับเซตรายการ ‘eb’ ได้เป็น $T^{eba}_{TDB_1}$ *= { 3, 5, 23, 25 } และ* $T^{eba}_{TDB_2}$ *= { 4, 5, 6, 7, 12, 16, 48, 52, 53, 54, 55 } จะเห็นว่ารายการ a ปรากฏขึ้นร่วมกับเซตรายการ ‘eb’ ทั้งสองฐานข้อมูล ดังนั้น จึงคำนวณอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น ที่ซึ่งมีค่าเท่ากับ 1.27 เท่า สังเกตได้ว่ารายการ ‘a’ ที่ปรากฏขึ้นร่วมกับเซตรายการ ‘eb’ มีอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นมากกว่าค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด จึงนำเซตรายการ ‘eba’ มาจัดเก็บเป็นผลลัพธ์ เนื่องจากเซตรายการ ‘eb’ มีเส้นทางเดียวจึงไม่สามารถสร้างคอนดิชันนอลอิคโคร-ทรีได้แล้ว จึงกลับมาพิจารณาที่คอนดิชันนอลอิคโคร-ทรีของรายการ ‘e’ แล้วทำการรวมเซตของหมายเลขทรานแซกชันของรายการ ‘b’ กับโหนดพ่อของรายการ ‘b’ แล้วจึงทำการลบรายการ ‘b’ ออกจากตารางรายการและคอนดิชันนอลอิคโคร-ทรีของรายการ ‘e’ แสดงดังภาพที่ 3.9 ที่ซึ่งสังเกตได้ว่า คอนดิชันนอลอิคโคร-ทรีรายการ ‘e’ มีเส้นทางเดียว จึงไม่สามารถสร้างคอนดิชันนอลอิคโคร-ทรีได้แล้ว ดังนั้นจะพิจารณาที่อิคโคร-ทรี โดยทำการรวมเซตของหมายเลขทรานแซกชันของรายการ ‘e’ กับโหนดพ่อของรายการ ‘e’ แล้วจึงทำการลบรายการ ‘e’ ออกจากตารางรายการและอิคโคร-ทรี แสดงดังภาพที่ 3.10*



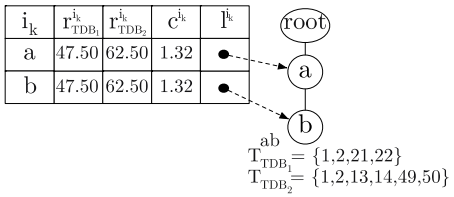
ภาพที่ 3.9 คอนดิชันนอลอิคโคร-ทรีของรายการ ‘*e*’ หลังจากลบรายการ ‘*b*’



ภาพที่ 3.10 อิคโคร-ทรีหลังจากที่มีการลบรายการ ‘*e*’

จากนั้นทำการพิจารณารายการ ‘c’ ที่อยู่ลำดับสุดท้ายในตารางรายการ โดยจะถูกท่องไปยังแต่ละโหนดของรายการ ‘c’ ในอิคโคร-ทรีตามตัวเชื่อมโยงในตารางรายการ ซึ่งเส้นทางแรก คือ

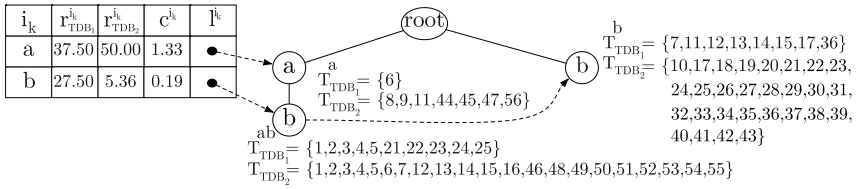
‘cba’ โดยที่รายการ ‘a’ และ ‘b’ จะถูกพิจารณาที่ปรากฏขึ้นร่วมกับรายการ ‘c’ ในทรานแซกชันที่ 1, 2, 21 และ 22 ของฐานข้อมูลรายการ TDB_1 และทรานแซกชันที่ 1, 2, 13, 14, 49 และ 50 ของฐานข้อมูลรายการ TDB_2 แล้วทำการสร้างคอนดิชันนอลอิคโคร-ทรีของรายการ ‘c’ แสดงดังภาพที่ 3.11



i_k	$r^{i_k}_{TDB_1}$	$r^{i_k}_{TDB_2}$	c^{i_k}	l^{i_k}
a	47.50	62.50	1.32	●
b	47.50	62.50	1.32	●

ภาพที่ 3.11 คอนดิชันนอลอิคโคร-ทรีของรายการ ‘c’

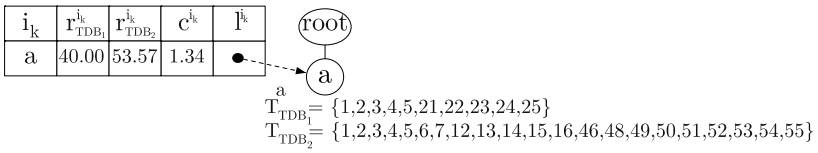
จากภาพที่ 3.11 จะเห็นได้ว่า คอนดิชันนอลอิคโคร-ทรีของรายการ ‘c’ มีเส้นทางเดียว จึงทำการคำนวณอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นของรายการ ‘c’ ได้เป็น 1.32 เท่า ที่ซึ่งมีค่ามากกว่าค่าขีดแบ่งการเปลี่ยนแปลง ดังนั้น จึงทำการหาสับเซตของเซตรายการในเส้นทางคอนดิชันนอลอิคโคร-ทรีกับรายการ ‘c’ ได้เป็น { ‘b’, ‘a’, ‘ba’} แล้วนำสมาชิกในสับเซตดังกล่าวมารวมกับรายการ ‘c’ แล้วจัดเก็บเป็นเซตของรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น ได้ดังนี้ { ‘cb’, ‘ca’, ‘cba’ } จากนั้นพิจารณาที่อิคโคร-ทรี โดยทำการรวมเซตของหมายเลขทรานแซกชันของรายการ ‘c’ กับโหนดพ่อของรายการ ‘c’ แล้วจึงทำการลบรายการ ‘c’ ออกจากตารางรายการและอิคโคร-ทรี แสดงดังภาพที่ 3.12



i_k	$r^{i_k}_{TDB_1}$	$r^{i_k}_{TDB_2}$	c^{i_k}	l^{i_k}
a	37.50	50.00	1.33	●
b	27.50	5.36	0.19	●

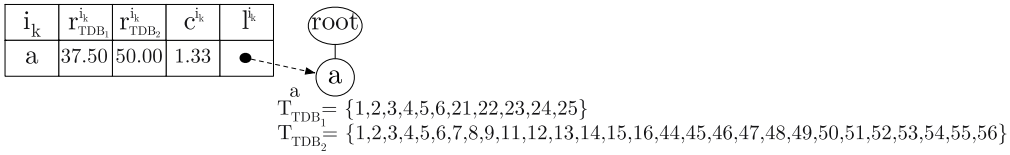
ภาพที่ 3.12 อิคโคร-ทรีหลังจากที่มีการลบรายการ ‘c’

จากนั้นทำการพิจารณารายการ ‘b’ ที่อยู่ลำดับสุดท้ายในตารางรายการ ที่ซึ่งแต่ละโหนดของรายการ ‘b’ ในอิคโคร-ทรีจะถูกท่องไปตามตัวเชื่อมโยงในตารางรายการ ซึ่งเส้นทางแรก คือ ‘ba’ โดยที่รายการ ‘a’ จะถูกพิจารณาที่ปรากฏขึ้นร่วมกับรายการ ‘b’ ในทรานแซกชันที่ 1, 2, 3, 4, 5, 21, 22, 23, 24 และ 25 ของฐานข้อมูลรายการ TDB_1 และทรานแซกชันที่ 1, 2, 3, 4, 5, 6, 7, 12, 13, 14, 15, 16, 46, 48, 49, 50, 51, 52, 53, 54 และ 55 ของฐานข้อมูลรายการ TDB_2 แล้วทำการสร้างคอนดิชันนอลอิคโคร-ทรีของรายการ ‘b’ แสดงดังภาพที่ 3.13



ภาพที่ 3.13 คอนดิชันนอลอิคโคร-ทรีของรายการ *‘b’*

จากภาพที่ 3.13 จะเห็นได้ว่า คอนดิชันนอลอิคโคร-ทรีของรายการ ‘b’ มีเส้นทางเดียว จึงทำการคำนวณอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นของรายการ ‘b’ ได้เป็น 1.34 เท่า ซึ่งมีค่ามากกว่าค่าขีดแบ่งการเปลี่ยนแปลง จึงนำเซตรายการ ‘ba’ มาจัดเก็บเป็นผลลัพธ์ จากนั้นจะทำการรวมเซตของหมายเลขทรานแซกชันของรายการ b กับโหนดพ่อของรายการ ‘b’ แล้วจึงทำการลบรายการ ‘b’ ออกจากตารางรายการและอิคโคร-ทรี แสดงดังภาพที่ 3.14



ภาพที่ 3.14 อิคโคร-ทรีหลังจากที่มีการลบรายการ *‘b’*

จากภาพที่ 3.14 สังเกตได้ว่าอิคโคร-ทรีมีเส้นทางเดียว และพิจารณาทุกรายการในตารางรายการแล้วจึงหยุดการทำงาน ที่ซึ่งจะได้เซตของรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น ดังนี้ { ‘a’, ‘c’, ‘eba’, ‘ca’, ‘cb’, ‘cba’, ‘ba’ } แสดงดังภาพที่ 3.15

i_k	$r^{i_k}_{TDB_1}$	$r^{i_k}_{TDB_2}$	c^{i_k}	$T^{i_k}_{TDB_1}$	$T^{i_k}_{TDB_2}$
a	37.50	50.00	1.33	{ 1, 2, 3, 4, 5, 6, 21, 22, 23, 24, 25 }	{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56 }
c	47.50	62.50	1.32	{ 1, 2, 21, 22 }	{ 1, 2, 13, 14, 49, 50 }
eba	45.00	57.14	1.27	{ 3, 5, 23, 25 }	{ 4, 5, 6, 7, 12, 16, 48, 52, 53, 54, 55 }
ca	47.50	62.50	1.32	{ 1, 2, 21, 22 }	{ 1, 2, 13, 14, 49, 50 }
cb	47.50	62.50	1.32	{ 1, 2, 21, 22 }	{ 1, 2, 13, 14, 49, 50 }
cba	47.50	62.50	1.32	{ 1, 2, 21, 22 }	{ 1, 2, 13, 14, 49, 50 }
ba	40.00	53.57	1.34	{ 1, 2, 3, 4, 5, 21, 22, 23, 24, 25 }	{ 1, 2, 3, 4, 5, 6, 7, 12, 13, 14, 15, 16, 46, 48, 49, 50, 51, 52, 53, 54, 55 }

ภาพที่ 3.15 ผลลัพธ์เซตของรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น

3.3 การวิเคราะห์ประสิทธิภาพของขั้นตอนวิธีไมโคร

ในงานวิจัยนี้ได้วิเคราะห์ความซับซ้อน (Complexity analysis) ของการค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นด้วยขั้นตอนวิธีไมโคร ใน 2 รูปแบบ คือ การวิเคราะห์ความซับซ้อนของเวลาที่ใช้ในการประมวลผลข้อมูล และการวิเคราะห์พื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลสูงสุด

ข้อเสนอที่ 3.3.1 การวิเคราะห์ความซับซ้อนของเวลาที่ใช้ในการประมวลผลข้อมูลในการค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นด้วยขั้นตอนวิธีไมโคร คือ $O((n(m+v)) + (n) + (2^p(m \log m+v \log v)))$ โดยที่ 1) n คือ จำนวนของรายการทั้งหมดในเซต I 2) m คือ จำนวนทรานแซกชันทั้งหมดในฐานข้อมูลรายการ TDB_1 3) v คือ จำนวนทรานแซกชันทั้งหมดในฐานข้อมูลรายการ TDB_2 และ 4) p คือ จำนวนรายการหลังจากที่ทำได้ทำการลดทอนข้อมูล

พิสูจน์ข้อเสนอที่ 3.3.1 การค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นด้วยขั้นตอนวิธีไมโคร เริ่มต้นจากการอ่านฐานข้อมูลรายการของทุกรายการ n ในทุกทรานแซกชัน m ของ TDB_1 จะใช้เวลาเป็น $n \times m$ และอ่านฐานข้อมูลรายการของทุกรายการ n ในทุกทรานแซกชัน v ของ TDB_2 จะใช้เวลาเป็น $n \times v$ ดังนั้นจะใช้เวลาสำหรับการอ่านฐานข้อมูลรายการเป็น $nm + nv$ เท่ากับ $n(m+v)$ หลังจากนั้นอ่านทุกรายการ n สำหรับการหาเซตรายการขนาด 1 รายการและลดทอนข้อมูลรายการที่ปรากฏขึ้นเพียงฐานข้อมูลรายการ TDB_1 หรือ TDB_2 จะได้รายการหลังจากการลดทอนข้อมูลเป็น p และในการค้นหาเซตรายการทั้งหมด ในแต่ละรายการจะทำการรวมกับรายการอื่น ๆ โดยที่จำนวนรวมของเซตรายการที่เป็นไปได้ทั้งหมด คือ 2^p และทำการรวมเซตของหมายเลขทรานแซกชันและเรียงลำดับข้อมูลของเซตรายการนั้นๆ ทั้งสองฐานข้อมูลรายการ ใช้เวลาเป็น $m \log m+v \log v$ และจะได้ความซับซ้อนของเวลาที่ใช้ในการประมวลผลข้อมูลในการค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นด้วยขั้นตอนวิธีไมโครทั้งหมด คือ $O((n(m+v)) + (n) + (2^p(m \log m+v \log v)))$

ข้อเสนอที่ 3.3.2 การวิเคราะห์ความซับซ้อนของพื้นที่หน่วยความจำในการจัดเก็บข้อมูลของการค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นด้วยขั้นตอนวิธีไมโคร คือ $O((n(m+v)) + (2^{(p-1)}(m+v)))$ โดยที่ 1) n คือ จำนวนของรายการทั้งหมดในเซต I 2) m คือ จำนวนทรานแซกชันทั้งหมดในฐานข้อมูลรายการ TDB_1 3) v คือ จำนวนทรานแซกชันทั้งหมดในฐานข้อมูลรายการ TDB_2 และ 4) p คือ จำนวนรายการหลังจากที่ทำได้ทำการลดทอนข้อมูล

พิสูจน์ข้อเสนอที่ 3.3.2 พื้นที่หน่วยความจำในการจัดเก็บข้อมูลของการค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นด้วยขั้นตอนวิธีไมโคร จะเริ่มพิจารณาจากการจัดเก็บข้อมูลเซตของหมายเลขของทรานแซกชันที่ปรากฏขึ้นของทุกรายการทั้งสองฐานข้อมูลรายการ จะใช้พื้นที่หน่วยความจำของทุกรายการเท่ากับ $nm + nv$ เท่ากับ $n(m+v)$

จากนั้นพิจารณารายการหลังจากการลดทอนข้อมูลจะได้เป็น p จากนั้นในส่วนของการค้นหาเซตรายการตั้งแต่ขนาด 2 รายการขึ้นไป โดยจะพิจารณาแต่ละเซตรายการที่มีรายการก่อนหน้าที่เหมือนกัน (Prefix) (หมายเหตุ ยกเว้นรายการสุดท้าย) แล้วนำไปรวมกับเซตรายการอื่น ๆ และแต่ละเซตรายการจะมีข้อมูลทรานแซกชันที่ปรากฏขึ้นของเซตรายการนั้นๆ ทั้งสองฐานข้อมูลรายการ ดังนั้นจะใช้พื้นที่หน่วยความจำสูงสุดในขั้นตอนวิธีนี้ คือ $2^{(p-1)}(m+v)$ ดังนั้นความซับซ้อนของพื้นที่หน่วยความจำในการจัดเก็บข้อมูลของการค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นด้วยขั้นตอนวิธีไมโคร คือ $O((n(m+v)) + (2^{(p-1)}(m+v)))$

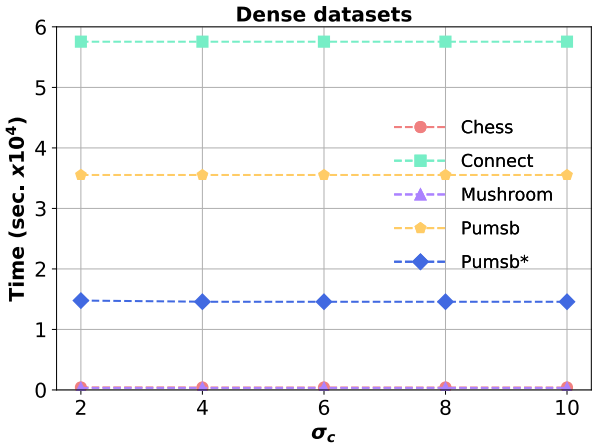
3.4 ผลการทดลอง

จากคุณลักษณะของฐานข้อมูลรายการที่ใช้ในการทดลอง ดังที่กล่าวในบทที่ 2 สามารถแบ่งฐานข้อมูลรายการได้ 2 ลักษณะข้อมูล ได้แก่ 1) แบบหนาแน่น (Dense) มีทั้งหมด 6 ฐานข้อมูลรายการ ได้แก่ Accidents, Chess, Connect, Mushroom, Pumsb และ Pumsb* 2) แบบเบาบาง (Sparse) มีทั้งหมด 4 ฐานข้อมูลรายการ ได้แก่ Kosarak, Retail, T10I4D100K และ T40I10D100K โดยในงานวิจัยนี้ได้แบ่งครึ่งฐานข้อมูลรายการออกเป็นสองส่วนเท่า ๆ กัน (ฐานข้อมูลรายการ TDB_1 และฐานข้อมูลรายการ TDB_2) และได้ใช้ Python 3.5.1 ด้วยโปรแกรม Pycharm บนเครื่องคอมพิวเตอร์ที่มีความเร็ว CPU 2.40 GHz, RAM 8 GB และ Windows 10

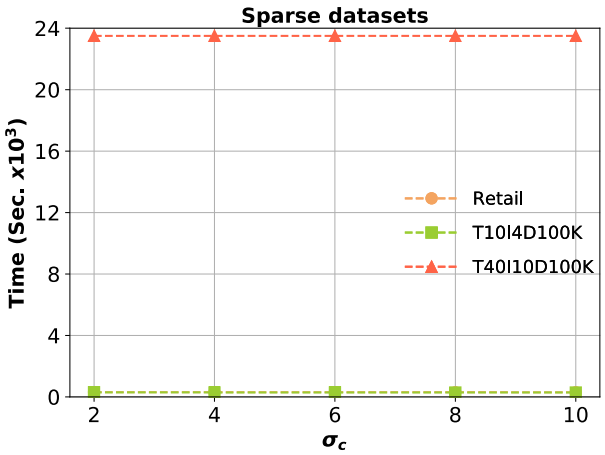
สำหรับการค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นด้วยขั้นตอนวิธีไมโคร โดยการพิจารณาความเปลี่ยนแปลงของพฤติกรรมการปรากฏขึ้นในแง่มุมความสม่ำเสมอที่เพิ่มขึ้นเมื่อเวลาผ่านไปภายใต้ค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด ที่ซึ่งกำหนดค่าขีดแบ่งการเปลี่ยนแปลงเริ่มตั้งแต่ 2 ถึง 10 และผลลัพธ์จะพิจารณาใน 3 แง่มุม ดังต่อไปนี้

3.4.1 เวลาที่ใช้ในการประมวลผลข้อมูล

ภาพที่ 3.16 และ ภาพที่ 3.17 แสดงผลการทดลองการค้นหาเซตรายการด้วยขั้นตอนวิธีไมโครในด้านเวลาของฐานข้อมูลรายการที่มีลักษณะข้อมูลหนาแน่นและเบาบาง ที่ซึ่งสังเกตได้ว่าเมื่อค่าขีดแบ่งการเปลี่ยนแปลงมีค่ามากขึ้นจะไม่มีผลกับเวลาที่ใช้ในการประมวลผลข้อมูล (หมายเหตุ เวลาที่ใช้ในการประมวลผลข้อมูลลดลงเล็กน้อยเมื่อค่าขีดแบ่งการเปลี่ยนแปลงมีค่ามากขึ้น) นอกจากนี้สามารถสังเกตได้ว่าฐานข้อมูลรายการใดที่มีข้อมูลขนาดใหญ่หรือมีจำนวนทรานแซกชันมากจะทำให้ใช้เวลาในการประมวลผลมากกว่าฐานข้อมูลรายการที่มีขนาดเล็ก และเนื่องจากได้ฐานข้อมูลรายการ Accidents และฐานข้อมูลรายการ Kosarak เป็นฐานข้อมูลรายการที่มีขนาดใหญ่มาก จึงมีผลทำให้ใช้เวลาในการประมวลผลข้อมูลนานมาก ด้วยเหตุนี้จึงทำให้ทั้งสองฐานข้อมูลรายการไม่สามารถแสดงเวลาที่ใช้ในการประมวลข้อมูล



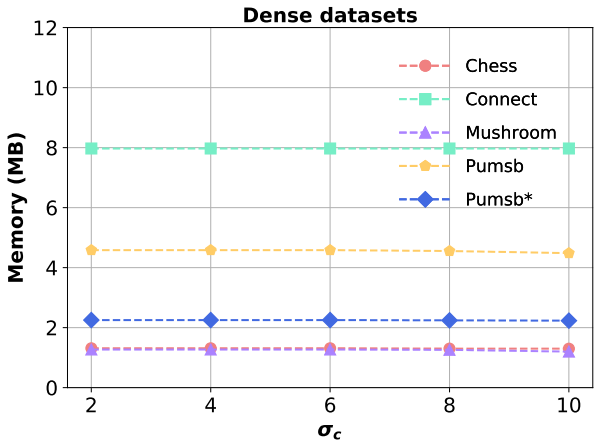
ภาพที่ 3.16 ผลการทดลองการค้นหาเซตรายการด้วยขั้นตอนวิธีไมโครในด้านเวลาที่ใช้ในการประมวลผลข้อมูลของฐานข้อมูลรายการที่มีลักษณะข้อมูลหนาแน่น



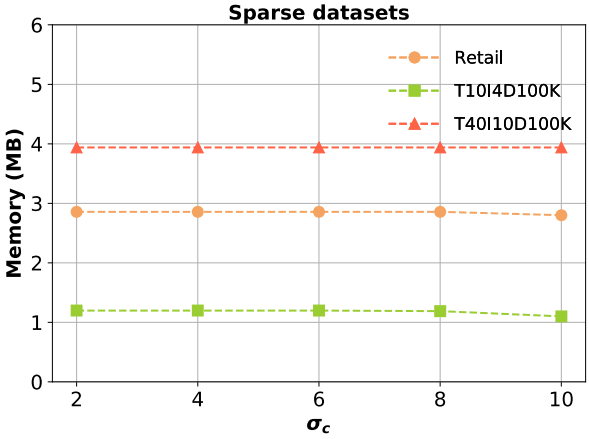
ภาพที่ 3.17 ผลการทดลองการค้นหาเซตรายการด้วยขั้นตอนวิธีไมโครในด้านเวลาที่ใช้ในการประมวลผลข้อมูลของฐานข้อมูลรายการที่มีลักษณะข้อมูลเบาบาง

3.4.2 พื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูล

ภาพที่ 3.18 และ ภาพที่ 3.19 แสดงผลการทดลองการค้นหาเซตรายการวิธีไมโครในด้านพื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลของทุกฐานข้อมูลรายการที่มีลักษณะข้อมูลหนาแน่นและเบาบาง ที่ซึ่งสังเกตได้ว่างานวิจัยนี้ได้ใช้พื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลที่น้อย เนื่องจากการค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นด้วยขั้นตอนวิธีไมโครนั้นได้มีการลดทอนของเซตรายการที่ปรากฏขึ้นแค่เพียงฐานข้อมูล TDB_1 หรือ TDB_2 ของขั้นตอนวิธีการสร้างอิคโคร-ทรี และ/หรือ คอนดิชันนอลของอิคโคร-ทรี แต่อย่างไรก็ตาม ถ้าฐานข้อมูลรายการมีขนาดใหญ่มากจะทำให้ใช้พื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลเยอะ ด้วยเหตุนี้จึงทำให้ฐานข้อมูลรายการ Accidents และฐานข้อมูลรายการ Kosarak ไม่สามารถแสดงพื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลได้ เนื่องจากเครื่องคอมพิวเตอร์ที่ใช้ทำการทดลองมีพื้นที่หน่วยความจำที่ใช้จัดเก็บข้อมูลไม่เพียงพอต่อความต้องของขั้นตอนวิธีนี้



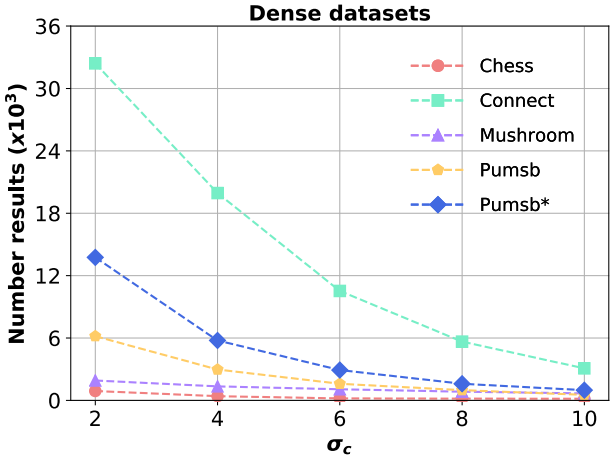
ภาพที่ 3.18 ผลการทดลองการค้นหาเซตรายการด้วยขั้นตอนวิธีไมโครในด้านพื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลของฐานข้อมูลรายการที่มีลักษณะข้อมูลหนาแน่น



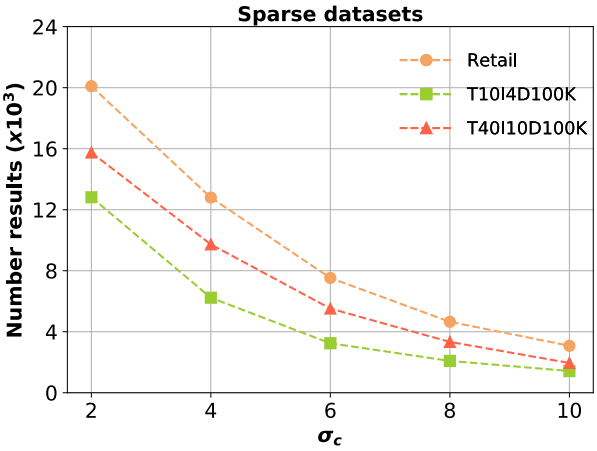
ภาพที่ 3.19 ผลการทดลองการค้นหาเซตรายการด้วยขั้นตอนวิธีไมโครในด้านพื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลของฐานข้อมูลรายการที่มีลักษณะข้อมูลเบาบาง

3.4.3 จำนวนผลลัพธ์เซตรายการที่ค้นพบ

ภาพที่ 3.20 และ ภาพที่ 3.21 แสดงผลการทดลองการค้นหาเซตรายการด้วยขั้นตอนวิธีไมโครในด้านจำนวนผลลัพธ์เซตรายการที่ค้นพบของทุกฐานข้อมูลรายการที่มีลักษณะข้อมูลหนาแน่นและเบาบาง ที่ซึ่งสังเกตได้ว่าเมื่อค่าอัตราการเปลี่ยนแปลงของความสม่ำเสมอน้อยโอกาสในการค้บพบเซตรายการก็จะมากขึ้น ในทางกลับกันเมื่อค่าอัตราการเปลี่ยนแปลงของความสม่ำเสมอมากโอกาสในการค้บพบเซตรายการก็จะน้อยลง นอกจากนี้ขนาดของฐานข้อมูลรายการก็มีผลต่อจำนวนผลลัพธ์เซตรายการที่ค้นพบ ด้วยเหตุนี้ จึงมีผลทำให้ฐานข้อมูลรายการ Accidents และฐานข้อมูลรายการ Kosarak ที่เป็นฐานข้อมูลขนาดใหญ่ ได้สร้างผลลัพธ์เซตรายการเป็นจำนวนมากจึงไม่สามารถแสดงจำนวนผลลัพธ์เซตรายการที่ค้นพบของทั้งสองฐานข้อมูลรายการนี้ได้



ภาพที่ 3.20 ผลการทดลองการค้นหาเซตรายการด้วยขั้นตอนวิธีไมโครในด้านจำนวนผลลัพธ์เซตรายการที่ค้นพบของฐานข้อมูลรายการที่มีลักษณะข้อมูลหนาแน่น



ภาพที่ 3.21 ผลการทดลองการค้นหาเซตรายการด้วยขั้นตอนวิธีไมโครในด้านจำนวนผลลัพธ์เซตรายการที่ค้นพบของฐานข้อมูลรายการที่มีลักษณะข้อมูลเบาบาง

บทที่ 4
การค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้น

การค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอที่ปรากฏขึ้นด้วยขั้นตอนวิธีไมโคร (MICRO) ดังที่กล่าวในบทที่ 3 ที่ซึ่งขั้นตอนวิธีนี้ได้มีการสร้างเซตรายการเป็นจำนวนมาก (Overwhelming) จึงเป็นเหตุให้ผู้ใช้ไม่สามารถนำเซตรายการดังกล่าวมาใช้ประโยชน์ได้ด้วยเหตุนี้ ในบทนี้จึงได้นำเสนอการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏ (Mining regular itemsets with interesting changes on regularity of occurrence) ด้วยการพิจารณาเซตรายการที่ปรากฏสม่ำเสมอภายใต้ค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด (Regularity threshold, σ_r) และแนวโน้มความเปลี่ยนแปลงของพฤติกรรมการปรากฏขึ้นอย่างสม่ำเสมอที่เพิ่มขึ้นเมื่อเวลาผ่านไปภายใต้ค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด (Change value threshold, σ_c) โดยการค้นหาเซตรายการดังกล่าวจะทำการกำหนดขอบเขตที่น่าสนใจเพื่อคัดกรองเซตรายการที่ไม่น่าสนใจ/ไม่สำคัญ นอกจากนั้นในบทนี้จะกล่าวถึงขั้นตอนวิธีการทำงานที่เรียกว่า ริคครอม (Regular Itemsets with interesting Changes on Regularity of Occurrence Miner, RICROM) และใช้โครงสร้างข้อมูลที่เรียกว่า *NIWS* (New Interval Word Segment structure) ที่ช่วยให้สามารถอ่านข้อมูลจากฐานข้อมูลรายการเพียงครั้งเดียวเท่านั้น และจัดเก็บข้อมูลที่ปรากฏขึ้นของแต่ละเซตรายการได้อย่างมีประสิทธิภาพ นอกจากนี้ยังมีการลดทอนปริภูมิจากการประยุกต์ใช้สมบัติปิดการลดลง เพื่อลดเวลาในการประมวลผลข้อมูลและลดพื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลการปรากฏขึ้นของเซตรายการได้อย่างมีประสิทธิภาพ การค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏขึ้นจะสามารถแบ่งออกเป็น 3 ขั้นตอนวิธีย่อย คือ ขั้นตอนวิธีการอ่านฐานข้อมูลรายการ (DB-scanning) ขั้นตอนวิธีการสร้างเซตรายการขนาด 2 รายการ (2-itemsets-generation) และการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏขึ้น (Regular-itemsets-mining) โดยนิยามและรายละเอียดขั้นตอนวิธีการทำงานสามารถอธิบายได้ ดังนี้

4.1 นิยาม

ในงานวิจัยนี้ได้ประยุกต์ใช้นิยามพื้นฐานดังที่กล่าวในบทที่ 3 บางส่วน อาทิเช่น การคำนวณอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น (c^X) (นิยามที่ 3.1) แต่สำหรับการพิจารณาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่เพิ่มขึ้นเมื่อเวลาผ่านไปภายใต้ค่าขีดแบ่งความสม่ำเสมอและค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนดจะมีนิยามที่แตกต่างจากเดิม ที่ซึ่งสามารถนิยามได้ ดังนี้

นิยามที่ 4.1 *เซตรายการ X จะเป็นเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้นก็ต่อเมื่อ 1) เซตรายการ X ปรากฏขึ้นทั้งสองฐานข้อมูลรายการ*

2) เซตรายการ X มีค่าความสม่ำเสมอ r^X น้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด และ 3) เซตรายการ X มีค่าความเปลี่ยนแปลง c^X มากกว่าหรือเท่ากับค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด

จากนิยามข้างต้นปัญหาการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้นจะเป็นการค้นหาเซตรายการที่ปรากฏขึ้นทั้งสองฐานข้อมูลรายการ มีอัตรา (ร้อยละ) ค่าความสม่ำเสมอน้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอ และมีอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นมากกว่าหรือเท่ากับค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด

ตัวอย่างที่ 4.1 *จากฐานข้อมูลรายการ TDB_1 ประกอบด้วย 40 ทรานแซกชัน และฐานข้อมูลรายการ TDB_2 ประกอบด้วย 56 ทรานแซกชัน แสดงดังภาพที่ 2.2 เมื่อทำการพิจารณารายการ 'a' สามารถระบุได้ถึงเซตของหมายเลขทรานแซกชันที่มีรายการ 'a' ปรากฏอยู่ในฐานข้อมูลรายการ TDB_1 ดังนี้ $T^a_{TDB_1} = \{ 1^a, 2^a, 3^a, 4^a, 5^a, 6^a, 21^a, 22^a, 23^a, 24^a, 25^a\}$ โดยอัตรา (ร้อยละ) ค่าความสม่ำเสมอ $r^a_{TDB_1}$ มีค่าเท่ากับ 37.5% ของทรานแซกชันทั้งหมด และสามารถระบุได้ถึงเซตของหมายเลขทรานแซกชันที่มีรายการ 'a' ปรากฏอยู่ในฐานข้อมูลรายการ TDB_2 ได้ดังนี้ $T^a_{TDB_2} = \{ 1^a, 2^a, 3^a, 4^a, 5^a, 6^a, 7^a, 8^a, 9^a, 11^a, 12^a, 13^a, 14^a, 15^a, 16^a, 44^a, 45^a, 46^a, 47^a, 48^a, 49^a, 50^a, 51^a, 52^a, 53^a, 54^a, 55^a, 56^a \}$ โดยอัตรา (ร้อยละ) ค่าความสม่ำเสมอ $r^a_{TDB_2}$ มีค่าเท่ากับ 50% ถ้าผู้ใช้กำหนดค่าขีดแบ่งความสม่ำเสมอมีค่าเท่ากับ 60% จะสังเกตุได้ว่าอัตรา (ร้อยละ) ค่าความสม่ำเสมอของทั้งสองฐานข้อมูลรายการมีค่าน้อยกว่าค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด จากนั้นสามารถคำนวณหาอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นของรายการ 'a' ที่ซึ่งมีค่าเท่ากับ 1.33 เท่า ถ้าผู้ใช้กำหนดค่าขีดแบ่งการเปลี่ยนแปลงมีค่าเท่ากับ 1.25 ดังนั้น สามารถกล่าวได้ว่ารายการ 'a' เป็นรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้น เนื่องจากมีอัตรา (ร้อยละ) ค่าความสม่ำเสมอน้อยกว่าค่าขีดแบ่งความสม่ำเสมอ และมีอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นมากกว่าค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด*

4.2 วิธีการดำเนินงานวิจัย

ในการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏขึ้น ผู้วิจัยได้นำเสนอขั้นตอนวิธีริคครอมที่จะใช้ โครงสร้างข้อมูล *NIWS* (New Interval Word Segment structure) สำหรับจัดเก็บข้อมูล โดยในส่วนนี้จะอธิบายถึงลักษณะโครงสร้างข้อมูล การอินเตอร์เซกชัน (Intersection) ระหว่าง $NIWS^X$ กับ $NIWS^Y$ ของเซตรายการ *X* และ *Y* ใด ๆ สำหรับการคำนวณหาเซตรายการที่ปรากฏขึ้นร่วมกัน การคำนวณอัตรา (ร้อยละ) ค่าความสม่ำเสมอจาก $NIWS^X$ สำหรับการคำนวณค่าความสม่ำเสมอของเซตรายการ X ที่ซึ่งบ่งบอกถึงช่วงหรือระยะห่างที่มากที่สุดของเซตรายการ X ที่ปรากฏขึ้นอย่างน้อยหนึ่งทรานแซกชันใน

ฐานข้อมูลรายการและขั้นตอนวิธีริคครอม (RICROM algorithm) ที่ใช้สำหรับการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏ ดังนี้

4.2.1 โครงสร้างข้อมูล *NIWS*

ในการค้นหาเซตรายการที่ปรากฏบ่อย บิตเวกเตอร์ (Bit-vectors) ได้ถูกนำมาประยุกต์ใช้ในการจัดเก็บข้อมูลที่ปรากฏขึ้นของรายการ/เซตรายการ โดยถ้าเซตรายการใดปรากฏในทรานแซกชัน j^{th} จะทำให้บิตในลำดับที่ j^{th} มีค่าเป็น 1 และในทางกลับกัน ถ้าเซตรายการใดไม่ปรากฏในทรานแซกชัน j^{th} จะทำให้บิตในลำดับที่ j^{th} มีค่าเป็น 0 (หมายเหตุ 8 บิต มีค่าเท่ากับ 1 ไบต์ (Byte))

ตัวอย่างที่ 4.2 *จากฐานข้อมูลรายการ TDB_1 ประกอบด้วย 40 ทรานแซกชัน และฐานข้อมูลรายการ TDB_2 ประกอบด้วย 56 ทรานแซกชัน แสดงดังภาพที่ 2.2 เมื่อทำการพิจารณารายการ 'a' สามารถระบุได้ถึงเซตของหมายเลขทรานแซกชันที่มีรายการ 'a' ปรากฏอยู่ในฐานข้อมูลรายการ TDB_1 ดังนี้ $T^{a}_{TDB_1}$ = { 1^a, 2^a, 3^a, 4^a, 5^a, 6^a, 21^a, 22^a, 23^a, 24^a, 25^a} จะได้บิตเวกเตอร์ในรูปแบบเลขฐานสองทั้งหมด 5 ไบต์ สำหรับจัดเก็บข้อมูลการปรากฏขึ้นของรายการ 'a' ในฐานข้อมูลรายการ TDB_1 ดังนี้ <11111100, 00000000 00001111, 10000000, 00000000> และสามารถจัดเก็บในรูปแบบเลขฐานสิบ ดังนี้ <$2^7+2^6+2^5+2^4+2^3+2^2$, 0, $2^3+2^2+2^1+2^0$, 2^7, 0> = <252, 0, 15, 128, 0 > (หมายเหตุ ไบต์ที่ 1 มีค่าเท่ากับ 252 ใช้แทนข้อมูลที่ปรากฏขึ้นในทรานแซกชัน t_1-t_8, ไบต์ที่ 2 มีค่าเท่ากับ 0 ใช้แทนข้อมูลที่ปรากฏขึ้นในทรานแซกชัน t_9-t_{16}, ไบต์ที่ 3 มีค่าเท่ากับ 15 ใช้แทนข้อมูลที่ปรากฏขึ้นในทรานแซกชัน t_{17}-t_{24}, ไบต์ที่ 4 มีค่าเท่ากับ 128 ใช้แทนข้อมูลที่ปรากฏขึ้นในทรานแซกชัน t_{25}-t_{32} และ ไบต์ที่ 5 มีค่าเท่ากับ 0 ใช้แทนข้อมูลที่ปรากฏขึ้นในทรานแซกชัน t_{33}-t_{40}) และสามารถระบุได้ถึงเซตของหมายเลขทรานแซกชันที่มีรายการ 'a' ปรากฏอยู่ในฐานข้อมูลรายการ TDB_2 ได้ดังนี้ $T^{a}_{TDB_2}$ = { 1^a, 2^a, 3^a, 4^a, 5^a, 6^a, 7^a, 8^a, 9^a, 11^a, 12^a, 13^a, 14^a, 15^a, 16^a, 44^a, 45^a, 46^a, 47^a, 48^a, 49^a, 50^a, 51^a, 52^a, 53^a, 54^a, 55^a, 56^a } จะได้บิตเวกเตอร์ในรูปแบบเลขฐานสองทั้งหมด 7 ไบต์ โดยในการจัดเก็บข้อมูลการปรากฏขึ้นของรายการ 'a' ในฐานข้อมูลรายการ TDB_2 ดังนี้ <11111111, 10111111 00000000, 00000000, 00000000, 00011111, 11111111> และสามารถจัดเก็บในรูปแบบเลขฐานสิบ (หมายเหตุ โดยมีการคำนวณเหมือนกันกับฐานข้อมูลรายการ TDB_1 ทุกประการ) ได้ดังนี้ <255, 191, 0, 0, 0, 31, 255>*

จากขั้นตอนวิธีดังกล่าว หากบิตเวกเตอร์มีไบต์เป็น 0 จำนวนมาก ก็จะทำให้สิ้นเปลืองพื้นที่ในการจัดเก็บข้อมูลรวมถึงสิ้นเปลืองเวลาในการประมวลผล ต่อมาจึงได้มีการจัดเก็บข้อมูลแบบไดนามิคบิตเวกเตอร์ (Dynamic bit-vectors) โดยทำการลดทอนไบต์ที่มีค่าเท่ากับ 0 ในหัวและท้ายของสายบิตเวกเตอร์ ที่ซึ่งสามารถใช้เวลาในการประมวลผลข้อมูลได้รวดเร็วขึ้นและลดพื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลได้เพียงบางส่วนเท่านั้น เพื่อหลีกเลี่ยงปัญหาดังกล่าวจึงได้มีการคิดค้นวิธีการจัดเก็บข้อมูลแบบอินเทอร์วอลเวิร์ดเซกเมนต์ (Interval word segments, *IWS*) โดยการลดทอนไบต์หรือเวิร์ด (Word) ที่เป็น 0 ทั้งหมด โดยสามารถกำหนด *IWS* ของเซตรายการ *X* ได้ดั ง นี้ IWS^X = { <wi_1, {$w_{1,1}$, $w_{1,2}$, ..., $w_{1,p}$}>, <wi_2, {$w_{2,1}$, $w_{2,2}$, ..., $w_{2,q}$}>, ... , <wi_u, {$w_{u,1}$, $w_{u,2}$, ..., $w_{u,v}$}> } โดยที่ แต่ละทูเพิล (Tuple) ของลำดับที่ y^{th} ได้แก่ <wi_y, W_y= {$w_{y,1}$, $w_{y,2}$, ...,

$w_{y,p}$}> จะประกอบด้วย 1) ตำแหน่งเวิร์ด (word index, wi_y) แสดงถึงตำแหน่งที่ไม่มีเวิร์ด 0 และ 2) แต่ละเวิร์ดที่ไม่ใช่เวิร์ด 0 (W_y= {$w_{y,1}$, $w_{y,2}$, ..., $w_{y,p}$}) ที่ซึ่ง IWS^X สามารถหลีกเลี่ยงการจัดเก็บเวิร์ด 0 โดยถ้าเจอเวิร์ด 0 แล้วเวิร์ดต่อมาที่ไม่ใช่เวิร์ด 0 ก็จะทำการแบ่งทูเพิลใหม่ ที่ซึ่งช่วยประหยัดพื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลและเวลาที่ใช้ในการประมวลผลข้อมูล

แต่อย่างไรก็ตามในการแบ่งทูเพิลใหม่ต่อเวิร์ด 0 ไม่สามารถช่วยประหยัดพื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลได้มากเท่าไหร่ และรวมถึงเวลาที่ในการประมวลผลข้อมูลในขั้นตอนวิธีการสร้างหรือจัดเก็บข้อมูลลงบนทูเพิลใหม่ด้วย ด้วยเหตุนี้ จึงทำการเปลี่ยนเงื่อนไขจากการแบ่งทูเพิลใหม่ต่อเวิร์ด 0 ไปเป็นการแบ่งทูเพิลใหม่เมื่อมีเวิร์ด 0 ปรากฏขึ้น 2 เวิร์ดขึ้นไป เนื่องจากข้อมูลที่ใช้ในการจัดเก็บต่อทูเพิลมี 2 ข้อมูล โดยจะเรียกโครงสร้างข้อมูลนี้ว่า โครงสร้างข้อมูล *NIWS* (New Interval Word Segment)

ตัวอย่างที่ 4.2 *จากฐานข้อมูลรายการ TDB_1 แสดงดังภาพที่ 2.2 เมื่อทำการพิจารณารายการ ‘a’ สามารถระบุได้ถึงเซตของหมายเลขทรานแซกชันที่มีรายการ ‘a’ ปรากฏอยู่ในฐานข้อมูลรายการ TDB_1 ดังนี้ $T^a_{TDB_1}$= { 1^a, 2^a, 3^a, 4^a, 5^a, 6^a, 21^a, 22^a, 23^a, 24^a, 25^a} และสามารถระบุได้ถึงเซตของหมายเลขทรานแซกชันที่มีรายการ ‘a’ ปรากฏอยู่ในฐานข้อมูลรายการ TDB_2 ได้ดังนี้ $T^a_{TDB_2}$ = { 1^a, 2^a, 3^a, 4^a, 5^a, 6^a, 7^a, 8^a, 9^a, 11^a, 12^a, 13^a, 14^a, 15^a, 16^a, 44^a, 45^a, 46^a, 47^a, 48^a, 49^a, 50^a, 51^a, 52^a, 53^a, 54^a, 55^a, 56^a }* โดยสามารถคำนวณ *NIWS ของรายการ ‘a’ ในฐานข้อมูลรายการ TDB_1 ($NIWS^a_{TDB_1}$) ดังนี้ รายการ ‘a’ ปรากฏขึ้นครั้งแรกในทรานแซกชัน t_1 แล้วเวิร์ดแรกจะได้ ‘1000 0000’ มีค่าเท่ากับ 2^7=128 (หมายเหตุ สำหรับการปรากฏขึ้นของรายการ ‘a’ ระหว่างทรานแซกชันที่ t_1 - t_8) จากนั้นสร้างทูเพิลแรกของ $NIWS^a_1$ ได้เป็น <1, {128}> (หมายเหตุ โดยที่ 1 คือ ตำแหน่งเวิร์ดของทูเพิลที่แสดงตำแหน่งเวิร์ดที่ไม่ใช่ 0 ครั้งแรก) ต่อมาการปรากฏขึ้นครั้งที่สองของรายการ ‘a’ คือ ทรานแซกชันที่ t_2 แล้วเวิร์ดแรกจะถูกอัปเดตโดยทรานแซกชันที่ t_2 ดังนี้ ‘1100 0000’ มีค่าเท่ากับ 2^7+ 2^6=192 และทูเพิลแรกของ ($NIWS^a_{TDB_1}$) ก็จะถูกอัปเดตด้วยเช่นกัน ได้เป็น <1, {192}> ต่อมาการปรากฏขึ้นครั้งที่สามถึงหกของรายการ ‘a’ คือทรานแซกชันที่ t_3 - t_6 แล้วเวิร์ดแรกจะถูกอัปเดต ดังนี้ ‘1111 1100’ มีค่าเท่ากับ2^7+ 2^6+2^5+2^4+2^3+2^2=252 และทูเพิลแรกของ $NIWS^a_1$ ก็จะถูกอัปเดต ได้เป็น <1, {252}> ต่อมาการปรากฏขึ้นครั้งที่เจ็ด คือ ทรานแซกชัน t_{21} จึงเป็นสาเหตุให้เวิร์ดที่สองเป็น ‘0000 000’ แล้วได้เวิร์ดที่สามเป็น ‘0000 1000’ มีค่าเท่ากับ 2^3 = 8 และอัปเดตค่าลงทูเพิลแรก ได้เป็น <1, {252, 0, 8}> ต่อมาพิจารณาการปรากฏขึ้นครั้งแปดถึงสิบเอ็ด ได้แก่ทรานแซกชันที่ t_{22} - t_{24} แล้วเวิร์ดที่สามจะถูกอัปเดตดังนี้ ‘0000 1111’ มีค่าเท่ากับ 2^3+2^2+2^1+2^0 = 15 และอัปเดตค่าลงทูเพิลแรก ได้เป็น <1, {252, 0, 15}> แล้วเวิร์ดที่สี่จะได้ ‘1000 000’ มีค่าเท่ากับ 2^7=128 แล้วอัปเดตลงทูเพิล เมื่อพิจารณาการปรากฏขึ้นของรายการ ‘a’ ทั้งหมดในฐานข้อมูลรายการ TDB_1 จะได้ ($NIWS^a_{TDB_1}$) = { < 1, {252, 0, 15, 128} > } และ NIWS ของรายการ a ในฐานข้อมูลรายการ TDB_2 จะได้ ($NIWS^a_{TDB_2}$) = { < 1, {255, 191} >, < 6, {31,*

255} > } (หมายเหตุ โดยมีขั้นตอนวิธีเดียวกันกับการคำนวณ NIWS ของรายการ 'a' ในฐานข้อมูลรายการ TDB_1) แสดงดังภาพที่ 4.1

$T^{a}_{TDB_1}$ = { 1, 2, 3, 4, 5, 6, 21, 22, 23, 24, 25 }

$T^{a}_{TDB_2}$ = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56 }

Word Index	1	2	3	4	5	6	7
Range of tids	1-8	9-16	17-24	25-32	33-40	41-48	49-56
Binary Value of TDB_1	1111 1100	0000 0000	0000 1111	1000 0000	0000 0000	-	-
Word Value of TDB_1	252	0	15	128	0	-	-
Binary Value of TDB_2	1111 1111	1011 1111	0000 0000	0000 0000	0000 0000	0001 1111	1111 1111
Word Value of TDB_2	255	191	0	0	0	31	255

$NIWS^{a}_{TDB_1}$= { <1, {252, 0, 15, 128}> } and $NIWS^{a}_{TDB_2}$= { <1, {255, 191}>, <6, {31, 255}> }

ภาพที่ 4.1 โครงสร้างข้อมูล *NIWS* ของรายการ *'a'*

4.2.2 การอินเตอร์เซกชันระหว่าง $NIWS^X$ กับ $NIWS^Y$

สำหรับการอินเตอร์เซกชันระหว่าง $NIWS^X$ กับ $NIWS^Y$ ของเซตรายการ X และ Y ใด ๆ จะเริ่มต้นจากการหาตำแหน่งของเวิร์ดที่มีค่าไม่เท่ากับศูนย์ (Non-zero bytes) แล้วพิจารณาตำแหน่งของแต่ละเวิร์ดที่ตรงกันของ $NIWS^X$ กับ $NIWS^Y$ จากนั้นใช้ตัวดำเนินการแบบบิต (Bitwise operation) โดยเครื่องหมาย AND (&) ในการดำเนินการเวิร์ดดังกล่าว แล้วดำเนินการไปจนกระทั่งไม่มีตำแหน่งเวิร์ดที่ตรงกันและจึงจัดเก็บเป็น $NIWS^{XY}$

ตัวอย่างที่ 4.3 *จากฐานข้อมูลรายการ TDB_1 ประกอบด้วย 40 ทรานแซกชัน และฐานข้อมูลรายการ TDB_2 ประกอบด้วย 56 ทรานแซกชัน แสดงดังภาพที่ 2.2 เมื่อทำการพิจารณารายการ 'a' สามารถระบุได้ถึงเซตของหมายเลขทรานแซกชันที่มีรายการ 'a' ปรากฏอยู่ในฐานข้อมูลรายการ TDB_1 ดังนี้ $T^{a}_{TDB_1}$ = { 1^a, 2^a, 3^a, 4^a, 5^a, 6^a, 21^a, 22^a, 23^a, 24^a, 25^a} โดย $NIWS^{a}_{TDB_1}$ มีค่าเท่ากับ { < 1, { 252, 0, 15, 128} > } และสามารถระบุได้ถึงเซตของหมายเลขทรานแซกชันที่มีรายการ 'b' ปรากฏอยู่ในฐานข้อมูลรายการ TDB_1 ได้ดังนี้ $T^{b}_{TDB_1}$ = { 1^a, 2^a, 3^a, 4^a, 5^a, 6^a, 7^a, 8^a, 9^a, 11^a, 12^a, 13^a, 14^a, 15^a, 16^a, 44^a, 45^a, 46^a, 47^a, 48^a, 49^a, 50^a, 51^a, 52^a, 53^a, 54^a, 55^a, 56^a } โดย $NIWS^{b}_{TDB_1}$ มีค่าเท่ากับ { < 1, {250, 62, 143, 128, 16} > } และสามารถคำนวณ $NIWS^{ab}_{TDB_1}$ โดยพิจารณาจากตำแหน่งเวิร์ดที่มีค่าไม่เท่ากับศูนย์และมีตำแหน่งเวิร์ดที่ตรงกันของ $NIWS^{a}_{TDB_1}$ และ $NIWS^{b}_{TDB_1}$ ดังนี้ 252&250 = 248, 0&62 = 0, 15&143 = 15, 128&128 =128 ดังนั้นจะได้ $NIWS^{ab}_{TDB_1}$ มีค่าเท่ากับ { < 1, {248, 0, 15, 128} > } และในการคำนวณ $NIWS^{ab}_{TDB_2}$ ก็สามารถทำการคำนวณเช่นเดียวกันกับฐานข้อมูลรายการ TDB_1 ได้ดังนี้ $NIWS^{ab}_{TDB_2}$ มีค่าเท่ากับ { < 1, {241, 31} >, < 6, {5, 254} > } แสดงดังภาพที่ 4.2*

$\mathrm{NIWS}^{a}_{\mathrm{TDB}_1} = \{ <1, \{252, 0, 15, 128\}> \}$
$\mathrm{NIWS}^{b}_{\mathrm{TDB}_1} = \{ <1, \{250, 62, 143, 128, 16\}> \}$
$\mathrm{NIWS}^{ab}_{\mathrm{TDB}_1} = \{ <1, \{248, 0, 15, 128\}> \}$

$\mathrm{NIWS}^{a}_{\mathrm{TDB}_2} = \{ <1, \{255, 191\}>, <6, \{31, 255\}> \}$
$\mathrm{NIWS}^{b}_{\mathrm{TDB}_2} = \{ <1, \{254, 95, 255, 255, 255, 229, 254\}> \}$
$\mathrm{NIWS}^{ab}_{\mathrm{TDB}_2} = \{ <1, \{254, 31\}>, <6, \{5, 254\}> \}$

ภาพที่ 4.2 การอินเตอร์เซกชันของ $NIWS^{ab}$

4.2.2 การคำนวณอัตรา (ร้อยละ) ค่าความสม่ำเสมอจาก $NIWS^{X}$

การคำนวณอัตรา (ร้อยละ) ค่าความสม่ำเสมอของเซตรายการ *X* จาก $NIWS^{X}$ จะใช้ตารางค้นหา (Look-up table) ที่มีทั้งหมด 256 ทูเพิล โดยแต่ละทูเพิลจะประกอบไปด้วย 3 ข้อมูล ดังต่อไปนี้

1. *pf* คือ ตำแหน่งของการปรากฏขึ้นครั้งแรกของบิต 1 (หมายเหตุ ถ้าบิตเป็น 0 ทั้งหมด ค่า *pf* จะมีค่าเท่ากับ 8)

2. *nl* คือ จำนวนของบิต 0 หลังจากการปรากฏขึ้นของบิต 1 ครั้งสุดท้ายของไบต์ (หมายเหตุ ถ้าบิตเป็น 0 ทั้งหมด ค่า *nl* จะมีค่าเท่ากับ 8)

3. *mg* คือ ช่วงหรือระยะห่างที่มากที่สุดระหว่างบิต 1 สองบิต (หมายเหตุ ถ้าบิตเป็น 1 ทั้งหมด ค่า *mg* จะมีค่าเท่ากับ 1)

จาก $NIWS^{X}$ และ ตารางค้นหา สามารถคำนวณอัตรา (ร้อยละ) ค่าความสม่ำเสมอของเซตรายการ *X* ได้จากการพิจารณาแต่ละเวิร์ด $w_{y,p}$ ในแต่ละลำดับที่ y^{th} ทูเพิลของ $NIWS^{X}$ โดยแบ่งได้เป็น 4 กรณี ดังต่อไปนี้

1. ถ้า $w_{y,p}$ เป็นเวิร์ดแรกของทูเพิลแรก แล้วค่าความสม่ำเสมอจะมีค่าเท่ากับ

$$r^{w,p} = max((((wi_y - 1) \times 8) + pf(w_{y,p})), mg(w_{y,p}))$$

2. ถ้า $w_{y,p}$ เป็นเวิร์ดแรกของทูเพิลลำดับที่ y^{th} แล้วค่าความสม่ำเสมอจะมีค่าเท่ากับ

$$r^{w,p} = max((nl(w_{y-1},|W_{y-1}|) + (((wi_y - 1) - (wi_{y-1} + |W_{y-1}|-1)) \times 8) + pf(W_{y,p})), mg(w_{y,p}))$$

3. ถ้า $w_{y,p}$ เป็นเวิร์ดที่ไม่ใช่เวิร์ดแรกของทูเพิลลำดับที่ y^{th} แล้วค่าความสม่ำเสมอจะมีค่าเท่ากับ

$$r^{w,p} = max((nl(w_{y,p-1}) + (nw_0 \times 8) + pf(w_{y,p})), mg(w_{y,p}))$$

4. ถ้า $w_{y,p}$ เป็นเวิร์ดสุดท้ายของทูเพิลสุดท้าย แล้วค่าความสม่ำเสมอจะมีค่าเท่ากับ

$$r^{w,p} = nl(w_{y,p}) + (lwo^{TDB} - (wi_y + |W_y| - 1)) \times 8$$

โดยที่ wi_y คือ ตำแหน่งเวิร์ดเริ่มต้นของทูเพิลลำดับที่ y^{th}

$pf(w_{y,p})$ คือ จำนวนบิต 0 ที่ติดกัน ก่อนการปรากฏบิต 1 ครั้งแรกในเวิร์ด $w_{y,p}$

mg($w_{y,p}$) คือ ช่วงหรือระยะห่างที่มากที่สุดระหว่างบิต 1 สองบิตของเวิร์ด $w_{y,p}$

nl(w_{y-1},$|W_{y-1}|$) คือ จำนวนบิต 0 จากการปรากฏขึ้นครั้งสุดท้ายของเซตรายการ *X* ในทูเพิลลำดับที่ $(y-1)^{th}$

wi_{y-1} คือ ตำแหน่งเวิร์ดเริ่มต้นของทูเพิลลำดับที่ $(y-1)^{th}$

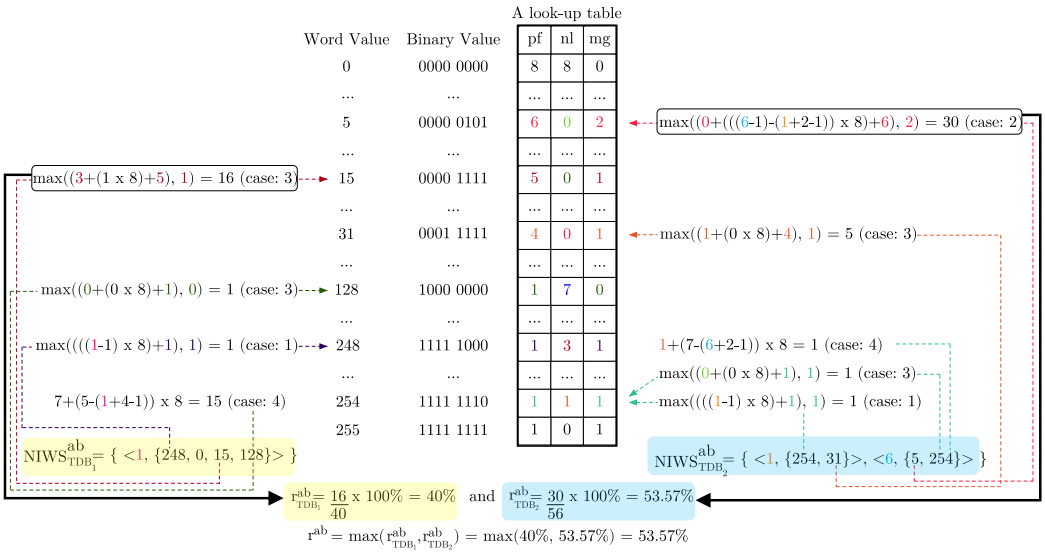
$|W_{y-1}|$ คือ จำนวนของเวิร์ดในทูเพิลลำดับที่ $(y-1)^{th}$

nl($w_{y,p-1}$) คือ จำนวนบิต 0 จากการปรากฏขึ้นครั้งสุดท้ายของเซตรายการ *X ในเวิร์ดลำดับที่* $(p-1)^{th}$ของทูเพิลลำดับที่ $(y-1)^{th}$

nw_0 คือ จำนวนเวิร์ด 0 ที่ปรากฏขึ้นติดกันก่อนการปรากกฏขึ้นของเวิร์ด $w_{y,p}$

nl($w_{y,p}$) คือ จำนวนของบิต 0 จากการปรากฏขึ้นครั้งสุดท้ายในเวิร์ด $w_{y,p}$

lwo^{TDB} คือ ตำแหน่งเวิร์ดสุดท้ายของฐานข้อมูลรายการ TDB



ภาพที่ 4.3 การคำนวณอัตรา (ร้อยละ) ค่าความสม่ำเสมอของเซตรายการ *'ab'* จาก $NIWS^{ab}$

ตัวอย่างที่ 4.4 *จากฐานข้อมูลรายการ TDB_1 แสดงดังภาพที่ 2.2 เมื่อทำการพิจารณารายการ 'ab' สามารถระบุได้ถึง $NIWS^{ab}_{TDB_1}$ เท่ากับ { < 1, {248, 0, 15, 128} > } และในการคำนวณอัตรา (ร้อยละ) ค่าความสม่ำเสมอของเซตรายการ 'ab' จาก $NIWS^{ab}_{TDB_1}$ โดยจะเริ่มพิจารณาจากเวิร์ดแรกของทูเพิลแรก (หมายเหตุ กรณีที่ 1) ได้แก่ 248 จะได้ ค่าความสม่ำเสมอมีค่าเท่ากับ $r^{w,p}$ = max(((wi_y - 1) x 8) + $pf(w_{y,p})$), $mg(w_{y,p})$) = max((((1-1) x 8) + pf(248), mg(248)) = max((((1-1) x 8) + 1), 1) = max(1, 1) = 1 จากนั้นพิจารณาเวิร์ดถัดไป (หมายเหตุ ที่ไม่ใช่เวิร์ด 0) มีค่าเท่ากับ 15 ที่ซึ่งไม่ใช่เวิร์ดแรกของทูเพิล (หมายเหตุ กรณีที่ 3) แล้วค่าความสม่ำเสมอจะมีค่าเท่ากับ $r^{w,p}$ = max((nl($w_{y,p-1}$) + (nw_0 x 8) + $pf(w_{y,p})$, $mg(w_{y,p})$) = max((nl(248) + (1 x 8) + pf(15)), mg(15)) = max((3+8+5), 1) = max(16, 1) = 16 จากนั้นพิจารณาเวิร์ดถัดไปมีค่าเท่ากับ 128 ที่ซึ่งไม่ใช่เวิร์ด*

แรกของทูเพิล (หมายเหตุ กรณีที่ 3) แล้วค่าความสม่ำเสมอจะมีค่าเท่ากับ $r^{w,p} = max((nl(w_{y,p-1}) + (nw_0 \times 8) + pf(w_{y,p}), mg(w_{y,p})) = max((nl(15) + (0 \times 8) + pf(128)), mg(128)) = max((0+0+1), 0) = max(1, 0) = 1$ *และเนื่องจากเป็นเวิร์ดสุดท้ายของทูเพิลสุดท้าย (หมายเหตุ กรณีที่ 4) แล้วค่าความสม่ำเสมอจะมีค่าเท่ากับ* $r^{w,p} = nl(w_{y,p}) + (lwo^{TDB} - (wi_y + |W_y| - 1)) \times 8 = nl(128) + (5-(1+4-1)) \times 8 = 7+8 = 15$ *จากนั้นหาค่าที่มากที่สุดของค่าความสม่ำเสมอจากทุกกรณีที่ซึ่งมีค่าเท่ากับ 16 ดังนั้น อัตรา (ร้อยละ) ค่าความสม่ำเสมอของเซตรายการ 'ab'* จาก $NIWS^{ab}_{TDB_1}$ *มีค่าเท่ากับ 40% และการคำนวณอัตรา (ร้อยละ) ค่าความสม่ำเสมอของเซตรายการ 'ab'* จาก $NIWS^{ab}_{TDB_2}$ *สามารถคำนวณได้เหมือนกันกับฐานข้อมูลรายการ* TDB_1 *ที่ซึ่งมีค่าเท่ากับ 53.57% แล้วทำการหาค่าที่มากที่สุดระหว่าง* $r^{ab}_{TDB_1}$ *และ* $r^{ab}_{TDB_2}$ *ดังนั้นอัตรา (ร้อยละ) ค่าความสม่ำเสมอของเซตรายการ 'ab' มีค่าเท่ากับ 53.57% แสดงดังภาพที่ 4.3*

4.2.4 ขั้นตอนวิธีริคครอม

สำหรับการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้น ที่เรียกว่า ริคครอม สามารถแบ่งลักษณะการทำงานออกเป็น 3 ขั้นตอนวิธี ดังนี้

1. ขั้นตอนวิธีการอ่านฐานข้อมูลรายการ (DB-scanning) เป็นกระบวนการการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้นขนาด 1 รายการ โดยจะอ่านข้อมูลจากฐานข้อมูลรายการ TDB_1 และ TDB_2 เพียงครั้งเดียวเท่านั้น และจัดเก็บข้อมูลที่ปรากฏขึ้นไปยังลิสต์รายการ ที่ซึ่งประกอบไปด้วย 6 ข้อมูล ดังต่อไปนี้

1. ชื่อรายการ
2. อัตรา (ร้อยละ) ค่าความสม่ำเสมอของรายการ i_k ในฐานข้อมูลรายการ TDB_1 ($r^{i_k}_{TDB_1}$)
3. อัตรา (ร้อยละ) ค่าความสม่ำเสมอของรายการ i_k ในฐานข้อมูลรายการ TDB_2 ($r^{i_k}_{TDB_2}$)
4. อัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นในรายการ i_k (c^{i_k})
5. *NIWS* ของรายการ i_k ในฐานข้อมูลรายการ TDB_1 ($NIWS^{i_k}_{TDB_1}$)
6. *NIWS* ของรายการ i_k ในฐานข้อมูลรายการ TDB_2 ($NIWS^{i_k}_{TDB_2}$)

Algorithm 1: *DB-scanning*

Input: TDB_1, TDB_2, σ_r, σ_c

Output: A set of regular items with interesting change on regularity of occurrence and *1List* containing regular items

1: • create a list named *1List* to maintain all single items
2: • create an entry for each item $i_k \in I$ in the *1List*
3: **for** each transaction t_p in database TDB_1 **do**
4: **for** each item i_k in transaction t_p **do**
5: • update $NIWS^{i_k}_{TDB_1}$ by tid p
6: • update $r^{i_k}_{TDB_1}$ by tid p
7: **for** each transaction t_q in database TDB_2 **do**
8: **for** each item i_k in transaction t_q **do**
9: • update $NIWS^{i_k}_{TDB_2}$ by tid q
10: • update $r^{i_k}_{TDB_2}$ by tid q
11: **for** each item i_k in *1List* **do**
12: • compute $r^{i_k} \leftarrow \max(r^{i_k}_{TDB_1}, r^{i_k}_{TDB_2})$
13: **if** $r^{i_k} > \sigma_r$ **then**
14: • remove the entry of i_k out of *1List*
15: **else**
16: • compute c^{i_k} by $\frac{r^{i_k}_{TDB_2}}{r^{i_k}_{TDB_1}}$
17: **if** $c^{i_k} \geq \sigma_c$ **then**
18: • identify i_k as a result

ภาพที่ 4.4 ขั้นตอนวิธีการอ่านฐานข้อมูลรายการ

ขั้นตอนวิธีการอ่านฐานข้อมูลรายการ แสดงดังภาพที่ 4.4 เริ่มต้นจากการสร้างลิสต์รายการขนาด 1 รายการ ที่เรียกว่า *1List* สำหรับจัดเก็บแต่ละรายการ $i_k \in I$ (บรรทัดที่ 1-2) จากนั้นอ่านข้อมูลในฐานข้อมูล TDB_1 และพิจารณาแต่ละทรานแซกชัน t_p และแต่ละรายการ $i_k \in t_p$ แล้วคำนวณ *NIWS* ของแต่ละรายการ i_k ในฐานข้อมูลรายการ TDB_1 ($NIWS^{i_k}_{TDB_1}$) และอัตรา (ร้อยละ) ค่าความสม่ำเสมอของแต่ละรายการ i_k ในฐานข้อมูลรายการ TDB_1 ($r^{i_k}_{TDB_1}$) เพื่ออัปเดตลงในลิสต์ *1List* (บรรทัดที่ 3-6) เมื่อพิจารณาครบทุกทรานแซกชัน t_p ในฐานข้อมูลรายการ TDB_1 แล้วต่อมาทำการอ่านข้อมูลจากฐานข้อมูลรายการ TDB_2 ที่ซึ่งมีขั้นตอนวิธีเหมือนกันกับฐานข้อมูล TDB_1 ทุกประการ (บรรทัดที่ 7-10) แต่ในขั้นตอนวิธีการคำนวณ *NIWS* ของแต่ละรายการ i_k ในฐานข้อมูลรายการจะเปลี่ยนจากการอัปเดตที่ $NIWS^{i_k}_{TDB_1}$ ไปเป็น $NIWS^{i_k}_{TDB_2}$ (บรรทัดที่ 9) และอัตรา (ร้อยละ) ค่าความสม่ำเสมอของแต่ละรายการ i_k ในฐานข้อมูลรายการจะเปลี่ยนจากการอัปเดตที่ $r^{i_k}_{TDB_1}$ ไปเป็น $r^{i_k}_{TDB_2}$ (บรรทัดที่ 10)

ขั้นตอนวิธีถัดไปจะทำการพิจารณาแต่ละรายการ i_k ในลิสต์ *1List* (บรรทัดที่ 11) แล้วคำนวณอัตรา (ร้อยละ) ค่าความสม่ำเสมอของจากค่าที่มากที่สุดระหว่าง$r^{i_k}_{TDB_1}$และ $r^{i_k}_{TDB_2}$ จากนั้นตรวจสอบ โดยพิจารณาดังนี้

- ถ้าอัตรา (ร้อยละ) ค่าความสม่ำเสมอมีค่ามากกว่าค่าขีดแบ่งความสม่ำเสมอ จะทำการลบรายการ i_k ออกจากลิสต์ *1List* (บรรทัดที่ 13-14)
- ถ้าอัตรา (ร้อยละ) ค่าความสม่ำเสมอมีค่าน้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอ จะทำการคำนวณอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นแล้วนำไปเปรียบเทียบกับค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด โดยถ้าอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นมีค่ามากกว่าหรือเท่ากับค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด แล้วจะทำการจัดเก็บรายการ i_k เป็นเซตของรายการที่มีความสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้น (บรรทัดที่ 15-18)

2. ขั้นตอนวิธีการสร้างเซตรายการขนาด 2 รายการ (2-itemsets-generation) เป็นกระบวนการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้นขนาด 2 รายการ

Algorithm 2: *2-itemsets-generation*

Input: *1List*, σ_r, σ_c
Output: A set of regular 2-itemsets with interesting change on regularity of occurrence and *2List* containing regular 2-itemsets

1: • create and initial a list called *2List* to maintain regular 2-itemsets
2: **for** each item i_j in *1List* **do**
3: **for** each item i_k in *1List* ($i_j \neq i_k$) **do**
4: • merge item i_j with item i_k to be itemset Z
5: • $NIWS^Z_{TDB_1} \leftarrow \text{intersect}(NIWS^{i_j}_{TDB_1}, NIWS^{i_k}_{TDB_1})$
6: • $NIWS^Z_{TDB_2} \leftarrow \text{intersect}(NIWS^{i_j}_{TDB_2}, NIWS^{i_k}_{TDB_2})$
7: • compute $r^Z_{TDB_1}$ from $NIWS^Z_{TDB_1}$ and $r^Z_{TDB_2}$ from $NIWS^Z_{TDB_2}$
8: • compute $r^Z \leftarrow \max(r^Z_{TDB_1}, r^Z_{TDB_2})$
9: **if** $r^Z \leq \sigma_r$ **then**
10: • create an entry of Z in *2List*
11: • compute cr^Z by $\frac{r^Z_{TDB_2}}{r^Z_{TDB_1}}$
12: **if** $c^Z \geq \sigma_c$ **then**
13: • identify Z as a result

ภาพที่ 4.5 ขั้นตอนวิธีการสร้างเซตรายการขนาด 2 รายการ

ขั้นตอนวิธีการสร้างเซตรายการขนาด 2 รายการ แสดงดังภาพที่ 4.5 เริ่มต้นจากการสร้างลิสต์รายการขนาด 2 รายการ ที่เรียกว่า *2List* สำหรับจัดเก็บเซตรายการที่ปรากฏสม่ำเสมอขนาด

2 รายการ โดยพิจารณาแต่ละรายการ i_j ในลิสต์ *1List* ร่วมกับรายการ i_k ในลิสต์ *1List* (หมายเหตุ โดยที่รายการ i_j ไม่เท่ากับรายการ i_k) จากนั้นทำการสร้างเซตรายการ Z โดยการรวมรายการ i_j กับรายการ i_k ($Z \leftarrow i_j \cup i_k$) (บรรทัดที่ 1-4) แล้วคำนวณ *NIWS* ของเซตรายการ Z ในฐานข้อมูลรายการ TDB_1 ($NIWS^Z_{TDB_1}$) จากการอินเตอร์เซกชันระหว่าง $NIWS^{i_j}_{TDB_1}$ และ $NIWS^{i_k}_{TDB_1}$ ($NIWS^Z_{TDB_1} \leftarrow NIWS^{i_j}_{TDB_1} \cap NIWS^{i_k}_{TDB_1}$) แล้วคำนวณ *NIWS* ของเซตรายการ Z ในฐานข้อมูลรายการ TDB_2 ($NIWS^Z_{TDB_2}$) จากการอินเตอร์เซกชันระหว่าง $NIWS^{i_j}_{TDB_2}$ และ $NIWS^{i_k}_{TDB_2}$ ($NIWS^Z_{TDB_2} \leftarrow NIWS^{i_j}_{TDB_2} \cap NIWS^{i_k}_{TDB_2}$) (บรรทัดที่ 5-6) จากนั้นคำนวณค่าความสม่ำเสมอของเซตรายการ Z ในฐานข้อมูลรายการ TDB_1 ($r^Z_{TDB_1}$) จาก $NIWS^Z_{TDB_1}$ และคำนวณค่าความสม่ำเสมอของเซตรายการ Z ในฐานข้อมูลรายการ TDB_2 ($r^Z_{TDB_2}$) จาก $NIWS^Z_{TDB_2}$ แล้วคำนวณอัตรา (ร้อยละ) ค่าความสม่ำเสมอของเซตรายการ Z จากค่าที่มากที่สุดระหว่าง $r^Z_{TDB_1}$ และ $r^Z_{TDB_2}$ (บรรทัดที่ 7-8) จากนั้นทำการตรวจสอบ ถ้าค่าอัตรา (ร้อยละ) ค่าความสม่ำเสมอมีค่าน้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอจะทำการเพิ่มเซตรายการ Z และข้อมูลไปยังลิสต์ *2List* (บรรทัดที่ 9-10) แล้วทำการคำนวณอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น แล้วนำไปเปรียบเทียบกับค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด โดยถ้าอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นมีค่ามากกว่าหรือเท่ากับค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด แล้วจะทำการจัดเก็บเซตรายการ Z เป็นผลลัพธ์ของเซตรายการที่มีความสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้น (บรรทัดที่ 11-13)

3. การค้นหาเซตรายการที่สม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้น (Regular-itemsets-mining) เป็นกระบวนการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้นตั้งแต่ขนาด 3 รายการขึ้นไป

Algorithm 3: *Regular-itemsets-mining*

Input: $kList$, σ_r, σ_c

Output: A complete set of itemsets with interesting change on regularity of occurrence

1: • create and initial $lList$ to maintain regular l-itemset (where $l = k + 1$)
2: **for** each itemset $X = \{i_a, \ldots, i_b\}$ in the $kList$ **do**
3: **for** each itemset $Y = \{i_a, \ldots, i_c\}$ in the $kList$ (where Y have the same prefix as X except only the last item) **do**
4: **if** there is an entry of the itemset i_b, i_c in *2List* **then**
5: • merge X and Y to be itemset Z
6: • $NIWS^Z_{TDB_1} \leftarrow \text{intersect}(NIWS^X_{TDB_1}, NIWS^Y_{TDB_1})$
7: • $NIWS^Z_{TDB_2} \leftarrow \text{intersect}(NIWS^X_{TDB_2}, NIWS^Y_{TDB_2})$
8: • compute $r^Z_{TDB_1}$ from $NIWS^Z_{TDB_1}$ and $r^Z_{TDB_2}$ from $NIWS^Z_{TDB_2}$
9: • compute $r^Z \leftarrow \max(r^Z_{TDB_1}, r^Z_{TDB_2})$
10: **if** $r^Z \leq \sigma_r$ **then**
11: • create an entry of Z in $lList$
12: • compute c^Z by $\frac{r^Z_{TDB_2}}{r^Z_{TDB_1}}$
13: **if** $c^Z \geq \sigma_c$ **then**
14: • identify Z as a result
15: **if** $lList$ contains more than one entry **then**
16: • repeat step of *Regular-itemsets-mining* by considering itemsets in $lList$
17: **else**
18: • empty $lList$
19: remove the entry of X out of $kList$

ภาพที่ 4.6 การค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏขึ้น

ขั้นตอนวิธีการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้น เริ่มต้นจากการสร้างลิสต์รายการขนาด l รายการ ที่เรียกว่า *lList* สำหรับจัดเก็บเซตรายการที่ปรากฏสม่ำเสมอขนาด l รายการ (หมายเหตุ โดยที่ $l = k+1$, เริ่มจากขนาดที่ l จนถึงขนาดที่ n) โดยพิจารณาแต่ละเซตรายการ $X = \{ i_a, \ldots, i_b \}$ ในลิสต์ *kList* กับแต่ละเซตรายการ $Y = \{ i_a, \ldots, i_c \}$ ในลิสต์ *kList* โดยที่เซตรายการ Y มีรายการก่อนหน้าที่เหมือนกับเซตรายการ X (หมายเหตุ ยกเว้นรายการสุดท้าย) (บรรทัดที่ 1-3) จากนั้นพิจารณารายการสุดท้าย $i_b \in X$ และรายการสุดท้าย $i_c \in Y$ แล้วรวมเป็นเซตรายการ '$i_b i_c$' จากนั้นนำไปใช้ตรวจสอบในลิสต์ *2List* ถ้าเซตรายการ '$i_b i_c$' มีอยู่ในลิสต์ *2List* จากนั้นทำการสร้างเซตรายการ Z โดยการรวมเซตรายการ X กับเซตรายการ Y ($Z \leftarrow X \cup Y$) แล้วคำนวณ *NIWS* ของเซตรายการ Z ในฐานข้อมูลรายการ TDB_1 ($NIWS^Z_{TDB_1}$) จากการอินเตอร์เซกชันระหว่าง $NIWS^X_{TDB_1}$ และ $NIWS^Y_{TDB_1}$ ($NIWS^Z_{TDB_1} \leftarrow NIWS^X_{TDB_1} \cap NIWS^Y_{TDB_1}$) แล้วคำนวณ *NIWS* ของเซตรายการ Z ในฐานข้อมูลรายการ TDB_2 ($NIWS^Z_{TDB_2}$) จากการอินเตอร์เซกชันระหว่าง $NIWS^X_{TDB_2}$ และ $NIWS^Y_{TDB_2}$ ($NIWS^Z_{TDB_2} \leftarrow NIWS^X_{TDB_2} \cap$

$NIWS^{Y}_{TDB_2}$) จากนั้นคำนวณค่าความสม่ำเสมอของเซตรายการ Z ในฐานข้อมูลรายการ TDB_1 ($r^{Z}_{TDB_1}$) จาก $NIWS^{Z}_{TDB_1}$ และคำนวณค่าความสม่ำเสมอของเซตรายการ Z ในฐานข้อมูลรายการ TDB_2 ($r^{Z}_{TDB_2}$) จาก $NIWS^{Z}_{TDB_2}$ แล้วคำนวณอัตรา (ร้อยละ) ค่าความสม่ำเสมอของเซตรายการ Z จากค่าที่มากที่สุดระหว่าง $r^{Z}_{TDB_1}$ และ $r^{Z}_{TDB_2}$ (บรรทัดที่ 4-9) จากนั้นทำการตรวจสอบ ถ้าค่าอัตรา (ร้อยละ) ค่าความสม่ำเสมอมีค่าน้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอจะทำการเพิ่มเซตรายการ Z และข้อมูลไปยังลิสต์ *lList* แล้วทำการคำนวณอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้น แล้วนำไปเปรียบเทียบกับค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนดโดยถ้าอัตราการเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นมีค่ามากกว่าหรือเท่ากับค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด แล้วจะทำการจัดเก็บเซตรายการ Z เป็นผลลัพธ์ของเซตรายการที่มีความสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้น (บรรทัดที่ 10-14)

หลังจากที่รวมเซตรายการ X กับเซตรายการทั้งหมดที่มีรายการก่อนหน้าที่เหมือนกับเซตรายการ X แล้ว จากนั้นจะทำการตรวจสอบ ถ้าลิสต์ *lList* มีมากกว่า 1 รายการก็จะทำซ้ำขั้นตอนวิธีการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้น และหลังจากที่พิจารณาเซตรายการ X และซูเปอร์เซตแล้ว เซตรายการ X จะถูกลบออกจาก *kList* จากนั้นพิจารณาเซตรายการอื่น ๆ ใน *kList* จนกระทั่งไม่สามารถพิจารณาได้แล้ว (บรรทัดที่ 15-19)

ตัวอย่างที่ 4.4 *กำหนดให้ค่าขีดแบ่งความสม่ำเสมอมีค่าเท่ากับ 60% และค่าขีดแบ่งการเปลี่ยนแปลงมีค่าเท่ากับ 1.25 โดยจะพิจารณาการค้นหาเซตรายการที่มีความสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของความสม่ำเสมอจากฐานข้อมูลรายการ TDB_1 และ TDB_2 แสดงดังภาพที่ 2.2 โดยเริ่มต้นจากการสร้างลิสต์ 1List สำหรับจัดเก็บทุกรายการขนาด 1 รายการ จากนั้นอ่านทรานแซกชัน t_1 = { a, b, c, d } จากฐานข้อมูลรายการ TDB_1 แล้วนำรายการที่ปรากฏขึ้นที่ทรานแซกชัน t_1 ไปคำนวณ NIWS ของแต่ละรายการ และอัตรา (ร้อยละ) ค่าความสม่ำเสมอ แล้วอัปเดตลงไปในลิสต์ 1List แสดงดังภาพที่ 4.7 และอ่านฐานข้อมูลรายการ TDB_1 จนครบทุกทรานแซกชัน แสดงดังภาพที่ 4.8*

i_k	$r^{i_k}_{TDB_1}$	$r^{i_k}_{TDB_2}$	c^{i_k}	$NIWS^{i_k}_{TDB_1}$	$NIWS^{i_k}_{TDB_2}$
a	2.5	-	-	{ <1, {128}> }	-
b	2.5	-	-	{ <1, {128}> }	-
c	2.5	-	-	{ <1, {128}> }	-
d	2.5	-	-	{ <1, {128}> }	-
e	-	-	-	-	-
f	-	-	-	-	-
g	-	-	-	-	-
h	-	-	-	-	-

ภาพที่ 4.7 ลิสต์ *1List* หลังจากอ่านทรานแซกชัน t_1 ของฐานข้อมูลรายการ TDB_1

i_k	$r^{i_k}_{TDB_1}$	$r^{i_k}_{TDB_2}$	c^{i_k}	$NIWS^{i_k}_{TDB_1}$	$NIWS^{i_k}_{TDB_2}$
a	37.50	-	-	{ <1, {252, 0, 15, 128}>}	-
b	27.50	-	-	{ <1, {250, 62, 143, 128, 16}> }	-
c	47.50	-	-	{ <1, {192, 0, 12}> }	-
d	15.00	-	-	{ <1, {241, 193, 127, 127, 239}> }	-
e	17.50	-	-	{ <1, {46, 191, 162, 232, 18}> }	-
f	22.50	-	-	{ <2, {128, 48, 15, 67}> }	-
g	-	-	-	-	-
h	-	-	-	-	-

ภาพที่ 4.8 ลิสต์ *1List* หลังจากอ่านครบทรานแซกชันของฐานข้อมูลรายการ TDB_1

จากนั้นอ่านฐานข้อมูลรายการ TDB_2 แล้วอัปเดต NIWS และอัตรา (ร้อยละ) ค่าความสม่ำเสมอลงไปในลิสต์ 1List แล้วทำการตรวจสอบรายการ ถ้ารายการใดมีอัตรา (ร้อยละ) ค่าความสม่ำเสมอ มากกว่า 60% หรือรายการใดไม่ปรากฏขึ้นทั้งสองฐานข้อมูลรายการ ก็จะถูกลบออกจากลิสต์ 1List แต่ถ้ารายการใดมีอัตรา (ร้อยละ) ค่าความสม่ำเสมอน้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอ และมีอัตราการเปลี่ยนแปลงมากกว่าค่าขีดแบ่งการเปลี่ยนแปลง ก็จะเก็บรายการนั้นเป็นผลลัพธ์ แสดงดังภาพที่ 4.9 และภาพที่ 4.10

i_k	$r^{i_k}_{TDB_1}$	$r^{i_k}_{TDB_2}$	c^{i_k}	$NIWS^{i_k}_{TDB_1}$	$NIWS^{i_k}_{TDB_2}$
a	37.50	50.00	1.33	{ <1, {252, 0, 15, 128}> }	{ <1, {255, 191}>, <6, {31, 255}> }
b	27.50	5.36	0.19	{ <1, {250, 62, 143, 128, 16}> }	{ <1, 254, 95, 255, 255, 255, 229, 254}> }
~~c~~	~~47.50~~	~~62.50~~	~~1.32~~	~~{ <1, {192, 0, 12}> }~~	~~{ <1, {192, 12}>, <7, {192}> }~~
~~d~~	~~15.00~~	~~-~~	~~-~~	~~{ <1, {241, 193, 127, 127, 239}> }~~	~~-~~
e	17.50	8.93	0.51	{ <1, {46, 191, 162, 232, 18}> }	{ <1, {31, 241, 255, 255, 240, 251, 31}> }
~~f~~	~~22.50~~	~~-~~	~~-~~	~~{ <2, {128, 48, 15, 67}> }~~	~~-~~
~~g~~	~~-~~	~~62.50~~	~~-~~	~~-~~	~~{ <1, {14, 48}>, <6, {3, 14}> }~~
~~h~~	~~-~~	~~21.43~~	~~-~~	~~-~~	~~{ <2, {16, 43, 46, 49, 225}> }~~

ภาพที่ 4.9 ลิสต์ *1List* จากฐานข้อมูลรายการ TDB_1 และ TDB_2

i_k	$r^{i_k}_{TDB_1}$	$r^{i_k}_{TDB_2}$	c^{i_k}	$NIWS^{i_k}_{TDB_1}$	$NIWS^{i_k}_{TDB_2}$
a	37.50	50.00	1.33	{ <1, {252, 0, 15, 128}> }	{ <1, {255, 191}>, <6, {31, 255}> }
b	27.50	5.36	0.19	{ <1, {250, 62, 143, 128, 16}> }	{ <1, 254, 95, 255, 255, 255, 229, 254}> }
e	17.50	8.93	0.51	{ <1, {46, 191, 162, 232, 18}> }	{ <1, {31, 241, 255, 255, 240, 251, 31}> }

ภาพที่ 4.10 ลิสต์ *1List* หลังจากลบรายการ *'c', 'd', 'f', 'g'* และ *'h'*

ขั้นตอนวิธีต่อมาคือ การสร้างเซตรายการขนาด 2 รายการ โดยพิจารณาจากแต่ละรายการในลิสต์ 1List ได้แก่ รายการ 'a' รวมกับรายการ 'b' จะได้เซตรายการ 'ab' แล้วทำการคำนวณ $NIWS^{ab}_{TDB_1}$ และ $NIWS^{ab}_{TDB_2}$ จากนั้นคำนวณอัตรา (ร้อยละ) ค่าความสม่ำเสมอทั้งสองฐานข้อมูลรายการ แล้วทำการตรวจสอบถ้าอัตรา (ร้อยละ) ค่าความสม่ำเสมอน้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอ ก็จะถูกจัดเก็บไปยังลิสต์ 2List จากนั้นคำนวณอัตราการเปลี่ยนแปลง ถ้ามีค่ามากกว่าค่าขีดแบ่งการเปลี่ยนแปลงจะจัดเก็บเซตรายการนั้นเป็นผลลัพธ์ จากนั้นทำซ้ำขั้นตอนวิธีการสร้างเซตรายการขนาด 2 รายการ โดยทำการรวมระหว่าง รายการ 'a' กับรายการ 'e' และทำการรวมรายการ 'b' กับรายการ 'e' สุดท้ายในขั้นตอนวิธีการสร้างเซตรายการขนาด 2 รายการ จะมีเซตรายการที่มีความสม่ำเสมอขนาด 2 รายการ ดังนี้ 'ab' , 'ae' และ 'be' แสดงดังภาพที่ 4.11

i_k	$r^{i_k}_{TDB_1}$	$r^{i_k}_{TDB_2}$	c^{i_k}	$NIWS^{i_k}_{TDB_1}$	$NIWS^{i_k}_{TDB_2}$
ab	40.00	53.57	1.34	{ <1, {248, 0, 15, 128}> }	{ <1, {254, 31}>, <6, {5, 254}> }
ae	42.50	50.00	1.18	{ <1, {44, 0, 2, 128}> }	{ <1, {31, 177}>, <6, {27, 31}> }
be	27.50	8.93	0.32	{ <1, {42, 62, 130, 128, 16}> }	{ <1, {30, 81, 255, 255, 240, 225, 30}> }

ภาพที่ 4.11 ลิสต์ *2List* จากลิสต์ *1List*

ขั้นตอนสุดท้าย คือ การค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้นตั้งแต่ขนาด 3 รายการขึ้นไป โดยเริ่มต้นจะพิจารณารายการ 'ab' และรวมกับรายการอื่น ๆ ที่มีรายการก่อนหน้าเหมือนกัน (หมายเหตุ ยกเว้นรายการสุดท้าย) อาทิเช่น เซตรายการ 'ab' รวมกับเซตรายการ 'ae' แล้วรายการ 'b' ที่ซึ่งเป็นรายการสุดท้ายของเซตรายการ 'ab' และรายการ 'e' ที่ซึ่งเป็นรายการสุดท้ายของเซตรายการ 'ae' เมื่อนำมารวมกันจะได้เซตรายการ 'be' จากนั้นนำเซตรายการ 'be' ไปตรวจสอบในลิสต์ 2List ว่ามีเซตรายการ 'be' อยู่หรือไม่ ถ้ามีก็ทำการรวมเซตรายการ 'ab' กับเซตรายการ 'ae' ได้เป็นเซตรายการ 'abe' จากนั้นทำการคำนวณ NIWS และอัตรา (ร้อยละ) ค่าความสม่ำเสมอ ทั้งสองฐานข้อมูลรายการ แล้วทำการตรวจสอบ ถ้าอัตรา (ร้อยละ) ค่าความสม่ำเสมอน้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอ ก็จะถูกจัดเก็บไปยังลิสต์ 3List จากนั้นคำนวณอัตราการเปลี่ยนแปลง ถ้ามีค่ามากกว่าค่าขีดแบ่งการเปลี่ยนแปลงก็จะจัดเก็บเป็นผลลัพธ์ แสดงดังภาพที่ 4.12 เมื่อไม่สามารถคำนวณได้แล้วจึงหยุดการพิจารณา และผลลัพธ์ทั้งหมดของการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏ แสดงดังภาพ 4.13

i_k	$r^{i_k}_{TDB_1}$	$r^{i_k}_{TDB_2}$	c^{i_k}	$NIWS^{i_k}_{TDB_1}$	$NIWS^{i_k}_{TDB_2}$
abe	45.00	57.14	1.27	{ <1, {40, 0, 2, 128}> }	{ <1, {30, 17}>, <6, {1, 30}> }

ภาพที่ 4.12 ลิสต์ *3List* จากลิสต์ *2List*

i_k	$r^{i_k}_{TDB_1}$	$r^{i_k}_{TDB_2}$	c^{i_k}	$NIWS^{i_k}_{TDB_1}$	$NIWS^{i_k}_{TDB_2}$
a	37.50	50.00	1.33	{ <1, {252, 0, 15, 128}> }	{ <1, {255, 191}>, <6, {31, 255}> }
ab	40.00	53.57	1.34	{ <1, {248, 0, 15, 128}> }	{ <1, {254, 31}>, <6, {5, 254}> }
abe	45.00	57.14	1.27	{ <1, {40, 0, 2, 128}> }	{ <1, {30, 17}>, <6, {1, 30}> }

ภาพที่ 4.13 ผลลัพธ์ทั้งหมดของการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏ

4.3 การวิเคราะห์ประสิทธิภาพของขั้นตอนวิธีริคครอม

ในงานวิจัยนี้ได้วิเคราะห์ความซับซ้อนของการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏขึ้นด้วยขั้นตอนวิธีริคครอม ใน 2 รูปแบบ คือ การวิเคราะห์ความซับซ้อนของเวลาที่ใช้ในการประมวลผลข้อมูล และการวิเคราะห์พื้นที่หน่วยความจำที่ใช้ในจัดเก็บข้อมูลสูงสุด

ข้อเสนอที่ 4.3.1 การวิเคราะห์ความซับซ้อนของเวลาที่ใช้ในการประมวลผลข้อมูลในการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏขึ้นด้วยขั้นตอนวิธีริคครอม คือ $O((n(m+v)) + (n) + (p^2) + (2^p 2(j + k)))$ โดยที่ 1) n คือ จำนวนของรายการทั้งหมดในเซต / 2) m คือ จำนวนทรานแซกชันทั้งหมดในฐานข้อมูลรายการ TDB_1 3) v คือ จำนวนทรานแซกชันทั้งหมดในฐานข้อมูลรายการ TDB_2 4) p คือ จำนวนรายการหลังจากที่ทำได้ทำการลดทอนข้อมูล 5) j คือ จำนวนทรานแซกชันทั้งหมดในฐานข้อมูลรายการ TDB_1 หารด้วย 8 (m/8) เนื่องจากได้จัดเก็บข้อมูลในรูปแบบบิต และ 6) k คือ จำนวนทรานแซกชันทั้งหมดในฐานข้อมูลรายการ TDB_2 หารด้วย 8 (v/8)

พิสูจน์ข้อเสนอที่ 4.3.1 การค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏขึ้นด้วยขั้นตอนวิธีริคครอม เริ่มต้นจากการอ่านฐานข้อมูลรายการของทุกรายการ n ในทุกทรานแซกชัน m ของ TDB_1 จะใช้เวลาเป็น $n \times m$ และ TDB_2 อ่านฐานข้อมูลรายการของทุกรายการ n ในทุกทรานแซกชัน v ของ TDB_2 จะใช้เวลาเป็น $n \times v$ ดังนั้นจะใช้เวลาสำหรับการอ่านฐานข้อมูลรายการเป็น $nm + nv$ เท่ากับ $n(m+v)$ หลังจากนั้นอ่านทุกรายการ n สำหรับการค้นหาเซตรายการขนาด 1 รายการและลดทอนข้อมูลรายการที่มีอัตรา (ร้อยละ) ค่าความสม่ำเสมอมากกว่าค่าขีดแบ่งความสม่ำเสมอ และลดทอนข้อมูลรายการที่ปรากฏขึ้นเพียงฐานข้อมูลรายการ TDB_1 หรือ TDB_2 แสดงเป็น p ต่อมาทำการค้นหาเซตรายการขนาด 2 รายการจากรายการที่ผ่านการพิจารณา รวมกับรายการอื่น ๆ ใช้เวลาเป็น p^2 และในส่วนของการค้นหาเซตรายการตั้งแต่ขนาด 3 รายการขึ้นไป จะมีเซตรายการที่นำไปรวมกับเซตรายการอื่น ๆ และที่เป็นไปได้ทั้งหมด คือ 2^p แล้วทำการอินเตอร์เซกชันเซตของหมายเลขทรานแซกชัน เพื่อให้ได้ทรานแซกชันที่ปรากฏขึ้นร่วมกันของเซตรายการนั้นๆ ของฐานข้อมูลรายการ TDB_1 ได้เป็น $j+j$ และฐานข้อมูลรายการ TDB_2 ได้เป็น $k+k$ ดังนั้น จะใช้เวลาในการค้นหาเซตรายการตั้งแต่ขนาด 3 รายการ

ขึ้นไปเป็น $2^p \times (2j + 2k)$ เท่ากับ $2^p \times 2(j + k)$ และจะได้ความซับซ้อนของเวลาที่ใช้ในการประมวลผลข้อมูลทั้งหมดของการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏขึ้นด้วยขั้นตอนวิธีริคครอม คือ $O((n(m+v) + (n) + (p^2) + (2^p 2(j + k)))$

ข้อเสนอที่ 4.3.2 การวิเคราะห์ความซับซ้อนของพื้นที่หน่วยความจำในการจัดเก็บข้อมูลของการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏขึ้นด้วยขั้นตอนวิธีริคครอม คือ $O((n(j+k)) + 2(p^2(j+k))$ โดยที่ 1) n คือ จำนวนของรายการทั้งหมดในเซต I 2) m คือ จำนวนทรานแซกชันทั้งหมดในฐานข้อมูลรายการ TDB_1 3) v คือ จำนวนทรานแซกชันทั้งหมดในฐานข้อมูลรายการ TDB_2 4) p คือ จำนวนรายการหลังจากที่ทำได้ทำการลดทอนข้อมูล 5) j คือ จำนวนทรานแซกชันทั้งหมดในฐานข้อมูลรายการ TDB_1 หารด้วย 8 (m/8) เนื่องจากได้จัดเก็บข้อมูลในรูปแบบบิต และ 6) k คือ จำนวนทรานแซกชันทั้งหมดในฐานข้อมูลรายการ TDB_2 หารด้วย 8 (v/8)

พิสูจน์ข้อเสนอที่ 4.3.2 พื้นที่หน่วยความจำในการจัดเก็บข้อมูลของการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏขึ้นด้วยขั้นตอนวิธีริคครอม โดยเริ่มแรกจะทำการอ่านฐานข้อมูลรายการ TDB_1 ในแต่ละรายการของแต่ละทรานแซกชันที่ซึ่งจัดเก็บข้อมูลในรูปแบบบิต ดังนั้นจะใช้พื้นที่หน่วยความจำในการจัดเก็บข้อมูลเป็น $n \times j$ ต่อมาอ่านฐานข้อมูลรายการ TDB_2 ในแต่ละรายการของแต่ละทรานแซกชัน ดังนั้นจะใช้พื้นที่หน่วยความจำในการจัดเก็บข้อมูลเป็น $n \times k$ ดังนั้นในขั้นตอนวิธีของการอ่านฐานข้อมูลรายการทั้งสองฐานข้อมูลรายการจะใช้พื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูล คือ $nj + nk$ เท่ากับ $n(j+k)$ จากนั้นทำการอ่านข้อมูลแต่ละรายการสำหรับการค้นหารายการขนาด 1 รายการ และการลดทอนข้อมูลที่มีค่าความสม่ำเสมอมากกว่าค่าขีดแบ่งความสม่ำเสมอ และ/หรือรายการนั้น ปรากฏแค่เพียงฐานข้อมูลรายการ TDB_1 หรือฐานข้อมูลรายการ TDB_2 ที่ซึ่งทำให้ได้รายการหลังการลดทอนข้อมูลแสดงเป็น p กับข้อมูลหมายเลขทรานแซกชันที่ปรากฏขึ้นของเซตรายการนั้นๆ ของทั้งสองฐานข้อมูลรายการ คือ $j + k$ ดังนั้นในขั้นตอนวิธีการสร้างเซตรายการขนาด 2 รายการจะใช้พื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูล คือ $p^2(j+k)$ จากนั้นในส่วนของการค้นหาเซตรายการตั้งแต่ขนาด 3 รายการขึ้นไป โดยจะพิจารณาแต่ละเซตรายการที่มีรายการก่อนหน้าที่เหมือนกัน แล้วนำไปรวมกับเซตรายการอื่น ๆ และแต่ละเซตรายการจะมีข้อมูลทรานแซกชันที่ปรากฏขึ้นของเซตรายการนั้นๆ ทั้งสองฐานข้อมูลรายการ ดังนั้น จะใช้พื้นที่หน่วยความจำสูงสุดในขั้นตอนวิธีนี้ คือ $(p-2)p(j+k)$ ดังนั้นจะได้ความซับซ้อนของพื้นที่หน่วยความจำในการจัดเก็บข้อมูลของการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏขึ้นด้วยขั้นตอนวิธีริคครอม คือ $O((n(j+k)) + p^2(j+k) + (p-2)p(j+k)) = O((n(j+k)) + 2(p^2(j+k))$

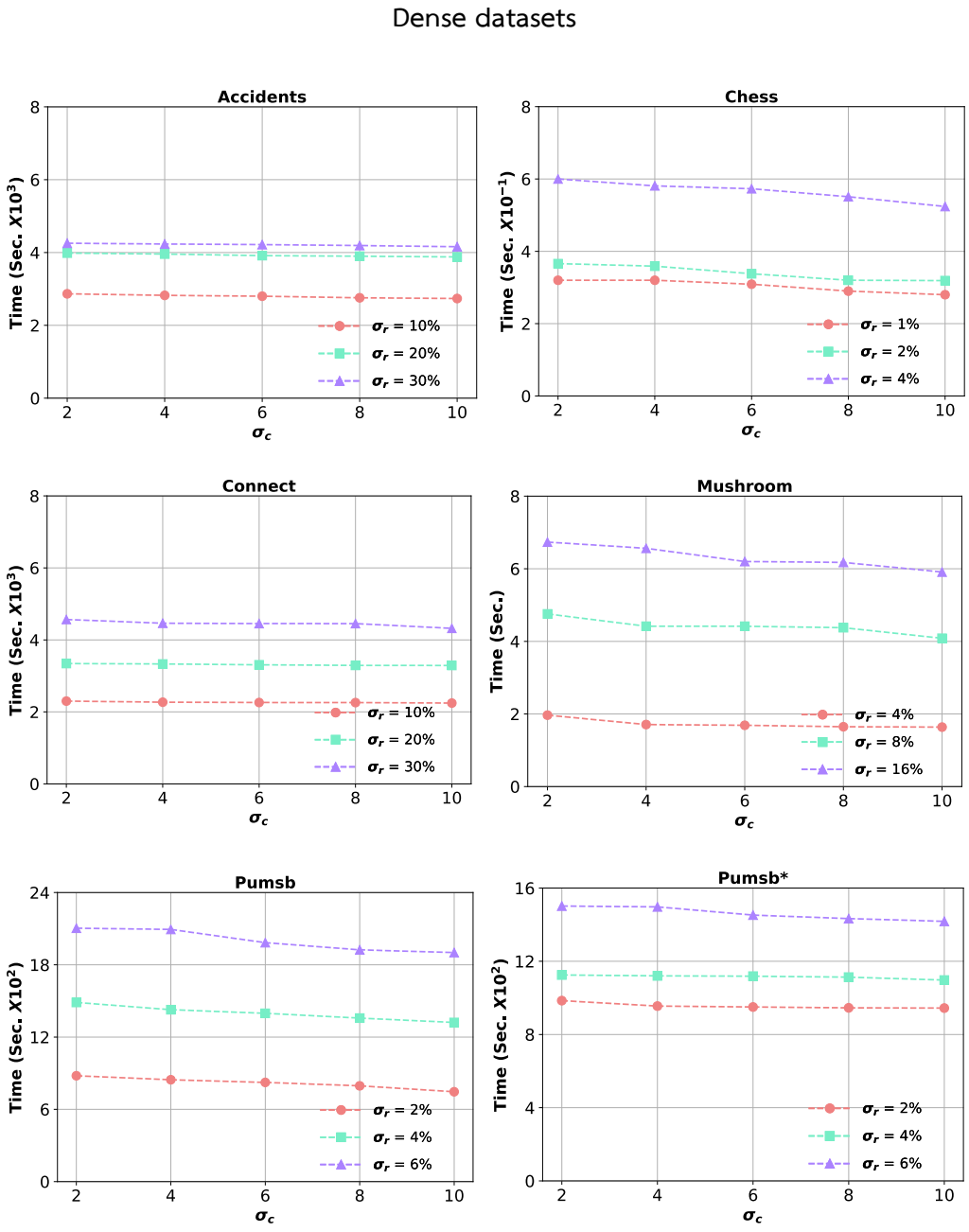
4.4 ผลการทดลอง

จากคุณลักษณะของฐานข้อมูลรายการที่ใช้ในการทดลอง ดังที่กล่าวในบทที่ 2 สามารถแบ่งฐานข้อมูลรายการได้ 2 ลักษณะข้อมูล ได้แก่ 1) แบบหนาแน่น (Dense) มีทั้งหมด 6 ฐานข้อมูลรายการ ได้แก่ Accidents, Chess, Connect, Mushroom, Pumsb และ Pumsb* 2) แบบเบาบาง (Sparse) มีทั้งหมด 4 ฐานข้อมูลรายการ ได้แก่ Kosarak, Retail, T10I4D100K และ T40I10D100K โดยในงานวิจัยนี้ได้แบ่งครึ่งฐานข้อมูลรายการออกเป็นสองส่วนเท่า ๆ กัน (ฐานข้อมูลรายการ TDB_1 และฐานข้อมูลรายการ TDB_2) และได้ใช้ Python 3.5.1 ด้วยโปรแกรม Pycharm บนเครื่องคอมพิวเตอร์ที่มีความเร็ว CPU 2.40 GHz, RAM 8 GB และ Windows 10

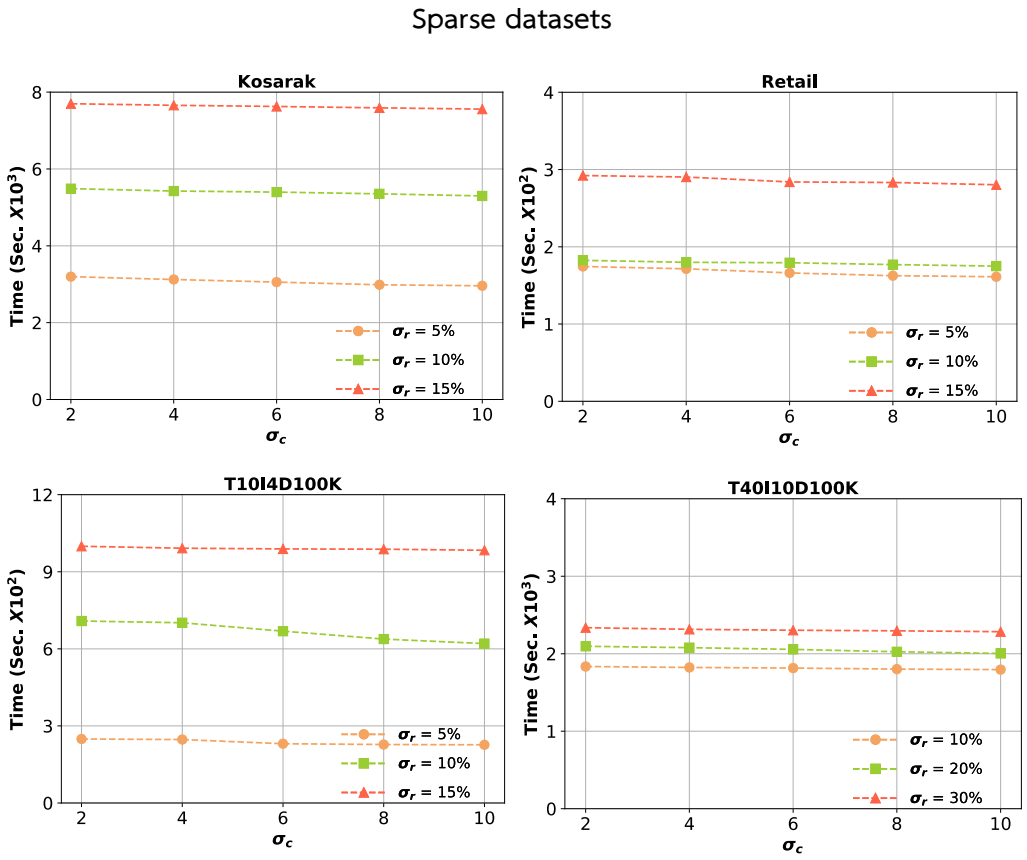
สำหรับการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏขึ้นด้วยขั้นตอนวิธีริคครอม โดยการพิจารณาเซตรายการที่ปรากฏสม่ำเสมอภายใต้ค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด และแนวโน้มความเปลี่ยนแปลงของพฤติกรรมการปรากฏขึ้นอย่างสม่ำเสมอที่เพิ่มขึ้นเมื่อเวลาผ่านไปภายใต้ค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนด ที่ซึ่งกำหนดค่าขีดแบ่งการเปลี่ยนแปลงเริ่มตั้งแต่ 2 ถึง 10 และผลลัพธ์จะพิจารณาใน 3 แง่มุม ดังต่อไปนี้

4.4.1 เวลาที่ใช้ในการประมวลผล

ภาพที่ 4.14 และ ภาพที่ 4.15 แสดงผลการทดลองการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏขึ้นด้วยขั้นตอนวิธีริคครอมในด้านเวลาที่ใช้ในการประมวลผลข้อมูลของฐานข้อมูลรายการที่มีลักษณะข้อมูลหนาแน่นและเบาบางด้วย ที่ซึ่งสังเกตได้ว่าฐานข้อมูลรายการที่มีลักษณะข้อมูลหนาแน่นจะใช้เวลาน้อยกว่าฐานข้อมูลรายการที่มีลักษณะข้อมูลเบาบาง แต่อย่างไรก็ตาม เวลาที่ใช้ในการประมวลผลข้อมูลจำขึ้นอยู่กับขนาดของฐานข้อมูลรายการและจำนวนทรานแซกชันด้วย นอกจากนี้เมื่อค่าขีดแบ่งความสม่ำเสมอของการปรากฏขึ้นมีค่ามากขึ้น จะทำให้เวลาที่ใช้ในการประมวลผลมากขึ้นตามไปด้วย



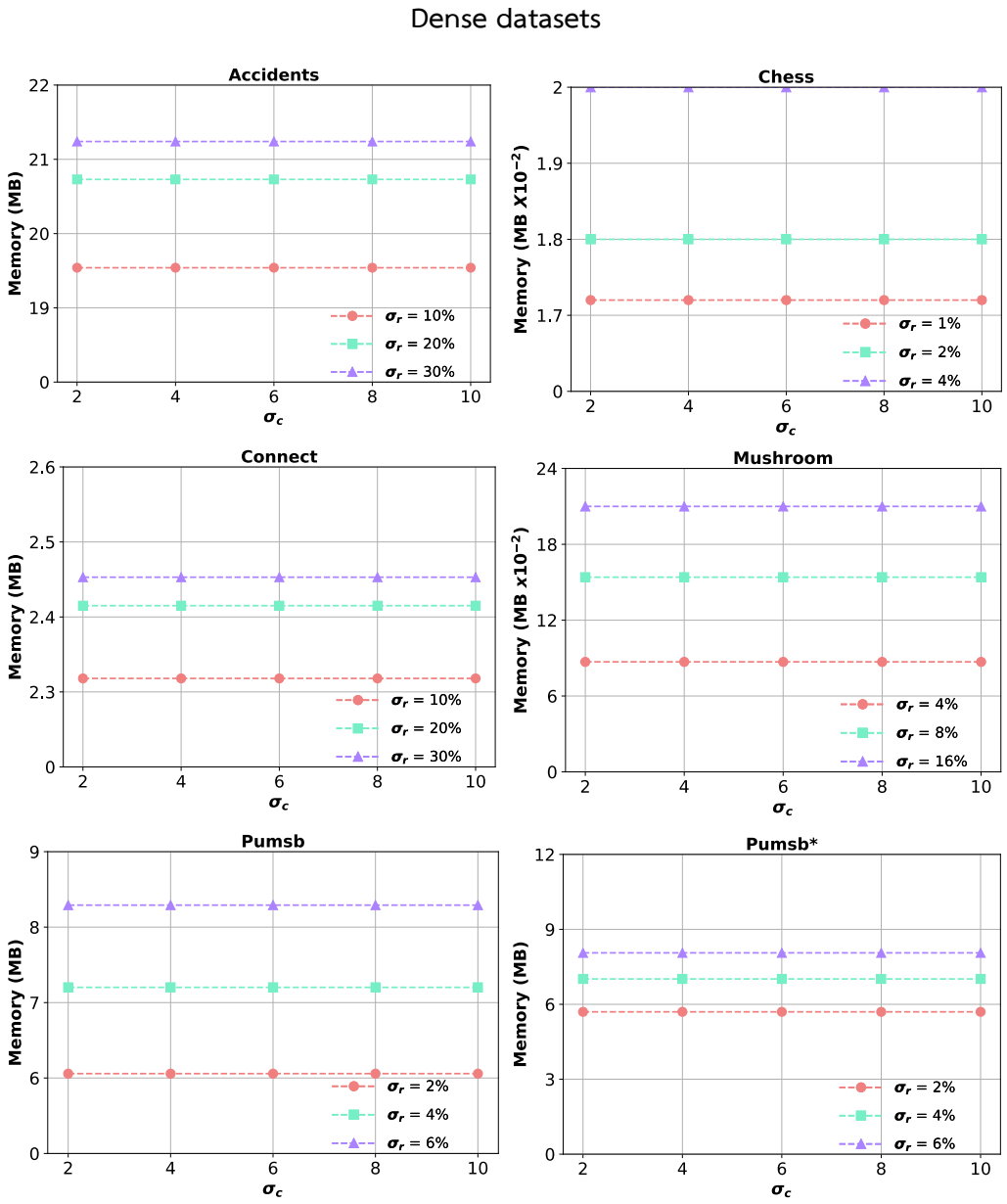
ภาพที่ 4.14 ผลการทดลองการค้นหาเซตรายการด้วยขั้นตอนวิธีริคครอมในด้านเวลาที่ใช้ในการประมวลผลข้อมูลของฐานข้อมูลรายการที่มีลักษณะข้อมูลหนาแน่น



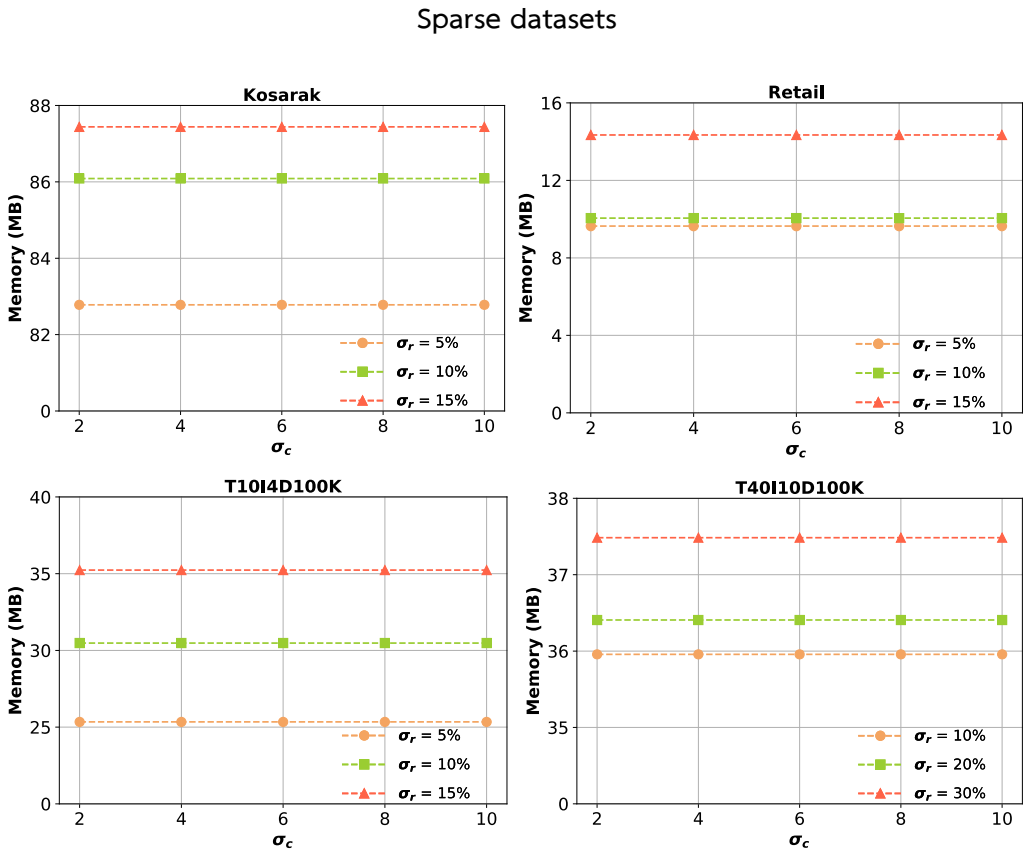
ภาพที่ 4.15 ผลการทดลองการค้นหาเซตรายการด้วยขั้นตอนวิธีริคครอมในด้านเวลาที่ใช้ในการประมวลผลข้อมูลของฐานข้อมูลรายการที่มีลักษณะข้อมูลเบาบาง

4.4.2 พื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูล

ภาพที่ 4.16 และ ภาพที่ 4.17 แสดงผลการทดลองการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏขึ้นด้วยขั้นตอนวิธีริคครอมในด้านพื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลของฐานข้อมูลรายการที่มีลักษณะข้อมูลหนาแน่นและเบาบาง ที่ซึ่งสังเกตได้ว่าฐานข้อมูลรายการที่มีลักษณะข้อมูลหนาแน่นจะใช้พื้นที่หน่วยความจำที่ใช้จัดเก็บข้อมูลน้อยกว่าฐานข้อมูลรายการที่มีลักษณะข้อมูลเบาบาง และสังเกตได้ว่าเมื่อค่าขีดแบ่งอัตราการเปลี่ยนแปลงมีค่าเพิ่มขึ้นจะไม่มีผลต่อพื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลด้วย



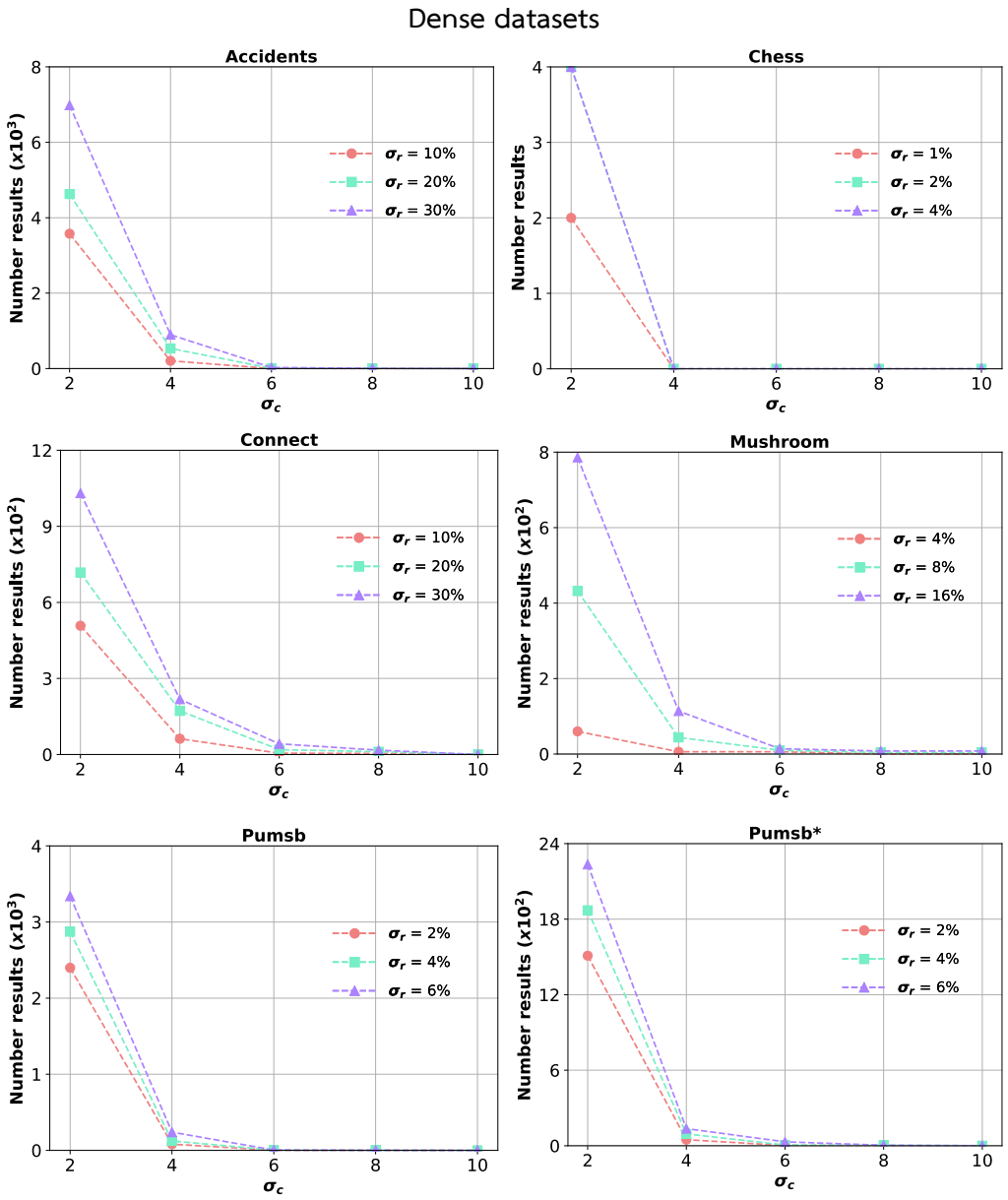
ภาพที่ 4.16 ผลการทดลองการค้นหาเซตรายการด้วยขั้นตอนวิธีริคครอมในด้านพื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลของฐานข้อมูลรายการที่มีลักษณะข้อมูลหนาแน่น



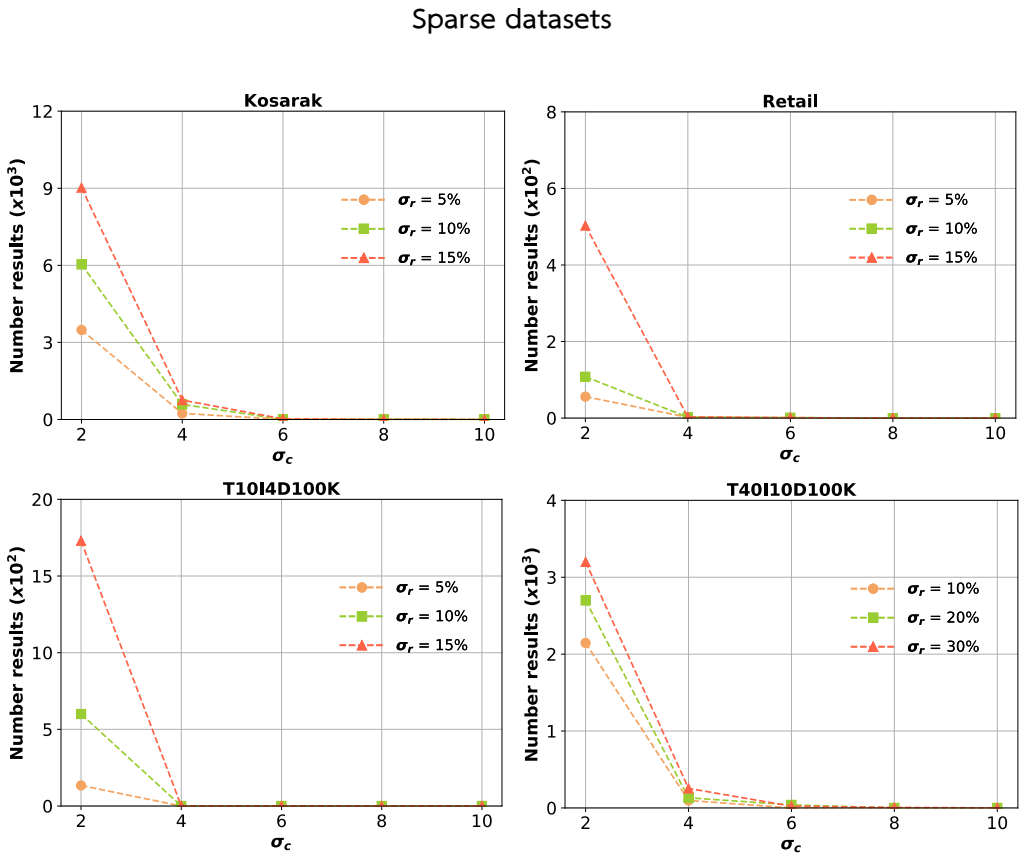
ภาพที่ 4.17 ผลการทดลองการค้นหาเซตรายการด้วยขั้นตอนวิธีริคครอมในด้านพื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลของฐานข้อมูลรายการที่มีลักษณะข้อมูลเบาบาง

4.4.3 จำนวนผลลัพธ์เซตรายการที่ค้นพบ

ภาพที่ 4.18 และ ภาพที่ 4.19 แสดงผลการทดลองการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอในการปรากฏขึ้นด้วยขั้นตอนวิธีในด้านจำนวนผลลัพธ์เซตรายการที่ค้นพบของฐานข้อมูลรายการที่มีลักษณะข้อมูลหนาแน่นและเบาบาง โดยถ้าค่าขีดแบ่งความสม่ำเสมอมีค่ามากขึ้นก็จะทำให้ได้ผลลัพธ์ที่มากขึ้น แต่เมื่อค่าขีดแบ่งการเปลี่ยนแปลงมีค่ามากขึ้นจะทำให้ผลลัพธ์น้อยลง



ภาพที่ 4.18 ผลการทดลองการค้นหาเซตรายการด้วยขั้นตอนวิธีริคครอมในด้านจำนวนผลลัพธ์เซตรายการที่ค้นพบของฐานข้อมูลรายการที่มีลักษณะข้อมูลหนาแน่น



ภาพที่ 4.19 ผลการทดลองการค้นหาเซตรายการด้วยขั้นตอนวิธีริคครอมในด้านจำนวนผลลัพธ์เซตรายการที่ค้นพบของฐานข้อมูลรายการที่มีลักษณะข้อมูลเบาบาง

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิทยานิพนธ์ฉบับนี้ได้นำเสนอการวิเคราะห์พฤติกรรมผู้บริโภคด้วยการวิเคราะห์ความเปลี่ยนแปลงของพฤติกรรมการซื้อสินค้า โดยได้มีหลายงานวิจัยนำเสนอการค้นหาเซตรายการภายใต้การวัดความน่าสนใจในแง่มุมต่าง ๆ ที่ซึ่งมีงานวิจัยหนึ่งที่สนใจเกี่ยวกับความเปลี่ยนแปลงของการปรากฏขึ้นในแง่มุมของความถี่ที่เพิ่มขึ้นเมื่อเวลาผ่านไป แต่อย่างไรก็ตาม การค้นหาเซตรายการภายใต้การเปลี่ยนแปลงลักษณะของการปรากฏอย่างสม่ำเสมอก็เป็นอีกแง่มุมที่น่าสนใจเช่นกัน ที่ซึ่งสามารถช่วยให้ทราบถึงพฤติกรรมการซื้อสินค้าที่ปรากฏขึ้นอย่างสม่ำเสมอ ที่จะช่วยให้บริษัทจะได้รับข้อมูลเพื่อใช้ประกอบการตัดสินใจ ที่ซึ่งจากการทราบถึงข้อมูลดังกล่าวจะทำให้ผู้บริหารสามารถคิดกลยุทธ์เพื่อสามารถกระตุ้นการซื้อสินค้าของผู้บริโภคอันนำมาซึ่งผลประกอบการบริษัทที่เพิ่มขึ้นได้ และสามารถนำมาประยุกต์ใช้งานได้ในหลาย ๆ แอปพลิเคชัน อาทิเช่น การเปลี่ยนแปลงค่าความสม่ำเสมอของการซื้อสินค้าในร้านค้าปลีก การเปลี่ยนแปลงค่าความสม่ำเสมอของผลกระทบหลังจากที่ผู้ป่วยมีการใช้ยา การเปลี่ยนแปลงค่าความสม่ำเสมอของเกณฑ์สำหรับการจองโรงแรมของนักท่องเที่ยวและอื่น ๆ

ดังนั้น ในงานวิทยานิพนธ์นี้จึงได้นำเสนอการค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นด้วยการพิจารณาความเปลี่ยนแปลงของพฤติกรรมการปรากฏขึ้นในแง่มุมความสม่ำเสมอที่เพิ่มขึ้นเมื่อเวลาผ่านไปภายใต้ค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนดด้วยขั้นตอนวิธีไมโคร ดังที่กล่าวในบทที่ 3 ที่ซึ่งประยุกต์ใช้โครงสร้างต้นไม้ที่เรียกว่า อิคโคร-ทรี ที่ช่วยให้อ่านข้อมูลจากฐานข้อมูลรายการเพียงครั้งเดียวเท่านั้น และยังมีการลดทอนข้อมูลจากสมบัติปิดการลดลง เพื่อลดเวลาในการประมวลผลข้อมูลและลดพื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลได้อย่างมีประสิทธิภาพ อย่างไรก็ตาม ขั้นตอนวิธีนี้ ได้มีสร้างผลลัพธ์เป็นจำนวนมาก ทำให้ผู้ใช้หรือผู้ที่สนใจไม่สามารถนำข้อมูลไปใช้งานได้หรือวิเคราะห์ได้ และผลลัพธ์ที่ได้ก็อาจจะไม่น่าสนใจหรือบ่งบอกถึงข้อมูลที่สำคัญได้

ด้วยเหตุนี้ จึงได้นำเสนอการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้น ด้วยการพิจารณาเซตรายการที่ปรากฏสม่ำเสมอภายใต้ค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนดและแนวโน้มความเปลี่ยนแปลงของพฤติกรรมการปรากฏขึ้นอย่างสม่ำเสมอที่เพิ่มขึ้นเมื่อเวลาผ่านไปภายใต้ค่าขีดแบ่งการเปลี่ยนแปลงที่ผู้ใช้กำหนดด้วยขั้นตอนวิธีริคครอม ดังที่กล่าวในบทที่ 4 ที่ซึ่งประยุกต์ใช้โครงสร้างข้อมูลที่เรียกว่า *NIWS* สำหรับจัดเก็บข้อมูลที่ปรากฏขึ้นของแต่ละรายการ/เซตรายการในกระบวนการของการค้นหาเซตรายการได้อย่างมีประสิทธิภาพ ที่ซึ่งผลการทดลองได้ใช้ข้อมูลที่เป็นข้อมูลจริงและข้อมูลที่ถูกสังเคราะห์ขึ้น โดยผลการทดลองแสดงให้เห็นว่าการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้นด้วยขั้นตอนวิธีริคครอมมีประสิทธิภาพในด้านของเวลาที่ใช้ในการประมวลผลข้อมูล พื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลและจำนวนผลลัพธ์เซตรายการที่ค้นพบ

5.2 ปัญหาและข้อจำกัดที่พบจากการวิจัย

จากผลการทดลองการค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นด้วยขั้นตอนวิธีไมโคร พบว่าเมื่อฐานข้อมูลรายการมีขนาดใหญ่จะทำให้ใช้เวลาในการประมวลผลและพื้นที่หน่วยความจำเป็นจำนวนมาก ด้วยเหตุนี้ ทำให้ขั้นตอนวิธีนี้ไม่สามารถทำการทดลองได้ทุกฐานข้อมูลรายการ เนื่องจากเครื่องคอมพิวเตอร์ที่ใช้ในการทดลองมีหน่วยความจำไม่เพียงพอต่อความต้องการของขั้นตอนวิธีดังกล่าว

นอกจากนี้ จากผลการทดลองการค้นหาเซตรายการทั้งการค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นด้วยขั้นตอนวิธีไมโครและการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้นด้วยขั้นตอนวิธีริคครอม พบว่า ในขั้นตอนวิธีการหนึ่ง ๆ ไม่เหมาะสมกับทุกฐานข้อมูลรายการ อาทิเช่น ขั้นตอนวิธีริคครอมอาจใช้เวลาในการประมวลข้อมูลที่น้อยแต่ใช้พื้นที่หน่วยความจำที่ใช้ในการจัดเก็บข้อมูลที่มากในบางฐานข้อมูลรายการ

5.3 ข้อเสนอแนะ

ในงานวิทยานิพนธ์นี้ ได้นำเสนอการค้นหาเซตรายการที่น่าสนใจภายใต้การเปลี่ยนแปลงค่าความสม่ำเสมอของการปรากฏขึ้นด้วยขั้นตอนวิธีไมโครและการค้นหาเซตรายการที่ปรากฏสม่ำเสมอภายใต้การเปลี่ยนแปลงที่น่าสนใจของค่าความสม่ำเสมอที่ปรากฏขึ้นด้วยขั้นตอนวิธีริคครอม ที่ซึ่งสามารถนำขั้นตอนวิธีดังกล่าวไปพัฒนาหรือปรับปรุงให้มีประสิทธิภาพมากยิ่งขึ้น และสามารถนำไปประยุกต์ใช้ในการตัดสินใจเชิงธุรกิจ ทางการแพทย์ และอื่น ๆ ต่อไปได้

บรรณานุกรม

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, (pp. 487–499).

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International conference on Management of data*, (pp. 207–216).

Amphawan, K., & Lenca, P. (2015). Mining top-k frequent-regular closed patterns. *Expert Systems with Applications*, *42*(21), 7882-7894.

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*, (pp. 18–29).

Chen, M. -C., Chiu, A. -L., & Chang, H. -H. (2006). Mining changes in customer behavior in retail marketing. *Expert Systems with Applications*, *28*(4), 773–781.

Coquin, L., Canipa, S. J., Drewe, W. C., Fisk, L., Gillet, V. J., Patel, M., Plante, J., Sherhod, R. J., & Vessey, J. D. (2015). New structural alerts for Ames mutagenicity discovered using emerging pattern mining techniques. *Toxicology Research*, *4*(1), 46-56.

Dong, G., & Li, J. (1999). Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* (pp. 43-52).

Dong, G., & Li, J. (2005). Mining border descriptions of emerging patterns from dataset pairs. *Knowledge and Information Systems, 8*(2), 178-202.

Dong, J., & Han, M. (2007) Bittable-FI An efficient mining frequent itemsets algorithm. *Knowledge-Based Systems*, *20*(4), 329–335.

Fang, G., & Deng, Z. –H., & Ma, H. (2009). Network Traffic Monitoring Based on Mining Frequent Patterns. In *Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery,* (pp. 571 – 575).

García-Vico, A. M., Montes, J., Aguilera, J., Carmona, C. J., & del Jesus, M. J. (2016). Analysing concentrating photovoltaics technology through the use of emerging pattern mining. In *International Joint Conference on SOCO'16-CISIS'16-ICEUTE'16*, (pp. 334-344).

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data. (*pp. 1-12).

Han, J., Wang, J., Lu, Y., & Tzvetkov P. (2002). Mining top-k frequent closed patterns without minimum support. In *Proceedings of the IEEE International Conference on Data Mining*, (pp. 211-218).

Huang, Z., Gan, C., Lu, X., & Huan, H. (2013). Mining the changes of medical behaviors for clinical pathways. *Studies in Health Technology and Informatics*, *192*(12), 117-121.

Kim, J. K., Song H. S., & Kim, H. K. (2005). Detecting the change of customer behavior based on decision tree analysis. *Expert Systems*, *22*(4), 193-205.

Li, G., Law, R., Vu, H. Q., Rong, J., & Zhao, X. R. (2015). Identifying emerging hotel preferences using emerging pattern mining technique. *Tourism Management*, *46*, 311-321.

Li, J., & Wong, L. (2002). Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns. *Bioinformatics*, *18*(10), 1406-1407.

Liu, B., Hsu, W., Han, H. -S., & Xia, Y. (2000). Mining changes for real-life applications. In *Proceedings of the 2nd International Conference on Data Warehousing and Knowledge Discovery,* (pp. 337-346).

Liu, W., Zheng, Y., Chawla, S., Yuan, J., & Xing, X. (2011). Discovering spatio-temporal causal interactions in traffic data streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 1010-1018).

M. Shah, H., & Kaur, N. (2014). Improve frequent pattern mining in data stream. *Engineering & Technology*, *2*(5), 143-152.

Mozafari, B., Thakkar, H., & Zaniolo, C. (2008). Verifying and mining frequent patterns from large windows over data streams. In *Proceedings of the IEEE 24th International Conference on Data Engineering*, (pp. 179-188).

Muzammal, M., & Raman, R. (2011). Mining sequential patterns from probabilistic databases. *Knowledge and Information Systems, 44(2)*, 325-358.

Neubarth, K., & Conklin, D. (2016). Contrast pattern mining in folk music analysis. In *International Conference on Computational Music Analysis*, (pp. 393-424).

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.

Sallaberry, A., Pecheur, N., Bringay, S., Roche, M. & Teisseire, M. (2011). Sequential patterns mining and gene sequence visualization to discover novelty from microarray data. *Biomedical Informatics*, *44*(5), 760-774.

Shih, M. -J., Liu, D.-R., & Hsu, M. -L., (2010). Discovering competitive intelligence by mining changes in patent trends. *Expert Systems with Applications*, *37*(4), 2882–2890.

Sim, K., Li, J., Gopalkrishnan, V., & Liu, G. (2009). Mining maximal quasi-bicliques: novel algorithm and applications in the stock market and protein networks. *Statistical Analysis and Data Mining, 2*(4), 255-273.

Song, H. S., Kim, J. K., & Kim, S. H. (2001). Mining the change of customer behavior in an internet shopping mall. *Expert Systems with Applications, 21*(3), 157-168.

Tanbeer, S. K., Ahmed, C. F., & Jeong, B. S. (2010). Mining regular patterns in data streams. In *proceedings of the 15th International Conference on Database Systems for Advanced Applications,* (pp. 399-413).

Tanbeer, S. K., Ahmed, C. F., & Jeong, B. S. (2010). Mining Regular Patterns in Incremental Transactional Databases. In *Proceedings of the 12th International Asia-Pacific Web Conference,* (pp. 375-377).

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*, (pp. 242-253).

Tanbeer, S. K., Hassan, M. M., Alrubaian, M., & Jeong, B. S. (2015). Mining Regularities in Body Sensor Network Data. In *International Conference on Internet and Distributed Computing Systems*, (pp. 88-99).

Tsai, C. Y., & Shieh, Y. C. (2009). A change detection method for sequential patterns. *Decision Support Systems*, *46*(2), 501-511.

Vo, B., Hong, T. -P., & Le, B. (2012). DBV-Miner: a dynamic bit-vector approach for fast mining frequent closed itemsets. *Expert Systems with Applications, 39*(8), 7196-7206.

Wang, G., Zhao, Y., Zhao, X., Wang, B., & Qiao, B. (2010). Efficient mining local conserved cluster from gene expression data. *Neurocomputing*, *73*(7), 1425-1437.

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

ภาคผนวก

ภาคผนวก ก

เอกสารรับรองผลการพิจารณาจริยธรรมการวิจัยในมนุษย์

ที่ ๔๙/๒๕๖๐

เอกสารรับรองผลการพิจารณาจริยธรรมการวิจัยในมนุษย์
มหาวิทยาลัยบูรพา

คณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์ มหาวิทยาลัยบูรพา ได้พิจารณาโครงการวิจัย

รหัสโครงการวิจัย Sci 013/2560
โครงการวิจัยเรื่อง การวิเคราะห์พฤติกรรมผู้บริโภคด้วยการวิเคราะห์ความเปลี่ยนแปลงของพฤติกรรมการซื้อสินค้า
หัวหน้าโครงการวิจัย นางสาวสุมาลี อิสริโยดม
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

เอกสารเผยแพร่ผลงานวิจัย

Discovering interesting itemsets based on change in regularity of occurrence

Sumalee Eisariyodom*, Komate Amphawan†
Computational Innovation Laboratory, Faculty of Informatics, Burapha University, Chonburi, 20131, Thailand
Email: *ann.eisariyodom@gmail.com, †komate@gmail.com

***Abstract*—Mining interesting itemsets/patterns is presented and utilized in a wide range of applications. Organizations and businesses have applied this to observe/track/monitor significant occurrence behavior of objects or events. Currently, with the emergence of new technologies, people may change their needs/behaviors in daily life. Thus, analysis of change on occurrence behavior of objects (or events) can be an important issue in several domains. In this paper, we propose to mine interesting itemsets based on change in regularity of occurrence (called *ICROs*) to capture change on behavior from actions performed by people. A single-pass algorithm, called *MICRO*, and a tree structure named *ICRO-tree* are designed to efficiently mine *ICROs*. Moreover, a pruning strategy is devised to cut-down search space, computation time and memory consumption. Experiments were done to investigate the performance of *MICRO* and to show efficiency of *MICRO* on runtime, memory usage and the number of discovered *ICROs*.**

***Keywords*—data mining; association rules; change detection; mining interesting itemsets; change on regularity of occurrence.**

I. Introduction

Mining frequent itemsets from transactional database is currently applied in a wide range of applications (such as retail-cross marketing, restaurants management, mobile commerce analysis, etc) that require to analyze behaviors of important people. Since proposed by Agrawal et al. [1], there are several extensions on frequent itemsets mining (*FIM*) in various aspects *e.g.* mining frequent itemsets without candidate generation [2], mining top-k frequent itemsets [3], mining frequent itemsets from incremental/data streams, mining weighted frequent itemsets. Moreover, Tanbeer et al. [4] proposed to observe occurrence behavior in the terms of frequency and regularity of occurrence by mining frequent-regular itemsets. This kind of itemsets can help to know about whether an itemset frequently, regularly and/or irregularly occur in database. Unfortunately, although the above mining approaches can discover valuable itemsets which have help for analyzing and taking actions, however, these approaches cannot identify significant changes in behavior which can help to gain a new kind of knowledge.

To address the above issue, Dong et al. [5] firstly proposed to discover emerging patterns (*EPs*) which can help to know trends and differences on occurrences of itemsets in the term of frequency. Since the first proposed of Dong et al., the task of mining *EPs* attracts a lot of extensions and is extended in various aspects. For example, internet shopping mall and retails tried to find change in customer behavior to help market analysts having better understanding on changes in customer needs and how those needs change and to establish effective promotion campaigns [6], [7]; manufacturing industry identified changes in patent trends to improve the organization competitiveness [8]; hotel business applied mining emerging patterns to observe change in travelers preferences which can help to gain insights into the behavior of travelers [9]. However, mining *EPs* only considers changing on frequency of occurrence of itemsets which may not sufficient to express change on regularity or irregularity of itemsets in some applications such as retail businesses, medical analysis, stock market, sensor networks, network monitoring, telecommunication, web site design and administration, web-click stream analysis, genetic data analysis, traffic data analysis, analyze comments on e-commerce, etc.

To solve the above limitation, we here propose a new approach to mine interesting itemsets based on their change on regularity of occurrence (also called mining *ICROs*) which can help to know change on behavior from actions performed by people in various applications. For example, in manufacturing and retail business, tracking change/fluctuation on purchasing on products of partners/customers may help to manage production process, manage inventory, create marketing strategies, design catalogs, etc; in medical analysis, finding change on effect of using medicines of patients based on regularity of effect can help to find solutions to alleviate severe pains of patients and mining changes in blood pressure of a patient can be useful information for doctors to provide proper treatment to a particular patient [10] and so on.

To mine *ICROs*, an efficient single-pass algorithm based on pattern-growth concept, called *MICRO* (*Mining interesting Itemsets based on Change on Regularity of Occurrence*). A FP-tree like structure is designed to efficiently maintain candidate itemsets during mining process. Moreover, we also devise a pruning strategy to cut-down search space and to reduce resource usage. Experiments on synthetic and real datasets were done to investigate performance of the proposed *MICRO* algorithm on computational time, memory usage and number of discovered *ICROs* and the results demonstrate that *MICRO* can efficiently discover interesting itemsets based on their change on regularity of occurrence from various databases.

II. Problem Statement

In this section, we first describe the concept of regularity of occurrence (as in [11], [12]) . Then, notations related to change on regularity of occurrence of an itemset and problem statement are thus introduced.

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

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

A transactional database 1

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

A transactional database 2

Fig. 1: A both transactional databases

Let $I = \{i_1, i_2, \dots i_n\}$ be a set of distinct items. A set $X = \{i_j, i_k, ..., i_l\}$ where $1 \leq j \leq k \leq l \leq n$ is called a pattern (also called an itemset or a k-itemset if X contains k items (*i.e.* $|X| = k$)). Given a transaction database $DB = \{t_1, t_2, \dots, t_m\}$ contains m transactions in which each transaction t_j has a unique transaction-identifier (called *tid* in short) and contains a set of items $Y = \{i_p, i_q, \dots, i_r\}$ (where $1 \leq p \leq q \leq r \leq n$) expressing objects/events that occur in the transaction. If an itemset $X \subseteq Y$, it can be said that X occurs in transaction t_j or t_j contains X. For an itemset X, $T^X = \{t_j, \dots t_k\}$ is the ordered set of transactions containing X. Then, the support of X, the ratio between transactions containing X and all transactions of database *DB*, can be defined as $s^X = \frac{|T^X|}{|DB|}$. To calculate regularity, Amphawan et al. [11] simplified basic definitions and notations as follows. A regularity of an itemset X based on its occurrence in transaction t_k can be computed in three cases: (*i*) if transaction t_k is the first transaction containing X, the regularity of X based on t_k can be defined as $r^X_{t_k} = k$ which express the number of transactions from the begining of database until the first occurrence of X, (*ii*) if X occurs in t_k after t_j, then $r^X_{t_k} = k - j$ *i.e.* the gap of occurrence of X between transaction t_j and t_k and (*iii*) if transaction t_k is the last transaction in *DB* that contains X, $r^X_{t_k}$ is calculated as $r^X_{t_k} = |DB| + 1 - k$ indicating the gap of occurrence between t_k and the last transaction of *DB*, respectively. Based on all occurrence of X in T^X, the regularity of occurrence of X can be computed as $r^X = max(r^X_{t_j}, r^X_{t_k}, \dots, r^X_{t_l})$ which is the maximum gap of repeated occurrence from all occurrences of X. The regularity of X can make a confirmation that X will occur at least once in every consecutive r^X transactions.

Example 1: From Fig 1, let's consider item 'd' in database DB_1 with its occurrence as $T^d = \{t_1, t_2, t_3, t_4, t_7, t_8, t_9, t_{10}\}$. Then, the regularity of item 'd' in DB_1 can be computed as $r^d_{DB_1} = max(r^d_{t_1}, r^d_{t_2}, r^d_{t_3}, r^d_{t_4}, r^d_{t_7}, r^d_{t_8}, r^d_{t_9}, r^d_{t_{10}}) = max(1, 1, 1, 1, 3, 1, 1, 1, 0) = 3$ (Notice that $r^d_{t_{10}}$ has two values because they can be computed as case (ii) and case (iii) as above). The regularity ($r^d_{DB_1} = 3$) lets us know that item 'd' occurs at least once in every three consecutive transactions of DB_1.

Thus, we then proposed to consider itemsets with significant increasing on regular of occurrence under user-given change-ratio on regularity threshold σ_{cr}.

Lets DB_1 and DB_2 be transactional databases at time t_1 and t_2. The change-ratio on regularity of occurrence of itemset X is the ratio of increasing on regularity of X in DB_1 and DB_2 which can be defined in the three cases as follow:

$$cr^X = \begin{cases} undefined & \text{, if X is not occur in } DB_1 \text{ and } DB_2 \\ 0 & \text{, if X is not occur only in } DB_2; \\ \frac{r^X_{DB_2}}{r^X_{DB_1}} & \text{, otherwise} \end{cases} \quad (1)$$

From eq. 1, change-ratio on regularity of occurrence of an itemset X can be simplified as follow:

- if the itemset X is not occur in both DB_1 and DB_2, *i.e.* $s^X_{DB_1}$ and $s^X_{DB_2}$ are equal to 0, its change-ratio on regularity of occurrence cannot be defined.
- if the itemset X occurs only in DB_1, *i.e.* $s^X_{DB_1} > 0$ but $s^X_{DB_2} = 0$, its change-ratio on regularity of occurrence can be computed as $\frac{0}{r^X_{DB_1}} = 0$.
- if the itemset X only occurs in DB_2 or in both DB_1 and DB_2 which can be separated into two cases: (i) if X only occurs in DB_2 which causes $r^X_{DB_1} = |DB_1| + 1$, then the change-ratio of X on regularity of occurrence can be expresses as $\frac{r^X_{DB_2}}{|DB_1|+1}$, otherwise, (ii) the change-ratio of X on regularity of occurrence can be defined as $\frac{r^X_{DB_2}}{r^X_{DB_1}}$, *i.e.* $r^X_{DB_1}$ and $s^X_{DB_2}$ are greater than 0, its change-ratio on regularity of occurrence can be defined as $\frac{r^X_{DB_2}}{r^X_{DB_1}}$.

Therefore, based on the change-ratio on regularity of X (cr^X), it can be said that X is an interesting itemset (pattern) if cr^X is no less than the user-given change-ratio on regularity threshold (σ_{cr}).

Example 2: As in Fig 1, the change-ratio on regularity of item 'b' can be computed as $cr^b = \frac{r^b_{DB_2}}{r^b_{DB_1}} = \frac{3}{2} = 1.5$. If the change-ratio threshold is set to be 1.25 (which means that the regularity of any itemset in DB_2 should be no less than its regularity in DB_1; the itemset is an interesting itemset if it has the maximum gap of occurrence in DB_2 larger than that of DB_1), then item 'b' is an interesting itemset which can be identified as a result.

Problem statement: Given two databases DB_1 and DB_2 and a change-ratio threshold σ_{cr}, the task of mining interesting itemsets based on change of regularity of occurrence is to find itemsets that occur in both databases and have change-ratio on regularity between DB_1 and DB_2 no less than σ_{cr}.

Property 1: For an itemset X, if X does not occur in DB_1 or DB_2, X and all supersets of X cannot be interesting itemsets based on their change on regularity of occurrence.

Proof: From problem statement, an interesting itemset should occur in both databases and has change-ratio on regularity between DB_1 and DB_2 no less than σ_{cr}. Then, based on *downward closure property* [13], if an itemset X is not occur in DB_1, all of its supersets also do not occur in DB_1 as well

(also the same for DB_2). Thus, X and all of X's supersets cannot be interesting itemsets. ■

III. Proposed Method

In this section, we here describe a tree-based structure named *ICRO-tree* used for maintaining candidate itemsets with their essential information. Then, an efficient single-pass algorithm based on pattern-growth concept named *MICRO* (*Mining interesting Itemsets based on their Change on Regularity of Occurrence*) is described in details. Last, an example of *MICRO* algorithm is given to simplify the task of mining a complete set of *ICROs*.

A. ICRO-tree structure

ICRO-tree is a *FP-tree* like structure with a header table used for maintaining candidate itemsets and single items during mining process. Each path P of *ICRO-tree* expresses an itemset X in which its occurrence information in DB_1 and DB_2 are stored only at the node of the last item (*i.e.* storing only at the leaf node). Each node in path P contains only one item in itemset X which the node can be classified into two types: (*i*) *internal node* containing item-name (i_k), links to parent and child nodes in the same path and a link to another node with the same item name, and (*ii*) *leaf node* (or *tail-node*) containing the same information as *internal node*, but it also contains $T^{i_k}_{DB_1}$ and $T^{i_k}_{DB_2}$ *i.e.* the occurrence information of itemset X in database DB_1 and DB_2. Meanwhile, the header table (a simple list) contains five information (see Fig. 2) that are (*i*) item-name (i_k), (*ii*) regularity of item i_k in database DB_1 ($r^{i_k}_{DB_1}$), (*iii*) regularity of item i_k in database DB_2 ($r^{i_k}_{DB_2}$), (*iv*) the change-ratio on regularity of item i_k (cr^{i_k}) and (*v*) a link to all nodes with item i_k, called *node-link* (l^{i_k}), respectively. With *ICRO-tree*, *MICRO* can apply the pattern-growth concept to efficiently mine interesting itemsets based on their change on regularity of occurrence.

B. MICRO algorithm

MICRO algorithm is a pattern-growth based where it scans database only once to reduce I/O cost. *MICRO* consists of two main steps *i.e.* (*i*) *ICRO-tree* construction (algo. 1), scaning database DB_1 and DB_2 once to capture essential information from the databases into *ICRO-tree* and (*ii*) *ICROs*-growth (algo. 2), pattern-growing to find a complete set of *ICROs*.

***ICRO-tree* construction** is the process of creating *ICRO-tree* by capturing candidate itemsets with their occurrence information through a scanning databases DB_1 and DB_2.

Firstly, *ICRO-tree* is initialized with a root node R and a header table for all items is created. Then, each transaction in DB_1 and DB_2 is scanned. To read each transaction $t_j = \{i_k, \ldots, i_l\}$ of database DB_1 (line $2-11$), a $tempNode$ (*i.e.* a buffer pointer used for checking whether there is a path in *ICRO-tree* for items in transaction t_j) is set to point to the root node R. Then, each item i_k in t_j is considered. The regularity $r^{i_k}_{DB_1}$ at the entry of item i_k in the header table is thus updated by *tid* j of the transaction t_j. Next, a node of item i_k is created and linked with the *node-link* of i_k in the header table, if there is none of a path in *ICRO-tree* that linked item i_k with the former items in t_j. Otherwise, the buffer pointer ($tempNode$) is moved to the node of i_k. After considered all items in transaction t_j, the buffer pointer $tempNode$ is now currently pointed to the node of the last item i_l of t_j. Then, the set of occurrence information $T^{i_l}_{DB_1}$ in the node of i_l is updated by adding *tid* j of transaction t_j (*i.e.* $T^{i_l}_{DB_1} \leftarrow T^{i_l}_{DB_1} \cup j$) in order to collect occurrence information of all items occurring in transaction t_j.

Next, database DB_2 is then read in the same manner as scanning of DB_1 in which the step of updating regularity of each item i_k in a transaction t_j (line 5) is changed from $r^{i_k}_{DB_1}$ to be $r^{i_k}_{DB_2}$, and the step of adding *tid* j of transaction t_j in the set of occurrence information $T^{i_l}_{DB_1}$ of the node of the last item i_l (line 11) is also changed to be storing into $T^{i_l}_{DB_2}$, respectively.

Last, the regularity $r^{i_k}_{DB_1}$ and $r^{i_k}_{DB_2}$ of each item i_k contained in the header table are then considered (line $13-19$). If $r^{i_k}_{DB_1} > |DB_1|$ and/or $r^{i_k}_{DB_2} > |DB_2|$ (*i.e.* this means that i_k does not occur in DB_1 and/or DB_2), then all nodes of item i_k are then eliminated from *ICRO-tree* and its child nodes are then linked with its parent nodes (see Eq. 1 and Property 1 for pruning property). Otherwise, the change-ratio on regularity of occurrence of i_k (cr^{i_k}) is computed and i_k is thus identified as an interesting with significant change on regularity of occurrence (also collected in the set of *ICROs*), if cr^{i_k} is no less than the user-given change-ratio threshold σ_{cr}.

***ICROs*-growth** is a recursive process on *ICRO-tree* where each iteration considers each item i_k and itemsets that occur together with i_k. The main computation in each iteration of *ICROs*-growth can be separated into two cases as follow.

- if the *ICRO-tree* contains only one path (also called single path) P with itemset Q (line $2-7$), then the change-ratio on regularity of occurrence of itemset Q (cr^Q) is calculated. If cr^Q is no less than the user-given σ_{cr} threshold, each sub-itemset β of itemset Q is generated and merged with the previous considered itemset X generated from previous iterations (*i.e.* $\beta \cup X$). Last, the itemset $\beta \cup X$ is then output by storing in the *ICROs* set.
- if the current considered *ICRO-tree* contains more than one path (line $8-29$), each item i_k in the header table (start from the last to the second one) is considered. The item i_k is then marked as considered by collecting i_k in the set of already considered itemset X and a simple-list named $iList_{i_k}$ is created for maintaining items occurring with i_k and previous considered items (line $10-11$). Next, each node n_{i_k} of item i_k that linked with the *node-link* of i_k in header table is thus traversed, each ancestor item i_l (*i.e.* the item (node) in the *ICRO-tree* located higher than item i_k) in the same paths as n_{i_k} is taken into account and then the set of occurrence information, $T^{i_l}_{DB_1}$ and $T^{i_l}_{DB_2}$, at the entry $iList^{i_l}_{i_k}$ of $iList_{i_k}$ are then updated by $T^{i_l}_{DB_1}$ and $T^{i_l}_{DB_2}$ of the node n_{i_k} (line $12-14$). After regarding all nodes n_{i_k} of item i_k and their ancestor items, each entry $iList^{i_l}_{i_k}$ of item i_l in $iList_{i_k}$ is then considered and checked whether i_l occurs together with itemset X (line $15-21$). If the item i_l occurs together with X in both database, $cr^{i_l \cup X}$ is computed from $T^{i_l}_{DB_1}$ and

Algorithm 1: *ICRO-tree* construction

Input: DB_1, DB_2, σ_{cr}

Output: *ICRO-tree*: a tree-structure contains sets of items occurring in transactions of DB_1 and DB_2, *ICROs*: a set of interesting items with significant change on regularity of occurrence

1: initial *ICRO-tree* with a root node as R, and initial a header table of all items
2: **for** each transaction t_j in database DB_1 **do**
3: set $tempNode$ as R
4: **for** each item i_k in transaction t_j **do**
5: update $r^{i_k}_{DB_1}$ at the entry of item i_k in the header table by considering *tid* j of t_j
6: **if** there is no a child node of $tempNode$ with item i_k **then**
7: create a new node for item i_k, set it to be a child node of $tempNode$ and link it with the header table
8: $tempNode \leftarrow$ the new node of item i_k
9: **else**
10: $tempNode \leftarrow$ a child node of $tempNode$ with item i_k
11: add *tid* j of t_j into $T^{i_k}_{DB_1}$ of $tempNode$ *i.e.* $T^{i_k}_{DB_1} \leftarrow T^{i_k}_{DB_1} \cup j$
12: read each transaction of DB_2 in the same manner as scanning of DB_1
13: **for** each item i_k in the header table **do**
14: **if** i_k does not occur in DB_1 or DB_2 **then**
15: remove all nodes of item i_k out of *ICRO-tree*
16: **else**
17: compute cr^{i_k} by $\frac{r^{i_k}_{DB_2}}{r^{i_k}_{DB_1}}$
18: **if** $cr^{i_k} \geq \sigma_{cr}$ **then**
19: collect i_k in *ICROs* as an interesting item with significant change on regularity of occurrence

Algorithm 2: *ICROs*-growth

Input: *ICRO-tree*, σ_{cr}

Output: *ICROs*: a complete set of interesting itemsets with significant change on regularity of occurrence

1: **Procedure**: *ICROs*-growth (*ICRO-tree* with a root node R, X, σ_{cr}) \\Note that X is a set of considered items from previous iterations (X is $\emptyset$ at the first mining)
2: **if** *ICRO-tree* contains only single path P **then**
3: compute cr^Q of itemsets Q of path P
4: **if** $cr^Q \geq \sigma_{cr}$ **then**
5: **for** each combination of items in path P (abbreviated as β) **do**
6: merge β with X and then compute $cr^{\beta \cup X}$ from T_{DB_1} and T_{DB_2} of the leaf node of path P
7: collect $\beta \cup X$ in *ICROs* as an interesting itemsets with significant change on regularity of occurrence
8: **else**
9: **for** each item i_k in the header table of *ICRO-tree* (start from the last to the second one) **do**
10: $X \leftarrow X \cup i_k$ \\collect i_k to be a member of already considered itemset
11: create and initial an item-list called $iList_{i_k}$ for maintaining all items (with their occurrence information) occurring together with item i_k
12: **for** each node n_{i_k} linked in *node-link* of item i_k **do**
13: **for** each item i_l located in the same vertical path as node n_{i_k} **do**
14: \\Note that $iList^{i_l}_{i_k}$ be an entry of item i_l in $iList_{i_k}$, merge $T^{i_l}_{DB_1}$ in the node n_{i_k} with $T^{i_k}_{DB_1}$ in $iList^{i_l}_{i_k}$ and merge $T^{i_l}_{DB_2}$ in the node n_{i_k} with $T^{i_k}_{DB_2}$ in $iList^{i_l}_{i_k}$
15: **for** each item i_l in $iList_{i_k}$ **do**
16: **if** i_l does not occur with X in DB_1 or DB_2 **then**
17: remove the entry of i_l from $iList_{i_k}$
18: **else**
19: compute $cr^{i_l \cup X}$ from $T^{i_l}_{DB_1}$ and $T^{i_l}_{DB_2}$ of $iList^{i_l}_{i_k}$
20: **if** $cr^{i_l \cup X} \geq \sigma_{cr}$ **then**
21: collect $i_l \cup X$ in *ICROs* as an interesting itemsets with significant change on regularity of occurrence
22: **if** $|iList_{i_k}| > 1$ **then**
23: create and initial *ICRO-tree* with root node as Z and a header table for items in $iList_{i_k}$
24: **for** each node n_{i_k} linked in *node-link* of item i_k **do**
25: set Y as a set of items in the same vertical path as n_{i_k} where there is an entry of the item in $iList_{i_k}$
26: update *ICRO-tree* with root Z by Y and $T^{i_k}_{DB_1}$ (also $T^{i_k}_{DB_2}$) in the node n_{i_k}
27: \\Note that $n_{i_{k-1}}$ is the parent node of the node n_{i_k} with item i_{k-1},merge $T^{i_k}_{DB_1}$ with $T^{i_{k-1}}_{DB_1}$ and $T^{i_k}_{DB_2}$ with $T^{i_{k-1}}_{DB_2}$ and then remove n_{i_k} with all of its information
28: call *ICROs*-growth (*ICRO-tree* with Z as root, X, σ_{cr})
29: $X \leftarrow X - i_k$ \\remove i_k out of the set of already considered items, since all itemsets consisting of i_k are already considered and remove $iList_{i_k}$

$T^{i_l}_{DB_2}$ in $iList^{i_l}_{i_k}$ and after that $i_l \cup X$ is thus identified and collected in the *ICROs* set, if $cr^{i_l \cup X}$ is no less than the σ_{cr} threshold. Otherwise, the entry $iList^{i_l}_{i_k}$ of item i_l will be removed from our consideration and from $iList_{i_k}$. Next, a new *ICRO-tree* with a root node Z is initialized, if the number of entries contained in $iList_{i_k}$ is more than one (line $22-23$). Then, each node n_{i_k} is traversed again to collect a set of items Y that still have entries in $iList_{i_k}$. With itemset Y, a path for Y in the new *ICRO-tree* is updated and the occurrence information in the node n_{i_k} is merged with that of the last node in the path of Y. After update the new *ICRO-tree* with itemset Y, the node n_{i_k} is not necessary. Thus, the occurrence information $T^{i_k}_{DB_1}$ and $T^{i_k}_{DB_2}$ in n_{i_k} are moved and merged with that of the parent node of n_{i_k} and n_{i_k} with all of its information is removed from the *ICRO-tree* (line 27). After considered all nodes n_{i_k} of item i_k, there is none of node of i_k in the *ICRO-tree* and we gain the new *ICRO-tree* with all candidate itemsets that occur together with item i_k and previous considered itemsets regarded in previous iterations. Last, *ICROs*-growth is then repeated again on the new *ICRO-tree*.

C. Example of MICRO algorithm on databases

Let's consider transactional databases DB_1 and DB_2 in Fig. 1 and the change-ratio threshold σ_{cr} is set to be 1.25. Our proposed *MICRO* algorithm will find a complete set of *ICROs* having different on regularities from DB_1 to DB_2 at least 1.25 times.

MICRO firstly performs on *ICRO-tree* construction in which a header table H and the root node R of *ICRO-tree* are initialized. Next, the transaction $t_1 = \{a, b, c, d\}$ of DB_1 is scanned. A path of t_1 is thus created in *ICRO-tree* in which the leaf node 'd' contains *tid* 1 and entries of items 'a', 'b', 'c' and 'd' in H are updated (see Fig. 2). The scanning is repeated for all transactions in DB_1 and we get *ICRO-tree* as shown in Fig. 3. Next, each transaction in DB_2 is also read. For the transaction $t_1 = \{a, b, c\}$, the entry of 'a', 'b', and 'c' are updated and a path of t_1 is created or updated with *tid* 1 contained in the node of 'c' (see Fig. 4). The transactions $t_2 - t_{12}$ are also read in order to update H and *ICRO-tree* as in Fig. 5. As indicated in Fig. 6, items (with its' entry and nodes) that do not occur in DB_1 and/or DB_2 are removed from our consideration, since these items and all of its supersets cannot meet change-ratio threshold (thanks to Property 1). Last, the change-ratios of the remaining items are calculated and items with change-ratio no less than σ_{cr} are identified as results.

To mine longer itemsets, *MICRO*-growth is executed. The last item 'e' in H is considered and keep in mind. Then, each node of 'e' in *ICRO-tree* is traversed by its node link. First, with the path of 'a,b,e', items 'a' and 'b' are considered as they occur together with 'e' in transactions t_3 and t_5 of DB_1 and transactions t_4, t_5, t_6, t_7 and t_{12} of DB_2. Next, the path of 'a,e' is visited and item 'a' is regarded to occur with 'e' in $T_1 = \{3, 5, 6\}$ and $T_2 = \{4, 5, 6, 7, 8, 9, 11, 12\}$. After traversing on all paths with item 'e', we can recognize that items 'a' and

'b' occur with 'e' in both databases. Then, a conditional-tree with 'e' as a prefix is created (see Fig. 7). Then, the last item 'b' is also keep in mind and collect to be a prefix as 'e,b' and the nodes of 'b' in the conditional-tree with 'e' is traversed in the same manner to create the conditional-tree with prefix 'eb'. The traversing and computing is repeated until the current considered conditional-tree has only single path. After traversal of all trees, we gain all *ICRO*s contained in the set of results.

IV. Experiment Results

In this section, we report experimental studies on our proposed *MICRO* algorithm for mining a complete set of *ICRO*s. Three benchmark datasets downloaded from http://fimi.ua.ac.be/data/ are used in our experiments (as used in [4]). As shown in Table I, each dataset is split into 2 small equal-size databases. *T10I4D100K*, a synthetic dataset with $100,000$ transactions, is divided into two of $50,000$-transactions databases. Meanwhile, *Chess* and *Mushroom* are real datasets which are also split into two of $1,598$- and $4,062$-transactions databases, respectively. Three experiments on the three datasets under a change-ratio on regularity threshold (ranging from 2 to 10) were conducted for investigating computational time, memory usage and number of generated results. Lastly, *MICRO* is implemented on Python 3.5.1 and run on a machine with a CPU speed 2.40 GHz, RAM 8 GB, and Windows 10.

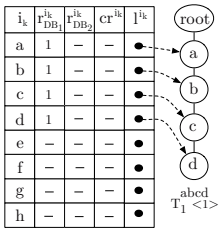


Fig. 2: *ICRO-tree* after reading t_1 of DB_1

The runtime of *MICRO* on the three datasets is shown in Fig. 9. From the figure, it can be observed that the runtime slightly decreases as the change-ratio on regularity threshold increases. The reason is that *MICRO* can prune items/itemsets which absence from DB_1 and/or DB_2 at the beginning of *ICRO-tree* construction process and the remaining items/itemsets are all considered in the *ICRO*s-growth. With the same amount of remaining items/itemsets (either the change-ratio on regularity threshold is high or low), the

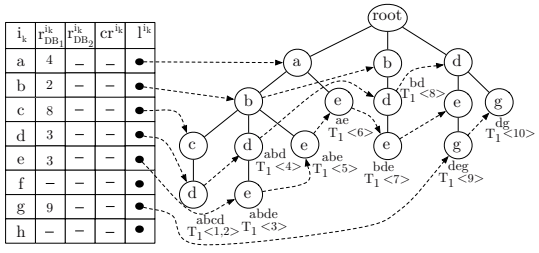


Fig. 3: *ICRO-tree* after reading all transactions of DB_1

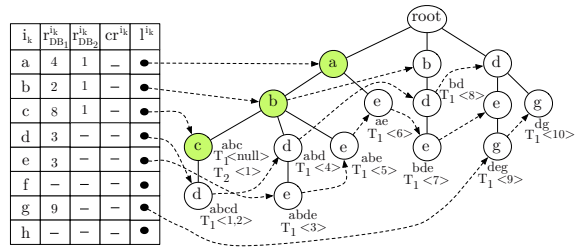


Fig. 4: *ICRO-tree* after reading t_1 of DB_2

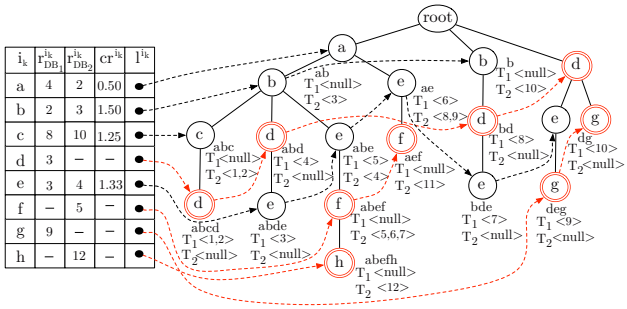


Fig. 5: *ICRO-tree* created from DB_1 and DB_2

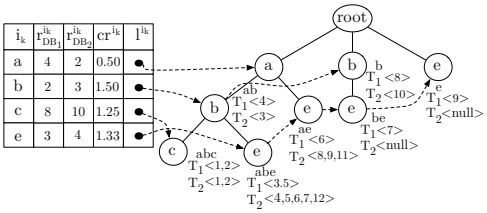


Fig. 6: *ICRO-tree* after remove items

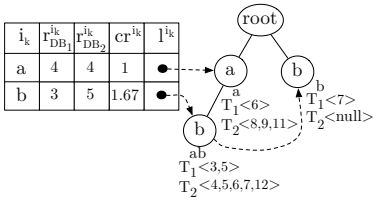


Fig. 7: Conditional-tree with 'e' as a prefix

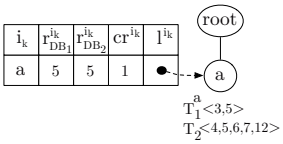


Fig. 8: conditional-tree with 'e,b' as a prefix

runtime is significantly different. In Fig. 10, the peak memory usage (*i.e.* the maximum amount of memory allocated by *MICRO* during mining process for maintaining all header

TABLE I: Data characteristics

Dataset	No. of items	Avg. transactions size	No. of DB_1	No. of DB_2
T10I4D100K	1,000	10	50,000	50,000
Chess	75	37	1,598	1,598
Mushroom	119	23	4,062	4,062

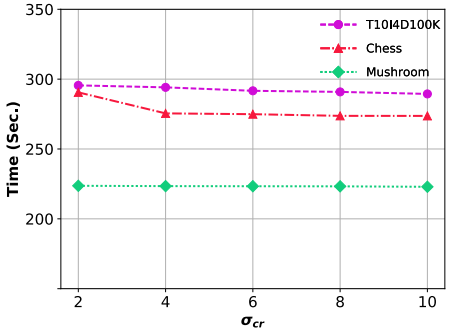


Fig. 9: Computational time of *MICRO*

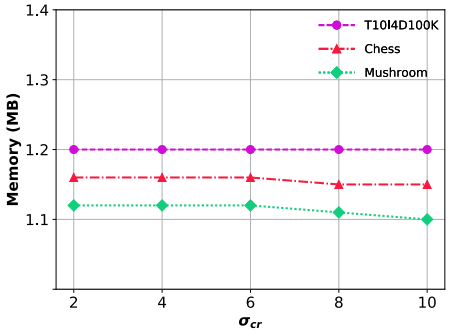


Fig. 10: Memory usage of *MICRO*

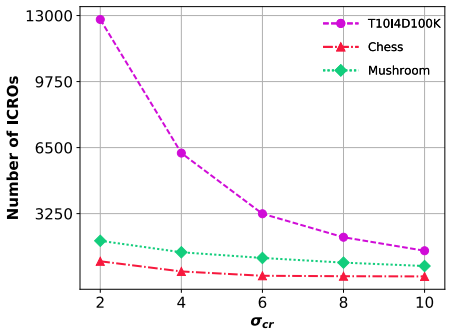


Fig. 11: Number of discovered *ICROs*

tables and *ICRO*-trees) is also observed. Similarly as above, *MICRO* uses nearly the same amount of memory on the low and high change-ratio on regularity threshold. Last, Fig. 11 shows the number of *ICRO*s generated from *MICRO*. With low change-ratio on regularity threshold, the number of *ICRO*s is numerous due to items/itemsets have more chance to satisfy the threshold. On the other hand, *MICRO* can generate fewer *ICRO*s on the high change-ratio on regularity threshold.

V. Conclusion and Future Work

In this paper, we have introduced a new approach to mine interesting itemsets based on their significant change on regularity of occurrence (also called *ICROs*). This kind of itemsets can occur in several real-life applications such as changing on regularity of purchasing in retails, changing on regularity of effect after using medicines of patients, changing on regularity of criteria for booking hotels of tourists, changing on regularity of reacting on banner advertisements and so on. From above applications, mining *ICROs* can also help to track, monitor and/or analyze changing on behavior of people that interact with organizations and businesses. To find such itemsets, we proposed an efficient single-pass algorithm based on pattern-growth concept named *MICRO*. A tree-based structure called *ICRO-tree* is also designed to efficiently maintain candidate itemsets with their essential information. A property used for pruning search space is also introduced in order to reduce resource usage during mining process. Experiments on real and synthetic datasets were done and the results demonstrate that our proposed algorithm with *ICRO-tree* structure is efficient on computational time, memory usage and number of discovered *ICROs* for mining interesting itemsets based on their change on regularity of occurrence.

In the future, we will improve the performance of *MICRO* algorithm and consider changes based on decreasing of regularity from DB_1 to DB_2, respectively.

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] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '00, 2000, pp. 1–12.

[3] J. Han, J. Wang, Y. Lu, and P. Tzvetkov, "Mining top-k frequent closed patterns without minimum support," pp. 211–218, 2002.

[4] 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.

[5] G. Dong and J. Li, "Efficient mining of emerging patterns: Discovering trends and differences," in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '99, 1999, pp. 43–52.

[6] H. S. Song, J. kyeong Kim, and S. H. Kim, "Mining the change of customer behavior in an internet shopping mall," *Expert Systems with Applications*, vol. 21, no. 3, pp. 157–168, 2001.

[7] M.-C. Chen, A.-L. Chiu, and H.-H. Chang, "Mining changes in customer behavior in retail marketing," *Expert Systems with Applications*, vol. 28, no. 4, pp. 773–781, 2005.

[8] M.-J. Shih, D.-R. Liu, and M.-L. Hsu, "Discovering competitive intelligence by mining changes in patent trends," *Expert Systems with Applications*, vol. 37, no. 4, pp. 2882–2890, 2010.

[9] G. Li, R. Law, H. Q. Vu, J. Rong, and X. R. Zhao, "Identifying emerging hotel preferences using emerging pattern mining technique," *Tourism Management*, vol. 46, pp. 311–321, 2015.

[10] S. K. Tanbeer, M. M. Hassan, M. Alrubaian, and B.-S. Jeong, "Mining regularities in body sensor network data," in *Proceedings of the 8th IDCS International Conference on Internet and Distributed Computing Systems*, 2015, pp. 88–99.

[11] 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.

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

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

Certificate of Contributions

Sumalee Eisariyodom and Komate Amphawan

Entitled

Discovering interesting itemsets based on change in regularity of occurrence

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

Mining regular itemsets with interesting changes on regularity of occurrence

Sumalee Eisariyodom*, Komate Amphawan†
Computational Innovation Laboratory, Faculty of Informatics, Burapha University, Chonburi, 20131, Thailand
Email: *ann.Eisariyodom@gmail.com, †komate@gmail.com

***Abstract*—Nowadays, customers' behavior analysis is a crucial issue in competitive business. There is a need to know customers' behavior including changes of customers' behavior. This leads to an emergence of discovering itemsets based on their change on regularity of occurrence. However, this approach often generates a large amount of such itemsets causing users cannot directly utilize these itemsets to gain interesting information and/or knowledge. To address this issue, the task of *mining regular itemsets with interesting changes on regularity of occurrence* is introduced in this paper. A regularity and a change thresholds are thus assigned to define scope of interest and to filter uninteresting/insignificant itemsets based on their change on regularity of occurrence. To discover such itemsets, an efficient single-pass algorithm called *Regular Itemsets with interesting Changes on Regularity of Occurrence Miner* (*RICROM*) is presented. A new structure named *New Interval Word Segment structure* (*NIWS*) is designed to efficiently maintain occurrence information of each item/itemset during mining process. Experimental studies are conducted to show efficiency of *RICROM* in the terms of runtime, memory usage and the number results in comparison with previous approaches.**

I. Introduction

Currently, frequent itemsets mining (*FIM*) [1] is widely applied in several business and applications. It aims to discover interesting sets of items (also called itemsets or patterns) based on observation of their occurrence behavior. An itemset is identified as interesting if it frequently occurs in a database (*i.e.* it occurs more frequent than a user-given support threshold). *FIM* can help to analyze buying behavior on retail, restaurants, mobile commerce, and so on.

As the popularity of *FIM*, there are several extensions and variations of *FIM* on mining interesting itemsets, for example, mining weighted frequent itemsets [2], mining top-k frequent itemsets [3], mining high utility itemsets [4], mining frequent-regular itemsets [5], [6], etc. Moreover, there exists an approach to mine emerging patterns (EPs) [7] which can help to extract trends and differences on frequency of occurrence of itemsets in different databases. This can be applied in shopping mall, retails, hotels to find changes on customer behavior which can help market analysts having better understanding about customer behavior [8], [9].

Recently, an alternative approach to mine interesting itemsets based on their change on regularity of occurrence [10] is proposed to observe interesting changes on behavior from actions performed by people. A single-pass algorithm named *MICRO* with a tree-structure is proposed to efficiently mine such kind of itemsets. However, without prior knowledge, the users may give inappropriate change threshold causing overwhelming of generated results and difficulties to the users. Hence, it is helpful to avoid this which can help users to be more efficient to look for interesting information and/or knowledge from these itemsets.

Therefore, to address this issue, we here propose to mine regular itemsets with interesting changes on regularity of occurrence. In this framework, a regularity and a change thresholds are thus applied to define scope of interest and to filter uninteresting/insignificant itemsets. To discover these itemsets, an efficient single-pass algorithm called *Regular Itemsets with interesting Changes on Regularity of Occurrence Miner* (*RICROM*) is introduced. A new structure named *NIWS* (*New Interval Word Segment structure*) is also designed to efficiently maintain occurrence information of each item/itemset during mining process. Experimental studies were done on benchmark datasets where the results show that *RICROM* is efficient in the terms of runtime, memory usage and the number results.

II. Problem definitions

In this section, the basic notations on an item/itemset and a transactional database are first described. Then, the regularity of occurrence, the change on regularity of occurrence and the problem statement are introduced as in [5], [10], [11].

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of distinct items. A set of items $X \subseteq I$ is called an itemset or a k-itemset if X contains k items. A transaction database $TDB = \{t_1, t_2, \ldots, t_m\}$ contains m transactions in which each transaction t_j has a unique transaction-identifier equal to j (called *tid* in short) and contains a set of items $Y \subseteq I$. If $X \subseteq Y$, it can be said that X occurs in the transaction t_j or the transaction t_j contains X, denoted as j^X. Therefore, the ordered set of tids containing X can be defined as $T^X = \{j^X, \ldots, k^X\}$.

Definition 1: The regularity of occurrence of an itemset X is the maximum number of consecutive transactions that X does not occur in the database, defined as $r^X = max(fr^X, rt_1^X, rt_2^X, \ldots, rt_{|T^X|-1}^X, lr^X)$ *where (i)* $fr^X = j^X - 1$ *is the number of transactions that X does not occur in database before its first occurrence in transaction t_j, (ii) each* $rt_p^X = q^X - p^X - 1$ *is the number of transactions between two consecutive occurrrences of X in transaction t_p and t_q, and (iii)* $lr^X = |TDB| - k^X - 1$ *is the number of transactions that X does not occur in database after its last occurrence in transaction t_k, respectively.*

978-1-5386-0572-1/18/$31.00 ©2018 IEEE

TDB_1

tid	Set of items	tid	Set of items	tid	Set of items	tid	Set of items	tid	Set of items
1	c, d, e, f	16	a, b, c, d, e	35	a, b, c, d	50	a, d, e	...	...
...	...	...	...	...	...	51	a, d	63	a, b, c, d, e, f
10	a, c, e	30	a, b, e	38	a, b, c, f	...	...	...	...
...	...	...	...	...	...	54	a, b, d, e	80	d

TDB_2

tid	Set of items	tid	Set of items	tid	Set of items	tid	Set of items	tid	Set of items
1	f, g	54	a, b, c, e, g	89	a, b, c, e	101	a	120	g
...	...	...	...	...	...	...	...		
24	a, b, c, e, f, g	56	a, c, e	99	a, b, c, e	116	a, c, e, f, g		
...	...	...	...	100	a	...	...		

Fig. 1: Two transactional databases TDB_1 and TDB_2

Definition 2: An itemset X is called a regular itemset if its regularity r^X is not greater than a user-given regularity threshold σ_r.

Definition 3: The change on regularity of occurrence of an itemset X (also called change value) is the different on regularity of occurrence of X between two given transaction databases i.e. TDB_1 and TDB_2, defined as

$$c^X = \begin{cases} \frac{r^X_{TDB_2}}{r^X_{TDB_1}} & \text{, if X occurs in both } TDB_1 \text{ and } TDB_2 \\ \textit{undefined} & \text{, otherwise} \end{cases}$$

***Problem statement**. Given two transaction databases i.e. TDB_1 and TDB_2, a regularity threshold σ_r and a change threshold σ_c, the task of mining regular itemset with interesting changes on regularity of occurrence is to find itemsets having (i) regularity not greater than a user-given regularity threshold (σ_r) and (ii) change value not less than a user-given change threshold (σ_c), respectively.*

III. Proposed Method

In this section, we here introduce *RICROM* algorithm for discovering regular itemsets with interesting changes on regularity of occurrence. *RICROM* consists of three main steps *i.e.* (*i*) *DB-scanning* which scans database once to capture occurrence information of each item and to identify regular items, (*ii*) *2-itemsets-generation* which considers and generates all of regular 2-itemsets and (*iii*) *Regular-itemsets-mining* which mines a complete set of regular itemsets with interesting changes on regularity of occurrence, respectively. Moreover, *RICROM* applies the new interval word segment structure (*NIWS*) to maintain occurrence information (*tidset*) of each item/itemset.

A. New Interval Word Segment structure

In frequent itemsets mining, a bit-vector is widely used for maintaining occurrence information of an item/itemset. A bit-vector of an itemset X is a sequence of bit where each j^{th} bit can be '0' (X do not occur in transaction t_j) or '1' (X occurs in transaction t_j), respectively. However, a bit-vector may contain a long sequence of bit '0' if X does not occur in a long period. This causes high memory consumption and computational time to process on bit-vector. Thus, the interval word segment structure (*IWS*) is designed to avoid the above issue. An *IWS* of an itemset X is a dynamic bit-vector which does not containing word of '0' (Noted that each word may contain 8 or 16 bits). Each occurrence of word of '0' causes splitting bit-vector from one to be two. For an itemset X, its *IWS* can be defined as $IWS^X = \{\langle wi_1, \{w_{1,1}, w_{1,2}, \ldots, w_{1,p}\}\rangle, \langle wi_2, \{w_{2,1}, w_{2,2}, \ldots, w_{2,q}\}\rangle, \ldots, \langle wi_u, \{w_{u,1}, w_{u,2}, \ldots, w_{u,v}\}\rangle\}$ which is a sequence of 2-tuples $\langle wi_y, W_y = \{w_{y,1}, w_{y,2}, \ldots, w_{y,p}\}\rangle$ where each y^{th} tuple contains (*i*) word index (wi_y) which expresses the first index of non-zero word of the sequence of bit-vector and (*ii*) a sequence of non-zero word $w_{y,1}, w_{y,2}, \ldots, w_{y,p}$. With this structure, IWS^X can avoid maintaining word '0' (if there is a zero word or sequence of zero word, the sequence of non-zero in the current tuple is split to be a new tuple). Then, it can help to safe memory consumption for maintaining sequence of word '0'.

However, splitting partition by one word '0' cannot save much memory to maintain occurrence information. Meanwhile, each time of splitting partition causes extra computation on creating and collecting information into the new partition. Then, we propose to change the condition on splitting partition from one word of word '0' to be three due to it compromises between the term of space and time usages.

To calculate regularity of an itemset X from its $NIWS^X$, a look-up table is applied. As shown in Fig. 5, the look-up table contains 256 tuples where each tuple contians $< pf, nl, mg >$ *i.e.* pf is the position of first occurrence of bit 1 (Note that if there are all bits 0 then $pf = 8$), nl is the number of bit 0 after the last occurrence of a bit 1 to the final of the byte (Note that if there are all bits 0 then $nl = 8$) and mg is the maximum position gap between two bits 1 (Note that if there are all bits 1 then $mg = 1$), respectively.

Therefore, based on the $NIWS^X$ and the look-up table, the regularity r^X can be calculated by considering each word $w_{y,p}$ in each y^{th} tuple of $NIWS^X$ as the following four cases :

1) if $w_{y,p}$ is the first word of the first tuple of $NIWS^X$, then $r^{w_{y,p}} = max((((wi_y - 1) \times 8) + pf(w_{y,p})), mg(w_{y,p})) - 1$;
2) if $w_{y,p}$ is the first word of the y^{th} tuple, then $r^{w_{y,p}} = max((nl(w_{y-1,|W_{y-1}|}) + ((wi_y - 1) - (wi_{y-1} + |W_{y-1}| - 1) \times 8) + pf(w_{y,p})), mg(w_{y,p})) - 1$;
3) if $w_{y,p}$ is a word (not the first word) of the y^{th} tuple, then $r^{w_{y,p}} = max((nl(w_{y,p-1}) + (nw_0 \times 8) + pf(w_{y,p})), mg(w_{y,p})) - 1$;
4) if $w_{y,p}$ is the last word of the last tuple, then $r^{w_{y,p}}$ $nl(w_{y,p}) + (lwo^{TDB} - (wi_y + |W_y| - 1) \times 8) - 1$.

where (*i*) wi_y is the start word index of the y^{th} tuple; (*ii*) $pf(w_{y,p})$ is the number of consecutive bit of 0 before the first bit of 1 in the word $w_{y,p}$; (*iii*) $mg(w_{y,p})$ is the maximum number of sequence of 0 between two bits of 1 of word $w_{y,p}$; (*iv*) $nl(w_{y-1,|W_{y-1}|})$ is the number of bits 0 from the last occurrence of X in the $(y-1)^{th}$ tuple; (*v*) wi_{y-1} is the start word index of the $(y-1)^{th}$ tuple; (*vi*) $|W_{y-1}|$ is the number of word in the $(y-1)^{th}$ tuple; (*vii*) $nl(w_{y,p-1})$ is the number of bits 0 from the last occurrence in the word $p-1^{th}$ of the y^{th} tuple; (*viii*) nw_0 is the number of consecutive of word 0 appearing before and close to the word $w_{y,p}$; (*ix*) $nl(w_{y,p})$ is the the number of bits 0 from the last occurrence in the word $w_{y,p}$; (*x*) lwo^{TDB} is the last word index of *TDB*, respectively.

Example 1: From the transactional databases of Fig. 1, the item 'a' occurs in transactions $t_{10}, t_{16}, t_{30}, t_{35}, t_{38}, t_{50}, t_{51}, t_{54}, t_{63}$ of TDB_1 and transactions $t_{24}, t_{54}, t_{56}, t_{89},$

T_1^a = { 10, 16, 30, 35, 38, 50, 51, 54, 63 } and T_2^a = { 24, 54, 56, 89, 99, 100, 101, 116 }

Word Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Range of tids	1-8	9-16	17-24	25-32	33-40	41-48	49-56	57-64	65-72	73-80	81-88	89-96	97-104	105-112	113-120
Binary Value of TDB_1	0000 0000	0100 0001	0000 0000	0000 0100	0010 0100	0000 0000	0110 0100	0000 0010	0000 0000	0000 0000	-	-	-	-	-
Word Value of TDB_1	0	65	0	4	36	0	100	2	0	0	-	-	-	-	-
Binary Value of TDB_2	0000 0000	0000 0000	0000 0001	0000 0000	0000 0000	0000 0000	0000 0101	0000 0000	0000 0000	0000 0000	0000 0000	1000 0000	0011 1000	0000 0000	0001 0000
Word Value of TDB_2	0	0	1	0	0	0	5	0	0	0	0	128	56	0	16

$NIWS_1^a$ = { <2, {65, 0, 4, 36, 0, 100, 2}> } and $NIWS_2^a$ = { <3, {1}, <7, {5}>, <12, {128, 56, 0, 16}> }

Fig. 2: An example of *NIWS* structure for an item 'a'

$t_{99}, t_{100}, t_{101}, t_{116}$ of *TDB*$_2$. Given the size of each word is 8 bits. Therefore, the new interval word segment of 'a' for *TDB*$_1$ can be sequentially calculated based on each occurrence of 'a', for example, 'a' first occurs in t_{10} then the first word contains '0000 0000' (for the occurrence of 'a' in the first eight transactions) and the second word contains '0100 0000' (for the occurrence of 'a' in between the 9^{th} - 16^{th} transactions). Thus, the first tuple of *NIWS*$_1^a$ is created as $\langle 2, \{64\}\rangle$ (where 2 is the word index of the tuple expressing index of the first non-zero word). Next, the second occurrence of 'a' is in t_{16}. Then, the second word is updated by tid 16 to be '0100 0001' and the first tuple of *NIWS*$_1^a$ is also updated to be $\langle 2, \{65\}\rangle$. The third occurrence in t_{30} causes the third word contains '0000 0000' and the fourth word contains '0000 0100'. Thus, the first tuple is updated as $\langle 2, \{65, 0, 4\}\rangle$. After considering all occurrence of 'a' in *TDB*$_1$, the *NIWS*$_1^a$ is defined as $\{\langle 2, \{65, 0, 4, 36, 0, 100, 2\}\rangle\}$. Similarly, the *NIWS*$_2^a$ can be defined from the occurrence of 'a' in *TDB*$_2$ as $\{\langle 3, \{1\}\rangle, \langle 7, \{5\}\rangle, \langle 12, \{128, 56, 0, 16\}\rangle\}$ (see Fig. 2), respectively.

B. DB-scanning

The first step of *RICROM* algorithm is *DB-scanning* which aims to identify and collect regular 1-itemset for further computation. As detailed in Algo. 1, a simple list named *1List* is first created. A new entry for each item $i_k \in I$ is created and inserted to the *1List* in order to maintain essential information of each item during computational process. Each transaction t_p of transactional database *TDB*$_1$ is then read and each item $i_k \in t_p$ is considered. The new interval word segment structure of item i_k in *TDB*$_1$ (*NIWS*$_1^{i_k}$) and the regularity of item i_k in *TDB*$_1$ ($r_1^{i_k}$) are thus updated by tid p of t_p. After reading all transactions of *TDB*$_1$, each transaction t_q of the transactional database *TDB*$_2$ is read in the same manner as *TDB*$_1$ where the new interval word segment structure of each item $i_k \in t_q$ (*NIWS*$_2^{i_k}$) and the regularity of the item i_k ($r_2^{i_k}$) are also updated by tid q of t_q. Next, each item i_k in the *1List* is considered and its regularity is calculated as $r^{i_k} = max(r_1^{i_k}, r_2^{i_k})$. The entry of i_k is thus eliminated from the *1List*, if its regularity r^{i_k} is greater than the user-given regularity threshold (σ_r). Last, the change c^{i_k} is computed and i_k is identified as a regular item with interesting change if its change c^{i_k} is not less than the user-given change threshold (σ_c).

C. 2-itemsets-generation

As detailed in Algo. 2, all of 2-itemsets that regularly occur in database are considered and generated. A simple list named *2List* is first created in order to collect and maintain all of regular 2-itemsets. Then, each item i_j in *1List* is considered and merged with another item i_k in *1List* (where $i_j \neq i_k$) in order to generate itemset $Z = i_j \cup i_k$. The new interval word segment of Z is calculated and collected by intersection between the new interval word segment of item i_j and i_k (line 5 and 6) and the regularity of Z is thus calculated from its new interval word segment (line 7 and 8). If the regularity of Z is not greater than the user-given regularity threshold, a new entry of Z is created with its new interval word segment and then inserted into *2List*. Next, the change of Z is calculated and if it is not less than the user-given change threshold, Z is thus identified as a regular itemset with interesting change. After performing this step, we gain *2List* containing all of regular 2-itemsets.

D. Regular-itemsets-mining

The step of *Regular-itemsets-mining* is for discovering all regular itemsets with 3 or more than 3 items and with interesting changes of regularity of occurrence. This step will perform after the step of *2-itemsets-generation* that creates 2*List* containing all of regular 2-itemsets (Note that in *Regular-itemsets-mining*, the *k*List is the list gain from previous step or previous computational which includes 2*List*).

To mine the complete set of results, a list, called lList, is created and initialized to maintain l-itemset (*i.e.* l starts from 3 to n). Then, each itemset $X = \{i_a, \ldots, i_b\}$ in kList is considered and merged with another itemset $Y = \{i_a, \ldots, i_c\}$ in *kList*. The last item $i_b \in X$ and the last item $i_c \in Y$ is then considered and grouped to be itemset i_b, i_c. It thus used for looking up in 2*List* in order to observe occurrence behavior of itemset i_b, i_c. Based on *downward closure property* [1], [5], if there is no itemset i_b, i_c contained in the *2List*, it then can conclude that itemset $X \cup Y$ and all of its supersets are not be regular itemsets. Otherwise, it can be said that $X \cup Y$ is a candidate itemset where it can be merged to be itemset Z.

If Z is a candidate itemset, its new interval word segment on both databases is then calculated and collected. Then, its regularity (r^Z) is computed. If the regularity r^Z is not greater than the regularity threshold σ_r, a new entry of itemset Z is created and inserted into *lList*. The change c^Z is thus calculated. The itemset Z is identified as a regular itemset with interesting changes on regularity of occurrence if its c^Z is not less than the change on regularity of occurrence threshold σ_c.

After merging X with all itemsets having the same prefix as X, *lList* is observed. If *lList* has more than one entry the step of *Regular-itemsets-mining* is repeatedly performed on *lList*. Otherwise, *lList* is emptied. Last, after considering X and its supersets, X is eliminated from *kList* and *RICROM* will move consideration to another itemset in *kList*, respectively.

E. Example of RICROM algorithm

Given the transaction databases as in Fig. 1, the regularity threshold (σ_r) to be 40 (*i.e.* 20% of total number of transactions

Algorithm 1: *DB-scanning*

Input: TDB_1, TDB_2, σ_r, σ_c
Output: A set of regular items with interesting changes on regularity of occurrence and *1List* containing regular items

• create a list named *1List* to maintain all single items
• create an entry for each item $i_k \in I$ in the *1List*
for each transaction t_p in database TDB_1 **do**
  for each item i_k in transaction t_p **do**
    • update $NIWS_1^{i_k}$ by tid p
    • update $r_1^{i_k}$ by tid p
for each transaction t_q in database TDB_2 **do**
  for each item i_k in transaction t_q **do**
    • update $NIWS_2^{i_k}$ by tid q
    • update $r_2^{i_k}$ by tid q
for each item i_k in *1List* **do**
  • compute $r^{i_k} \leftarrow \max(r_1^{i_k}, r_2^{i_k})$
  if $r^{i_k} > \sigma_r$ **then**
    • remove the entry of i_k out of *1List*
  else
    • compute c^{i_k} by $\frac{r_2^{i_k}}{r_1^{i_k}}$
    if $c^{i_k} \geq \sigma_c$ **then**
      • identify i_k as a result

Algorithm 2: 2-itemsets-generation

Input: *1List*, σ_r, σ_c
Output: A set of regular 2-itemsets with interesting changes on regularity of occurrence and *2List* containing regular 2-itemsets

• create and initial a list called *2List* to maintain regular 2-itemsets
for each item i_j in *1List* **do**
  for each item i_k in *1List* ($i_j \neq i_k$) **do**
    • merge item i_j with item i_k to be itemset Z
    • $NIWS_1^Z \leftarrow \text{intersect}(NIWS_1^{i_j}, NIWS_1^{i_k})$
    • $NIWS_2^Z \leftarrow \text{intersect}(NIWS_2^{i_j}, NIWS_2^{i_k})$
    • compute r_1^Z from $NIWS_1^Z$ and r_2^Z from $NIWS_2^Z$
    • compute $r^Z \leftarrow \max(r_1^Z, r_2^Z)$
    if $r^Z \leq \sigma_r$ **then**
      • create an entry of Z in *2List*
      • compute c^Z by $\frac{r_2^Z}{r_1^Z}$
      if $c^Z \geq \sigma_c$ **then**
        • identify Z as a result

Algorithm 3: *Regular-itemsets-mining*

Input: *kList*, σ_r, σ_c
Output: A complete set of itemsets with interesting changes on regularity of occurrence

• create and initial *lList* to maintain regular l-itemset (where $l = k + 1$)
for each itemset $X = \{i_a, \ldots, i_b\}$ in the *kList* **do**
  for each itemset $Y = \{i_a, \ldots, i_c\}$ in the *kList* (where Y have the same prefix as X except only the last item) **do**
    if there is an entry of the itemset i_b, i_c in *2List* **then**
      • merge X and Y to be itemset Z
      • $NIWS_1^Z \leftarrow \text{intersect}(NIWS_1^X, NIWS_1^Y)$
      • $NIWS_2^Z \leftarrow \text{intersect}(NIWS_2^X, NIWS_2^Y)$
      • compute r_1^Z from $NIWS_1^Z$ and r_2^Z from $NIWS_2^Z$
      • compute $r^Z \leftarrow \max(r_1^Z, r_2^Z)$
      if $r^Z \leq \sigma_r$ **then**
        • create an entry of Z in *lList*
        • compute c^Z by $\frac{r_2^Z}{r_1^Z}$
        if $c^Z \geq \sigma_c$ **then**
          • identify Z as a result
  if *lList* contains more than one entry **then**
    • repeat step of *Regular-itemsets-mining* by considering itemsets in *lList*
  else
    • empty *lList*
  • remove the entry of X out of *kList*

in both database) and the change on regularity of occurrence threshold (σ_c) to be 1.5, respectively.

In the *DB-scanning*, the transaction $t_1 = \{c, d, e, f\}$ of TDB_1 is first read. Then, the entry of items 'c', 'd', 'e' and 'f' in the *1List* are updated by tid 1 as shown in the Fig. 3(a). All transactions of TDB_1 is read in the same manner, and then we get the *1List* as in Fig. 3(b). All transactions of TDB_2 is also read to update regularity value and *NIWS* of each item. Then, items do not occur in both database or having regularity greater than 40 are eliminated from *1List* (as the items strike-through in red in Fig. 3(c)).

Next, the *2-itemsets-generation* is performed by first considering item 'a' in *1List*. Then, the item 'a' is merged with item 'b' to generate the itemset 'ab'. As in Fig. 4, the $NIWS_1^{ab}$ is computed from the intersection between $NIWS_1^a$ and $NIWS_1^b$, *i.e.* $NIWS_1^{ab} = NIWS_1^a \cap NIWS_1^b = \{\langle 2, \{65, 0, 4, 36, 0, 100, 2\}\rangle\} \cap \{\langle 2, \{1, 2, 4, 36, 0, 4, 2, 4\}\rangle\} = \{\langle 2, \{1, 0, 4, 36, 0, 4, 2\}\rangle\}$. Similarly, the $NIWS_2^{ab}$ is calculated as $NIWS_2^{ab} = NIWS_2^a \cap NIWS_2^b = \{\langle 3, \{1\}\rangle, \langle 7, \{5\}\rangle, \langle 12, \{128, 56, 0, 16\}\rangle\} \cap \{\langle 3, \{1\}\rangle, \langle 7, \{4, 0, 32, 0, 64, 128, 32\}\rangle\} = \{\langle 3, \{1\}\rangle, \langle 7, \{4\}\rangle, \langle 12, \{128, 32\}\rangle\}$. The regularity $r_1^{ab} = 16$ is calculated from $NIWS_1^{ab}$ and the regularity $r_2^{ab} = 34$ is calculated from $NIWS_2^{ab}$ (see Fig. 5). The regularity r^{ab} can thus computed as $r^{ab} = \max(r_1^{ab}, r_2^{ab}) = 34$. The entry of itemset '$ab$' is created and inserted in *2List* since its regularity is less than 40. The change on regularity of 'ab' is calculated by $c^{ab} = \frac{r_2^{ab}}{r_1^{ab}} = \frac{34}{16} = 2.13$. Since c^{ab} is greater than σ_c, the itemset 'ab' is thus identified as a regular itemset with interesting change. Next, the *2-itemsets-generation* repeatedly performs on merging between the item 'a' and items 'b', 'c', 'e', merging between the item 'b' and 'c', 'e' and merging between the item 'c' and 'e', respectively. Hence, as shown in Fig. 3(d), the *2List* at the end of this step contains all regular 2-itemsets 'ab', 'ac', 'ae', 'bc', and 'be', respectively.

Last, the step of *Regular-itemsets-mining* is performed. The itemset 'ab' is first considered and merged with another itemset with the same prefix *i.e.* 'ac' and 'ae'. For the merging between 'ab' and 'ac', the last item 'b' of 'ab' and the last item 'c' of 'ac' is merged to be 'bc'. Then, the 'bc' is used for looking up in *2List* in order to observe that "*whether there exists an entry of 'bc' in 2List*". Since there is the entry of 'bc', the itemsets 'ab' and 'ac' are then merged to be the itemset 'abc'. The $NIWS_1^{ab}$ is intersected with $NIWS_1^{ac}$ and the $NIWS_2^{ab}$ is intersected with $NIWS_2^{ac}$ in order to generate $NIWS_1^{abc}$ and $NIWS_2^{abc}$. The regularity $r^{abc} = 34$ is thus calculated and a new entry for the itemset 'abc' is created and inserted into *3List*, since $r^{abc} < \sigma_r$ (as shown in Fig. 3(e)). The all of merging is sequentially performs in the same manner. At the end of *Regular-itemsets-mining* step, we gain a complete set of results as shown in Fig. 3(f).

IV. Experimental study

In this section, experimental studies on *RICROM* algorithm is investigated and reported. Experiments were conducted on three benchmark databases (*i.e. Chess*, *Mushroom* and *T10I4D100K*) from http://fimi.ua.ac.be/data. *Chess* and *Mushroom* are real dense datasets containing 3,196 and 8,124 transactions. Meanwhile, *T10I4D100K* is a synthetic sparse dataset containing 100,000 transactions. In our experiments, each dataset is divided into 2 equal groups of transactions in which the first group is for the transaction database TDB_1 and the second group is for the transaction database TDB_2, respectively. The previous algorithm named *MICRO* [10] is used for comparing performance. The regularity and change

i_k	$r_1^{i_k}$	$r_2^{i_k}$	c^{i_k}	$NIWS_1^{i_k}$	$NIWS_2^{i_k}$
a	-	-	-	-	-
b	-	-	-	-	-
c	0	-	-	{ <1, {128}> }	-
d	0	-	-	{ <1, {128}> }	-
e	0	-	-	{ <1, {128}> }	-
f	0	-	-	{ <1, {128}> }	-
g	-	-	-	-	-

(a) 1List after reading t_1 of TDB_1

i_k	$r_1^{i_k}$	$r_2^{i_k}$	c^{i_k}	$NIWS_1^{i_k}$	$NIWS_2^{i_k}$
a	16	-	-	{ <2, {65, 0, 4, 36, 0, 100, 2}> }	-
b	15	-	-	{ <2, {1, 2, 4, 36, 0, 4, 2, 4}> }	-
c	24	-	-	{ <1, {128, 65, 0, 0, 36, 0, 0, 2}> }	-
d	19	-	-	{ <1, {128, 1, 0, 0, 33, 0, 100, 4, 0, 1}> }	-
e	9	-	-	{ <1, {128, 65, 4, 5, 1, 0, 68, 66, 132, 4}> }	-
f	26	-	-	{ <1, {128, 0, 0, 16, 4, 0, 16, 2, 132, 4}> }	-
g	-	-	-	-	-

(b) 1List after reading all transaction of TDB_1

i_k	$r_1^{i_k}$	$r_2^{i_k}$	c^{i_k}	$NIWS_1^{i_k}$	$NIWS_2^{i_k}$
a	16	32	2.00	{ <2, {65, 0, 4, 36, 0, 100, 2}> }	{ <3, {1}>, <7, {5}>, <12, {128, 56, 0, 16}> }
b	15	29	1.93	{ <2, {1, 2, 4, 36, 0, 4, 2, 4}> }	{ <3, {1}>, <7, {4, 0, 32, 0, 64, 128, 32}> }
c	24	29	1.21	{ <1, {128, 65, 0, 0, 36, 0, 0, 2}> }	{ <3, {1}>, <7, {5, 0, 32, 0, 0, 128, 32, 0, 16}> }
d	19	-	-	{ <1, {128, 1, 0, 0, 33, 0, 100, 4, 0, 1}> }	-
e	9	22	2.44	{ <1, {128, 65, 4, 5, 1, 0, 68, 66, 132, 4}> }	{ <3, {3, 0, 1, 0, 141, 8, 2, 0, 17, 192, 96, 8, 16}> }
f	26	92	-	{ <1, {128, 0, 0, 16, 4, 0, 16, 2, 132, 4}> }	{ <3, {1}>, <15, {16}> }
g	-	62	-	-	{ <3, {1}>, <7, {4}>, <15, {16}>}

(c) 1List created from TDB_1 and TDB_2

i_k	$r_1^{i_k}$	$r_2^{i_k}$	c^{i_k}	$NIWS_1^{i_k}$	$NIWS_2^{i_k}$
ab	16	34	2.13	{ <2, {1, 0, 4, 36, 0, 4, 2}> }	{ <3, {1}>, <7, {4}>, <12, {128, 32}> }
ac	24	32	1.33	{ <2, {65, 0, 0, 36, 0, 0, 2}> }	{ <3, {1}>, <7, {5}>, <12, {128, 32, 0, 16}> }
ae	19	32	1.68	{ <2, {65, 0, 4, 0, 0, 68, 2}> }	{ <3, {1}>, <7, {5}>, <12, {128, 32, 0, 16}> }
bc	24	29	1.21	{ <2, {1, 0, 0, 36, 0, 0, 2}> }	{ <3, {1}>, <7, {4, 0, 32, 0, 0, 128, 32}> }
be	23	34	1.48	{ <2, {1, 0, 4, 0, 0, 4, 2, 4}> }	{ <3, {1}>, <7, {4}>, <12, {128, 32}> }
ce	46	32	-	{ <1, {128, 65}>, <8, {2}> }	{ <3, {1}>, <7, {5}>, <12, {128, 32, 0, 16}> }

(d) 2List created from 1List

i_k	$r_1^{i_k}$	$r_2^{i_k}$	c^{i_k}	$NIWS_1^{i_k}$	$NIWS_2^{i_k}$
abc	24	34	1.42	{ <2, {1, 0, 0, 36, 0, 0, 2}> }	{ <3, {1}>, <7, {4}>, <12, {128, 32}> }
abe	23	34	1.48	{ <2, {1, 0, 4, 0, 0, 4, 2}> }	{ <3, {1}>, <7, {4}>, <12, {128, 32}> }

(e) 3List created from 2List

i_k	$r_1^{i_k}$	$r_2^{i_k}$	c^{i_k}	$NIWS_1^{i_k}$	$NIWS_2^{i_k}$
a	16	32	2.00	{ <2, {65, 0, 4, 36, 0, 100, 2}> }	{ <3, {1}>, <7, {5}>, <12, {128, 56, 0, 16}> }
b	15	29	1.93	{ <2, {1, 2, 4, 36, 0, 4, 2, 4}> }	{ <3, {1}>, <7, {4, 0, 32, 0, 64, 128, 32}> }
e	9	22	2.44	{ <1, {128, 65, 4, 5, 1, 0, 68, 66, 132, 4}> }	{ <3, {3, 0, 1, 0, 141, 8, 2, 0, 17, 192, 96, 8, 16}> }
ab	16	34	2.13	{ <2, {1, 0, 4, 36, 0, 4, 2}> }	{ <3, {1}>, <7, {4}>, <12, {128, 32}> }
ae	19	32	1.68	{ <2, {65, 0, 4, 0, 0, 68, 2}> }	{ <3, {1}>, <7, {5}>, <12, {128, 32, 0, 16}> }

(f) A complete set of results

Fig. 3: An example of *RICROM* algorithm

$NIWS_1^a$ = { <2, {65, 0, 4, 36, 0, 100, 2}> }

$NIWS_1^b$ = { <2, {1, 2, 4, 36, 0, 4, 2, 4}> }

$NIWS_1^{ab}$ = { <2, {1, 0, 4, 36, 0, 4, 2}> }

$NIWS_2^a$ = { <3, {1}>, <7, {5}>, <12, {128, 56, 0, 16}> }

$NIWS_2^b$ = { <3, {1}>, <7, {4, 0, 32, 0, 64, 128, 32}> }

$NIWS_2^{ab}$ = { <3, {1}>, <7, {4}>, <12, {128, 32}> }

Fig. 4: An example of intersection for $NIWS^{ab}$

thresholds are assigned in range $[1, 16]$ and $[2, 10]$ (as in [11], [10]).

Figure 6 shows the computational time of *RICROM* with different value of regularity thresholds. It is also compared with *MICRO* algorithm. From the figure, we can observe that *RICROM* uses less computational time than *MICRO* in four times on *Chess* and *Mushroom* which are dense datasets. However, on *T10I4D100K* which is sparse dataset, *RICROM* take more time than *MICRO* in some cases. In addition, we can also investigate that the higher regularity threshold causes the increasing of computational time. On high regularity threshold, there are more and more items/itemsets that meet the threshold. Then, *RICROM* needs to take more time to consider these items/itemsets.

The memory usage of *RICROM* with different value of regularity thresholds and that of *MICRO* algorithm are shown in Fig. 7. Similarly as above, the memory usage of *RICROM* is less than that of *MICRO* in three times on *Chess* and *Mushroom*. However, *RICROM* uses more memory than *MICRO* in some cases. Moreover, we can observe that the increasing of the change threshold is not effect the memory usage of both algorithms. The memory usage of both algorithms is stable based on the variation of the change threshold.

Last, the number of results discovered from *RICROM* and *MICRO* are illustrated in Fig. 8. With the regularity threshold, *RICROM* can generate only few results in comparison with that of *MICRO*. This can help to scope the regularity of occurrence of results and also help users easier to look for interesting information and/or hidden knowledge.

V. Conclusion

This paper introduces a new approach for mining regular itemsets with interesting changes on their regularity of occurrence in order to generate a compact set of results based on the user-given regularity and change thresholds. This approach can alleviate overwhelming of results causing difficulties to look for interesting information and/or hidden knowledge. It then can be applied in several real-life applications such as tracking changes of buying behavior, monitoring changes of effects on patients after using medicines, observing changes of reacting on banner advertisements, etc. An efficient single-pass algorithm named *RICROM* and a new interval word segment structure called *NIWS* are designed to efficiently mine such itemsets and maintain occurrence information of each itemset. Experiments were done in real and synthetic datasets. The results illustrate the efficiency of *RICROM* with *NIWS* is faster than and the number of memory usage and discovered itemsets are also reduced from the previous approach especially on dense datasets.

References

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

[2] C. H. Cai, A. W. C. Fu, C. H. Cheng, and W. W. Kwong, "Mining association rules with weighted items," in *Database Engineering and Applications Symposium, 1998. Proceedings. IDEAS'98. International*, 1998, pp. 68–77.

[3] J. Han, J. Wang, Y. Lu, and P. Tzvetkov, "Mining top-k frequent closed patterns without minimum support," pp. 211–218, 2002.

[4] R. Chan, Q. Yang, and Y.-D. Shen, "Mining high utility itemsets," in *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, 2003, pp. 19–26.

[5] 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.

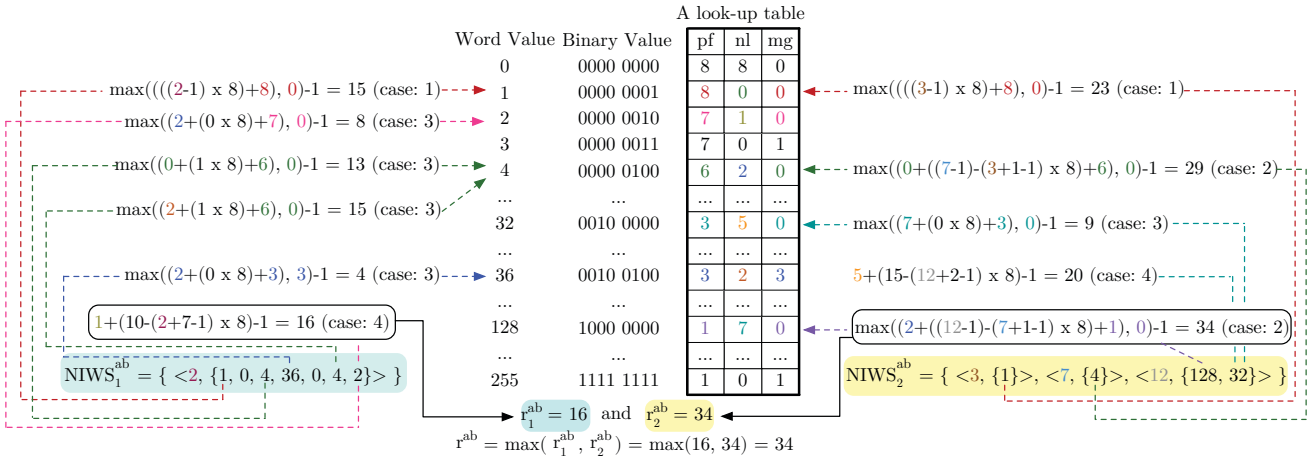


Fig. 5: An example of regularity calculation for an itemset '*ab*'

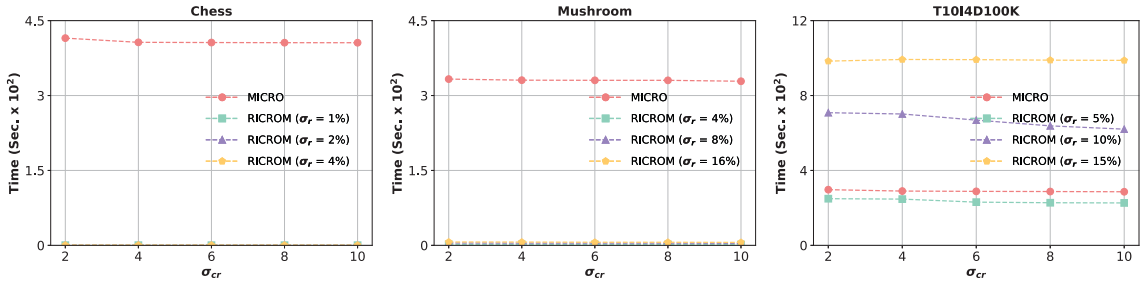


Fig. 6: Computational time of *Chess*, *Mushroom* and *T10I4D100K*

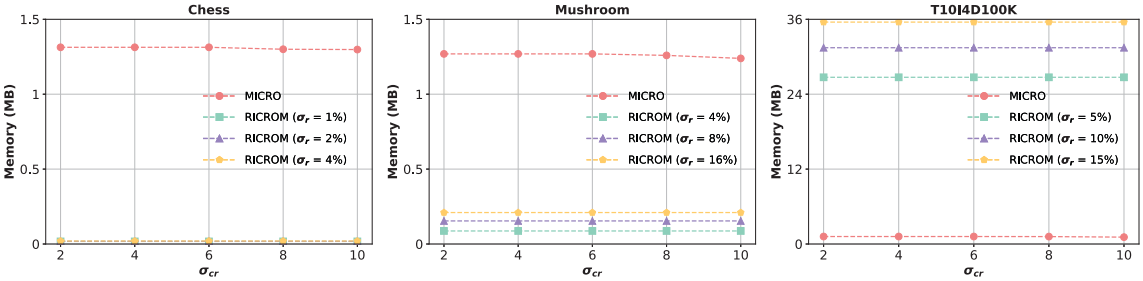


Fig. 7: Memory usage of of *Chess*, *Mushroom* and *T10I4D100K*

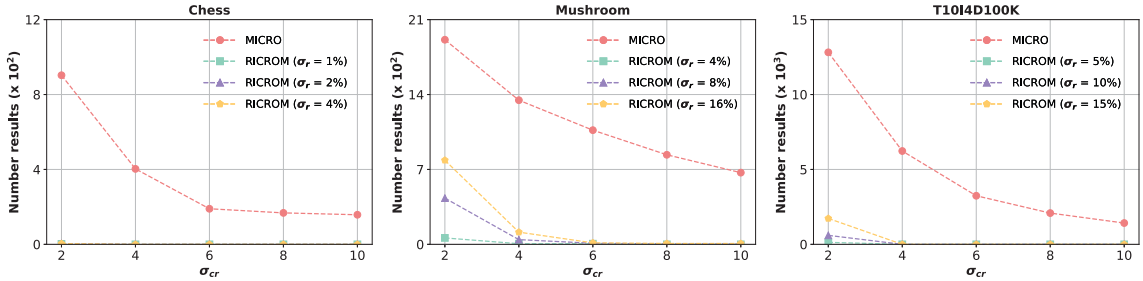


Fig. 8: Number results of *Chess*, *Mushroom* and *T10I4D100K*

[6] 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.

[7] G. Dong and J. Li, "Efficient mining of emerging patterns: Discovering trends and differences," in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '99, 1999, pp. 43–52.

[8] M.-C. Chen, A.-L. Chiu, and H.-H. Chang, "Mining changes in customer behavior in retail marketing," *Expert Systems with Applications*, vol. 28, no. 4, pp. 773–781, 2005.

[9] G. Li, R. Law, H. Q. Vu, J. Rong, and X. R. Zhao, "Identifying emerging hotel preferences using emerging pattern mining technique," *Tourism Management*, vol. 46, pp. 311–321, 2015.

[10] S. Eisariyodom and K. Amphawan, "Discover interesting itemsets based on change in regularity of occurrence," in *Proceedings of the 9th International Conference on Knowledge and Smart Technology*, 2017, pp. 138–143.

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

iCDAMT International Conference on Digital Arts, Media and Technology

3 ECTI NCON 2018

Wisdom for Community Empowerment

Certification of Appreciation
presented to

Sumalee Eisariyodom

as Speaker of

Mining Regular Itemsets with Interesting Changes on Regularity of Occurrence

at International Conference on Digital Arts, Media and Technology
University of Phayao Chiang Rai Campus, THAILAND

25 - 28 February 2018

Assoc. Prof. Somsak Choomchuay, Ph.D.

General Chair of ICDAMT 2018