รายงานการวิจัยฉบับสมบูรณ์

โครงการ “การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงที่ปรากฏอย่างสม่ำเสมอเพื่อการวิเคราะห์พฤติกรรมผู้บริโภค (Mining high utility patterns with regular occurrence for customers’ behavior analysis)”

คณะผู้วิจัย

นายโกเมศ	อัมพวัน	หัวหน้าโครงการ
นายอรรถสิทธิ์	สรุฤกษ์	ผู้ร่วมวิจัย
นายอนุชิต	จิตพัฒนกุล	ผู้ร่วมวิจัย

โครงการวิจัยประเภทงบประมาณเงินรายได้

จากเงินอุดหนุนรัฐบาล (งบประมาณแผ่นดิน)

ปีงบประมาณ พ.ศ. ๒๕๕๙

มหาวิทยาลัยบูรพา

รหัสโครงการ 2559A10802045

สัญญาเลขที่ 47/2559

รายงานการวิจัยฉบับสมบูรณ์

โครงการ “การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงที่ปรากฏอย่างสม่ำเสมอเพื่อการวิเคราะห์พฤติกรรมผู้บริโภค (Mining high utility patterns with regular occurrence for customers’ behavior analysis)”

คณะผู้วิจัย

นายโกเมศ	อัมพวัน	หัวหน้าโครงการ*
นายอรรถสิทธิ์	สรุฤกษ์	ผู้ร่วมวิจัย**
นายอนุชิต	จิตพัฒนกุล	ผู้ร่วมวิจัย***

*ห้องปฏิบัตการวิจัยนวัตกรรมการประมวลผล คณะวิทยาการสารสารเทศ มหาวิทยาลัยบูรพา

**ห้องปฏิบัติการทางวิศวกรรมระบบนับได้เชิงทฤษฎี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

*** คณะวิทยาศาสตร์ประยุกต์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

กันยายน 2560

บทคัดย่อ

การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงเป็นหัวข้องานวิจัยหนึ่งภายใต้การทำเมืองข้อมูลที่น่าสนใจ การค้นหารูปแบบดังกล่าวสามารถประยุกต์ใช้ในแอพพลิเคชันต่างๆอย่างแพร่หลาย ตัวอย่างเช่น การประยุกต์ใช้ในธุรกิจค้าปลีกเพื่อทำการค้นหาเซตของสินค้าที่ถูกซื้อจากลูกค้า โดยเซตของสินค้าดังกล่าวจะเป็นรายการสินค้าต่างๆที่ถูกซื้อร่วมกันที่จะให้ผลกำไรสูงหรือต้นทุนที่ต่ำเป็นต้น แต่อย่างไรก็ตาม การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงจะทำการพิจารณาเพียงแค่ค่าคุณประโยชน์ของรายการต่างๆเท่านั้นที่ซึ่งการดำเนินการดังกล่าวอาจไม่เพียงพอต่อการสังเกตุ/วิเคราะห์พฤติกรรมการซื้อสินค้าของผู้บริโภค ด้วยเหตุนี้ งานวิจัยนี้จึงมุ่งเน้นที่จะทำการเพิ่มเติมเงื่อนไขการพิจารณารูปแบบโดยจะทำการเพิ่มเติมเงื่อนไขของการปรากฏอย่างสม่ำเสมอร่วมกับการพิจารณาค่าคุณประโยชน์ของรายการต่างๆ ภายใต้แนวคิดข้างต้น รูปแบบที่น่าสนใจจะเป็นรูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏขึ้นในชุดข้อมูลอย่างสม่ำเสมอ

ในการค้นหารูปแบบใหม่ที่นำเสนอ ผู้วิจัยได้เสนอขั้นตอนวิธีที่มีประสิทธิภาพที่ชื่อว่า HURI-UL ที่ซึ่งจะทำการอ่านข้อมูลจากฐานข้อมูลเพียงครั้งเดียว และทำการประยุกต์ใช้แนวความคิดเกี่ยวกับค่าคุณประโยชน์ที่หลงเหลือและค่าคุณประโยชน์แบบประมาณเพื่อช่วยลดทอนปริภูมิสถานะของการค้นหารูปแบบ นอกจากนั้นยังประยุกต์ใช้โครงสร้างลิสต์คุณประโยชน์เพื่อใช้ในการจัดเก็บค่าคุณประโยชน์และข้อมูลการปรากฏขึ้นของรูปแบบหนึ่งๆ ยิ่งไปกว่านั้น ผู้วิจัยได้นำเสนอโครงสร้างลิสต์คุณประโยชน์ใหม่ ที่เรียกว่า New Utility List structure (NUL) ที่ซึ่งเพิ่มการจัดเก็บคุณประโยชน์สำหรับ prefix items ที่ซึ่งจะช่วยลดเวลาในการคำนวณค่าคุณประโยชน์ที่แท้จริงของรูปแบบ/เซตรายการได้ จากการคิดค้นโครงสร้างลิสต์คุณประโยชน์ใหม่ก่อให้เกิดการนำเสนอขั้นตอนวิธี MHUIRA ที่ซึ่งประยุกต์ใช้โครงสร้างลิสต์คุณประโยชน์ใหม่เพื่อทำการค้นรูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างสม่ำเสมอ โดยในการทดสอบประสิทธิภาพของขั้นตอนวิธีที่นำเสนอ เราจะสังเกตุได้ว่าขั้นตอนวิธีที่นำเสนอสามารถค้นหารูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างสม่ำเสมอได้อย่างมีประสิทธิภาพ

Abstract

High utility itemsets mining (HUIM) is an interesting topic in data mining which can be applied in a wide range of applications, for example on retail marketing to find sets of sold products that give high profit, low cost, etc. However, HUIM only considers utility values of items/itemsets which may be insufficient to observe buying behavior of customers. To address this issue, we here introduce an approach on pushing regularity constraint on high utility itemsets mining to observe occurrence behavior of high utility itemsets. Based on this approach, sets of co-occurrence items with (*i*) high utility values and (*ii*) regular occurrence, called *high utility-regular itemsets* (*HURI*s), are regarded as interesting. To mine HURIs, an efficient single-pass algorithm, called HURI-UL, is proposed. HURI-UL applies the concept of remaining and overestimated utilities of itemsets to early prune search space and also utilizes utility list structure to efficiently maintain utility values and occurrence information of itemsets. Moreover, to enhance efficiency of HURI-UL, a new utility list structure (also called NUL) is designed and developed. NUL is an extension of the utility list which adding utility value of prefix items into each entry of the list. This can help to quickly calculate actual utility of itemsets. Thus, based on NUL, a new algorithm named MHUIRA (Mining Hihg-Uitlity Itemset with Regular Appearance) is developed. MHUIRA performs one scan of database as HURI-UL but it applied NUL to maintain essential information during mining process. Experimental results on real datasets show that our proposed approach is efficient to discover high utility itemsets with regular occurrence.

สารบัญ

สารบัญรูปภาพ

สารบัญตาราง

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในยุคปัจจุบันเป็นยุคที่การดำเนินธุรกิจการค้ามีการแข่งขันกันค่อนข้างสูง และมีธุรกิจขนาดกลางและขนาดเล็กก่อตั้งขึ้นเป็นจำนวนมาก ด้วยเหตุนี้จึงเป็นเหตุให้หลายธุรกิจให้ความสนใจกับการประยุกต์ใช้ข้อมูลข่าวสารที่เป็นข้อมูลเชิงกลยุทธ์เพื่อประกอบการตัดสินใจในการดำเนินธุรกิจและการประยุกต์ใช้เทคโนโลยีคอมพิวเตอร์หรือระบบพื้นฐานต่าง ๆ เพื่อช่วยในการดำเนินธุรกิจ จากความต้องการดังกล่าวจึงเป็นเหตุให้มีนักวิจัยพยายามที่จะทำการศึกษารูปแบบพฤติกรรมผู้บริโภคด้วยการค้นหารูปแบบที่ปรากฏบ่อย (Frequent pattern mining) เพื่อบ่งบอกถึงความสัมพันธ์ของสิ่งของหรือเหตุการณ์ที่กฎขึ้นพร้อมกันบ่อย ๆ ตัวอย่างเช่น ในธุรกิจห้างสรรพสินค้าหรือธุรกิจค้าปลีกจะทำการหาความสัมพันธ์ของรายการสินค้าที่ถูกซื้อร่วมกันบ่อย ๆ เพื่อช่วยในการจัดทำโปรโมชันสินค้า ช่วยในการจัดชั้นวางสินค้าให้สินค้าที่ถูกซื้อร่วมกันบ่อย ๆ ให้อยู่ในพื้นที่ใกล้ ๆ กันเพื่ออำนวยความสะดวกให้แก่ลูกค้าและยังช่วยกระตุ้นการจับจ่ายใช้สอยของลูกค้า นอกจากนั้นยังช่วยในการจัดทำแคตาล็อกสินค้าให้สินค้าที่ถูกซื้อร่วมกันบ่อย ๆ ให้อยู่ใกล้ ๆ กัน

แนวความคิดเบื้องต้นของการค้นหารูปแบบที่ปรากฏบ่อยจะประยุกต์ใช้ค่าสนับสนุน (ค่าความถี่หรือจำนวนครั้งในการเกิดขึ้นของรูปแบบนั้น ๆ) เป็นตัวชี้วัดความสำคัญหรือความน่าสนใจของรูปแบบ แต่อย่างไรก็ตามการใช้เพียงแค่ค่าสนับสนุนอาจจะไม่เพียงพอต่อการค้นหารูปแบบที่มีความหลากหลาย โดยแนวความคิดนี้ถูกพัฒนาอย่างต่อเนื่องในหลายแง่มุม อาทิเช่น การค้นหารูปแบบที่มีการเรียงลำดับที่ปรากฏบ่อย (Frequent sequential pattern mining) การค้นหารูปแบบที่ปรากฏบ่อยภายใต้ค่าน้ำหนักของแต่ละรายการ (Frequent weighted pattern mining) การค้นหารูปแบบที่มีค่าคุณประโยชน์สูง (High utility pattern mining) การค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ (Frequent-regular pattern mining) และ อื่น ๆ

จากงานวิจัยข้างต้น มีหัวข้องานวิจัยหนึ่งที่ทำการพิจารณาค่าคุณประโยชน์ของรูปแบบ (utility of patterns) ที่ซึ่งค่าคุณประโยชน์ของรูปแบบนั้นอาจหมายถึง ผลกำไรที่ได้รับจากการขายสินค้าชิ้นหนึ่ง ๆ ของ

รายการสินค้าหนึ่ง ๆ หรือการบริการหนึ่ง ๆ เมื่อเราทำการคำนวณค่าคุณประโยชน์ทั้งหมดของรายการสินค้านั้นจะทำให้เราสามารถทราบได้ถึงจำนวนผลกำไร/ขาดทุนที่ได้จากรายการสินค้านั้น ๆ และยังทราบถึงรายการสินค้าที่ให้ผลตอบแทนที่สูง แต่อย่างไรก็ดีการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงไม่ได้ทำการพิจารณาถึงพฤติกรรมการปรากฏขึ้นของรูปแบบหรือรายการนั้น ๆ ว่ามีพฤติกรรมการปรากฏขึ้นอย่างสม่ำเสมอ ไม่สม่ำเสมอ สม่ำเสมอในบางช่วงเวลาหรือไม่ ซึ่งการตรวจสอบหรือการค้นหาพฤติกรรมการปรากฏขึ้นของรูปแบบอาจช่วยให้เราทราบถึงช่วงเวลาที่รายการสินค้าได้รับความนิยม อันนำมาซึ่งการตัดสินใจในการเพิ่ม-ลดกำลังการผลิต การเตรียมวัตถุดิบเพื่อให้สอดคล้องกับกำลังการผลิต การจัดทำโปรโมชันให้กับรายการสินค้าที่พิจารณาเพื่อช่วยกระตุ้นยอดขาย และอื่น ๆ เป็นต้น

ดังนั้น ในงานวิจัยนี้ได้นำเสนอกรอบความคิดที่จะทำการค้นหารูปแบบที่มีประโยชน์สูงที่ปรากฏขึ้นอย่างสม่ำเสมอที่จะสามารถทำให้ผู้ที่ทำการวิเคราะห์หรือต้องการค้นหารูปแบบดังกล่าวทราบถึงพฤติกรรมการเกิดขึ้นของรูปแบบที่มีค่าคุณประโยชน์สูงว่ามีพฤติกรรมการปรากฏขึ้นของรูปแบบอย่างสม่ำเสมอหรือไม่ ที่ซึ่งจะทำให้ธุรกิจต่าง ๆ สามารถประยุกต์ใช้รูปแบบดังกล่าวไปวิเคราะห์หาสาเหตุของการเกิดขึ้นของรูปแบบ อันนำไปซึ่งการปรับปรุงและการพัฒนาผลิตภัณฑ์ หรือ กระบวนการดำเนินธุรกิจที่จะทำให้ได้ผลประกอบการที่ดียิ่งขึ้น

1.2 วัตถุประสงค์ของโครงการวิจัย

1. เพื่อศึกษาการวิเคราะห์พฤติกรรมหรือรูปแบบการบริโภคของผู้บริโภคภายใต้กรอบแนวคิดเกี่ยวกับรูปแบบที่มีค่าคุณประโยชน์สูงและการปรากฏขึ้นของรูปแบบอย่างสม่ำเสมอ ที่จะนำไปสู่การค้นหาสาเหตุของการเกิดขึ้นของพฤติกรรมหรือรูปแบบการบริโภคเหล่านั้นที่ซึ่งจะสามารถนำไปเป็นส่วนหนึ่งในการวิเคราะห์สำหรับการพัฒนาผลิตภัณฑ์หรือขั้นตอนการดำเนินธุรกิจสืบไป
2. เพื่อสร้างนวัตกรรมใหม่ในการตรวจสอบพฤติกรรมของการบริโภค
3. เพื่อให้ผู้ที่สนใจสามารถนำแนวคิดที่นำเสนอ ไปศึกษาเพื่อทำการพัฒนาหรือประยุกต์ใช้ในงานวิจัยหรือประยุกต์ใช้ในการดำเนินธุรกิจของตนเองต่อไป

1.3 ขอบเขตของโครงการวิจัย

การวิจัยครั้งนี้มุ่งที่จะศึกษาและพัฒนาการค้นหารูปแบบที่มีคุณประโยชน์สูงที่ปรากฏอย่างสม่ำเสมอ โดยมีขอบเขตดังนี้

1. ผู้ที่ต้องการค้นหารูปแบบที่มีคุณประโยชน์สูงและปรากฏขึ้นอย่างสม่ำเสมอจะต้องกำหนดค่าพารามิเตอร์สองค่าด้วยกันคือ 1) ค่าขีดแบ่งคุณประโยชน์ (Utility threshold) และ 2) ค่าขีดแบ่งความสม่ำเสมอ (Regularity threshold) เพื่อใช้เป็นมาตรวัดความสำคัญของรูปแบบที่จะทำการค้นหา
2. ฐานข้อมูลที่ใช้ในการค้นหารูปแบบจะต้องมีจำนวนรายการที่ปรากฏขึ้นในแต่ละทรานแซกชันแนบอยู่ด้วย และแต่ละรายการจะต้องมีค่าคุณประโยชน์ที่กำหนดไว้ก่อนหน้า (โดยค่าคุณประโยชน์อาจหมายถึง ยอดขาย ต้นทุน หรือผลกำไรที่จะได้รับต่อการขายสินค้าชิ้นหนึ่งๆ)
3. การทดสอบประสิทธิภาพของขั้นตอนวิธีในการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงที่ปรากฏอย่างสม่ำเสมอจะวัดในเชิงเวลาและจำนวนหน่วยความจำที่ใช้ในการคำนวณ

1.4 ประโยชน์ที่ได้รับ

1. ได้แนวทางในการวิเคราะห์รูปแบบที่มีประโยชน์สูงและปรากฏขึ้นอย่างสม่ำเสมอ ที่สามารถนำไปวิเคราะห์หาสาเหตุของการเกิดขึ้นของรูปแบบ ที่ซึ่งจะช่วยให้ผู้บริหารกิจการ บริษัท หรือเจ้าของธุรกิจ จะสามารถทำการปรับเปลี่ยนวิธีหรือกลยุทธ์ในการดำเนินธุรกิจเพื่อให้ธุรกิจที่ทำอยู่สามารถดำเนินไปได้ด้วยดี
2. ได้ขั้นตอนวิธีต้นแบบในการค้นหารูปแบบที่มีประโยชน์สูงและปรากฏขึ้นอย่างสม่ำเสมอ
3. สามารถนำขั้นตอนวิธีข้างต้นไปพัฒนาระบบซอฟต์แวร์ เพื่อใช้ในการวิเคราะห์พฤติกรรมของลูกค้าหรือผู้บริโภค ที่จะทำให้กิจการ บริษัท ห้างร้านต่าง ๆ สามารถปรับตัวตามพฤติกรรมของผู้บริโภคได้
4. ขั้นตอนวิธีที่นำเสนอสามารถถูกใช้เป็นต้นแบบในการศึกษาและวิธีขั้นสูงต่อไป

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

งานวิจัยนี้เกิดจากการนำสองแนวความคิดหลักมาผสมผสานกันเพื่อทำการค้นหารูปแบบที่มีประโยชน์สูงและปรากฏขึ้นอย่างสม่ำเสมอ คือ 1) การค้นหารูปแบบที่มีประโยชน์สูง และ 2) การค้นหารูปแบบที่ปรากฏขึ้นสม่ำเสมอ ที่ซึ่งเป็นการพัฒนาต่อยอดมาจากการค้นหารูปแบบที่ปรากฏบ่อยที่ถูกพัฒนาในวงกว้าง โดยทั้งสามแนวคิดหลักสามารถอธิบายได้ดังนี้

2.1.1 การค้นหารูปแบบที่ปรากฏบ่อย (Mining frequent patterns from transactional databases)

การค้นหารูปแบบที่ปรากฏบ่อยเป็นการค้นหารูปแบบโดยมุ่งเน้นที่จะพิจารณาจำนวนครั้ง/ความบ่อย/ความถี่ในการปรากฏขึ้นของรูปแบบเหล่านั้นในฐานข้อมูล โดยปัญหาการค้นหารูปแบบที่ปรากฏบ่อยสามารถนิยามได้ดังนี้

กำหนดให้ เซต $I = \{i_1, i_2, ..., i_m\}$ เป็นเซตของรายการ (items) ที่ใช้แทนสิ่งของหรือเหตุการณ์ที่ต้องการหาความสัมพันธ์ เซต $X = \{i_p, i_{p+1}, ...,i_q\} \subseteq I$ จะเรียกว่าว่าเป็น เซตรายการ (set of items, an itemset หรือ a pattern) และ จะเรียกว่า k-itemset หรือ k-patterns เมื่อ เซต X ประกอบไปด้วยรายการทั้งสิ้น k รายการ กำหนดให้ *TDB* = $\{t_1, t_2, ..., t_n\}$ คือ ฐานข้อมูลรายการหรือฐานข้อมูลแบบทรานแซกชัน (transactional database) ที่ซึ่งแต่ละทรานแซกชัน $t_j \in TDB$ จะประกอบด้วยหมายเลขกำกับทรานแซกชัน (unique transaction identifier, *tid*) tid = j และ เซตของรายการ Y ที่ถูกบรรจุอยู่ในทรานแซกชันนั้นๆ (ดังแสดงตัวอย่างในตารางที่ 1) ถ้าเซตรายการ $X \subseteq Y$ เราจะสามารถสรุปได้ว่าเซตรายการ X ปรากฏขึ้นในทรานแซกชัน t_j หรือทรานแซกชัน t_j มี X บรรจุอยู่ เราสามารถเขียนสัญลักษณ์แทนได้ว่า t^X_j ดังนั้นเมื่อทำการตรวจสอบรูปแบบ X ว่า

ปรากฏขึ้นในทรานแซกชันใดบ้างในฐานข้อมูล *TDB* เราจะทราบถึง $T^X = \{t^X_j, t^X_{j+1}, ..., t^X_k\}$ ซึ่งก็คือ เซตของหมายเลขทรานแซกชัน (tid) ที่ถูกเรียงลำดับที่ซึ่งมี X ปรากฏอยู่ในทรานแซกชันเหล่านั้น (สามารถเรียกเซต T^X ใดๆได้เป็น tidset ของเซตรายการ X) ดังนั้นเราจะสามารถทราบถึงจำนวนครั้งในการปรากฏขึ้นของรูปแบบ X ในฐานข้อมูล (ค่าความถี่หรือค่าสนับสนุน) โดยสามารถคำนวณได้เป็น $s^X = |T^X|$ จากนิยามข้างต้น ปัญหาการค้นหารูปแบบที่ปรากฏบ่อยจะเป็นการค้นหารูปแบบที่มีค่าความถี่มากกว่าหรือเท่ากับค่าขีดแบ่งสนับสนุน (support threshold) ที่ผู้ใช้กำหนด

ตารางที่ 1 ตัวอย่างฐานข้อมูลรายการที่ประกอบไปด้วยหมายเลขทรานแซกชันและเซตรายการที่ปรากฏในทรานแซกชัน

หมายเลขทรานแซกชัน (tid)	เซตรายการที่ปรากฏในทรานแซกชัน (a set of items or an itemset)
1	a b c d
2	a c d
3	a b d
4	b c d e
5	a b c e
6	a e
7	a b c
8	b c d e
9	a b d e
10	a e

2.1.2 การค้นหารูปแบบที่มีประโยชน์สูง (Mining high utility patterns from transactional databases)

การค้นหารูปแบบที่มีประโยชน์สูงถูกพัฒนามาจากการค้นหารูปแบบที่ปรากฏบ่อย โดยทำการเปลี่ยนการพิจารณาความน่าสนใจของรูปแบบจากเดิมที่เป็นค่าความถี่ของการปรากฏไปเป็นค่าคุณประโยชน์ของรูปแบบนั้นๆ โดยค่าคุณประโยชน์อาจเป็นผลกำไร ต้นทุน ความเสี่ยง และอื่นๆ โดยการค้นหารูปแบบที่มีประโยชน์สูงสามารถนิยามได้ดังนี้

กำหนดให้ เซต $I = \{i_1, i_2, ..., i_m\}$ เป็นเซตของรายการ (items) โดยแต่ละรายการ i_p ($1 \leq p \leq m$) ในเซต I จะมีค่าคุณประโยชน์ (utility of item) แนบอยู่ด้วย (เขียนแทนด้วย $eu(i_p)$) โดยค่าคุณประโยชน์อาจหมายถึงค่าผลกำไร คุณค่าของสิ่งของเหล่านั้น และ อื่นๆ ดังแสดงในตารางที่ 2 รายการ ‘a’ จะมีค่าคุณประโยชน์เท่ากับ 10 ดังนั้นเมื่อมีรายการ ‘a’ ปรากฏขึ้นหนึ่งครั้งในฐานข้อมูล จะทำให้เราทราบว่า a มีค่าคุณประโยชน์ที่ปรากฏขึ้นในฐานข้อมูลเท่ากับ 10

เซต $X = \{i_p, i_{p+1}, ...,i_q\} \subseteq I$ จะเรียกว่าเป็น เซตรายการ (set of items or an itemset) และ จะเรียกว่า k-itemset หรือ k-patterns เมื่อ เซต X ประกอบไปด้วยรายการทั้งสิ้น k รายการ นอกจากนี้ยังกำหนดให้ *TDB* = $\{t_1, t_2, ..., t_n\}$ คือ ฐานข้อมูลรายการหรือฐานข้อมูลแบบทรานแซกชัน (transactional database) ที่ซึ่งแต่ละทรานแซกชัน $t_j \in TDB$ จะประกอบด้วยหมายเลขกำกับทรานแซกชัน (unique transaction identifier, *tid*) tid = j และ เซตของรายการที่ถูกบรรจุอยู่ในทรานแซกชันนั้นๆ ดังแสดงในตารางที่ 3 ทรานแซกชันที่ 1 กล่าวคือ t_1 = {a(3), b(6)} จะประกอบด้วย 2 รายการคือ ‘a’ และ ‘b’ โดยทรานแซกชันที่ 1 จะมีรายการ ‘a’ ปรากฏขึ้นทั้งหมด 3 ครั้ง และ รายการ ‘b’ ปรากฏขึ้นทั้งหมด 6 ครั้ง ตามลำดับ จากตัวอย่างเราสามารถพูดได้ว่าเซตรายการ ‘ab’ ปรากฏขึ้นหรือถูกบรรจุอยู่ในทรานแซกชันที่ 1 โดยที่เซตรายการ ‘ab’ จะมีค่าคุณประโยชน์ที่ปรากฏขึ้นในทรานแซกชันที่ 1 เท่ากับ (3×10) + (6×5) = 60 เป็นต้น

จากนิยามข้างต้น ปัญหาการค้นหารูปแบบที่มีประโยชน์สูง อาจหมายถึง การค้นหารูปแบบรายการสินค้าที่มีผลกำไรสูง ที่จะทำการพิจารณาค่าคุณประโยชน์ของแต่ละรายการและจำนวนครั้งที่เกิดขึ้นของแต่ละรายการในแต่ละทรานแซกชัน โดยที่เซตรายการ X ใดๆจะเป็นเซตรายการที่มีประโยชน์สูงก็ต่อเมื่อ X มีค่าคุณประโยชน์มากกว่าหรือเท่ากับค่าขีดแบ่งคุณประโยชน์ (utility threshold) ที่ผู้ที่ต้องทำการค้นหารูปแบบเป็นผู้กำหนด โดย

ค่าคุณประโยชน์ของเซตรายการ X หาได้จากผลรวมของค่าคุณประโยชน์ของแต่ละรายการที่เป็นสมาชิกของ X โดยค่าคุณประโยชน์ของแต่ละการ $x_i \in X$ จะสามารถคำนวณได้จากผลรวมของจำนวนทั้งหมดที่ตาละรายการ x_i ปรากฏขึ้นในฐานข้อมูลคูณกับค่าคุณประโยชน์ของ x_i นั้นๆ ดังนั้น เราสามารถสรุปได้ว่าปัญหาการค้นหารูปแบบที่มีประโยชน์สูงคือ การค้นหารูปแบบหรือเซตรายการที่มีค่าคุณประโยชน์มากกว่าหรือเท่ากับค่าขีดแบ่งคุณประโยชน์

ตารางที่ 2 ตัวอย่างตารางแสดงค่าคุณประโยชน์ของแต่ละรายการ

รายการ	a	b	c	d	e
ค่าคุณประโยชน์	10	5	3	2	7

ตารางที่ 3 ตัวอย่างฐานข้อมูลรายการที่ประกอบไปด้วยหมายเลขทรานแซกชันและเซตรายการที่ปรากฏในทรานแซกชันที่มีการจำนวนของการปรากฏขึ้นของแต่ละรายการ

หมายเลขทรานแซกชัน (tid)	เซตรายการที่ปรากฏในทรานแซกชัน (a set of items or an itemset)
1	a(3) b(6)
2	a(2) c(1) d(3)
3	a(7) b(1) d(5)
4	b(2) c(1) d(3) e(2)
5	a(1) b(1) c(2) e(2)
6	a(2) e(2)
7	a(3) b(2) c(4)
8	b(4) c(1) d(3) e(2)
9	a(3) b(2) d(4) e(1)
10	a(2) e(7)

2.1.3 การค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ (Mining frequent-regular patterns from transactional databases)

การค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอจะเป็นการค้นหารูปแบบที่พัฒนาต่อยอดจากการค้นหารูปแบบที่ปรากฏบ่อย โดยพัฒนาการวัดความน่าสนใจของรูปแบบที่จากเดิมจะวัดความน่าสนใจโดยพิจารณาจากความถี่ในการปรากฏขึ้นของรูปแบบไปเป็นการวัดความสนใจโดยการตรวจสอบจากพฤติกรรมการปรากฏขึ้นของรูปแบบ โดยจะทำการพิจารณาความถี่และความสม่ำเสมอของการปรากฏขึ้นของรูปแบบ ที่ซึ่งสามารถนิยามได้ดังนี้

กำหนดให้ เซต $I = \{i_1, i_2, ..., i_m\}$ เป็นเซตของรายการ (items) เซต $X = \{i_p, i_{p+1}, ..., i_q\} \subseteq I$ จะเรียกว่าว่าเป็น เซตรายการ (set of items or an itemset) และ จะเรียกว่า k-itemset หรือ k-patterns เมื่อ เซต X ประกอบไปด้วยรายการทั้งสิ้น k รายการ กำหนดให้ *TDB* $= \{t_1, t_2, ..., t_n\}$ คือ ฐานข้อมูลรายการหรือฐานข้อมูลแบบทรานแซกชัน (transactional database) ที่ซึ่งแต่ละทรานแซกชัน $t_j \in TDB$ จะประกอบด้วยหมายเลขกำกับทรานแซกชัน (unique transaction identifier, *tid*) tid = j และ เซตของรายการ Y ที่ถูกบรรจุอยู่ในทรานแซกชันนั้นๆ ถ้า $X \subseteq Y$ เราจะสามารถสรุปได้ว่าเซตรายการ X ปรากฏขึ้นในทรานแซกชัน t_j หรือ t_j มี X อยู่ในทรานแซกชัน ที่ซึ่งเราสามารถเขียนสัญลักษณ์แทนได้ว่า t^X_j ดังนั้นเมื่อทำการตรวจสอบรูปแบบ X ว่าปรากฏขึ้นในทรานแซกชันใดบ้างในฐานข้อมูล *TDB* เราจะทราบถึง $T^X = \{t^X_j, t^X_{j+1}, ..., t^X_k\}$ ซึ่งก็คือ เซตของหมายเลขทรานแซกชัน (tid) ที่ถูกเรียงลำดับที่ซึ่งมี X อยู่ในทรานแซกชัน (สามารถเขียนย่อๆได้เป็น tidset) ดังนั้นเราจะสามารถทราบถึงจำนวนครั้งในการปรากฏขึ้นของรูปแบบ X ในฐานข้อมูล โดยสามารถคำนวณได้เป็น $s^X = |T^X|$

ในการที่จะศึกษาถึงพฤติกรรมการปรากฏขึ้นของรูปแบบว่ามีพฤติกรรมการปรากฏขึ้นอย่างสม่ำเสมอหรือไม่ เราจะต้องทำการพิจารณาเซตของหมายเลขทรานแซกชันที่มี X ปรากฏขึ้น โดยเริ่มจากการพิจารณาแต่ละคู่ของหมายเลขทรานแซกชัน t^X_j และ t^X_{j+1} ที่อยู่ในลำดับติดกัน แล้วทำการหาจำนวนทรานแซกชันที่ไม่มี X ปรากฏระหว่างสองทรานแซกชันนั้นๆ ที่ซึ่งสามารถคำนวณได้เป็น $rtt^Xj = t^X_{j+1} - t^X_j$ แต่สำหรับการปรากฏขึ้นในครั้งแรกและครั้งสุดท้ายของ X จะมีวิธีการคำนวณที่แตกต่างจากการปรากฏขึ้นครั้งอื่นๆที่ซึ่งสามารถคำนวณได้เป็น $fr^X = t^X_1$ (เมื่อ t^X_1 คือ หมายเลขทรานแซกชันที่ X ปรากฏขึ้นครั้งแรก) และ $lr^X = n - t^X_{|T^X|}$ (เมื่อ n คือจำนวนทรานแซกชันทั้งหมดในฐานข้อมูล และ $t^X_{|T^X|}$ คือ หมายเลขทรานแซกชันที่ X ปรากฏขึ้นครั้งสุดท้าย) จากที่กล่าวข้างต้น เราสามารถหาจำนวนทรานแซกชันที่ติดกันสูงที่สุดที่ไม่มี X ปรากฏ โดยสามารถคำนวณได้เป็น $r^X =$

max (fr^X, rtt^X_1, rtt^X_2, ..., lr^X) ที่ซึ่งสามารถบอกได้ถึงช่วงเวลาที่ยาวนานที่สุดที่ไม่มี X ปรากฏขึ้นในฐานข้อมูล และยังสามารถการันตีได้ว่าเซตรายการ X จะปรากฏขึ้นอย่างน้อยหนึ่งครั้งในทุกๆ r^X ทรานแซกชันที่เรียงต่อกัน ซึ่งจากการพิจารณาค่า r^X จะทำให้ทราบถึงพฤติกรรมการปรากฏขึ้นของ X ได้

จากนิยามที่กล่าวมาข้างต้น ปัญหาการค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอจะเป็นการค้นหารูปแบบที่มีค่าความถี่ในการปรากฏมากกว่าหรือเท่ากับค่าขีดแบ่งสนับสนุน (support/ frequency threshold) และมีการช่วงเวลาที่ยาวนานที่สุดที่รูปแบบนั้นๆไม่ปรากฏขึ้นในฐานข้อมูลไม่มากไปกว่าค่าขีดแบ่งความสม่ำเสมอ (regularity threshold)

2.2 งานวิจัยที่เกี่ยวข้อง

S.K. Tanbeer และ คณะ (Tanbeer, 2009) นำเสนองานวิจัยเรื่อง “Discovering periodic-frequent patterns in transactional databases” ที่ชี้ให้เห็นว่าการค้นหารูปแบบปรากฏบ่อยจากฐานข้อมูลโดยใช้ค่าสนับสนุน (จำนวนครั้งของการเกิดขึ้นของรูปแบบเหล่านั้นในฐานข้อมูล) อาจจะไม่เพียงพอต่อการค้นหารูปแบบที่น่าสนใจ จึงได้ทำการเสนอแนวความคิดในการค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ เพื่อที่จะได้ทราบถึงพฤติกรรมการเกิดขึ้นของรูปแบบเหล่านั้น โดยแนวความคิดนี้สามารถนำไปประยุกต์ใช้ได้ในงานหลายๆด้าน อาทิเช่น ผู้จัดการหรือผู้บริหารของธุรกิจค้าปลีกอาจจะสนใจรายการสินค้าที่ถูกซื้อบ่อยๆและถูกซื้ออย่างสม่ำเสมอมากกว่ารายการสินค้าที่ถูกซื้อบ่อยๆเพียงอย่างเดียว เพื่อที่จะทำการจัดเตรียมสินค้าให้พอเหมาะกับความต้องการของผู้บริโภค และยังสามารถช่วยในการจัดทำโปรโมชันสำหรับสินค้าที่ถูกซื้อบ่อยๆร่วมกับสินค้าที่ถูกซื้อไม่บ่อยได้อีกด้วย ในส่วนของการพัฒนาการออกแบบเว็บไซต์หรือการดูแลรักษาเว็บไซต์ ผู้ดูแลเว็บไซต์อาจจะสนใจความสม่ำเสมอของการคลิกเพื่อเรียกดูข้อมูลในเว็บเพจที่ต่อเนื่องกันเพื่อนำไปปรับปรุงข้อความหรือเนื้อหาของเว็บไซต์ให้มีความน่าสนใจยิ่งขึ้น ในส่วนของการวิเคราะห์ข้อมูลทางพันธุกรรม กลุ่มของยีนส์ที่ปรากฏบ่อยและสม่ำเสมออาจบ่งบอกถึงข้อมูลที่สำคัญให้แก่นักวิทยาศาสตร์ได้ ในส่วนตลาดหุ้น กลุ่มของหุ้นที่มีดัชนีที่มีการเพิ่มขึ้นอย่างสม่ำเสมออาจจะได้รับความน่าสนใจจากนักลงทุนต่างๆ และ อื่นๆ

ในการหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ ผู้ใช้จะต้องทำการกำหนดค่าพารามิเตอร์ 2 ค่าด้วยกันคือ 1) ค่าขีดแบ่งสนันสนุน และ 2) ค่าขีดแบ่งความสม่ำเสมอ เพื่อใช้วัดความน่าสนใจหรือความสำคัญของรูปแบบภายใต้พฤติกรรมการเกิดขึ้นของรูปแบบเหล่านั้น แต่อย่างไรก็ดี เป็นที่ทราบกันดีว่า “ถ้าเราไม่ได้มีความรู้ในข้อมูลมาก่อน การกำหนดค่าขีดแบ่งสนับสนุนเพื่อที่จะได้รับรูปแบบที่น่าสนใจและมีความสำคัญมากที่สุดจะเป็น

เรื่องที่ยุ่งยากและลำบาก” โดยที่ถ้าเรากำหนดค่าขีดแบ่งสนับสนุนสูงเกินไป อาจทำให้เราได้ผลลัพธ์เป็นจำนวนน้อยหรืออาจจะไม่ได้ผลลัพธ์เลย ในกรณีนี้ เราจำเป็นต้องคาดเดาค่าขีดแบ่งให้มีค่าน้อยลงแล้วทำการค้นหาผลลัพธ์ใหม่อีกครั้งที่ซึ่งอาจจะได้รับหรือไม่ได้รับผลลัพธ์ที่ดีขึ้นก็เป็นได้ แต่ในกรณีที่ค่าขีดแบ่งถูกกำหนดให้มีค่าน้อย อาจทำให้เราได้ผลลัพธ์ออกมาเป็นจำนวนมากเกินกว่าที่เราจะทำการพิจารณาองค์ความรู้ได้ และการค้นหาผลลัพธ์จะใช้เวลาค่อนข้างมากอีกด้วย

จากปัญหาข้างต้นดังกล่าว จึงมีงานวิจัยที่ทำการพัฒนาต่อยอดจากงานของ Tanbeer โดยมีวัตถุประสงค์ที่จะหลีกเลี่ยงการกำหนดค่าขีดแบ่งสนับสนุน โดยการกำหนดให้ผู้ใช้ทำการกำหนดจำนวนผลลัพธ์ (รูปแบบ) ที่ต้องการแทนด้วยการนำแนวคิดของการค้นหารูปแบบที่ปรากฏขึ้นบ่อยสุดเคอันดับแรกมาประยุกต์ใช้ (Fu, 2000)(Wang, 2005)(Yang, 2008)(Li, 2009)(Ke, 2009)(Fournier-Viger, 2013) เป็นต้น โดยในงานวิจัยนั้นได้เสนอปัญหาการค้นหารูปแบบที่ปรากฏขึ้นบ่อยและสม่ำเสมอเคอันดับแรก (Mining top-k frequent-regular pattern) (Amphawan, 2009) เพื่อทำการหารูปแบบทั้งสิ้นเครูปแบบ ที่ซึ่งปรากฏในฐานข้อมูลอย่างสม่ำเสมอและปรากฏบ่อยที่สุด ภายใต้ปัญหานี้ ผู้ที่ต้องการค้นหารูปแบบจะต้องทำการกำหนดค่าพารามิเตอร์ 2 ค่าด้วยกัน คือ 1) ค่าขีดแบ่งความสม่ำเสมอ (σ_r) และ 2) จำนวนผลลัพธ์ที่ต้องการ (k) โดยในการค้นหารูปแบบดังกล่าวได้อย่างรวดเร็ว ผู้วิจัยได้เสนอ 3 อัลกอริทึมที่มีประสิทธิภาพ ได้แก่ MTKPP (Amphawan, 2009), TR-CT (Amphawan, 2011) และ TKRIMPE (Amphawan, 2012) ตามลำดับ นอกเหนือจากหลีกเลี่ยงความยุ่งยากในการกำหนดค่าขีดแบ่งสนับสนุน การค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอได้ถูกพัฒนาอย่างต่อเนื่องในหลายๆแง่มุม อาทิเช่น การค้นหารูปแบบที่ปรากฏบ่อยและสม่ำเสมอจากฐานข้อมูลที่มีการเพิ่มเติมข้อมูล (Mining frequent-regular patterns on incremental transactional databases)(Tanbeer, 2010) และจากฐานข้อมูลที่เป็นแบบสายข้อมูล (Mining frequent-regular patterns on data stream)(Tanbeer, 2010) การหารูปแบบที่เกิดขึ้นอย่างสม่ำเสมอที่ประกอบด้วยรูปแบบที่ปรากฏบ่อยและปรากฏไม่บ่อย (Mining frequent-regular patterns consisting of both frequent and rare items)(Surana, 2012), การค้นหารูปแบบที่ปรากฏบ่อยและสม่ำเสมอด้วยการกำหนดเงื่อนไขเกี่ยวกับค่าสนับสนุน (Mining periodic-frequent patterns with maximum items' support constraints)(Kiran, 2010), และ อื่นๆ

ในส่วนของการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงจะเริ่มจาก H. Yao และชาวคณะ (Yao, 2004) ได้นำเสนอปัญหาและนิยามการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงที่ซึ่งจะทำการวัดความน่าสนใจหรือความสำคัญ

ของรูปแบบจากความมีประโยชน์ของรูปแบบนั้นๆ โดยความมีประโยชน์อาจเกี่ยวข้องกับ ผลกำไร ยอดขาย หรือต้นทุน เป็นต้น โดยรูปแบบที่มีค่าคุณประโยชน์สูงจะมีบทบาทสำคัญในการตัดสินใจต่างๆในการดำเนินธุรกิจ อาทิ เช่น การเพิ่มรายได้ การลดค่าใช้จ่ายด้านการตลาดและการจัดการคลังสินค้า และอื่นๆ นอกเหนือจากการประยุกต์ใช้ในแวดวงธุรกิจ การค้นหารูปแบบในลักษณะนี้ยังสามารถนำไปประยุกต์ใช้กับงานในหลายๆด้าน อาทิ เช่น การวิเคราะห์ทางพันธุกรรม(Biological gene analysis) การเข้าถึงเว็บแบบมีลำดับ(Web-click sequence analysis) การตรวจสอบการขึ้นลงของดัชนีหุ้น การวัดประสิทธิภาพของการจราจร การตรวจสอบการเข้าใช้งานเซิร์ฟเวอร์ การวิเคราะห์ข้อมูลที่ได้จากเซ็นเซอร์เน็ตเวิร์ค และ การวิเคราะห์ข้อมูลการใช้โทรศัพท์ทางไกล เป็นต้น โดยในตอนเริ่มต้น (Yao, 2004)(Yao, 2006)(Liu, 2005) ได้พยายามที่จะคิดต้นขั้นตอนวิธีที่จะทำการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงได้อย่างมีประสิทธิภาพ โดยพยายามที่จะคิดค้นขั้นตอนวิธีที่จะลดทอนปริมาณข้อมูลหรือจำนวนรูปแบบที่ต้องทำการพิจารณา (itemset lattice หรือ search space of itemsets) แต่อย่างไรก็ตามขั้นตอนที่คิดค้นขึ้นยังคงทำการอ่านข้อมูลจากฐานข้อมูลอยู่หลายครั้ง ที่ซึ่งทำให้ใช้เวลาในการคำนวณค่อนข้างมาก ต่อมา (Tseng, 2010) ได้ทำการคิดค้นโครงสร้างต้นไม้ที่มีชื่อว่า UP-tree (Utility Pattern Tree) เพื่อใช้ในการจัดเก็บข้อมูลรูปแบบมีค่าคุณประโยชน์สูงระหว่างการคำนวณ โดยที่การใช้โครงสร้างต้นไม้ดังกล่าวจะสามารถลดจำนวนครั้งในการอ่านฐานข้อมูลเหลือเพียง 3 ครั้งเท่านั้น แต่อย่างไรก็ดีในปี 2012 (Liu, 2012) ได้เสนอโครงสร้างข้อมูลที่มีชื่อว่า Utility-list ที่ซึ่งเป็นลิงค์ลิสที่ใช้ในการจัดเก็บรูปแบบมีค่าคุณประโยชน์สูงในระหว่างการคำนวณ ซึ่งจากการประยุกต์ใช้ utility-list จะทำให้สามารถลดการอ่านฐานข้อมูลเหลือเพียง 2 ครั้งเท่านั้น

จากที่ได้กล่าวมาทั้งหมดการค้นหารูปแบบที่มีคุณประโยชน์สูงยังคงได้รับความสนใจจากนักวิจัยเป็นจำนวนมากที่ซึ่งพยายามพัฒนาขั้นตอนวิธีสำหรับข้อมูลที่มีการเพิ่มข้อมูลทรานแซกชัน (Incremental transactional database) และ ฐานข้อมูลแบบสตรีม (Data stream) เป็นต้น

บทที่ 3
วิธีดำเนินการวิจัย

จากที่กล่าวข้างต้น การค้นหารูปแบบปรากฏบ่อยและปรากฏสม่ำเสมอจะไม่สามารถบ่งบอกถึงความสำคัญหรือคุณประโยชน์ของรูปแบบได้ แต่สำหรับการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงจะไม่สามารถบ่งบอกถึงพฤติกรรมการปรากฏขึ้นของรูปแบบได้ ด้วยเหตุนี้ ในบทนี้จะนำเสนอการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏสม่ำเสมอจากฐานข้อมูลรายการ (mining high-utility regular itemsets from transactional database) ที่ซึ่งจะทำการพิจารณาค่าคุณประโยชน์พร้อมกับพฤติกรรมการปรากฏของรูปแบบภายใต้ความสม่ำเสมอ โดยรูปแบบในลักษณะนี้จะสามารถบ่งบอกถึงพฤติกรรมผู้บริโภคที่เกี่ยวเนื่องกับการซื้อสินค้าที่ให้ผลกำไรสูงโดยสินค้าเหล่านั้นถูกซื้ออย่างสม่ำเสมอ จากองค์ความรู้ดังกล่าว จะทำให้ทราบถึงความต้องการสินค้าของผู้บริโภค และมีส่วนช่วยเป็นข้อมูลประกอบการตัดสินใจเกี่ยวกับการบริหารจัดการคลังสินค้า การจัดทำโปรโมชันเพื่อส่งเสริมการขาย และอื่น ๆ

โดยในตอนเริ่มต้นจะเป็นส่วนของการให้นิยามเกี่ยวกับปัญหาการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างสม่ำเสมอ จากนั้นจะเป็นการนำเสนอขั้นตอนวิธี HURI-UL ซึ่งประยุกต์ใช้โครงสร้างข้อมูล utility list ในการจัดเก็บข้อมูลระหว่างการประมวลผล และนำเสนอขั้นตอนวิธี MHUIRA ซึ่งประยุกต์ใช้โครงสร้างข้อมูล new utility list ที่ซึ่งพัฒนา/ดัดแปลงมากจากโครงสร้างข้อมูล utility list ที่ซึ่งจะช่วยให้สามารถประมวลผลได้รวดเร็วยิ่งขึ้น

3.1 นิยามที่เกี่ยวข้องกับการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏสม่ำเสมอ

กำหนดให้

- เซต $I = \{i_1, i_2, ..., i_m\}$ เป็นเซตของรายการ (items)
- แต่ละรายการ $i_j \in I$ จะมีค่าคุณประโยชน์ต่อ i_j หนึ่งๆ (เรียกว่า external utility) อาทิเช่น ผลกำไรจากการสินค้าชิ้นหนึ่งๆ ต้นทุนของสินค้าชิ้นหนึ่งๆ หรือ อื่นๆ ซึ่งจะแทนด้วยสัญลักษณ์ $eu(i_j)$
- เซต $X = \{i_p, i_{p+1}, ...,i_q\} \subseteq I$ จะเรียกว่าว่าเป็น เซตรายการ (set of items or an itemset) และ จะเรียกว่า k-itemset หรือ k-patterns เมื่อ เซต X ประกอบไปด้วยรายการทั้งสิ้น k รายการ
- *TDB* = $\{t_1, t_2, ..., t_n\}$ คือ ฐานข้อมูลรายการหรือฐานข้อมูลแบบทรานแซกชัน (transactional database) ที่ซึ่งแต่ละทรานแซกชัน $t_p \in TDB$ จะประกอบด้วยหมายเลขกำกับทรานแซกชัน (unique transaction identifier, *tid*) tid = p และ เซตของรายการ Y ที่ถูกบรรจุอยู่ในทรานแซกชันนั้นๆ โดยที่แต่ละ $i_j \in Y$ จะมีจำนวนชิ้นของ i_j ที่ปรากฏในทรานแซกชันนั้นๆ (เรียกว่า internal utility) สามารถแทนได้ด้วย $iu(i_j, t_p)$
- ถ้า $X \subseteq Y$ ของ t_p เราจะสามารถสรุปได้ว่าเซตรายการ X ปรากฏขึ้นในทรานแซกชัน t_p หรือ t_p มี X อยู่ในทรานแซกชัน ที่ซึ่งเราสามารถเขียนสัญลักษณ์แทนได้ว่า t^X_p ดังนั้นเมื่อทำการตรวจสอบรูปแบบ X ว่าปรากฏขึ้นในทรานแซกชันใดบ้างในฐานข้อมูล *TDB* เราจะทราบถึง $T^X = \{t^X_p, t^X_{p+1}, ..., t^X_q\}$ ซึ่งก็คือ เซตของหมายเลขทรานแซกชัน (tid) ที่ถูกเรียงลำดับที่ซึ่งมี X อยู่ในทรานแซกชัน (สามารถเขียนย่อๆได้เป็น tidset)
- จำนวนครั้งในการปรากฏขึ้นของรูปแบบ X ในฐานข้อมูล โดยสามารถคำนวณได้เป็น $s^X = |T^X|$
- ค่าคุณประโยชน์ของรายการ i_j ที่ปรากฏในทรานแซกชัน t_p จะเป็นผลคูณระหว่าง จำนวนชิ้นของ i_j ที่ปรากฏในทรานแซกชัน t_p กับค่าคุณประโยชน์ต่อ i_j สามารถคำนวณและแทนด้วยสัญลักษณ์ $u(i_j, t_p) = iu(i_j, t_p) \times eu(i_j)$

- ค่าคุณประโยชน์ของเซตรายการ X ที่ปรากฏในทรานแซกชัน t_p จะเป็นผลรวมของค่าคุณประโยชน์ของทุกรายการที่เป็นสมาชิกของเซตรายการ X ที่ปรากฏในทรานแซกชัน t_p สามารถคำนวณและแทนด้วยสัญลักษณ์ $\mathrm{u}(\mathrm{X}, \mathrm{t}_p) = \sum_{i_j \in X} iu(i_j, t_p) \times \mathrm{eu}(i_j)$
- ค่าคุณประโยชน์ของเซตรายการ X ที่ปรากฏในฐานข้อมูลรายการ *TDB* จะเป็นผลรวมของค่าคุณประโยชน์ของเซตรายการ X ที่ปรากฏในทุกทรานแซกชันของฐานข้อมูลรายการ *TDB* สามารถคำนวณและแทนด้วยสัญลักษณ์ $\mathrm{u}(\mathrm{X}) = \sum_{i_j \in X, X \subseteq t_p} \mathrm{u}(\mathrm{X}, \mathrm{t}_p)$

จากนิยามและสัญลักษณ์ทั้งหมดข้างต้น ปัญหาการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงจะเป็นการค้นหารูปแบบ X ใดๆ ที่มีค่าคุณประโยชน์ $\mathrm{u}(\mathrm{X})$ มากกว่าหรือเท่ากับค่าขีดแบ่งคุณประโยชน์ σ_u ที่ผู้ใช้กำหนด แต่อย่างไรก็ตาม การค้นหารูปแบบตามที่มีคุณสมบัติข้างต้นจะมีอุปสรรคตรงที่ไม่สามารถประยุกต์ใช้ downward closure property ในการลดทอนปริภูมิสถานะ เนื่องจากรูปแบบที่เป็นซุปเปอร์เซตของรูปแบบที่มีค่าคุณประโยชน์ต่ำอาจมีค่าคุณประโยชน์สูง นี้จึงเป็นเหตุเราไม่สามารถตัดรูปแบบที่มีค่าคุณประโยชน์ต่ำออกจากการพิจารณาได้ ด้วยเหตุนี้ Liu (Liu, 2005) จึงได้เสนอแนวคิดเกี่ยวกับ “transaction-weighted utility (twu)” ที่ซึ่งจะเป็นค่าประมาณของค่าคุณประโยชน์ที่ทำให้สามารถประยุกต์ใช้สามารถประยุกต์ใช้ downward closure property ในการลดทอนปริภูมิสถานะได้ โดย twu จะสามารถนิยามได้ดังนี้

- ค่าคุณประโยชน์ของทรานแซกชัน t_p จะเป็นผลรวมของค่าคุณประโยชน์ของทุกๆรายการที่ปรากฏในทรานแซกชัน t_p สามารถคำนวณและแทนด้วยสัญลักษณ์ $\mathrm{tu}(\mathrm{t}_p) = \sum_{i_j \in t_p} \mathrm{u}(i_j, \mathrm{t}_p)$
- ค่า twu ของเซตรายการ X ในฐานข้อมูลรายการ *TDB* จะเป็นค่าประมาณคุณประโยชน์ของ X ที่เกิดจากผลรวมของค่าคุณประโยชน์ของทุกทรานแซกชันในฐานข้อมูลรายการ *TDB* ที่มี X ปรากฏ สามารถคำนวณและแทนด้วยสัญลักษณ์ $\mathrm{twu}(\mathrm{X}) = \sum_{X \in D, X \subseteq t_p} \mathrm{tu}(\mathrm{t}_p)$

จากนิยามเกี่ยวกับ twu ของรูปแบบหนึ่งๆ จะทำให้เราสามารถประยุกต์ใช้ downward closure property ในการลดทอนปริภูมิสถานะได้ ก็ต่อเมื่อ เซตรายการ X ใดๆ มีค่า twu(X) น้อยกว่าค่าขีดแบ่งคุณประโยชน์แล้ว จะทำให้ทุกๆเซตรายการที่เป็นซุปเปอร์เซตของ X จะมีค่าคุณประโยชน์น้อยกว่าค่าขีดแบ่งคุณประโยชน์ด้วยเช่นกัน นี้จึงเป็นเหตุให้ เราสามารถตัดการพิจารณาเซตรายการ X และเซตรายการ

ที่เป็นซุปเปอร์เซตของ X ออกจากการพิจารณาได้ เนื่องจาก เซตรายการ X และ ทุกๆเซตรายการที่เป็นซุปเปอร์เซตของ X จะมีค่าคุณประโยชน์น้อย

แม้ว่า twu ของรูปแบบจะมีส่วนช่วยในการลดทอนปริภูมิสถานะได้ แต่อย่างไรก็ตาม twu ของรูปแบบหนึ่งๆจะเป็นค่าประมาณคุณประโยชน์ที่มีค่าสูงกว่าค่าคุณประโยชน์จริงค่อนข้างมาก ด้วยเหตุนี้จึงเป็นเหตุให้ Liu (Liu, 2012) คิดค้นแนวความคิดเกี่ยวกับค่าประมาณคุณประโยชน์ที่มีความกระชับ (มีค่าเกินจริงน้อยกว่าค่า twu) ที่ซึ่งสามารถนิยามได้ดังนี้

- กำหนดให้ $>$ แสดงถึงลำดับของรายการในเซตรายการ I
- ค่าคุณประโยชน์ส่วนที่เหลือ (remaining utility) ของเซตรายการ X ในทรานแซกชัน t_p หมายถึงผลรวมของค่าคุณประโยชน์ของทุกรายการที่ปรากฏในทรานแซกชัน t_p และรายการเหล่านั้นมีลำดับหลังจาก X สามารถคำนวณและแทนด้วยสัญลักษณ์ $\mathrm{ru}(\mathrm{X}, \mathrm{t}_p) = \sum_{i_j \in t_p, X<i_j} \mathrm{u}(i_j, \mathrm{t}_p)$
- ค่าคุณประโยชน์ส่วนที่เหลือของเซตรายการ X ในฐานข้อมูลรายการ *TDB* จะเป็นค่าผลรวมของค่าคุณประโยชน์ส่วนที่เหลือของเซตรายการ X ในทุกทรานแซกชัน ที่มี X ปรากฏ สามารถคำนวณและแทนด้วยสัญลักษณ์ $\mathrm{ru}(\mathrm{X}) = \sum_{X \in D, X \subseteq t_p} \mathrm{ru}(\mathrm{X}, \mathrm{t}_p)$
- ค่าประมาณคุณประโยชน์แบบกระชับของเซตรายการ X ในฐานข้อมูล *TDB* จะเป็นค่าผลรวมระหว่างค่าคุณประโยชน์จริงของเซตรายการ X กับค่าคุณประโยชน์ส่วนที่เหลือของเซตรายการ X ในฐานข้อมูลรายการ *TDB* สามารถคำนวณและแทนด้วยสัญลักษณ์ $\mathrm{ou}(\mathrm{X}) = \mathrm{u}(\mathrm{X}) + \mathrm{ru}(\mathrm{X})$

จากแนวคิดและนิยามข้างต้นเราสามารถบอกได้ว่า ถ้าเซตรายการ X มีค่าประมาณคุณประโยชน์แบบกระชับน้อยกว่าค่าขีดแบ่งคุณประโยชน์แล้ว เซตรายการใดที่เกิดจากการรวมกันระหว่างเซตรายการ X และรายการ i_j ใดๆที่มีลำดับหลังจาก X จะมีค่าคุณประโยชน์น้อยกว่าค่าขีดแบ่งคุณประโยชน์เสมอ ซึ่งจากข้อสรุปดังกล่าวจะทำให้เราสามารถลดทอนการพิจารณาเซตรายการ X และซุปเปอร์เซตของ X ที่เกิดจากการรวมกันระหว่างเซตรายการ X และรายการ i_j ใดๆที่มีลำดับหลังจาก X ได้ นี้อันนำมาซึ่งการลดทอนปริภูมิสถานะ

ในการที่จะศึกษาถึงพฤติกรรมการปรากฏขึ้นของรูปแบบว่ามีพฤติกรรมการปรากฏขึ้นอย่างสม่ำเสมอหรือไม่ เราจะต้องทำการพิจารณาเซตของหมายเลขทรานแซกชันที่มี X ปรากฏขึ้น โดยเริ่มจากการพิจารณาแต่ละคู่ของหมายเลขทรานแซกชัน t^X_j และ t^X_{j+1} ที่อยู่ในลำดับติดกันในเซต T^X แล้วทำการหาจำนวนทรานแซกชันที่ไม่มี

X ปรากฏระหว่างสองทรานแซกชันนั้นๆ ที่ซึ่งสามารถคำนวณได้เป็น $rtt^X j = t^X_{j+1} - t^X_j$ แต่สำหรับการปรากฏขึ้นในครั้งแรกและครั้งสุดท้ายของ X จะมีวิธีการคำนวณที่แตกต่างจากการปรากฏขึ้นครั้งอื่นๆที่ซึ่งสามารถคำนวณได้เป็น $fr^X = t^X_1$ (เมื่อ t^X_1 คือ หมายเลขทรานแซกชันที่ X ปรากฏขึ้นครั้งแรก) และ $lr^X = n - t^X_{|T^X|}$ (เมื่อ n คือจำนวนทรานแซกชันทั้งหมดในฐานข้อมูล และ $t^X_{|T^X|}$ คือ หมายเลขทรานแซกชันที่ X ปรากฏขึ้นครั้งสุดท้าย) จากที่กล่าวข้างต้น เราสามารถหาจำนวนทรานแซกชันที่ติดกันสูงที่สุดที่ไม่มี X ปรากฏ โดยสามารถคำนวณได้เป็น $r^X = \max(fr^X, rtt^X_1, rtt^X_2, ..., lr^X)$ ที่ซึ่งสามารถบอกได้ถึงช่วงเวลาที่ยาวนานที่สุดที่ไม่มี X ปรากฏขึ้นในฐานข้อมูล และยังสามารถการันตีได้ว่าเซตรายการ X จะปรากฏขึ้นอย่างน้อยหนึ่งครั้งในทุกๆ r^X ทรานแซกชันที่เรียงต่อกัน ซึ่งจากการพิจารณาค่า r^X จะทำให้ทราบถึงพฤติกรรมการปรากฏขึ้นของ X ได้ ดังนั้น เซตรายการ X ใดๆจะเป็นเซตรายการที่ปรากฏสม่ำเสมอก็ต่อเมื่อ ค่าความสม่ำเสมอ r^X มีค่าน้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอ σ_r ที่ผู้ใช้กำหนด จากนิยามทั้งหมดข้างต้น รูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏสม่ำเสมอสามารถนิยามได้ดังนี้

นิยาม เซตรายการ X หนึ่งๆจะเป็นเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏสม่ำเสมอก็ต่อเมื่อ X มีค่าคุณประโยชน์ $u(X)$ ไม่น้อยกว่า σ_u และ X มีค่าความสม่ำเสมอ r^X ไม่เกินกว่า σ_r ที่ผู้ใช้กำหนด

3.2 ขั้นตอนวิธีที่นำเสนอ HURI-UL

ภายใต้การพิจารณาเกี่ยวกับปัญหาการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏสม่ำเสมอจากฐานข้อมูลรายการภายใต้ค่าขีดแบ่ง σ_u และ σ_r ที่ผู้ใช้กำหนด ผู้วิจัยได้นำเสนอขั้นตอนวิธีที่มีชื่อว่า HURI-UL ที่ซึ่งสามารถลดทอนการอ่านข้อมูลจากฐานข้อมูล (วิธีอื่นๆต้องอ่านข้อมูลจากฐานข้อมูล 2 ครั้ง) ให้ทำการอ่านข้อมูลเพียงครั้งเดียว โดยทำการจัดเก็บค่าคุณประโยชน์ของทุกทรานแซกชันในฐานข้อมูลไว้ในอะเรย์ที่เรียกว่า tu-List นอกจากนั้น HURI-UL ได้ประยุกต์ใช้แนวคิดค่าประมาณคุณประโยชน์แบบกระชับของเซตรายการในการลดทอนและประยุกต์ใช้โครงสร้างข้อมูล utility list เพื่อทำการจัดเก็บข้อมูลการปรากฏพร้อมกับค่าคุณประโยชน์ของรายการหนึ่งๆที่ปรากฏในทรานแซกชันหนึ่งๆ จากการประยุกต์ใช้โครงสร้างข้อมูลดังกล่าว จะทำให้ HURI-UL สามารถค้นหารูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏสม่ำเสมอจากฐานข้อมูลรายการได้อย่างมีประสิทธิภาพ

3.2.1 โครงสร้างข้อมูล Utility list

จากการกำหนดลำดับของรายการที่เป็นสมาชิกของเซตรายการ $I = \{i_1, i_2, ..., i_n\}$ เราสามารถสร้าง utility list ของเซตรายการ X จากฐานข้อมูล TDB ได้เป็นลิสต์ของ 3-tuple คือ $\langle p, u(X, t_p), ru(X, t_p)\rangle$ เมื่อ 1) p คือ

หมายเลขทรานแซกชันที่มีเซตรายการ X ปรากฏ 2) u(X, t_p) คือ ค่าคุณประโยชน์ของเซตรายการ X ในทรานแซกชัน t_p และ 3) ru(X, t_p) คือ ค่าคุณประโยชน์ส่วนที่เหลือ (remaining utility) ของเซตรายการ X ในทรานแซกชัน t_p ตัวอย่างเช่น พิจารณารายการ ‘a’ ที่มีค่าคุณประโยชน์ในตารางที่ 4 เท่ากับ 3 และปรากฏในฐานข้อมูลในตารางที่ 5 เป็น $T^a = \{t_1, t_3, t_5, t_7, t_8\}$ ตามลำดับ ถ้าลำดับของรายการทั้งหมดเป็น a < b < c < d < e < f < g < f เราจะสามารถสร้าง utility list ของรายการ ‘a’ ได้เป็น {<1, 9, 73>, <3, 6, 64>, <5, 6, 19>, <7, 15, 42>, <8, 9, 41>} โดยที่สมาชิกอันดับแรกของ utility list จะบ่งบอกได้ว่า รายการ ‘a’ ปรากฏในทรานแซกชันที่ 1 รายการ ‘a’ มีค่าคุณประโยชน์ในทรานแซกชันที่ 1 เท่ากับ 9 และมีค่าคุณประโยชน์ส่วนเหลือในทรานแซกชันที่ 1 เท่ากับ 73 ตามลำดับ แต่สำหรับสมาชิกลำดับอื่นๆของ utility list ก็จะบ่งบอกถึงการปรากฏขึ้นของรายการ ‘a’ หลังจากทรานแซกชันที่ 1 ตามลำดับ

ตารางที่ 4 ตัวอย่างตารางแสดงค่าคุณประโยชน์ของแต่ละรายการ

รายการ	a	b	c	d	e	f	g	f
ค่าคุณประโยชน์	3	2	1	30	5	3	4	15

ตารางที่ 5 ตัวอย่างฐานข้อมูลรายการที่ประกอบไปด้วยหมายเลขทรานแซกชันและเซตรายการที่ปรากฏในทรานแซกชันที่มีการจำนวนของการปรากฏขึ้นของแต่ละรายการ

หมายเลขทรานแซกชัน (tid)	เซตรายการที่ปรากฏในทรานแซกชัน (a set of items or an itemset)
1	a(3), c(8), d(2), e(1)
2	b(5), f(3), g(5), h(20)
3	a(2), c(4), d(1)
4	c(5), e(1), f(1)
5	a(2), b(3), c(1), f(4)
6	d(1), g(5), h(1)
7	a(5), b(1) c(4)
8	b(4) c(1) d(3) e(2)
9	a(3) b(2) d(4) e(1)
10	a(2) e(7)

3.2.2 ขั้นตอนวิธี HURI-UL

ดังที่กล่าวข้างต้น ขั้นตอนวิธี HURI-UL ไม่เพียงแต่ประยุกต์แนวความคิดค่าคุณประโยชน์ส่วนเหลือ และค่าประมาณคุณประโยชน์แบบกระชับเพื่อทำการลดทอนปริภูมิสถานะ แต่ยังทำการประยุกต์ใช้โครงสร้าง utility list เพื่อใช้ในการจัดเก็บข้อมูลเกี่ยวกับการปรากฏขึ้นของรายการ/เซตรายการหนึ่งๆ พร้อมกับค่าคุณประโยชน์ของรายการ/เซตรายการนั้นๆที่ปรากฏในทรานแซกชันหนึ่งๆ ระหว่างการค้นหาผลลัพธ์ นอกจากนั้น HURI-UL ยังทำการลดทอนการอ่านฐานข้อมูลให้เหลือเพียงครั้งเดียวด้วยการใช้อะเรย์หนึ่งมิติสำหรับจัดเก็บค่าคุณประโยชน์ของทุกทรานแซกชัน (เรียกว่า tu-List) โดยในการหาผลลัพธ์จากขั้นตอนวิธี HURI-UL จะประกอบไปด้วย 2 ขั้นตอนการคำนวณหลักคือ

1) การระบุถึงรายการที่ปรากฏสม่ำเสมอและคาดว่าน่าจะมีค่าคุณประโยชน์สูง (เมื่อรวมกับรายการอื่นๆ) ที่ซึ่งจะทำการสร้างลิสต์หลายมิติเพื่อใช้เก็บรายการและเซตของรายการที่มีคุณสมบัติข้างต้น (เรียกว่า i-List) โดยการระบุถึงรายการที่ปรากฏสม่ำเสมอและคาดว่าน่าจะมีค่าคุณประโยชน์สูง (เมื่อรวมกับรายการอื่นๆ)จากการอ่านฐานข้อมูลหนึ่งครั้งจะถูกจัดเก็บอยู่ในลิสต์มิติแรก (เรียกว่า 1-List)
2) การค้นหาผลลัพธ์จากลิสต์ที่สร้างขึ้นในขั้นตอนแรก

ดังแสดงในรูปที่ 1 การระบุถึงรายการที่ปรากฏสม่ำเสมอและคาดว่าน่าจะมีค่าคุณประโยชน์สูง (เมื่อรวมกับรายการอื่นๆ) จะเริ่มจากการสร้างอะเรย์ tu-List เพื่อใช้ในการจัดเก็บค่าคุณประโยชน์ของทุกทรานแซกชันในฐานข้อมูล ทำการสร้างลิสต์หลายมิติ i-List และทำการสร้างลิสต์ที่ใช้สำหรับจัดเก็บรายการที่ปรากฏสม่ำเสมอและคาดว่าน่าจะมีค่าคุณประโยชน์สูง (เมื่อรวมกับรายการอื่นๆ) 1-List (บรรทัดที่ 1) จากนั้นแต่ละทรานแซกชัน t_p ในฐานข้อมูลจะถูกอ่านเพื่อทำการคำนวณค่าคุณประโยชน์ $tu(t_p)$ และ จัดเก็บ $tu(t_p)$ ใน tu-List ต่อมา HURI-UL จะทำการพิจารณาแต่ละรายการ i_j ที่ปรากฏในทรานแซกชัน t_p และทำการอัพเดท utility list ของรายการ i_j ที่ถูกจัดเก็บใน 1-List ด้วย $\langle p, u(i_j, t_p), 0\rangle$ พร้อมกับทำการคำนวณค่าความสม่ำเสมอจากการปรากฏขึ้นของ i_j ในทรานแซกชัน t_p (บรรทัดที่ 2 -4)

ขั้นตอนต่อไปจะเป็นการตรวจสอบการปรากฏขึ้นของแต่ละรายการ i_j ที่ถูกจัดเก็บใน 1-List ว่ามีความสม่ำเสมอหรือไม่? โดยทำการตรวจสอบค่าความสม่ำเสมอ $\boldsymbol{r^{i_j}}$ ของรายการ i_j ว่ามีค่ามากกว่าค่าขีดบางความสม่ำเสมอ σ_r ที่ผู้ใช้กำหนดหรือไม่ ถ้าค่าความสม่ำเสมอ $\boldsymbol{r^{i_j}}$ มีค่ามากกว่า σ_r HURI-UL จะลบรายการ i_j ออกจากการพิจารณา แต่ก่อนที่จะทำการลบ i_j ออกจากการพิจารณาจะทำการลดทอนค่าคุณประโยชน์ของแต่ละท

รานแซกชันที่มีรายการ i_j ปรากฏเพื่อทำการลดทอนค่าประมาณคุณประโยชน์ โดยทำการพิจารณาแต่ละสมาชิกใน utility list ของรายการ i_j ที่ซึ่งมีลักษณะเป็น $<p, u(i_j, t_p), 0>$ และทำการลดทอนค่า $tu(t_p)$ ที่ถูกจัดเก็บใน tu-List ด้วยค่า $u(i_j, t_p)$ และเมื่อทำการพิจารณาทุกสมาชิกใน utility list ของรายการ i_j แล้ว จะสามารถลบข้อมูลทั้งหมดของ i_j ที่ถูกจัดเก็บอยู่ใน 1-list ออกจากหน่วยความจำและการพิจารณาได้ (บรรทัดที่ 6 – 10)

ขั้นตอนต่อไปจะเป็นการคำนวณค่าคุณประโยชน์ส่วนเหลือในแต่ละสมาชิกใน utility list ของแต่ละรายการ i_j รวมถึงทำการคำนวณหาค่าคุณประโยชน์ที่แท้จริง $u(i_j)$ ของรายการ i_j และ ทำการคำนวณค่าคุณประโยชน์ส่วนเหลือ $ru(i_j)$ ของรายการ i_j (บรรทัดที่ 12 – 20) โดยในการพิจารณาจะทำการพิจารณาทีละรายการ i_j (จากลำดับของรายการที่ทราบก่อนหน้าแล้ว) จากนั้นทำการพิจารณาแต่ละสมาชิก $<p, u(i_j, t_p), 0>$ ใน utility list ของรายการ i_j จากนั้นทำการอัพเดทค่าคุณประโยชน์ของทรานแซกชัน t_p ที่ถูกจัดเก็บอยู่ใน tu-List ด้วย $tu(i_j, t_p) = tu(i_j, t_p) - u(i_j, t_p)$ เพื่อที่จะทราบถึงค่าคุณประโยชน์ของทุกรายการในทรานแซกชันที่อยู่ในลำดับถัดไปจาก i_j ซึ่งค่า $tu(i_j, t_p)$ หลังการอัพเดทจะหมายถึงค่า $ru(i_j, t_p)$ โดยเมื่อทราบถึงค่าดังกล่าว จะทำการอัพเดทสมาชิกใน utility list ของรายการ i_j ที่พิจารณาให้มีค่าเป็น $<p, u(i_j, t_p), ru(i_j, t_p)>$ และทำการอัพเดทค่า $u(i_j)$ และ $ru(i_j)$ ของรายการ i_j ด้วย $u(i_j, t_p)$ และ $ru(i_j, t_p)$ ตามลำดับ และท้ายสุด ถ้าค่า $u(i_j)$ ของรายการ i_j มีค่าไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด HURI-UL จะทำการระบุและจัดเก็บรายการ i_j ว่าเป็นรูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างสม่ำเสมอ

Algorithm 1 1-HURIs identification

Input: D : transactional database, σ_u : a minimum utility threshold, σ_r : a maximum regularity threshold

Output: *i-List* : a two-dimension list containing 1-HURIs and non 1-HURIs (with potentail to be HURIs with other items) in *1-List*

- create *tu-List*, *i-List* and then initial *1-List* with all single items

for each transaction t in database D **do**
- compute $tu(t)$ and then collect $tu(t)$ in *tu-List*(t)

for each item i in t **do**
- update utility list of i in *1-List* on utiltiy and regularity value

for each item i in *1-List* **do**
if $r^i > \sigma_r$ **then**
for each entry e in utility list of i **do**
- decrease utility value in *tu-List* by e's tid and e's $u(i, t_{tid})$
- remove i and all of its information from *1-List*

- sort *1-List* based on order of items $\succ$

for each item i in *1-List* **do**
- set $u(i)$ and $ru(i)$ to be zero

for each entry $e = \langle tid_e, u(i)_e, ru(i)_e \rangle$ in utility list of item i **do**
- update *tu-List*(tid_e) by *tu-List*$(tid_e) - u(i)_e$
- set $ru(i)_e$ to be equal to *tu-List*$(tid_e) - u(i)_e$
- increase $ru(i)$ by $ru(i)_e$
- increase $u(i)$ by $u(i, t_{tid})$

if $u(i) \geq \sigma_u$ **then**
- HURIs $=$ HURIs $\cup i$

รูปที่ 1 ขั้นตอนการระบุรายการที่มีค่าคุณประโยชน์สูงและปรากฏสม่ำเสมอ

หลังจากทำการสร้างลิสต์ของรายการที่ปรากฏอย่างสม่ำเสมอและมีแนวโน้มที่จะมีค่าคุณประโยชน์สูง (เมื่อรวมกับรายการอื่นๆ) ที่ซึ่งเก็บไว้ใน 1-List แล้ว ขั้นตอนต่อไปจะเป็นการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏสม่ำเสมอทั้งหมดจาก 1-List ที่สร้างขึ้น (ดังแสดงในขั้นตอนวิธีในรูปที่ 2) โดยในขั้นแรก จะเป็นการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏสม่ำเสมอที่ประกอบไปด้วย 2 รายการ โดยเริ่มทำการพิจารณาแต่ละรายการ i_j ที่ถูกจัดเก็บใน 1-List (โดยในการพิจารณาจะพิจารณาแบบเรียงลำดับตามลำดับของรายการที่เป็นสมาชิกของเซต I) จากนั้นจะทำการสร้าง 2-List เพื่อใช้ในการจัดเก็บเซตรายการที่ประกอบด้วย 2 รายการที่ซึ่งเกิดจากการรวมกันระหว่างรายการ i_j กับรายการอื่นๆที่อยู่ในลำดับถัดๆไป โดยในตอนเริ่มแรกจะกำหนดให้ 2-List ไม่มีข้อมูลใดๆ ต่อมาจะเป็นการรวมรายการ i_j เข้ากับรายการ i_k ที่อยู่ในลำดับถัดๆไปจากรายการ i_j เพื่อสร้างการพิจารณาเซตรายการ $i_j \cup i_k$ จากนั้นจะทำการอินเทอร์เซกชัน utility list ของรายการ i_j กับ utility list ของ

รายการ i_k เพื่อทำการจัดเก็บข้อมูลการปรากฏขึ้น ค่าคุณประโยชน์ และค่าคุณประโยชน์ส่วนเหลือสำหรับทรานแซกชันหนึ่งๆ ที่เซตรายการ $i_j \cup i_k$ ปรากฏ หลังจากขั้นตอนอินเทอร์เซกชันเสร็จสิ้น เราจะได้ utility list ของเซตรายการ $i_j \cup i_k$ ที่จะสามารถใช้ utility list ดังกล่าวในการคำนวณค่าคุณประโยชน์ ค่าความสม่ำเสมอ และค่าประมาณคุณประโยชน์แบบกระชับของเซตรายการ $i_j \cup i_k$ ได้

หลังจากคำนวณค่าคุณประโยชน์ ค่าความสม่ำเสมอ และค่าประมาณคุณประโยชน์แบบกระชับของเซตรายการ $i_j \cup i_k$ แล้ว จะทำการตรวจสอบว่า เซตรายการ $i_j \cup i_k$ ปรากฏอย่างสม่ำเสมอหรือไม่ และทำการตรวจสอบว่า เซตรายการ $i_j \cup i_k$ มีแนวโน้มที่จะมีค่าคุณประโยชน์สูงหรือไม่ ถ้าเซตรายการ เซตรายการ $i_j \cup i_k$ ผ่านเงื่อนไขทั้งสองข้อข้างต้น HURI-UL จะทำการจัดเก็บเซตรายการ เซตรายการ $i_j \cup i_k$ พร้อมทั้งข้อมูลทั้งหมดที่เกี่ยวข้องกับเซตรายการ $i_j \cup i_k$ ไว้ใน 2-List เพื่อทำการพิจารณาเซตรายการที่ประกอบด้วย 3 รายการต่อไป โดยหลังจากทำการรวมรายการ i_j เข้ากับ i_k แล้ว ขั้นตอนวิธี HURI-UL จะดำเนินการรวมรายการ i_j เข้ากับ i_l ที่ถูกบรรจุอยู่ใน 1-List ในลำดับหลังจากรายการ i_k และจะดำเนินการตามกระบวนการต่างๆข้างต้น เมื่อทำการรวมรายการ i_j เข้ากับทุกรายการใน 1-List แล้ว เราจะได้ 2-List ที่บรรจุไปด้วยเซตรายการที่ประกอบไปด้วย 2 รายการ และเป็นเซตรายการ i_j เป็นรายการขึ้นต้น จากนั้นจะทำการตรวจสอบจำนวนเซตรายการที่บรรจุอยู่ใน 2-List ซึ่งถ้ามีจำนวนเซตรายการมากกว่า 1 จะทำการวนซ้ำการทำงานเพื่อหาเซตรายการที่ประกอบด้วย 3 รายการ โดยการดำเนินการจะดำเนินการเช่นเดียวกับการพิจารณาเซตรายการที่ประกอบไปด้วย 2 รายการ (ส่วนของ Procedure Gen-Longer-Itemsets)

Algorithm 2 Mining HURIs

Input: *i-List* : a list of itemsets
Output: HURIs : a complete set of HURIs

for each item i in *1-List* **do**
- initial *2-List* to be empty

for each item j in *1-List* ($i \prec j$) **do**
- merge i and j to be itemset Z
- intersect utility lists of u and v to compute r^Z, $u(Z)$, $ru(Z)$, $ou(Z)$ and then to collect occurrence information and utilities in UL^Z

if $r^Z \leq \sigma_r$ or $ou(Z) \geq \sigma_u$ **then**
- create an entry of itemset Z in *2-List with* r^Z, $u(Z)$, $ru(Z)$, $ou(Z)$ and UL^Z

if $u(Z) \geq \sigma_u$ **then**
- HURIs = HURIs $\cup Z$

if |2-List| > 1 **then**
- Gen-Longer-Itemsets(2, *i-List*)

Procedure Gen-Longer-Itemsets(k, *i-List*)
- initial *(k+1)-List* to be empty

for each entry u in *k-List* **do**
for each entry v in *k-List* ($u \prec v$) **do**
- merge itemsets in entry u and v to create itemset Z
- intersect utility lists of u and v to compute r^Z, $u(Z)$, $ru(Z)$, $ou(Z)$ and then to collect occurrence information and utilities in UL^Z

if $r^Z \leq \sigma_r$ or $ou(Z) \geq \sigma_u$ **then**
- create an entry for itemset Z in *(k+1)-List* with r^Z, $u(Z)$, $ru(Z)$, $ou(Z)$ and UL^Z

if $u(Z) \geq \sigma_u$ **then**
- HURIs = HURIs $\cup Z$

if |*(k+1)-List*| > 1 **then**
- Gen-Long-Itemsets(k+1, *i-List*)

รูปที่ 2 ขั้นตอนการหารูปแบบทั้งหมดที่มีค่าคุณประโยชน์สูงและปรากฏสม่ำเสมอ

3.3 ขั้นตอนวิธีที่นำเสนอ MHUIRA

ขั้นตอนวิธี MHUIRA จะเป็นขั้นตอนที่พัฒนามาจากขั้นตอนวิธี HURI-UL ที่ซึ่งจะเป็นการพัฒนาในส่วนของโครงสร้างข้อมูลในการจัดเก็บข้อมูลระหว่างการประมวลผล โดยทำการพัฒนาโครงสร้างข้อมูล utility list ให้ทำการจัดเก็บค่าคุณประโยชน์ของเซตรายการ prefix ของเซตรายการ X ใดๆ ที่ปรากฏในทรานแซกชันหนึ่งๆ (เรียกโครงสร้างที่ถูกพัฒนาขึ้นใหม่ว่า New Utility List structure (NUL)) ซึ่งจากการพัฒนาดังกล่าว จะทำให้ MHUIRA สามารถลดการประมวลผลรวมถึงเวลาในการประมวลผลได้

3.3.1 โครงสร้างข้อมูล New utility list structure (NUL)

จากหัวข้อ 3.2.1 ข้างต้น เราสามารถทราบได้ว่า utility list ของเซตรายการ X ใดๆ จะถูกใช้สำหรับจัดเก็บข้อมูลการปรากฏและค่าคุณประโยชน์ของ และมีลักษณะเป็นลิสต์ของ element เรียงต่อกัน โดยที่แต่ละ element จะประกอบไปด้วย 3-tuple คือ $<p, u(X, t_p), ru(X, t_p)>$ เมื่อ

1) p คือหมายเลขทรานแซกชันที่มีเซตรายการ X ปรากฏ 2
2) $u(X, t_p)$ คือ ค่าคุณประโยชน์ของเซตรายการ X ในทรานแซกชัน t_p และ
3) 3) $ru(X, t_p)$ คือ ค่าคุณประโยชน์ส่วนที่เหลือ (remaining utility) ของเซตรายการ X ในทรานแซกชัน t_p

โดยข้อมูลในลิสต์จะถูกเรียงลำดับตามหมายเลขทรานแซกชันที่มีเซตรายการ X ปรากฏ (เรียงลำดับจากน้อยไปมาก) แต่อย่างไรก็ตาม ถึงแม้ว่า utility list จะสามารถจัดเก็บข้อมูลการปรากฏขึ้น (หมายเลขทรานแซกชันที่เซตรายการปรากฏขึ้น) และค่าคุณประโยชน์ของเซตรายการ X ได้ แต่เมื่อทำการประมวลผลกับ utility list ด้วยการอินเทอร์เซกชันเพื่อจัดเก็บข้อมูลการปรากฏและค่าคุณประโยชน์ของเซตรายการที่มีขนาดใหญ่ขึ้นจะต้องใช้กระบวนการประมวลผลที่ค่อนข้างซับซ้อน และสิ้นเปลืองเวลาในการประมวลผล ตัวอย่างเช่น กำหนดให้เซตรายการ X และ Y จะมีเซตรายการ prefix ที่เหมือนกัน โดยจะมีความแตกต่างกันเฉพาะรายการสุดท้ายเท่านั้น (กล่าวคือ เซตรายการ $X = W \cup i_j$ และ เซตรายการ $Y = W \cup i_k$ ที่ซึ่งเซตรายการ X และ Y มีเซตรายการ prefix ที่เหมือนกันคือเซตรายการ W และจะแตกต่างกันเฉพาะรายการสุดท้ายที่ $i_j \neq i_k$) ดังนั้น เมื่อทำการอินเทอร์เซกชันระหว่าง UL^X ของเซตรายการ X และ UL^Y ของเซตรายการ Y เพื่อทำการคำนวณหาค่าคุณประโยชน์และจัดเก็บข้อมูลการปรากฏของเซตรายการ $X \cup Y$ (เรียกย่อๆว่า เซตรายการ XY) จะไม่สามารถคำนวณหาค่าคุณประโยชน์ที่แท้จริงของเซตรายการ XY ได้โดยตรง แต่เราจะต้องทำการลบทอนค่าคุณประโยชน์ในแต่ละทรานแซกชันที่มีเซตรายการที่เป็น prefix ของเซตรายการ X ปรากฏอยู่ จึงจะสามารถทราบถึงค่าคุณประโยชน์ที่แท้จริงได้ ตัวอย่างเช่น สำหรับทรานแซกชัน t_p ที่มีเซตรายการ X และ Y ปรากฏ จะสามารถคำนวณหาค่าคุณประโยชน์ $u(XY, t_p)$ จะได้จากการอินเทอร์เซกชันกันระหว่าง $<u(X, t_p), ru(X, t_p)>$ และ $<u(Y, t_p), ru(Y, t_p)>$ $= u(X, t_p) + u(Y, t_p) = u(W, t_p) + u(i_j, t_p) + u(W, t_p) + u(i_k, t_p) = (2 \times u(W, t_p)) + u(i_j, t_p) + u(i_k, t_p)$ ซึ่งเราจะสังเกตุได้ว่าเป็นค่าคุณประโยชน์ที่ไม่ถูกต้อง ด้วยเหตุนี้ เราจำเป็นที่ต้องทำการลดทอนค่าคุณประโยชน์ $u(XY, t_p)$ ด้วย $u(W, t_p)$ ซึ่งจะทำให้ค่า คุณประโยชน์ $u(XY, t_p)$ มีค่าเท่ากับ $u(W, t_p) + u(i_j, t_p) + u(W, t_p) + u(i_k, t_p) - u(W, t_p)$

= $u(W, t_p) + u(i_j, t_p) + u(i_k, t_p)$ ซึ่งเป็นค่าที่ถูกต้อง แต่อย่างไรก็ตาม กระบวนการลดทอนค่าคุณประโยชน์เพื่อทำให้ได้ค่าคุณประโยชน์ที่ถูกต้องเป็นขั้นตอนที่ใช้เวลาในการประมวลผลเพิ่มเติมจากการอินเทอร์เซกชันเบื้องต้น และจะทำให้สิ้นเปลืองเวลาในการค้นหารูปแบบที่น่าสนใจ ด้วยเหตุนี้ ผู้วิจัยจึงได้มีแนวคิดที่จะลดทอนกระบวนการดังกล่าว ด้วยการออกแบบ utility list ใหม่ที่มีชื่อเรียกว่า New Utility List (NUL) ให้มีการการจัดเก็บข้อมูลค่าคุณประโยชน์ของเซตรายการ prefix เข้าไปด้วยซึ่งจะทำให้มีการเก็บข้อมูลเพิ่มขึ้น แต่สามารถลดการประมวลผลลงได้ ซึ่งจากการออกแบบดังกล่าวจะทำให้ NUL เป็นลิสต์ของ 4-tuple คือ <p, $u(X, t_p)$, $ru(X, t_p)$, $up (X, t_p)$> เมื่อ

1) p คือหมายเลขทรานแซกชันที่มีเซตรายการ X ปรากฏ

2) $u(X, t_p)$ คือ ค่าคุณประโยชน์ของเซตรายการ X ในทรานแซกชัน t_p

3) $ru(X, t_p)$ คือ ค่าคุณประโยชน์ส่วนที่เหลือ (remaining utility) ของเซตรายการ X ในทรานแซกชัน t_p

4) $up (X, t_p)$ คือ ค่าคุณประโยชน์ของเซตรายการ prefix ของเซตรายการ X ในทรานแซกชัน t_p

ตัวอย่าง 3.X พิจารณาการปรากฏขึ้นของเซตรายการ ‘a, b’ และ ‘a, c’ ในตารางที่ 3-x ถ้าลำดับของรายการทั้งหมดเป็น a < b < c < d < e < f < g < f เราจะสามารถสร้าง new utility list ของ ‘a, b’ และ ‘a, c’ ได้เป็น $NUL^{‘a,b’}$ = { <5, 12 (= (2 × 3) + (3 × 2)), 13 (= (1 × 1) + (4 × 3)), 6 (=2 × 3)>, <7, 17, 4, 15>, <9, 13, 125, 9>} และ $NUL^{‘a,c’}$ = { <1, 17 (= (3 × 3) + (8 × 1)), 65 (= (2 × 30) + (1 × 5)), 9 (=3 × 3)>, <3, 10, 30, 6>, <5, 7, 12, 6>} ตามลำดับ ดังนั้นเมื่อเราต้องการพิจารณาถึงเซตรายการ ‘a, b, c’ เราจะสามารถพิจาณาถึงการปรากฏขึ้นและค่าคุณประโยชน์ของเซตรายการดังกล่าวได้จากการอินเทอร์เซกชันระหว่าง $NUL^{‘a,b’}$ และ $NUL^{‘a,c’}$ ที่ซึ่งจะสามารถสร้าง $NUL^{‘a, b, c’}$ = {<5, 13 (= 12 + 7 - 6), 12, 12>} ที่ซึ่งสามารถบ่งบอกถึงข้อมูลได้อย่างถูกต้อง อาทิเช่น

1) เซตรายการ ‘a, b, c’ ปรากฏในทรานแซกชันที่ 5 ของฐานข้อมูลเท่านั้น
2) ค่าคุณประโยชน์ของเซตรายการ ‘a, b, c’ ในทรานแซกชันที่ 5 จะมีค่าเท่ากับ 13 ซึ่งสามารถคำนวณได้จาก $u(‘a, b’, t_5) + u(‘a, c’, t_5) - u(‘a’, t_5) = 12 + 7 – 6 = 13$

3) ค่าคุณประโยชน์ส่วนที่เหลือของเซตรายการ ‘a, b, c’ ในทรานแซกชันที่ 5 จะมีค่าเท่ากับ 12 ซึ่งสามารถกำหนดได้เป็น ru(‘a, b, c’, t_5) = ru(‘a, c’, t_5) = 12 (เนื่องจากเซตรายการ ‘a, b, c’ และ ‘a, c’ ลงท้ายด้วยรายการ ‘c’ เหมือนกัน ดังนั้นเซตรายการที่ตามหลังรายการ ‘c’ จะเหมือนกัน ด้วยเหตุนี้จึงสามารถกำหนดให้ค่าคุณประโยชน์ส่วนที่เหลือของเซตรายการ ‘a, b, c’ มีค่าเท่ากับค่าคุณประโยชน์ส่วนที่เหลือของเซตรายการ ‘a, c’ ได้โดยตรง)
4) ค่าคุณประโยชน์ของเซตรายการ prefix ของเซตรายการ ‘a, b, c’ ในทรานแซกชันที่ 5 ซึ่งสามารถพิจารณาได้เป็นค่าคุณประโยชน์ของเซตรายการ ‘a, b’ จะมีค่าเท่ากับ 12 ซึ่งสามารถกำหนดได้เป็น up (‘a, b, c’, t_5) = u(‘a, b’, t_5) = 12

จากตัวอย่างข้างต้น เราจะสังเกตุได้ว่า เราสามารถคำนวณหาค่าคุณประโยชน์ที่แท้จริงของเซตรายการ ‘a, b, c’ ในทรานแซกชันที่ 5 ได้จาก u(‘a, b’, t_5) + u(‘a, c’, t_5) - u(‘a’, t_5) ที่ซึ่งจะสามารถลดการคำนวณจากการประยุกต์ใช้ utility list ที่ไม่มีการจัดเก็บค่าคุณประโยชน์ของเซตรายการ prefix ของเซตรายการ ‘a, b, c’ ได้

3.3.2 ขั้นตอนวิธี MHUIRA

ขั้นตอนวิธี MHUIRA จะมี 2 ขั้นตอนการทำงานเหมือนกับขั้นตอนวิธี HURI-UL คือ 1) 1-HUIR identification และ 2) Mining HUIRs ตามลำดับ แต่อย่างไรก็ตาม MHUIRA จะมีส่วนที่แตกต่างจาก HURI-UL ที่ซึ่งจะแตกต่างที่ MHUIRA ประยุกต์ใช้ NUL (New Utility List structure) ในการจัดเก็บข้อมูลการปรากฏและค่าคุณประโยชน์ของเซตรายการ แต่ HURI-UL ประยุกต์ใช้ UL (Utility List structure) ในการจัดเก็บข้อมูลการปรากฏและค่าคุณประโยชน์ของเซตรายการ ดังนั้นเมื่อมีการใช้โครงการข้อมูลที่แตกต่างกัน การดำเนินการของขั้นตอนวิธีจะมีการเปลี่ยนแปลงเล็กน้อย ดังแสดงในรูปที่ 3 และ 4 (หมายเหตุ ในส่วนที่มีการเพิ่ม/ แก้ไข / เปลี่ยนแปลงจะเป็นตัวหนังสือสีเขียว)

การทำงานของขั้นตอน 1-HUIR identification ดังแสดงในรูปที่ 3 จะเริ่มจากการสร้างอะเรย์ t_{List} เพื่อใช้ในการจัดเก็บค่าคุณประโยชน์ของทุกทรานแซกชันในฐานข้อมูล ทำการสร้างลิสต์หลายมิติ i_{List} และทำการสร้างลิสต์ที่ใช้สำหรับจัดเก็บรายการที่ปรากฏสม่ำเสมอและคาดว่าน่าจะมีค่าคุณประโยชน์สูง (เมื่อรวมกับรายการอื่นๆ) 1_{List} (บรรทัดที่ 1) จากนั้นจะทำการอ่านแต่ละทรานแซกชัน t_p ในฐานข้อมูลเพื่อทำการคำนวณค่าคุณประโยชน์ tu(t_p) และ จัดเก็บ tu(t_p) ใน tu-List ต่อมา MHUIRA จะทำการพิจารณาแต่ละรายการ i_j ที่ปรากฏในทรานแซกชัน t_p เพื่อทำการคำนวณหาค่าคุณประโยชน์ของรายการ i_j ในทรานแซกชัน t_p (กล่าวคือ คำนวณหาค่า u(i_j, t_p))

และ ทำการอัพเดท new utility list ของรายการ i_j ที่ถูกจัดเก็บใน 1_{List} ด้วย <p, u(i_j, t_p), 0, 0> พร้อมกับทำการคำนวณค่าความสม่ำเสมอจากการปรากฏขึ้นของ i_j ในทรานแซกชัน t_p และ ทำการคำนวณค่า TWU^i โดยกำหนดให้ $TWU^i = TWU^i + tu(t_p)$ (บรรทัดที่ 2 - 6)

ขั้นตอนต่อไปจะเป็นการตรวจสอบการปรากฏขึ้นของแต่ละรายการ i_j ที่ถูกจัดเก็บใน 1_{List} ว่า 1) มีความสม่ำเสมอหรือไม่? และ 2) มีแนวโน้มที่จะมีค่าคุณประโยชน์สูงหรือไม่? โดยทำการตรวจสอบค่าความสม่ำเสมอ r^{i_j} ของรายการ i_j ว่ามีค่ามากกว่าค่าขีดบางความสม่ำเสมอ σ_r ที่ผู้ใช้กำหนดหรือไม่ และทำการตรวจสอบค่า twu^{i_j} ว่ามีค่าน้อยกว่าค่าขีดแบ่งคุณประโยชน์ σ_u หรือไม่ ถ้าค่าความสม่ำเสมอ r^{i_j} มีค่ามากกว่า σ_r หรือค่าประมาณคุณประโยชน์ twu^{i_j} ว่ามีค่าน้อยกว่า σ_u MHUIRA จะทำการลบรายการ i_j ออกจากการพิจารณา แต่ก่อนที่จะทำการลบ i_j ออกจากการพิจารณาจะทำการลดทอนค่าคุณประโยชน์ของแต่ละทรานแซกชันที่มีรายการ i_j ปรากฏเพื่อทำการลดทอนค่าประมาณคุณประโยชน์ โดยทำการพิจารณาแต่ละสมาชิกใน new utility list ของรายการ i_j ที่ซึ่งมีลักษณะเป็น <p, u(i_j, t_p), 0, 0> และทำการลดทอนค่า tu(t_p) ที่ถูกจัดเก็บใน t_{List} ด้วยค่า u(i_j, t_p) และเมื่อทำการพิจารณาทุกสมาชิกใน new utility list ของรายการ i_j แล้ว จะสามารถลบข้อมูลทั้งหมดของ i_j ที่ถูกจัดเก็บอยู่ใน 1_{List} ออกจากหน่วยความจำและการพิจารณาได้ (บรรทัดที่ 7 – 11) จากนั้น จะทำการคำนวณค่าคุณประโยชน์ส่วนเหลือในแต่ละสมาชิกใน new utility list ของแต่ละรายการ i_j รวมถึงทำการคำนวณหาค่าคุณประโยชน์ที่แท้จริง u(i_j) ของรายการ i_j และ ทำการคำนวณค่าคุณประโยชน์ส่วนเหลือ ru(i_j) ของรายการ i_j (บรรทัดที่ 13 – 20) โดยในการพิจารณาจะทำการพิจารณาทีละรายการ i_j (จากลำดับของรายการที่ทราบก่อนหน้าแล้ว) จากนั้นทำการพิจารณาแต่ละสมาชิก <p, u(i_j, t_p), 0, 0> ใน new utility list ของรายการ i_j จากนั้นทำการอัพเดทค่าคุณประโยชน์ของทรานแซกชัน t_p ที่ถูกจัดเก็บอยู่ใน t_{List} ด้วย tu(i_j, t_p) = tu(i_j, t_p) - u(i_j, t_p) เพื่อที่จะทราบถึงค่าคุณประโยชน์ของทุกรายการในทรานแซกชันที่อยู่ในลำดับถัดไปจาก i_j ซึ่งค่า tu(i_j, t_p) หลังการอัพเดทจะหมายถึงค่า ru(i_j, t_p) โดยเมื่อทราบถึงค่าดังกล่าว จะทำการอัพเดทสมาชิกใน new utility list ของรายการ i_j ที่พิจารณาให้มีค่าเป็น <p, u(i_j, t_p), ru(i_j, t_p), 0> และทำการอัพเดทค่า u(i_j) และ ru(i_j) ของรายการ i_j ด้วย u(i_j, t_p) และ ru(i_j, t_p) ตามลำดับ และท้ายสุด ถ้าค่า u(i_j) ของรายการ i_j มีค่าไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด HURI-UL จะทำการระบุและจัดเก็บรายการ i_j ว่าเป็นรูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างสม่ำเสมอ

Algorithm 1. *1-HUIR* identification

Input. D : transactional database, σ_{u}: a utility threshold, σ_{r}: regularity threshold
Output. i_{List}: a 2D list containing items with potentially to be *HUIR* in 1_{List}

1) create t_{List} , i_{List} and then create and initial 1_{List} in i_{List} for all items
2) **for each** transaction t_p in database D
3) compute and collect $u(t_p)$ of transaction t_p in t_{List}
4) **for each** item i in transaction t_{p}
5) compute $u(i, t_{\mathrm{p}})$ and then **update NUL^i of i with $< p, u(i, t_{\mathrm{p}}), 0, 0 >$**
6) compute TWU^i by $u(t_p)$ and regularity $r(i)$ of i by t_p
7) **for each** item i in 1_{List}
8) **if** $r(i) > \sigma_{\mathrm{r}}$ or $TWU^i < \sigma_{\mathrm{u}}$
9) **for each** entry $e = < tid_e, u(i, t_{tid_e})_e, ru(t_{tid_e})_e, up(t_{tid_e})_e >$ **in NUL^i**
10) decrease utility $u(t_{tid_e})$ of transaction t_{tid_e} in t_List by $u(i, t_{tid_e})_e$ of e
11) remove entry of i out of 1_{List}
12) sort 1_{List} based on the order of items $\succ$
13) **for each** item i in 1_{List}
14) initial value of $u(i)$ and $ru(i)$ to be 0
15) **for each** entry $e = < tid_e, u(i, t_{tid_e})_e, ru(t_{tid_e})_e, up(t_{tid_e})_e >$ **in NUL^i**
16) decrease utility $u(t_{tid_e})$ of transaction t_{tid_e} in t_{List} by $u(i, t_{tid_e})_e$
17) set $ru(t_{tid_e})_e$ to be $u(t_{tid_e})$ of t_{List}
18) increase $u(i)$ by $u(i, t_{tid_e})_e$ and increase $ru(i)$ by $ru(t_{tid_e})_e$
19) **if** $u(i) \geq \sigma_{\mathrm{u}}$
20) $HUIR = HUIR \cup i$

รูปที่ 3 ขั้นตอนวิธี MHUIRA : ขั้นตอน 1-HUIR identification

หลังจากทำการประมวลผลในขั้นตอน 1-HUIR Identification อัลกอริทึม MHUIRA ได้ทำการสร้าง 1_{List} ที่บรรจุไปด้วยรายการต่างๆที่มีค่าความสม่ำเสมอของการปรากฏน้อยกว่าเท่ากับค่าขีดแบ่งความสม่ำเสมอ และมีค่าประมาณคุณประโยชน์มากกว่าหรือเท่ากับค่าขีดแบ่งคุณประโยชน์ ขั้นตอนต่อไปจะเป็นการหารูปแบบที่น่าสนใจที่ปรากฏอย่างสม่ำเสมอและมีค่าคุณประโยชน์สูงจาก 1_{List} ที่สร้างก่อนหน้า (ดังแสดงในขั้นตอนวิธี Mining HUIRs ในรูปที่ 4) โดยเริ่มจากการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏสม่ำเสมอที่ประกอบไปด้วย 2 รายการ โดยเริ่มทำการพิจารณาแต่ละรายการ i_j ที่ถูกจัดเก็บใน 1_{List} (โดยในการพิจารณาจะพิจารณาแบบเรียงลำดับตามลำดับของรายการที่เป็นสมาชิกของเซต I) จากนั้นจะทำการสร้าง 2_{List} เพื่อใช้ในการจัดเก็บเซตรายการที่ประกอบด้วย 2 รายการที่ซึ่งเกิดจากการรวมกันระหว่างรายการ i_j กับรายการอื่นๆที่อยู่ในลำดับถัดๆไป โดยในตอนเริ่มแรกจะกำหนดให้ 2_{List} ไม่มีข้อมูลใดๆ ต่อมาจะเป็นการรวมรายการ i_j เข้ากับรายการ i_k ที่อยู่ในลำดับถัดๆไปจากรายการ i_j เพื่อสร้างการพิจารณาเซตรายการ $i_j \cup i_k$ จากนั้นจะทำการอินเทอร์เซกชัน new utility list ของรายการ i_j (NUL^{i_j}) กับ new utility list ของรายการ i_k (NUL^{i_k}) เพื่อทำการจัดเก็บข้อมูล

การปรากฏขึ้น ค่าคุณประโยชน์ ค่าคุณประโยชน์ส่วนเหลือ และค่าคุณประโยชน์ของเซตรายการ prefix สำหรับทรานแซกชันหนึ่งๆ ที่เซตรายการ $i_j \cup i_k$ ปรากฏ หลังจากขั้นตอนอินเทอร์เซกชันเสร็จสิ้น เราจะได้ new utility list (NUL^{i_j,i_k}) ของเซตรายการ $i_j \cup i_k$ ที่จะสามารถใช้ new utility list ดังกล่าวในการคำนวณค่าคุณประโยชน์ ค่าความสม่ำเสมอ และค่าประมาณคุณประโยชน์แบบกระชับของเซตรายการ $i_j \cup i_k$ ได้ หลังจากคำนวณค่าคุณประโยชน์ ค่าความสม่ำเสมอ และค่าประมาณคุณประโยชน์แบบกระชับของเซตรายการ $i_j \cup i_k$ แล้ว จะทำการตรวจสอบว่า เซตรายการ $i_j \cup i_k$ ปรากฏอย่างสม่ำเสมอหรือไม่ และทำการตรวจสอบว่า เซตรายการ $i_j \cup i_k$ มีแนวโน้มที่จะมีค่าคุณประโยชน์สูงหรือไม่ ถ้าเซตรายการ เซตรายการ $i_j \cup i_k$ ผ่านเงื่อนไขทั้งสองข้อข้างต้น HURI-UL จะทำการจัดเก็บเซตรายการ เซตรายการ $i_j \cup i_k$ พร้อมทั้งข้อมูลทั้งหมดที่เกี่ยวข้องกับเซตรายการ $i_j \cup i_k$ ไว้ใน 2_{List} เพื่อทำการพิจารณาเซตรายการที่ประกอบด้วย 3 รายการต่อไป โดยหลังจากทำการรวมรายการ i_j เข้ากับ i_k แล้ว ขั้นตอนวิธี MHUIRA จะดำเนินการรวมรายการ i_j เข้ากับ i_l ที่ถูกบรรจุอยู่ใน 1_{List} ในลำดับหลังจากรายการ i_k และจะดำเนินการตามกระบวนการต่างๆข้างต้น เมื่อทำการรวมรายการ i_j เข้ากับทุกรายการใน 1_{List} แล้ว เราจะได้ 2_{List} ที่บรรจุไปด้วยเซตรายการที่ประกอบไปด้วย 2 รายการ และเป็นเซตรายการ i_j เป็นรายการขึ้นต้น จากนั้นจะทำการตรวจสอบจำนวนเซตรายการที่บรรจุอยู่ใน 2_{List} ซึ่งถ้ามีจำนวนเซตรายการมากกว่า 1 จะทำการวนซ้ำการทำงานเพื่อหาเซตรายการที่ประกอบด้วย 3 รายการ โดยการดำเนินการจะดำเนินการเช่นเดียวกับการพิจารณาเซตรายการที่ประกอบไปด้วย 2 รายการ (ส่วนของ Procedure Gen-Longer-Itemsets)

Algorithm 2. Mining *HUIR*s

Input. i_{List}: a 2D-list of itemsets, σ_u: a utility threshold, σ_r: a regularity threshold
Output. $HUIR$: a complete set of *HUIR*s

1) **for each** item i in 1_{List} of i_{List} where $tou(i) \geq \sigma_u$
2) create 2_{List} in i_{List} *and* initial 2_{List} to be empty
3) **for each** item j in 1_{List} of i_{List} (where $i \prec j$)
4) **intersect NUL^i and NUL^j of item i and j in order to calculate $r(ij)$, $u(ij)$, $ru(ij)$, $up(ij)$and to collect NUL^{ij} for further computation**
5) calculate $tou(ij)$ as $tou(ij) = u(ij) + ru(ij) - up(ij)$
6) **if** $r(ij) \leq \sigma_r$
7) **create an entry of itemset ij in 2_{List} with its $r(ij)$, $u(ij)$, $ru(ij)$, $up(ij)$ and NUL^{ij}**
8) **if** $u(ij) \geq \sigma_u$
9) $HUIR = HUIR \cup ij$
10) **if** 2_{List} contains more than one itemsets
11) $GenLongerItemsets(2, i_{List})$

Procedure $GenLongerItemsets(k, i_{List})$
1) **for each** itemset u in k_{List} of $i_{List}$where $tou(ij) \geq \sigma_u$
2) create $(k+1)_{List}$ in i_{List} *and* initial $(k+1)_{List}$ to be empty
3) **for each** itemset v in k_{List} of i_{List} (where $u \prec v$)
4) **intersect NUL^u and NUL^v of itemset u and v in order to calculate $r(uv)$, $u(uv)$, $ru(uv)$, $up(uv)$and to collect NUL^{uv} for further computation**
5) calculate $tou(uv)$ as $tou(uv) = u(uv) + ru(uv)$
6) **if** $r(uv) \leq \sigma_r$
7) **create an entry of itemset uv in $(k+1)_{List}$ with its $r(uv)$, $u(uv)$, $ru(uv)$, $up(uv)$ and NUL^{uv}**
8) **if** $u(uv) \geq \sigma_u$
9) $HUIR = HUIR \cup uv$
10) **if** $(k+1)_{List}$ contains more than one itemset
11) $GenLongerItemsets(k+1, i_{List})$

รูปที่ 4 ขั้นตอนวิธี MHUIRA : ขั้นตอน Mining HUIRs

บทที่ 4

ผลการทดลอง

ในบทนี้จะนำเสนอการทดสอบประสิทธิภาพการค้นหารูปแบบที่ปรากฏอย่างสม่ำเสมอและมีค่าคุณประโยชน์สูงที่ค้นหาจากขั้นตอนวิธี HURI-UL และขั้นตอนวิธี MHUIRA โดยจากการศึกษางานวิจัยที่เกี่ยวข้องพบว่ายังไม่มีงานวิจัยใดที่ทำการพิจารณาความน่าสนใจของรูปแบบภายใต้การพิจารณาความสม่ำเสมอของการปรากฏร่วมกับคุณประโยชน์ของรูปแบบนั้น ๆ ดังนั้น การดำเนินการทดลองจะทำการเปรียบเทียบประสิทธิภาพของขั้นตอนวิธี HURI-UL กับขั้นตอนวิธี MHUIRA เท่านั้น

แต่อย่างไรก็ตาม ในการดำเนินการทดลอง ผู้วิจัยได้กำหนดพารามิเตอร์ให้มีความใกล้เคียงกับงานวิจัยที่เกี่ยวข้อง กล่าวคือ ค่าขีดแบ่งความสม่ำเสมอจะกำหนดให้มีค่าอยู่ระหว่าง 1 – 30% ของจำนวนทรานแซกชันทั้งหมดในฐานข้อมูล (กล่าวคือ รูปแบบจะเป็นรูปแบบที่ปรากฏอย่างสม่ำเสมอ จะต้องมีค่าความสม่ำเสมอไม่เกิน 1 – 30% ของจำนวนทรานแซกชันทั้งหมดในฐานข้อมูล) และ ค่าขีดแบ่งคุณประโยชน์จะกำหนดให้มีค่าระหว่าง 0.001 – 22% ของค่าคุณประโยชน์ทั้งหมดของรูปแบบที่ปรากฏในฐานข้อมูล (กล่าวคือ รูปแบบจะเป็นรูปแบบที่มีค่าคุณประโยชน์สูงจะต้องมีค่าคุณประโยชน์ไม่น้อยกว่า 0.001 – 22% ของค่าคุณประโยชน์ทั้งหมดของรูปแบบที่ปรากฏในฐานข้อมูล) ตามลำดับ โดยผู้วิจัยได้ทำการเขียนโปรแกรมการคำนวณตามขั้นตอนวิธี HURI-UL และ MHUIRA ด้วยภาษาซี และทำการทดสอบประสิทธิภาพในเครื่อง Xeon® 2.4 GHz ที่มีปริมาณหน่วยความจำ 64 GB

ในการทดสอบประสิทธิภาพจะดำเนินการทดสอบกับชัดข้อมูลจริง 4 ชุดข้อมูล โดยชุดข้อมูลที่ใช้ทดสอบสามารถดาว์นโหลดได้จาก P. F. Viger, "SPMF: An Open-Source Data Mining Library" โดยแต่ละชุดข้อมูลจะมีรายละเอียด ดังแสดงในตารางที่ 6

ตารางที่ 6 คุณลักษณะของชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของขั้นตอนวิธี HURI-UL และ MHUIRA

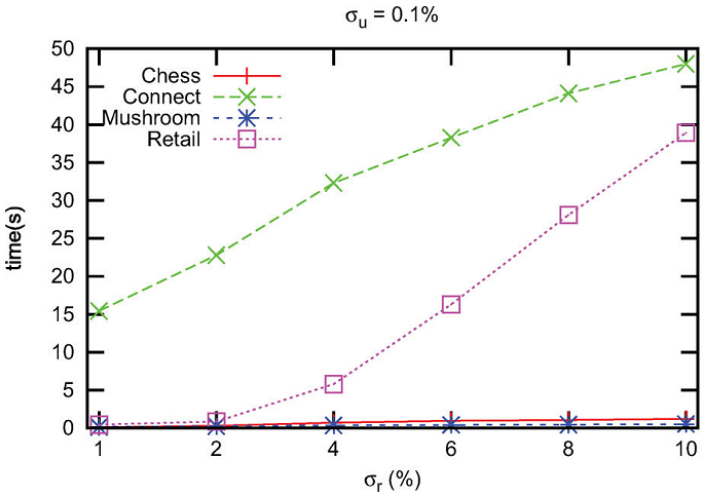
ชื่อฐานข้อมูล	จำนวนรายการที่ปรากฏ	จำนวนทรานแซกชัน	ความยาวเฉลี่ยของทรานแซกชัน	ชนิดของฐานข้อมูล
Chess	75	3,196	37	หนาแน่น
Foormart2000	1,559	36,869	11	เบาบาง
Mushroom	119	8,124	23	หนาแน่น
Retail	16,469	88,162	10.3	เบาบาง

การทดลองที่ผู้วิจัยได้ทำการทดสอบประสิทธิภาพของขั้นตอนวิธี HURI-UL และ MHUIRA สามารถแบ่งได้เป็น 4 กรณี คือ 1) การทดสอบเวลาในการคำนวณเมื่อทำการกำหนดค่าขีดแบ่งคุณประโยชน์แบบตายตัวและกำหนดค่าขีดแบ่งความสม่ำเสมอแบบแปรปรวน 2) การทดสอบเวลาในการคำนวณเมื่อทำการกำหนดค่าขีดแบ่งคุณประโยชน์แบบแปรปรวนและกำหนดค่าขีดแบ่งความสม่ำเสมอแบบตายตัว 3) การพิจารณาจำนวนผลลัพธ์ที่ขั้นตอนวิธี HURI-UL สามารถค้นหาได้เมื่อทำการกำหนดค่าขีดแบ่งคุณประโยชน์แบบตายตัวและกำหนดค่าขีดแบ่งความสม่ำเสมอแบบแปรปรวน และ 4) การพิจารณาจำนวนผลลัพธ์ที่ขั้นตอนวิธี HURI-UL สามารถค้นหาได้เมื่อทำการกำหนดค่าขีดแบ่งคุณประโยชน์แบบแปรปรวนและกำหนดค่าขีดแบ่งความสม่ำเสมอแบบตายตัวตามลำดับ

4.1 การทดสอบประสิทธิภาพของขั้นตอนวิธี HURI-UL

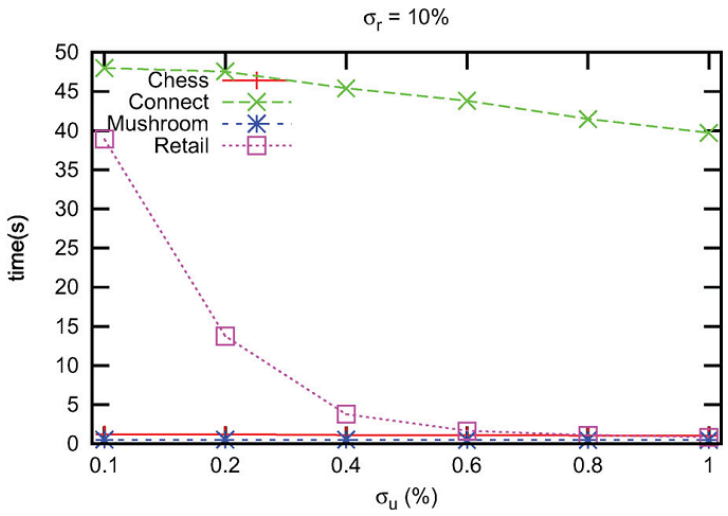
รูปที่ 5 แสดงเวลาที่ใช้ในการประมวลผลของขั้นตอนวิธี HURI-UL กับ 4 ชุดข้อมูลภายใต้การกำหนดค่าขีดแบ่งคุณประโยชน์เท่ากับ 0.1% ซึ่งเป็นค่าต่ำสุดของค่าขีดแบ่งคุณประโยชน์ที่ทำการพิจารณาที่ซึ่งจะทำให้มีรูปแบบเป็นจำนวนมากมีค่าคุณประโยชน์มากกว่าหรือเท่ากับค่าขีดแบ่งคุณประโยชน์ โดยค่าดังกล่าวจะทำให้การค้นหารูปแบบใช้เวลามาก ซึ่งจะสะท้อนให้เห็นถึงประสิทธิภาพของขั้นตอนวิธี HURI-UL เมื่อต้องพิจารณารูปแบบเป็นจำนวนมาก นอกจากนั้นการทดลองในรูปข้างต้นจะมีการแปรปรวนของค่าขีดแบ่งความสม่ำเสมอเพื่อทำการศึกษาถึงผลกระทบของการกำหนดค่าขีดแบ่งความสม่ำเสมอกับประสิทธิภาพการคำนวณของขั้นตอนวิธี HURI-UL โดยจากรูป จะสังเกตุได้ว่าเมื่อค่าขีดแบ่งความสม่ำเสมอมีค่าเพิ่มขึ้นจะทำให้ขั้นตอนวิธี HURI-UL ใช้เวลาในการคำนวณเพิ่มขึ้น เนื่องจากเมื่อค่าขีดแบ่งความสม่ำเสมอมีค่าเพิ่มขึ้นจะทำให้มีรูปแบบเป็นจำนวนมากขึ้น

มีค่าความสม่ำเสมอน้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอ ซึ่งจะส่งผลให้ขั้นตอนวิธี HURI-UL ใช้เวลาในการพิจารณารูปแบบต่าง ๆ เพิ่มขึ้นด้วยเช่นกัน



รูปที่ 5 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HURI-UL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ

ในทางตรงกันข้าม รูปที่ 6 จะแสดงเวลาที่ใช้ในการประมวลผลของขั้นตอนวิธี HURI-UL กับ 4 ชุดข้อมูล ภายใต้การกำหนดค่าขีดแบ่งความสม่ำเสมอแบบตายตัวให้มีค่าเท่ากับ 10% โดยจากการกำหนดดังกล่าวจะทำให้มีรูปแบบเป็นจำนวนมากมีค่าความสม่ำเสมอน้อยกว่าหรือเท่ากับค่าขีดแบ่งสม่ำเสมอที่กำหนด ซึ่งจะทำให้เราได้ทราบถึงเวลาที่ใช้ในการประมวลผลของขั้นตอนวิธี HURI-UL เมื่อต้องทำการพิจารณารูปแบบเป็นจำนวนมาก นอกจากนั้นการทดลองในรูปข้างต้นยังเป็นการทดลองภายใต้การกำหนดค่าขีดแบ่งคุณประโยชน์แบบแปรปรวนที่มีการกำหนดค่าขีดแบ่งคุณประโยชน์ตั้งแต่ 0.1 – 1% โดยจากรูปจะสามารถสังเกตุได้ว่า เมื่อค่าขีดแบ่งคุณประโยชน์มีค่าเพิ่มขึ้นจะทำให้เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HURI-UL น้อยลง เนื่องจากเมื่อค่าขีดแบ่งคุณประโยชน์มีค่าสูงขึ้นที่ซึ่งจะทำให้มีรูปแบบเป็นจำนวนน้อยมีค่าคุณประโยชน์สูงกว่าหรือเท่ากับค่าขีดแบ่งคุณประโยชน์ และเมื่อจำนวนรูปแบบที่ต้องทำการพิจารณามีจำนวนน้อยลง ขั้นตอนวิธี HURI-UL จึงใช้เวลาในการคำนวณน้อยลงด้วยเช่นกัน

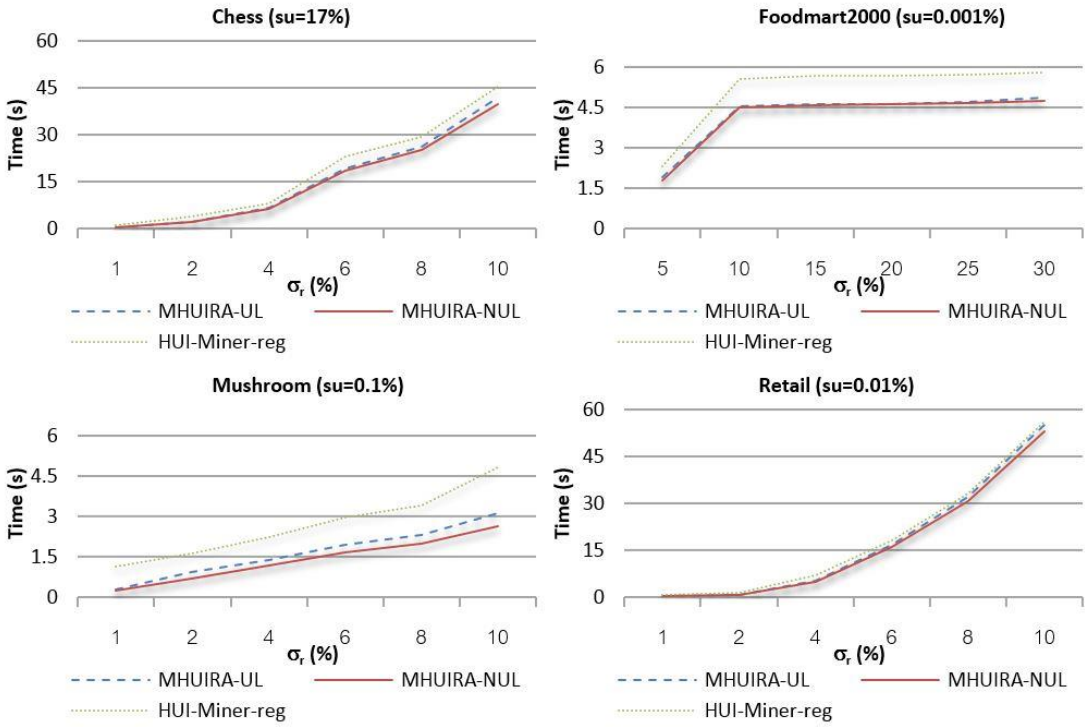


รูปที่ 6 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HURI-UL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์

4.2 การทดสอบประสิทธิภาพของขั้นตอนวิธี MHUIRA

การทดสอบประสิทธิภาพของขั้นตอนวิธี MHUIRA จะทำการเปรียบเทียลกับขั้นตอนวิธี HURI-UL และ HUI-Miner ที่เพิ่มการพิจารณาความสม่ำเสมอของการปรากฏ เรียกขั้นตอนวิธีนี้ว่า HUI-Miner-reg (หมายเหตุ HUI-Miner เป็นขั้นตอนวิธีที่ใช้สำหรับค้นหารูปแบบที่มีค่าคุณประโยชน์สูง แต่ไม่ได้ทำการพิจารณาเกี่ยวกับความสม่ำเสมอของการปรากฏ อีกทั้ง HUI-Miner เป็นขั้นตอนวิธีที่ประยุกต์ใช้ utility-list ในการจัดเก็บข้อมูลที่สำคัญระหว่างการประมวลผล ประยุกต์ใช้ค่าคุณประโยชน์คงเหลือ (remaining utility) และค่าประมาณคุณประโยชน์ (overestimated utility) เพื่อทำการลดทอนปริภูมิสถานะเช่นกัน)

รูปที่ 7 จะแสดงถึงเวลาในการหาผลลัพธ์ของทั้ง 3 ขั้นตอนวิธีกับ 4 ชุดข้อมูล (คือ Chess, Footmart2000, Mushroom และ Retail) ภายใต้การกำหนดค่าขีดแบ่งความสม่ำเสมอให้มีความเปลี่ยนแปลง (ค่าขีดแบ่งคุณประโยชน์จะถูกกำหนดให้มีค่าตายตัว) โดยจากรูปเราจะสังเกตุได้ว่าเมื่อค่าขีดแบ่งความสม่ำเสมอเพิ่มขึ้น ทั้ง 3 ขั้นตอนวิธีจะใช้เวลาในการประมวลผลเพิ่มขึ้น โดยเมื่อค่าขีดแบ่งความสม่ำเสมอเพิ่มขึ้นจะเป็นเหตุให้มีรูปแบบเป็นจำนวนมากขึ้นที่มีค่าคุณความสม่ำเสมอของการปรากฏมากกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอ และจะทำให้ใช้เวลาในการประมวลเพิ่มขึ้นจากจำนวนรูปแบบที่เพิ่มขึ้นด้วยเช่นกัน นอกจากนั้น เรายังสังเกตุได้ว่าขั้นตอนวิธี HURI-UL และ MHUIRA สามารถประมวลผลได้รวดเร็วกว่าขั้นตอนวิธี HUI-Miner-reg ที่ซึ่งทำการอ่านฐานข้อมูล 2 ครั้ง และ ขั้นตอนวิธี MHUIRA สามารถประมวลผลได้รวดเร็วกว่าขั้นตอนวิธี HURI-UL เล็กน้อย (เนื่องจากสามารถใช้ประโยชน์จากการใช้โครงสร้างข้อมูล NUL)



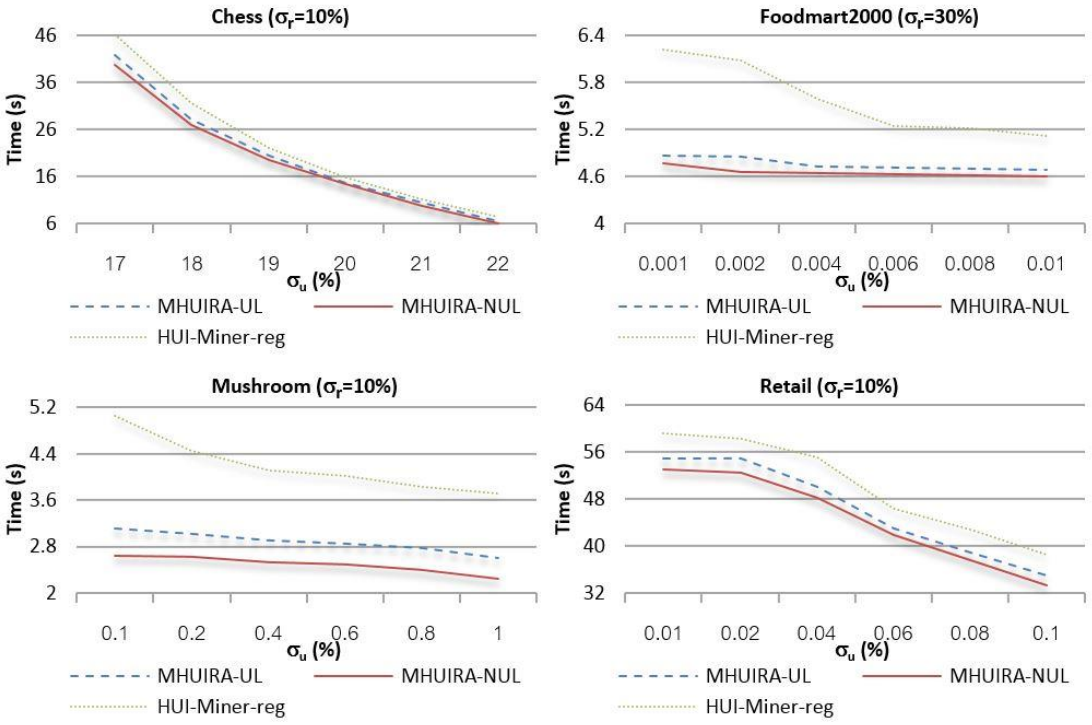
รูปที่ 7 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี MHUIRA เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ

รูปที่ 8 จะแสดงถึงเวลาในการหาผลลัพธ์ของทั้ง 3 ขั้นตอนวิธีกับ 4 ชุดข้อมูล ภายใต้การกำหนดค่าขีดแบ่งคุณประโยชน์ (ค่าขีดแบ่งความสม่ำเสมอจะถูกกำหนดให้มีค่าตายตัว) โดยจากรูปจะสังเกตุได้ว่าทั้ง 3 ขั้นตอนวิธีจะใช้เวลาในการประมวลผลลดลงเมื่อค่าขีดแบ่งคุณประโยชน์เพิ่มขึ้น สาเหตุเนื่องจาก เมื่อค่าขีดแบ่งคุณประโยชน์เพิ่มขึ้นจะทำให้มีจำนวนรายการและเซตรายการที่มีค่าคุณประโยชน์มากกว่าหรือเท่ากับค่าขีดแบ่งคุณประโยชน์เพิ่มขึ้นลดลง ซึ่งจะทำให้ปริภูมสถานะลดลงและเวลาที่ใช้ในการประมวลผลลดลงตามไปด้วย นอกเหนือจากข้างต้น เรายังสังเกตุได้อีกว่าขั้นตอนวิธี HURI-UL และ MHUIRA สามารถประมวลผลได้รวดเร็วกว่าขั้นตอนวิธี HUI-Miner-reg อย่างชัดเจน และ ขั้นตอนวิธี MHUIRA จะสามารถประมวลผลได้รวดเร็วกว่าขั้นตอนวิธี HURI-UL เนื่องจากการได้รับประโยชน์จากการประยุกต์ใช้ NUL

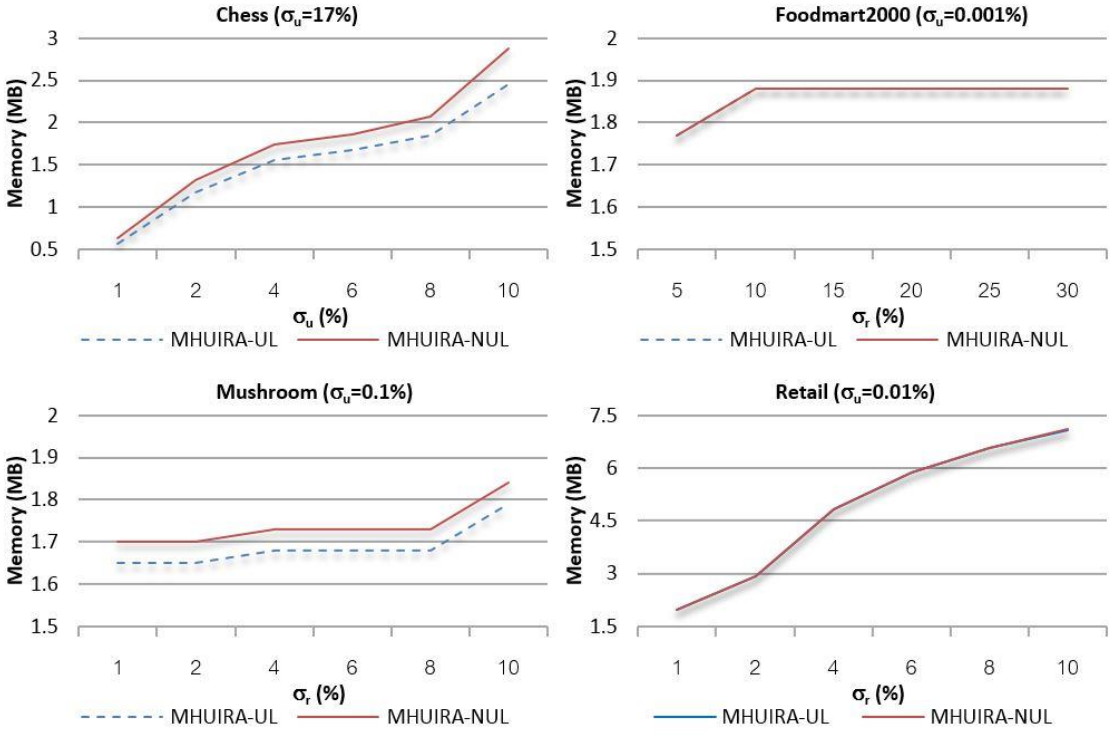
ในส่วนของการพิจารณาถึงการใช้หน่วยความจำระหว่างการประมวลผล รูปที่ 9 จะแสดงถึงหน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี MHUIRA เทียบกับขั้นตอนวิธี HURI-UL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ (ค่าขีดแบ่งคุณประโยชน์จะถูกกำหนดให้มีค่าตายตัว) โดยจากรูปเราจะสังเกตุได้ว่าเมื่อค่าขีดแบ่งความสม่ำเสมอเพิ่มขึ้น การใช้หน่วยความจำของขั้นตอนวิธี HURI-UL และ MHUIRA จะเพิ่มขึ้น เหตุผลคือ เมื่อค่าขีดแบ่งความสม่ำเสมอเพิ่มขึ้น จะทำให้มีรายการและเซตรายการที่มีค่าความสม่ำเสมอน้อยกว่าหรือเท่ากับค่าขีดแบ่ง

ความสม่ำเสมอเพิ่มขึ้น นี้จึงเป็นเหตุให้ทั้งสองขั้นตอนวิธีจะต้องทำการจัดเก็บรายการและเซตรายการ (รวมถึง utility list/ new utility list ของรายการ/เซตรายการเหล่านั้น) เพิ่มขึ้น แต่อย่างไรก็ตาม เมื่อทำการเปรียบเทียบการใช้หน่วยความจำของทั้งสองขั้นตอนวิธี เราจะสังเกตุเพิ่มเติมได้อีกว่า ขั้นตอนวิธี MHUIRA จะใช้หน่วยความจำมากกว่าขั้นตอนวิธี HURI-UL เล็กน้อย เหตุเนื่องมากจาก ขั้นตอนวิธี MHUIRA ได้ทำการประยุกต์ใช้ new utility list ในการจัดเก็บข้อมูล ที่ซึ่งจะมีการจัดเก็บค่าคุณประโยชน์ของเซตรายการ prefix เพิ่มเติม นี้จึงก่อให้เกิดความผกผันระหว่างเวลาในการประมวลผลที่ลดลง แต่จะใช้หน่วยความจำที่เพิ่มขึ้น

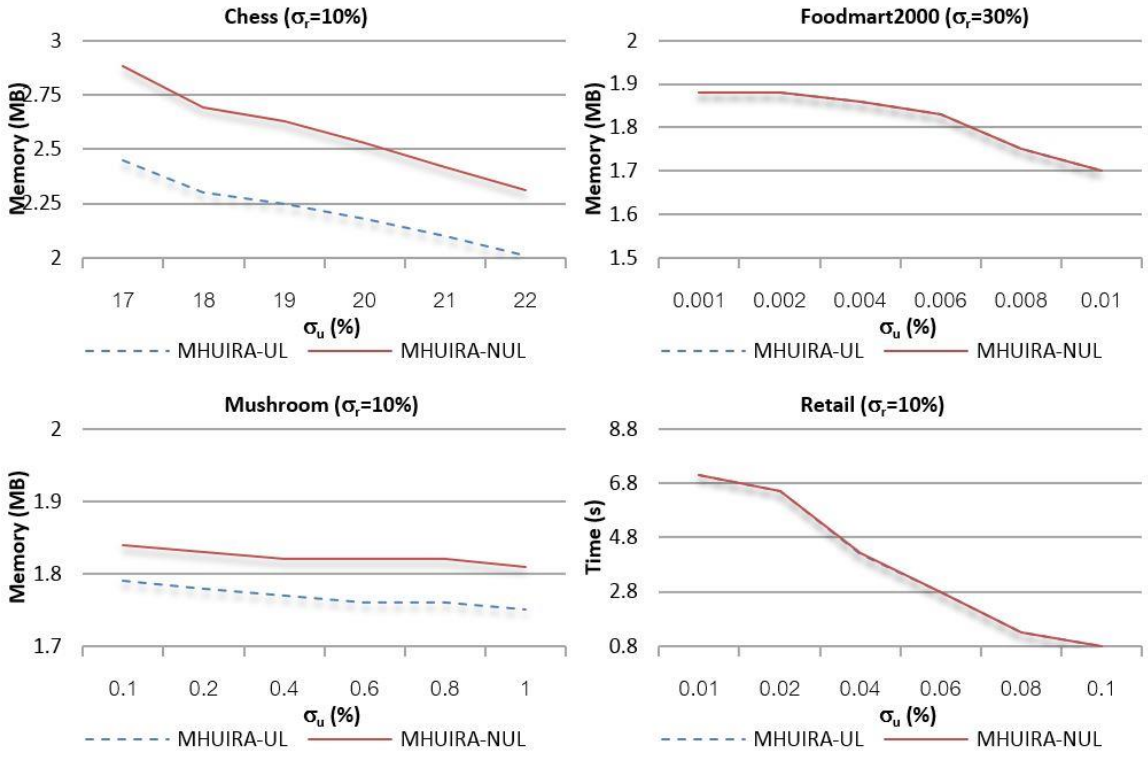
รูปที่ 10 จะแสดงถึงหน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี MHUIRA เทียบกับขั้นตอนวิธี HURI-UL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ (ค่าขีดแบ่งความสม่ำเสมอจะถูกกำหนดให้มีค่าตายตัว) โดยจากรูปเราจะสังเกตุได้ว่าเมื่อค่าขีดแบ่งคุณประโยชน์เพิ่มขึ้น การใช้หน่วยความจำของขั้นตอนวิธี HURI-UL และ MHUIRA จะลดลง เหตุผลเนื่องจาก เมื่อค่าขีดแบ่งความสม่ำเสมอเพิ่มขึ้น จะทำให้มีรายการและเซตรายการที่มีค่าคุณประโยชน์มากกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอลดลง นี้จึงเป็นเหตุให้ทั้งสองขั้นตอนวิธีจะต้องทำการจัดเก็บรายการและเซตรายการ (รวมถึง utility list/ new utility list ของรายการ/เซตรายการเหล่านั้น) ลดลง แต่อย่างไรก็ตาม เมื่อทำการเปรียบเทียบการใช้หน่วยความจำของทั้งสองขั้นตอนวิธี เราจะสังเกตุเพิ่มเติมได้อีกว่า ขั้นตอนวิธี MHUIRA จะใช้หน่วยความจำมากกว่าขั้นตอนวิธี HURI-UL เล็กน้อย เหตุเนื่องมากจาก ขั้นตอนวิธี MHUIRA ได้ทำการประยุกต์ใช้ new utility list ในการจัดเก็บข้อมูล ที่ซึ่งจะมีการจัดเก็บค่าคุณประโยชน์ของเซตรายการ prefix เพิ่มเติม นี้จึงก่อให้เกิดความผกผันระหว่างเวลาในการประมวลผลที่ลดลง แต่จะใช้หน่วยความจำที่เพิ่มขึ้น



รูปที่ 8 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี MHUIRA เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์



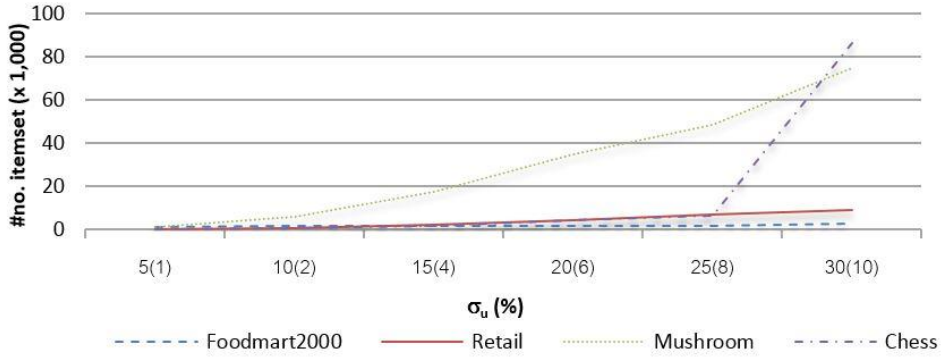
รูปที่ 9 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี MHUIRA เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ



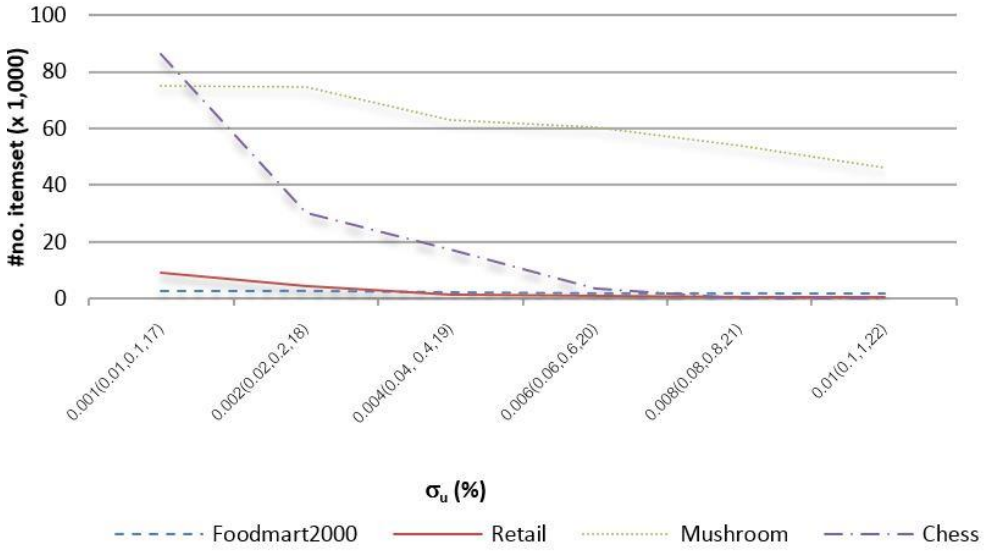
รูปที่ 10 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี MHUIRA เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์

4.3 จำนวนผลลัพธ์ที่ค้นหาได้จากขั้นตอนวิธี MHUIRA

ในส่วนของการพิจารณาจำนวนผลลัพธ์ที่สามารถค้นหาได้จากขั้นตอนวิธี MHUIRA ภายใต้การกำหนดค่าขีดแบ่งคุณประโยชน์แบบตายตัวและกำหนดค่าขีดแบ่งความสม่ำเสมอแบบแปรปรวน และการกำหนดค่าขีดแบ่งคุณประโยชน์แบบแปรปรวนและกำหนดค่าขีดแบ่งความสม่ำเสมอแบบตายตัว จะแสดงได้ดังรูปที่ 11 และ 12 ที่จะสังเกตุได้ว่าจำนวนผลลัพธ์จะมีจำนวนเพิ่มขึ้นอย่างมีนัยสำคัญเมื่อทำการกำหนดค่าขีดแบ่งความสม่ำเสมอเพิ่มขึ้น และจะมีจำนวนลดลงเมื่อทำการกำหนดค่าขีดแบ่งคุณประโยชน์เพิ่มขึ้น



รูปที่ 11 ผลลัพธ์ที่ได้จากขั้นตอนวิธี MHUIRA เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ



รูปที่ 12 ผลลัพธ์ที่ได้จากขั้นตอนวิธี MHUIRA เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์

บทที่ 5

สรุปผลการวิจัย

งานวิจัยนี้ได้นำเสนอแนวความคิดในการค้นหารูปแบบที่น่าสนใจภายใต้การพิจารณาความสม่ำเสมอของการปรากฏและค่าคุณประโยชน์ของรูปแบบนั้น ๆ (โดยค่าคุณประโยชน์อาจอยู่ในรูปแบบของผลกำไร ต้นทุน ความเสี่ยงหรือ อื่น ๆ) การค้นหารูปแบบภายใต้การพิจารณาข้างต้นจะมีส่วนช่วยในการสังเกตพฤติกรรมการซื้อของลูกค้าที่ทำการซื้อสินค้าที่ให้ผลกำไรสูงที่ซึ่งจะเป็นข้อมูลที่จะช่วยให้ผู้บริหารสามารถวิเคราะห์ความต้องการลูกค้า บริหารจัดการคลังสินค้า และสามารถวางแผนการตลาดได้ดียิ่งขึ้น จากกรอบความคิดข้างต้น รูปแบบหนึ่ง ๆ จะเป็นรูปแบบที่น่าสนใจก็ต่อเมื่อรูปแบบนั้น ๆ เป็นรูปแบบที่ปรากฏ (ถูกซื้อ) อย่างสม่ำเสมอ (กล่าวคือ รูปแบบมีค่าความสม่ำเสมอไม่มากกว่าค่าขีดแบ่งความสม่ำเสมอ) และ เป็นรูปแบบที่มีค่าคุณประโยชน์ที่ได้รับจากการปรากฏขึ้นของรูปแบบ (กล่าวคือ รูปแบบมีค่าคุณประโยชน์ไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์)

ในการค้นหารูปแบบรูปแบบที่ปรากฏอย่างสม่ำเสมอและมีค่าคุณประโยชน์สูงอย่างมีประสิทธิภาพ ผู้วิจัยได้นำเสนอขั้นตอนวิธีที่เรียกว่า HURI-UL (High Utility Regular Itemsets using Utility-List) ที่ทำการอ่านข้อมูลจากฐานข้อมูลเพียงครั้งเดียว ขั้นตอนวิธี HURI-UL ที่นำเสนอประยุกต์ใช้แนวความคิดเกี่ยวกับ ค่าคุณประโยชน์คงเหลือ (remaining utility) และ ค่าประมาณคุณประโยชน์ (overestimated utility) เพื่อทำการลดทอนปริภูมิสถานะ นอกจากนั้น HURI-UL ได้ประยุกต์ใช้โครงสร้างข้อมูลในรูปแบบลิสต์ (utility list) เพื่อทำการจัดเก็บข้อมูลการปรากฏขึ้นและค่าคุณประโยชน์ของรูปแบบหนึ่ง และ ท้ายสุด HURI-UL ประยุกต์ใช้การค้นหาผลลัพธ์แบบเชิงลึก (depth first search) เพื่อทำการค้นหาผลลัพธ์ได้อย่างมีประสิทธิภาพ

แต่อย่างไรก็ตาม จากการที่ HURI-UL ประยุกต์ใช้โครงสร้าง utility list เพื่อจัดเก็บข้อมูลการปรากฏขึ้นและค่าคุณประโยชน์ของรูปแบบหนึ่ง ๆ ระหว่างการประมวลผล ที่ซึ่งจะต้องการการประมวลผลที่ค่อนข้างซับซ้อนในการดำเนินการอินเทอร์เซกชันกันระหว่าง 2 utility list ใด ๆ เพื่อคำนวณหาค่าคุณประโยชน์ที่แท้จริงของเซตรายการที่สร้างขึ้นจากการอินเทอร์เซกชัน ด้วยเหตุนี้ ผู้วิจัยจึงได้นำเสนอการพัฒนา utility list ให้มีการจัดเก็บข้อมูลที่เพิ่มขึ้น (เรียกว่า New Utility List structure, NUL) โดยจะทำการเพิ่มการจัดเก็บค่าคุณประโยชน์ของเซตรายการ prefix ด้วย ซึ่งจากการเพิ่มการจัดเก็บข้อมูลดังกล่าว จะทำให้สามารถลดเวลาที่ใช้ในการประมวลผลได้ เมื่อทำการพัฒนา NUL จะทำให้การทำงานของขั้นตอนวิธี HURI-UL จะต้องมีการปรับเปลี่ยนเล็กน้อย ผู้วิจัยจึง

ได้นำเสนอการพัฒนาขั้นตอนวิธี MHUIRA ซึ่งพัฒนามาจากขั้นตอนวิธี HURI-UL ที่ประยุกต์ใช้ NUL ในการจัดเก็บข้อมูลการปรากฏขึ้นและค่าคุณประโยชน์ของรูปแบบหนึ่ง ๆ โดย MHUIRA จะทำการอ่านข้อมูลเพียงครั้งเดียว ทำการประยุกต์ใช้แนวความคิดเกี่ยวกับ ค่าคุณประโยชน์คงเหลือ (remaining utility) และ ค่าประมาณคุณประโยชน์ (overestimated utility) เพื่อทำการลดทอนปริภูมิสถานะ และ ประยุกต์ใช้การค้นหาผลลัพธ์แบบเชิงลึก (depth first search) เพื่อทำการค้นหาผลลัพธ์

ผู้วิจัยได้ทำการทดลองเพื่อทดสอบประสิทธิภาพของตอนวิธีที่เสนอกับข้อมูลจริง 4 ชุดข้อมูล โดยทำการทดลองในสองแง่มุมด้วยกันคือ เวลาที่ใช้ในการหาผลลัพธ์ และจำนวนผลลัพธ์ที่สามารถค้นหาได้ โดยการทดลองชี้ให้เห็นว่า ขั้นตอนวิธี HURI-UL และขั้นตอนวิธี MHUIRA ที่ประยุกต์ใช้โครงสร้างข้อมูล NUL สามารถค้นหารูปแบบที่ปรากฏอย่างสม่ำเสมอและมีค่าคุณประโยชน์สูงได้อย่างมีประสิทธิภาพ

บรรณานุกรม

1. Tanbeer, S.K., Ahmed, C.F., Jeong, B.S., and Lee, Y.K. (2009). Discovering periodic-frequent patterns in transactional databases. Proceedings of PAKDD.
2. Fu, A.W.C., w. Kwong, R.W., and Tang, J. (2000). Mining n-most interesting itemsets. Proceedings of the 12th International Symposium on Foundations of Intelligent Systems, Springer-Verlag.
3. Wang, J., Han, J., Lu, Y., and Tzvetkov, P. 2005. Tfp: an efficient algorithm for mining top-k frequent closed itemsets. Proceedings of the IEEE Transactions on Knowledge and Data Engineering. Volume 17.
4. Yang, B., Huang, H., Wu, Z. 2008. Topsis: Finding top-k significant n-itemsets in sliding windows adaptively. Journal of Knowledge-Based Systems. 21(6).
5. Li, H.F. 2009. Mining top-k maximal reference sequences from streaming web click sequences with a damped sliding window. Journal of Expert Systems with Applications 36(8).
6. Ke, Y., Cheng, J., and Yu, J.X. 2009. Top-k correlative graph mining. Proceedings of SDM, SIAM.
7. Fournier-Viger, P., and Tseng, V.S. 2013. Tns: mining top-k non-redundant sequential rules. Proceedings of SAC, ACM.
8. Amphawan, K., Lenca, P., and Surarerks, A. 2009. Mining top-k periodic-frequent patterns without support threshold. Proceeding of IAIT. Volume 55 of CCIS., Springer.
9. Amphawan, K., Lenca, P., and Surarerks, A. 2011. Efficient mining top-k regular-frequent itemset using compressed tidsets. Proceedings of New Frontiers in Applied Data Mining. Volume 7104 of Lecture Notes in Computer Science.
10. Amphawan, K., Lenca, P., and Surarerks, A. 2012. Mining top-k regular-frequent itemsets using database partitioning and support estimation. Journal of Expert Systems with Applications 39(2).
11. Tanbeer, S.K., Ahmed, C.F., and Jeong, B.S. 2010. Mining regular patterns in incremental transactional databases. Proceedings of Int. Asia-Pacific Web Conference, IEEE Computer Society.
12. Tanbeer, S.K., Ahmed, C.F., and Jeong, B.S. 2010. Mining regular patterns in data streams. Proceedings of DASFAA. Volume 5981 of LNCS., Springer.
13. Surana, A., Kiran, R.U., and, Reddy, P.K. 2012. An efficient approach to mine periodic-frequent patterns in transactional databases. Proceedings of PAKDD Workshops.
14. Kiran, R.U., and Reddy, P.K. 2010. Mining periodic-frequent patterns with maximum items' support constraints. Proceedings of the Third Annual ACM Bangalore Conference. COMPUTE '10.

15. Yao, H., Hamilton, H. J. and Butz, C. J. 2004. A Foundational Approach to Mining Itemset Utilities from Databases. Proceedings of the 4th SIAM International Conference on Data Mining.
16. Yao, H., and Hamilton, H.J. 2006. Mining itemset utilities from transaction databases. Journal of Data & Knowledge Engineering.
17. Liu, Y., Liao, W.-K., and Choudhary, A. 2005. A Two Phase algorithm for fast discovery of High Utility of Itemsets. Proceedings of PAKDD.
18. Tseng, V., Wu, C. W., Shie, B. E. and Yu, P. 2010. UP-Growth: An Efficient Algorithm for High Utility Itemsets Mining. Proceedings of the 16th ACM SIG KDD Conference on Knowledge Discovery and Data Mining.
19. Liu, M. and Qu, J. 2012. Mining High Utility Itemsets without Candidate Generation. Proceedings of the ACM international conference on Information and Knowledge Management (CIKM).