



รายงานการวิจัยฉบับสมบูรณ์

โครงการ “การวิเคราะห์พฤติกรรมผู้บริโภคจากการวิเคราะห์ข้อมูลการซื้อสินค้า
(Consumers’ behavior analysis based on buying behavior analysis)”

คณะผู้วิจัย

นายโกเมศ อัมพวัน	หัวหน้าโครงการ
นายอรรถสิทธิ์ สุรฤกษ์	ผู้ร่วมวิจัย
นายอนุชิต จิตพัฒนกุล	ผู้ร่วมวิจัย
นายณัฐนนท์ ลีลาตระกูล	ผู้ร่วมวิจัย
นางสาวสุนิสา रिमเจริญ	ผู้ร่วมวิจัย
นายวรารุฒิ ผ้าเจริญ	ผู้ร่วมวิจัย

โครงการวิจัยประเภทงบประมาณเงินรายได้
จากเงินอุดหนุนรัฐบาล (งบประมาณแผ่นดิน)

ปีงบประมาณ พ.ศ. ๒๕๖๐

มหาวิทยาลัยบูรพา

รหัสโครงการ 2560A10802209

สัญญาเลขที่ 65/2560

รายงานการวิจัยฉบับสมบูรณ์

โครงการ “การวิเคราะห์พฤติกรรมผู้บริโภคจากการวิเคราะห์ข้อมูลการซื้อสินค้า
(Consumers’ behavior analysis based on buying behavior analysis)”

คณะผู้วิจัย

นายโกเมศ อัมพวัน	หัวหน้าโครงการ
นายอรรถสิทธิ์ สุรฤกษ์	ผู้ร่วมวิจัย
นายอนุชิต จิตพัฒนกุล	ผู้ร่วมวิจัย
นายณัฐนนท์ ลีลาตระกูล	ผู้ร่วมวิจัย
นางสาวสุนิสา रिमเจริญ	ผู้ร่วมวิจัย
นายวรารุฒิ ผ้าเจริญ	ผู้ร่วมวิจัย

*ห้องปฏิบัติการวิจัยนวัตกรรมการประมวลผล คณะวิทยาการสารสนเทศ

มหาวิทยาลัยบูรพา

**ห้องปฏิบัติการทางวิศวกรรมระบบนับได้เชิงทฤษฎี คณะวิศวกรรมศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย

***คณะวิทยาศาสตร์ประยุกต์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

กันยายน 2561

บทคัดย่อ

การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงเป็นหัวข้องานวิจัยหนึ่งที่น่าสนใจภายใต้การทำเหมืองข้อมูลที่สามารถประยุกต์ใช้อย่างแพร่หลาย ตัวอย่างเช่น การประยุกต์ใช้ในธุรกิจค้าปลีก เพื่อทำการค้นหารายการสินค้าที่ถูกซื้อจากลูกค้า ที่ซึ่งสินค้าดังกล่าวจะเป็นรายการสินค้าที่ถูกซื้อร่วมกันและให้ผลตอบแทนที่สูง เป็นต้น แต่อย่างไรก็ตาม การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงจะทำการพิจารณาเพียงแค่ค่าคุณประโยชน์ของรายการต่าง ๆ เท่านั้น ที่ซึ่งการค้นหารูปแบบดังกล่าวอาจไม่เพียงพอต่อการสังเกตถึงพฤติกรรมการซื้อสินค้าของผู้บริโภค ด้วยเหตุนี้ งานวิจัยจึงมุ่งเน้นที่จะทำการเพิ่มเติมเงื่อนไขการพิจารณารูปแบบ โดยจะทำการเพิ่มเติมเงื่อนไขของการปรากฏอย่างไม่สม่ำเสมอร่วมกับการพิจารณาค่าคุณประโยชน์ของรูปแบบต่าง ๆ โดยภายใต้แนวคิดใหม่ข้างต้น รูปแบบที่น่าสนใจจะเป็นรูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏขึ้นในชุดข้อมูลอย่างไม่สม่ำเสมอ โดยจะเรียกรูปแบบดังกล่าวว่า “รูปแบบที่มีค่าคุณประโยชน์สูงที่ปรากฏไม่สม่ำเสมอ (*High-utility-irregular itemsets, HUII*)”

ในการค้นหารูปแบบดังกล่าว งานวิจัยนี้ได้เสนอขั้นตอนวิธีที่มีประสิทธิภาพ ที่เรียกว่า “*High-Utility Itemsets with Irregular Occurrence Miner, HUIIM*” ซึ่งจะทำการอ่านข้อมูลจากฐานข้อมูลเพียงครั้งเดียว และทำการประยุกต์ใช้ปรับปรุงโครงสร้างการเก็บข้อมูล (*New modified utility-list, NUL*) สำหรับจัดเก็บข้อมูลการปรากฏขึ้นและค่าคุณประโยชน์ของเซตรายการหนึ่ง ๆ ให้มีประสิทธิภาพ นอกจากนี้ ขั้นตอนวิธี *HUIIM* ได้ประยุกต์ใช้แนวความคิดเกี่ยวกับค่าประมาณคุณประโยชน์ (*transaction-weighted utility, TWU*), ค่าคุณประโยชน์คงเหลือ (*remaining utility*) และค่าประมาณคุณประโยชน์แบบกระชับ (*tight over-estimated utility, tou*) เพื่อทำการลดทอนปริภูมิสถานะของการค้นหาเซตรายการ การทดลองได้ถูกจัดทำขึ้นเพื่อทดสอบประสิทธิภาพของขั้นตอนวิธี *HUIIM* ที่ถูกนำเสนอ โดยจากผลการทดลองจะสามารถสังเกตได้ว่าขั้นตอนวิธี *HUIIM* ที่นำเสนอสามารถประมวลผลได้อย่างมีประสิทธิภาพทั้งด้านเวลาและหน่วยความจำที่ใช้ในการประมวลผล

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, 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) irregular occurrence, called “*high utility-irregular itemsets, HUII*”, are regarded as interesting.

To mine such itemsets, an efficient algorithm called “*High-Utility Itemsets with Irregular Occurrence Miner (HUIIM)*” is designed and introduced. *HUIIM* scans database once to capture occurrence information and utility value of single items into the *new-modified utility list* structure (*NUL*) used for maintaining occurrence information of an item/itemset. The concept *transaction weighted utility (TWU)*, *remaining utility (ru)* and *tight over-estimated utility (tou)* of an item/itemset are applied and utilized to prune search space. Experimental studies are conducted to investigate performance of the proposed methods and the results show that *HUIIM* can effectively mine high utility itemsets with irregular occurrence in both computational time and memory usage.

สารบัญ

บทคัดย่อ.....	I
Abstract.....	II
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของโครงการวิจัย.....	3
1.3 ขอบเขตของโครงการวิจัย.....	4
1.4 ประโยชน์ที่ได้รับ.....	5
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	6
2.1 ทฤษฎีที่เกี่ยวข้อง.....	6
2.1.1 การค้นหารูปแบบที่ปรากฏบ่อย (<i>Mining frequent patterns from transactional database</i>).....	6
2.1.2 การค้นหารูปแบบที่มีค่าคุณประโยชน์สูง (<i>Mining high utility patterns from transaction database</i>).....	8
2.1.3 การค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ (<i>Mining frequent regular patterns from transaction database</i>).....	10
2.2 งานวิจัยที่เกี่ยวข้อง.....	12
บทที่ 3 วิธีการดำเนินการวิจัย.....	16
3.1 นิยามที่เกี่ยวข้องกับการค้นหารูปแบบที่ให้ค่าคุณประโยชน์สูงโดยปรากฏอย่างไม่สม่ำเสมอ.....	16
3.2 ขั้นตอนวิธีการนำเสนอ HUIIM.....	19
3.2.1 โครงสร้างข้อมูล <i>New-modified Utility List (NUL)</i>	20
3.2.2 ขั้นตอนวิธี HUIIM.....	20
3.3 ตัวอย่างขั้นตอนวิธี HUIIM.....	25

บทที่ 4 ผลการทดลอง.....	34
4.1 เวลาที่ใช้ในการคำนวณ.....	36
4.2 หน่วยความจำที่ใช้ในการคำนวณ.....	47
4.3 ผลลัพธ์.....	58
บทที่ 5 สรุปผลวิจัย.....	60
บรรณานุกรม	61

สารบัญภาพ

ภาพที่ 3-1 ขั้นตอนการอ่านฐานข้อมูลและระบุรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่าเสมอ	21
ภาพที่ 3-2 ขั้นตอนการหารูปแบบทั้งหมดที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่าเสมอ.....	23
ภาพที่ 3-3 อ่านข้อมูลทรานแซกชันที่ 1 ในฐานข้อมูล.....	26
ภาพที่ 3-4 อ่านข้อมูลทรานแซกชันที่ 2 ในฐานข้อมูล.....	27
ภาพที่ 3-5 อ่านข้อมูลครบทุกทรานแซกชันในฐานข้อมูล	27
ภาพที่ 3-6 ตัดการพิจารณารายการ 'e' ที่มี TWU ไม่ผ่านค่าขีดแบ่งคุณประโยชน์	28
ภาพที่ 3-7 หลังจากอ่านข้อมูลทุกทรานแซกชันในฐานข้อมูล	29
ภาพที่ 3-8 <i>HUII-tree</i> ที่บรรจุ 2- <i>HUIIs</i>	31
ภาพที่ 3-9 <i>HUII-tree</i> ที่บรรจุ 3- <i>HUII</i>	33
ภาพที่ 3-10 เซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่าเสมอจากขั้นตอน <i>HUIIM</i>	33
ภาพที่ 4-1 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี <i>HUIIM</i> และ <i>MHUIRA-NUL</i> เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่าเสมอของฐานข้อมูล <i>Accidents</i>	36
ภาพที่ 4-2 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี <i>HUIIM</i> และ <i>MHUIRA-NUL</i> เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่าเสมอของฐานข้อมูล <i>BMS</i>	37
ภาพที่ 4-3 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี <i>HUIIM</i> และ <i>MHUIRA-NUL</i> เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่าเสมอของฐานข้อมูล <i>Chainstore</i>	37
ภาพที่ 4-4 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี <i>HUIIM</i> และ <i>MHUIRA-NUL</i> เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่าเสมอของฐานข้อมูล <i>Chess</i>	38
ภาพที่ 4-5 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี <i>HUIIM</i> และ <i>MHUIRA-NUL</i> เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่าเสมอของฐานข้อมูล <i>Connect</i>	38
ภาพที่ 4-6 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี <i>HUIIM</i> และ <i>MHUIRA-NUL</i> เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่าเสมอของฐานข้อมูล <i>Foodmart2000</i>	39
ภาพที่ 4-7 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี <i>HUIIM</i> และ <i>MHUIRA-NUL</i> เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่าเสมอของฐานข้อมูล <i>Kosarak</i>	40
ภาพที่ 4-9 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี <i>HUIIM</i> และ <i>MHUIRA-NUL</i> เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่าเสมอของฐานข้อมูล <i>PUMSB</i>	40

ภาพที่ 4-38 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี <i>HUIIM</i> และ <i>MHUIRA-NUL</i> เมื่อทำการเปลี่ยนแปลงค่าคุณประโยชน์ของฐานข้อมูล <i>Mushroom</i>	56
ภาพที่ 4-39 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี <i>HUIIM</i> และ <i>MHUIRA-NUL</i> เมื่อทำการเปลี่ยนแปลงค่าคุณประโยชน์ของฐานข้อมูล <i>PUMSB</i>	57
ภาพที่ 4-40 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี <i>HUIIM</i> และ <i>MHUIRA-NUL</i> เมื่อทำการเปลี่ยนแปลงค่าคุณประโยชน์ของฐานข้อมูล <i>Retail</i>	57
ภาพที่ 4-41 จำนวนผลลัพธ์ที่ค้นหาได้จากขั้นตอนวิธี <i>HUIIM</i> เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ	58
ภาพที่ 4-42 จำนวนผลลัพธ์ที่ค้นหาได้จากขั้นตอนวิธี <i>HUIIM</i> เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์.....	59

สารบัญตาราง

ตารางที่ 2-1 ตัวอย่างฐานข้อมูลรายการที่ประกอบไปด้วยหมายเลขทรานแซกชันและเซตรายการที่ปรากฏในแต่ละทรานแซกชัน	7
ตารางที่ 2-2 ตัวอย่างค่าคุณประโยชน์ (ผลกำไร) ของแต่ละรายการ	9
ตารางที่ 2-3 ตัวอย่างข้อมูลการซื้อสินค้าของลูกค้า	9
ตารางที่ 3-1 ฐานข้อมูลรายการที่ประกอบไปด้วยหมายเลขทรานแซกชันและเซตรายการที่ปรากฏในทรานแซกชันที่มีจำนวนของการปรากฏขึ้นของแต่ละรายการ.....	17
ตารางที่ 3-2 ตารางค่าคุณประโยชน์ของแต่ละรายการ	17
ตารางที่ 4-1 คุณลักษณะของชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพ.....	35

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

ในยุคปัจจุบันเป็นยุคของข้อมูล ข่าวสาร และสารสนเทศมีการเปิดเผยอย่างแพร่หลาย ประกอบกับ ณ ปัจจุบันมีแอปพลิเคชันหรือโปรแกรมที่ทำการจัดเก็บข้อมูลได้หลากหลาย ดังนั้นจึงเป็นเหตุให้ธุรกิจ SME ธุรกิจขนาดกลาง และธุรกิจขนาดใหญ่ มีการประยุกต์ใช้ข้อมูล ข่าวสาร สารสนเทศ และองค์ความรู้ต่าง ๆ ในการประกอบการตัดสินใจเกี่ยวกับการดำเนินธุรกิจมากยิ่งขึ้น จากการใช้ข้อมูลดังกล่าว วิธีการวิเคราะห์ทั้งในเชิงปริมาณและคุณภาพที่เหมาะสมกับธุรกิจต่าง ๆ ได้ถูกคิดค้นโดยนักวิจัยในหลายแขนงสาขา แต่อย่างไรก็ตาม มีแนวคิดหนึ่งที่ถูกประยุกต์ใช้อย่างแพร่หลายคือ การวิเคราะห์พฤติกรรมผู้บริโภคด้วยการค้นหารูปแบบที่ปรากฏบ่อย (*Frequent Itemset Mining, FIM*) ที่สามารถบ่งบอกถึงสิ่งของหรือเหตุการณ์ที่ปรากฏขึ้นพร้อมกันบ่อย ๆ ตัวอย่างเช่น ในธุรกิจห้างสรรพสินค้าหรือธุรกิจค้าปลีกจะทำการหาความสัมพันธ์ของรายการสินค้าที่ถูกซื้อพร้อมกันบ่อย ๆ เพื่อช่วยในการจัดทำโปรโมชั่นสินค้า ช่วยในการจัดชั้นวางสินค้าให้สินค้าที่ถูกซื้อพร้อมกันให้อยู่ในพื้นที่ใกล้เคียงกัน เพื่ออำนวยความสะดวกให้แก่ลูกค้าและยังช่วยกระตุ้นการจับจ่ายใช้สอยของลูกค้าอีกด้วย นอกจากนี้ยังช่วยในการจัดทำแคตตาล็อกสินค้าให้สินค้าที่ถูกซื้อพร้อมกันบ่อย ๆ ให้อยู่ใกล้กัน ซึ่งการดำเนินการทั้งหมดนี้จะช่วยให้ห้างสรรพสินค้ารักษารฐานลูกค้าให้ยังคงซื้อสินค้ากับห้างสรรพสินค้าต่อไป

แนวความคิดเบื้องต้นของการค้นหารูปแบบที่ปรากฏบ่อยจะประยุกต์ใช้ค่าสนับสนุน (ค่าความถี่หรือจำนวนครั้งในการเกิดขึ้นของรูปแบบนั้น ๆ) เป็นตัวชี้วัดความสำคัญหรือความน่าสนใจของรูปแบบ แต่อย่างไรก็ตาม การใช้เพียงค่าสนับสนุนอาจไม่เพียงพอต่อการค้นหารูปแบบที่มีความหลากหลาย โดยแนวความคิดนี้ถูกพัฒนาอย่างต่อเนื่องในหลายๆ แง่มุม อาทิเช่น การค้นหารูปแบบที่การปรากฏบ่อยแบบเรียงลำดับ (*Frequent sequential pattern mining, Han, J. และคณะ, 2000*), การค้นหารูปแบบที่ปรากฏบ่อยภายใต้ค่าน้ำหนักของแต่ละรายการ (*Frequent weighted pattern mining, Yun, U. และ Leggett J., 2005*), การค้นหารูปแบบที่มีค่าคุณประโยชน์สูง (*High utility pattern mining, Chan, R และคณะ, 2003 และ Yao, H. และคณะ, 2004*), การค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ (*Frequent-regular pattern mining, Tanbeer, S.K. และคณะ, 2009*) และอื่น ๆ

จากงานวิจัยข้างต้นจะมีหัวข้องานวิจัยหนึ่งที่ทำการศึกษาคุณค่าคุณประโยชน์ของรูปแบบ (Utility of patterns) ที่ซึ่งค่าคุณประโยชน์ของรูปแบบอาจหมายถึง ผลกำไรที่ได้รับจากการขายสินค้าชิ้นหนึ่ง ๆ ของรายการสินค้าหนึ่ง ๆ หรือการบริการหนึ่งๆ เมื่อเราทำการคำนวณค่าคุณประโยชน์ทั้งหมดของรายการสินค้า จะทำให้เราสามารถทราบได้ถึงจำนวนผลกำไร/ขาดทุนที่ได้จากรายการสินค้านั้นๆ และยังทราบถึงรายการสินค้าที่ให้คุณประโยชน์สูง แต่อย่างไรก็ดี การค้นหาแบบที่มีค่าคุณประโยชน์สูงไม่ได้ทำการพิจารณาถึงพฤติกรรมการปรากฏขึ้นของรูปแบบหรือรายการนั้นๆ ว่ามีพฤติกรรมการปรากฏขึ้นอย่างสม่ำเสมอ ไม่สม่ำเสมอ สม่ำเสมอในบางช่วงเวลาหรือไม่

จากปัญหาข้างต้นจึงได้มีนักวิจัยได้คิดค้นวิธีการค้นหาแบบที่มีคุณประโยชน์สูงที่ปรากฏขึ้นอย่างสม่ำเสมอ (*Mining high-utility-regular itemsets*, Amphawan, K. และ Surarerks, A., 2015-b) ได้เพิ่มเติมเงื่อนไขการพิจารณารูปแบบ โดยทำการเพิ่มเติมการพิจารณาเงื่อนไขของการปรากฏอย่างสม่ำเสมอร่วมกับการพิจารณาค่าคุณประโยชน์ของรายการต่าง ๆ ซึ่งจะทำให้เราทราบถึงแบบที่มีค่าคุณประโยชน์สูงและมีพฤติกรรมการปรากฏขึ้นอย่างสม่ำเสมอ อาจช่วยให้เราทราบถึงช่วงเวลาที่รายการสินค้าได้รับความนิยม อันนำมาซึ่งการจัดทำโปรโมชั่นให้กับรายการสินค้าที่พิจารณาเพื่อช่วยกระตุ้นยอดขาย และอื่น ๆ เป็นต้น แต่อย่างไรก็ตาม การทราบถึงพฤติกรรมการเกิดอย่างสม่ำเสมอ อาจยังไม่เพียงพอสำหรับการค้นหาแบบใหม่ๆ ที่มีความหลากหลาย

ดังนั้นในงานวิจัยนี้จึงมุ่งเน้นที่การค้นหาแบบที่ให้คุณประโยชน์สูงโดยปรากฏอย่างไม่สม่ำเสมอ (ยกตัวอย่างจากข้อมูลห้างสรรพสินค้าแห่งหนึ่ง มีพฤติกรรมการซื้อสินค้าของผู้บริโภคจะซื้อ {กล้อง, เลนส์} มีคุณประโยชน์สูงแต่พฤติกรรมการปรากฏจะไม่สม่ำเสมอ) ซึ่งจากการทราบถึงแบบที่ให้คุณประโยชน์สูงดังกล่าว จะทำให้ผู้บริหารสามารถคิดกลยุทธ์เพื่อสามารถกระตุ้นการซื้อสินค้าของผู้บริโภคอันนำมาซึ่งผลประโยชน์ของบริษัทที่เพิ่มขึ้นได้ รวมไปถึงการวางแผนจัดการสินค้าคงคลัง เพราะการลงทุนในสินค้าคงคลังต้องใช้เงินจำนวนมากและมีโอกาสที่วัตถุดิบจะเสื่อมสภาพ เช่น อุปกรณ์อิเล็กทรอนิกส์ เครื่องใช้ไฟฟ้า อาจส่งผลกระทบต่อสภาพคล่องของธุรกิจได้

การวิเคราะห์พฤติกรรมผู้บริโภคที่น่าเสนอเหมาะกับทุกธุรกิจการค้าที่มีรายการสินค้าที่หลากหลายและมีการขายสินค้าอย่างต่อเนื่อง อาทิเช่น ธุรกิจค้าปลีก ธุรกิจค้าส่ง ธุรกิจร้านอาหาร ธุรกิจเกี่ยวกับเฟอร์นิเจอร์ ธุรกิจเกี่ยวกับอุปกรณ์ไฟฟ้า ห้างสรรพสินค้า ธุรกิจขายตรง ธุรกิจขายอะไหล่รถจักรยานยนต์หรือรถยนต์ ธุรกิจเกี่ยวกับอาหารทะเล ธุรกิจเกี่ยวกับวัสดุอุปกรณ์การเกษตร ธุรกิจเกี่ยวกับวัสดุอุปกรณ์ก่อสร้าง ธุรกิจเกี่ยวกับผลิตภัณฑ์ทารกและเด็ก ธุรกิจเกี่ยวกับอุปกรณ์เครื่องเขียนและสิ่งพิมพ์ ธุรกิจเกี่ยวกับอุปกรณ์คอมพิวเตอร์และโทรศัพท์เคลื่อนที่ ธุรกิจเกี่ยวกับดอกไม้และผลิตภัณฑ์ทางการเกษตร ธุรกิจเกี่ยวกับเบเกอรี่ ธุรกิจเกี่ยวกับเวชกรรมและอุปกรณ์ทางการแพทย์ ธุรกิจเกี่ยวกับเครื่องประดับ ธุรกิจสิ่งทอและเครื่องแต่งกาย ธุรกิจเกี่ยวกับเว็บไซต์ต่าง

ๆ ที่ถูกเข้าถึงอย่างต่อเนื่องที่ซึ่งจะทำให้เจ้าของธุรกิจทราบถึงความนิยมของเว็บไซต์ที่ทางบริษัทจัดทำขึ้น ทางกายภาพและอื่น ๆ

ตัวอย่างการประยุกต์ใช้งานด้านธุรกิจ เช่น บริษัท Amazon.com¹ เป็นศูนย์กลางของผู้บริโภคทั่วโลกที่จะหาซื้อสินค้าใด ๆ ที่ต้องการผ่านทางระบบออนไลน์ในราคาที่สมเหตุสมผลเท่าที่จะเป็นไปได้ จุดขายคือ ความสะดวกสบาย สินค้ามีมากมายให้เลือกซื้อ ฤดูกาลจำหน่ายที่สำคัญของบริษัท Amazon.com Inc. คือช่วงเทศกาล Black Friday (Slice Intelligence² บริษัทเก็บข้อมูลด้านอีคอมเมิร์ซได้เปิดเผยว่าจากเทศกาล Black Friday ในปี 2015 ยอดขายเฉพาะของ Amazon คิดเป็น 36% ของยอดขายออนไลน์ทั้งหมด) ถ้าเราทราบถึงสินค้าที่ให้คุณประโยชน์สูงที่ถูกซื้อเฉพาะช่วงเทศกาลต่าง ๆ อาจช่วยให้เราทราบถึงช่วงเวลาที่ยอดขายสินค้าได้รับความนิยม อันนำมาซึ่งการตัดสินใจในการเพิ่มหรือลดการผลิต การเตรียมวัตถุดิบเพื่อให้สอดคล้องกับกำลังการผลิต ลดปัญหาในการสต็อกสินค้า

การประยุกต์ใช้งานในด้านการแพทย์ ปัจจุบันพบว่าโรคหลายชนิดมีสาเหตุสืบเนื่องมาจากความผิดปกติของยีน เช่น โรคมะเร็ง การวิเคราะห์ข้อมูลทางพันธุกรรมของยีน DNA Microarray Technology³ ที่มีการปรากฏร่วมกันอย่างสม่ำเสมอและมีความผิดปกติในลำดับ DNA จะทำให้นักวิทยาศาสตร์ทราบถึงกลไกการเกิดความผิดปกติอันนำมาซึ่งสาเหตุการเกิดโรคที่ร้ายแรงได้และอาจทราบถึงวิธีการรักษาโรคต่าง ๆ แต่อย่างไรก็ตามรูปแบบยีนที่ปรากฏร่วมกันอย่างสม่ำเสมอ ยังไม่เพียงพอในการวิเคราะห์ความผิดปกติของยีนที่ทำให้เกิดโรคได้ ซึ่งถ้าเราทราบถึงยีนที่ปรากฏร่วมกันอย่างไม่สม่ำเสมอ และมีความผิดปกติ อาจช่วยให้นักวิทยาศาสตร์หรือแพทย์ทราบถึงแนวโน้มที่เป็นสาเหตุที่ทำให้เกิดโรคต่าง ๆ เพิ่มมากขึ้นและ โอกาสในการรักษาโรคเพิ่มขึ้นได้

1.2 วัตถุประสงค์ของโครงการวิจัย

1. เพื่อสร้างแนวคิดใหม่ในการตรวจสอบพฤติกรรมของการบริโภคที่สามารถนำไปประยุกต์ใช้ในธุรกิจต่าง ๆ ได้จริง โดยแนวคิดดังกล่าวสามารถประยุกต์ใช้ได้กับ ธุรกิจค้าปลีก ธุรกิจค้าส่ง ธุรกิจร้านอาหาร ธุรกิจเกี่ยวกับเฟอร์นิเจอร์ ธุรกิจเกี่ยวกับอุปกรณ์ไฟฟ้า ห้างสรรพสินค้า ธุรกิจขายตรง ธุรกิจขายอะไหล่รถจักรยานยนต์หรือรถยนต์ ธุรกิจเกี่ยวกับอาหารทะเล ธุรกิจเกี่ยวกับวัสดุอุปกรณ์การเกษตร ธุรกิจเกี่ยวกับวัสดุอุปกรณ์ก่อสร้าง ธุรกิจเกี่ยวกับผลิตภัณฑ์ทารกและเด็ก ธุรกิจเกี่ยวกับอุปกรณ์เครื่องเขียนและสิ่งพิมพ์ ธุรกิจเกี่ยวกับอุปกรณ์คอมพิวเตอร์และโทรศัพท์เคลื่อนที่ ธุรกิจเกี่ยวกับดอกไม้และผลิตภัณฑ์ทางการเกษตร ธุรกิจ

¹ <https://www.amazon.com>

² <https://intelligence.slice.com>

³ <http://ejournals.swu.ac.th/index.php/pharm>

เกี่ยวกับเบเกอร์รี่ ธุรกิจเกี่ยวกับเวชกรรมและอุปกรณ์ทางการแพทย์ ธุรกิจเกี่ยวกับเครื่องประดับ ธุรกิจสิ่งทอและเครื่องแต่งกาย ธุรกิจเกี่ยวกับเว็บไซต์ต่างๆที่ถูกเข้าถึงอย่างต่อเนื่องที่ซึ่งจะทำให้เจ้าของธุรกิจทราบถึงความนิยมของเว็บไซต์ที่ทางบริษัทจัดทำขึ้น ทาง การแพทย์ และอื่น ๆ

2. เพื่อศึกษาการวิเคราะห์พฤติกรรมหรือรูปแบบการบริโภคของผู้บริโภคภายใต้กรอบแนวคิด เกี่ยวกับการค้นหารูปแบบที่ทำให้คุณประโยชน์สูงและการค้นหารูปแบบที่ปรากฏอย่างไม่ สม่าเสมอ
3. เพื่อทราบถึงกลุ่มของรายการสินค้าที่ถูกซื้อไม่สม่ำเสมอแต่ให้คุณประโยชน์สูง ซึ่งจากข้อมูล ดังกล่าวจะนำไปสู่การค้นหสาเหตุของการเกิดขึ้นของพฤติกรรมหรือรูปแบบการบริโภคที่ซึ่ง จะสามารถนำข้อมูลดังกล่าวไปประกอบการตัดสินใจเพื่อที่จะสามารถทำการพัฒนาผลิตภัณฑ์ และขั้นตอนการดำเนินธุรกิจได้
4. เพื่อพัฒนาระบบเทคโนโลยีสารสนเทศสำหรับการวิเคราะห์พฤติกรรมผู้บริโภคภายใต้การ วิเคราะห์ข้อมูลการซื้อสินค้า เพื่อนำไปประกอบการตัดสินใจในการพัฒนา/ปรับปรุง กระบวนการในการดำเนินธุรกิจเพื่อให้ผู้ที่สนใจสามารถนำแนวคิดที่นำเสนอ ไปศึกษาเพื่อทำ การพัฒนาหรือประยุกต์ใช้ในงานวิจัยหรือประยุกต์ใช้ในการดำเนินธุรกิจของตนเองต่อไป

1.3 ขอบเขตของโครงการวิจัย

1. ข้อมูลการซื้อสินค้าที่จะทำการพิจารณา จะต้องมึลักษณะเป็นแบบทรานแซกชันที่ประกอบ ไปด้วยรายการสินค้าและ จำนวนชิ้นของแต่ละสินค้าที่ถูกซื้อในแต่ละทรานแซกชัน (ข้อมูลที่ จะนำมาวิเคราะห์จะสามารถค้นหาได้จากธุรกิจที่มีรายการสินค้าที่หลากหลายและ มีการ สั่งซื้อสินค้าต่อเนื่อง)
2. การวิเคราะห์พฤติกรรมผู้บริโภคจะเป็นการวิเคราะห์ภายใต้การค้นหารูปแบบที่ทำให้ คุณประโยชน์สูงโดยปรากฏอย่างไม่สม่ำเสมอ ซึ่งผู้ที่ต้องการผลลัพธ์จะต้องทำการกำหนดค่า ชิดแบ่ง (threshold) เพื่อใช้เป็นเกณฑ์สำหรับวัดความน่าสนใจของรูปแบบที่จะทำการค้นหา จากข้อมูลที่ต้องการวิเคราะห์ อาทิเช่น ค่าขีดแบ่งคุณประโยชน์/ค่าขีดแบ่งค่าคุณประโยชน์ และ ค่าขีดแบ่งความสม่ำเสมอ
3. การวัดผลของการวิเคราะห์สามารถดำเนินการได้ใน 3 แง่มุม คือ 1) เวลาที่ใช้ในการ ประมวลผล 2) หน่วยความจำที่ใช้ในการคำนวณ และ 3) จำนวนผลลัพธ์ที่สามารถค้นหาได้ ตามลำดับ โดยผลลัพธ์ที่ได้จากการวิเคราะห์สามารถนำไปประกอบการตัดสินใจได้ แต่การ ตัดสินใจและการติดตามผลการตัดสินใจไม่สามารถดำเนินการได้

1.4 ประโยชน์ที่ได้รับ

1. ได้โมเดลต้นแบบวิเคราะห์พฤติกรรมกรรมการบริโภคของลูกค้า ซึ่งเป็นข้อมูลเพื่อสนับสนุนการตัดสินใจที่จะดำเนินการกระตุ้นพฤติกรรมการใช้จ่ายใช้สอยของผู้บริโภคได้ อาทิเช่น การจัดทำโปรโมชั่น การนำเสนอรายการสินค้าใหม่ ๆ และอื่น ๆ โดยการวิเคราะห์พฤติกรรมกรรมการบริโภคที่นำเสนอจะเหมาะกับทุกธุรกิจการค้าที่มีรายการสินค้าที่หลากหลายและ มีการขายสินค้าต่อเนื่อง อาทิเช่น ธุรกิจค้าปลีก ธุรกิจค้าส่ง ธุรกิจร้านอาหาร ธุรกิจเกี่ยวกับเฟอร์นิเจอร์ ธุรกิจเกี่ยวกับอุปกรณ์ไฟฟ้า ห้างสรรพสินค้า ธุรกิจขายอะไหล่รถจักรยานยนต์หรือรถยนต์ ธุรกิจเกี่ยวกับอาหารทะเล ธุรกิจเกี่ยวกับวัสดุอุปกรณ์การเกษตร ธุรกิจเกี่ยวกับวัสดุอุปกรณ์ก่อสร้าง ธุรกิจเกี่ยวกับผลิตภัณฑ์ทารกและเด็ก ธุรกิจเกี่ยวกับอุปกรณ์เครื่องเขียนและสิ่งพิมพ์ ธุรกิจเกี่ยวกับอุปกรณ์คอมพิวเตอร์และโทรศัพท์เคลื่อนที่ ธุรกิจเกี่ยวกับดอกไม้และผลิตภัณฑ์ทางการเกษตร ธุรกิจเกี่ยวกับเบเกอรี่ ธุรกิจเกี่ยวกับเวชกรรมและอุปกรณ์ทางการแพทย์ ธุรกิจเกี่ยวกับเครื่องประดับ ธุรกิจสิ่งทอและเครื่องแต่งกาย ธุรกิจเกี่ยวกับเว็บไซต์ต่าง ๆ ที่ถูกเข้าถึงอย่างต่อเนื่องซึ่งจะทำให้เจ้าของธุรกิจทราบถึงความนิยมของเว็บไซต์ที่ทางบริษัทจัดทำขึ้นและอื่น ๆ
2. ได้ขั้นตอนวิธีสำหรับการวิเคราะห์พฤติกรรมกรรมการบริโภคที่มีประสิทธิภาพในแง่มุมมองของความถูกต้อง เวลาที่ใช้ในการประมวลผล และหน่วยความจำที่ใช้สำหรับการคำนวณ
3. ได้ผลงานวิจัยที่ตีพิมพ์ในงานประชุมวิชาการ “*Knowledge and Smart Technology*” (KST), 2017 9th International Conference on. IEEE, 2017
4. สามารถนำแนวความคิดวิเคราะห์พฤติกรรมกรรมการบริโภคและขั้นตอนวิธีที่นำเสนอไปพัฒนาเป็นซอฟต์แวร์เพื่อใช้จริงได้
5. สามารถนำแนวความคิดวิเคราะห์พฤติกรรมกรรมการบริโภคและขั้นตอนวิธีที่นำเสนอไปต่อยอดเพื่อการดำเนินการวิจัยขั้นสูงต่อไป

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

แนวความคิดที่นำเสนอในโครงการวิจัยนี้เป็นแนวคิดที่พัฒนาต่อจากการค้นหารูปแบบที่น่าสนใจภายใต้การวัดความน่าสนใจ 3 แนวมุม ดังนี้

1. การค้นหารูปแบบที่ปรากฏบ่อย (*Mining frequent patterns from transactional database*)
2. การค้นหารูปแบบที่มีค่าคุณประโยชน์สูง (*Mining high utility patterns from transaction database*)
3. การค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ (*Mining frequent regular patterns transaction database*)

2.1.1 การค้นหารูปแบบที่ปรากฏบ่อย (*Mining frequent patterns from transactional database*)

การค้นหารูปแบบที่ปรากฏบ่อยเป็นการค้นหารูปแบบที่น่าสนใจภายใต้การพิจารณาจำนวนครั้ง (หรือ ความบ่อย หรือ ความถี่) ในการปรากฏขึ้นของรูปแบบเหล่านั้น โดยปัญหาการค้นหารูปแบบที่ปรากฏบ่อยจะมุ่งเน้นที่การค้นหาเซตของรายการสินค้าที่ถูกซื้อพร้อมกันบ่อยๆ ซึ่งจะทำให้บริษัท/ห้างร้าน/สถานประกอบการสามารถทราบถึงปริมาณการซื้อที่ซึ่งสามารถนำข้อมูลดังกล่าวไปเป็นส่วนประกอบในการจัดทำโปรโมชั่น การจัดการคลังสินค้า การทำแคตตาล็อกสินค้า การจัดวางชั้นสินค้าและอื่น ๆ โดยปัญหาดังกล่าวสามารถนิยามได้ ดังนี้

นิยามที่ 1 กำหนดให้เซต $I = \{i_1, i_2, \dots, i_m\}$ เป็นเซตของรายการ (items) ที่อาจหมายถึงสิ่งของหรือเหตุการณ์ที่ต้องการหาความสัมพันธ์

นิยามที่ 2 กำหนดให้เซต $X = \{i_j, i_{j+1}, \dots, i_k\} \subseteq I$ เรียกว่า เซตรายการ (set of items, an itemset หรือ a pattern) ซึ่งประกอบด้วยหลายรายการ

นิยามที่ 3 กำหนดให้ $TDB = \{t_1, t_2, \dots, t_n\}$ คือ ฐานข้อมูลรายการหรือฐานข้อมูลแบบทรานแซกชัน (transactional database) ที่ซึ่งแต่ละทรานแซกชัน $t_p \in TDB$ จะประกอบด้วย 1) หมายเลข

กำกับทรานแซกชัน (unique transaction identifier, tid) $tid = p$ และ 2) เซตของรายการ X ที่ถูกบรรจุอยู่ในทรานแซกชันนั้น ๆ (ดังแสดงตัวอย่างในตารางที่ 2-1)

ตารางที่ 2-1 ตัวอย่างฐานข้อมูลรายการที่ประกอบไปด้วยหมายเลขทรานแซกชันและเซตรายการที่ปรากฏในแต่ละทรานแซกชัน

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

ถ้าเซตรายการ $X \subseteq Y$ สามารถสรุปได้ว่า เซตรายการ X ปรากฏขึ้นในทรานแซกชัน t_p หรือทรานแซกชัน t_p มี X บรรจุอยู่ สามารถแสดงในรูปแบบของสัญลักษณ์ได้เป็น p^X ดังนั้นเมื่อทำการตรวจสอบเซตรายการ X ว่าปรากฏขึ้นในทรานแซกชันใดบ้างในฐานข้อมูล TDB จะทำให้ทราบถึงเซตของหมายเลขทรานแซกชันที่มี X ปรากฏ สามารถนิยามได้ดังนี้

นิยามที่ 4 กำหนดให้ $T^X = \{p^X, (p+1^X), \dots, k^X\}$ เมื่อ $1 \leq j < k \leq |TDB|$ คือเซตของหมายเลขทรานแซกชัน (tid) ที่ถูกมี X ปรากฏ โดยสมาชิก (tid) ใน T^X จะถูกเรียงลำดับจากน้อยไปมากเพื่อเพิ่มประสิทธิภาพในการประมวลผล (สามารถเรียกโดยย่อได้เป็น tidset)

นิยามที่ 5 กำหนดให้ s^X คือจำนวนครั้งการปรากฏของเซตรายการ X หรือเรียกว่าค่าสนับสนุนของเซตรายการ X สามารถคำนวณได้เป็น $s^X = |T^X|$

ตัวอย่างที่ 1 จากตารางที่ 2-1 รายการ 'a' ปรากฏอยู่ในทรานแซกชัน $t_1, t_2, t_3, t_5, t_6, t_7, t_9, t_{10}$ ที่ซึ่งจะทำให้ระบุเซตของหมายเลขทรานแซกชันที่มีรายการ 'a' ปรากฏอยู่ ดังนี้ $T^a = \{1^a, 2^a, 3^a, 5^a, 6^a, 7^a, 9^a, 10^a\}$ ดังนั้นค่าสนับสนุนของเซตรายการ 'a' สามารถคำนวณได้ดังนี้ $s^a = |T^a| = |\{1^a, 2^a, 3^a, 5^a, 6^a, 7^a, 9^a, 10^a\}| = 8$

จากนิยามข้างต้นการค้นหาเซตรายการที่ปรากฏที่ปรากฏบ่อยสามารถนิยามได้ดังนี้

นิยามที่ 6 เซตรายการ X จะเป็นเซตรายการที่ปรากฏบ่อยก็ต่อเมื่อ s^X มีค่ามากกว่าหรือเท่ากับค่าขีดแบ่งสนับสนุน (support threshold, σ_s)

2.1.2 การค้นหารูปแบบที่มีค่าคุณประโยชน์สูง (Mining high utility patterns from transaction database)

การค้นหารูปแบบที่มีค่าคุณประโยชน์สูง ถูกพัฒนามาจากการค้นหารูปแบบที่ปรากฏบ่อย โดยการค้นหารูปแบบดังกล่าวจะทำการพิจารณาเกี่ยวกับค่าคุณประโยชน์ของรูปแบบที่อยู่ในรูปของผลกำไร ต้นทุน ค่าความเสี่ยง ค่าความผิดพลาด และอื่น ๆ การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงจะสามารถบอกได้ถึงรายการสินค้าที่ให้ผลกำไรสูงหรือต้นทุนต่ำ นอกจากนั้นยังสามารถบอกได้ถึงปริมาณ/จำนวนสินค้าที่ถูกซื้อได้ ซึ่งจะทำให้บริษัท/ห้างร้าน/สถานประกอบการสามารถทราบถึงปริมาณการซื้อสินค้า แล้วสามารถนำข้อมูลดังกล่าวไปเป็นส่วนประกอบในการจัดทำโปรโมชั่น การจัดการคลังสินค้า การทำแคตตาล็อกสินค้า การจัดวางชั้นสินค้า โดยปัญหาการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงสามารถนิยามได้ ดังนี้

นิยามที่ 7 กำหนดให้แต่ละรายการ $i_j \in I$ (ดังนิยามที่ 1) จะมีค่าคุณประโยชน์ต่อการปรากฏขึ้นหนึ่งครั้งของรายการ i_j เรียกว่า external utility อาทิเช่น ผลกำไรจากสินค้าชิ้นหนึ่ง ๆ ต้นทุนของสินค้าชิ้นหนึ่ง ๆ หรืออื่น ๆ สามารถแทนด้วยสัญลักษณ์ $eu(i_j)$

ตารางที่ 2-2 ตัวอย่างค่าคุณประโยชน์ (ผลกำไร) ของแต่ละรายการ

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

นิยามที่ 8 กำหนดให้แต่ละทรานแซกชัน $t_p = \{i_j, \dots, i_k\}$ ประกอบไปด้วยเซตของรายการ $Y = \{i_j, \dots, i_k\}$ ที่ถูกบรรจุอยู่ในทรานแซกชัน t_p โดยแต่ละ $i_j \in Y$ จะมีจำนวนครั้งของการปรากฏขึ้นของรายการ i_j ในทรานแซกชัน $t_p = \{i_j, \dots, i_k\}$ เรียกว่า *internal utility* สามารถแทนได้ด้วยสัญลักษณ์ $iu(i_j, t_p)$

ตารางที่ 2-3 ตัวอย่างข้อมูลการซื้อสินค้าของลูกค้า

หมายเลขทรานแซกชัน	เซตรายการที่ปรากฏในทรานแซกชัน
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)

นิยามที่ 9 ค่าคุณประโยชน์ของรายการ i_j ที่ปรากฏในทรานแซกชัน t_p จะเป็นผลคูณครั้งของการปรากฏขึ้นของรายการ i_j ในทรานแซกชัน t_p กับค่าคุณประโยชน์ต่อการปรากฏขึ้นหนึ่งครั้งของรายการ i_j สามารถคำนวณและแทนด้วยสัญลักษณ์ $u(i_j, t_p) = iu(i_j, t_p) \times eu(i_j)$

นิยามที่ 10 ค่าคุณประโยชน์ของรายการ X ที่ปรากฏในทรานแซกชัน t_p จะเป็นผลรวมของคุณประโยชน์ของทุกรายการที่เป็นสมาชิกของเซตรายการ X ที่ปรากฏในทรานแซกชัน t_p สามารถคำนวณและแทนด้วยสัญลักษณ์ $u(X, t_p) = \sum_{i_j \in X, X \in t_p} iu(X, t_p) \times eu(i_j)$

นิยามที่ 11 ค่าคุณประโยชน์ของเซตรายการ X ที่ปรากฏในฐานข้อมูลรายการจะเป็นผลรวมของค่าคุณประโยชน์ทั้งหมดของ X ในทรานแซกชันทั้งหมดที่มี X ปรากฏ สามารถคำนวณและแทนสัญลักษณ์ $u(X) = \sum_{X \in t_p, t_p \in TDB} u(X, t_p)$

ตัวอย่างที่ 2 จากตารางที่ 2-2 และ 2-3 ที่ประกอบไปด้วยค่าคุณประโยชน์ของแต่ละรายการและฐานข้อมูลรายการจำนวน 10 ทรานแซกชัน ค่าคุณประโยชน์ของเซตรายการ 'ab' ในทรานแซกชัน t_1 สามารถคำนวณได้ดังนี้ $u(ab, t_1) = (iu(a, t_1) \times eu(a)) + (iu(b, t_1) \times eu(b)) = (3 \times 10) + (6 \times 5) = 60$ ผลรวมของค่าคุณประโยชน์ทั้งหมดของ 'ab' ในทรานแซกชันที่มี 'ab' ปรากฏ (กล่าวคือ $T^{ab} = \{t_1, t_3, t_5, t_7, t_9\}$) สามารถคำนวณได้ ดังนี้ $u(ab) = u(ab, t_1) + u(ab, t_3) + u(ab, t_5) + u(ab, t_7) + u(ab, t_9) = 60 + 75 + 15 + 40 + 40 = 230$

จากนิยามข้างต้นการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงสามารถนิยามได้ ดังนี้

นิยามที่ 12 เซตรายการ X จะเป็นเซตรายการที่มีคุณค่าคุณประโยชน์สูง (high utility itemset) ก็ต่อเมื่อ X มีค่าคุณประโยชน์ $u(X)$ มากกว่าหรือเท่ากับค่าขีดแบ่งคุณประโยชน์ (utility threshold, σ_u) ที่ผู้ใช้กำหนด

2.1.3 การค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ (Mining frequent regular patterns from transaction database)

การค้นหารูปแบบที่ปรากฏบ่อยและสม่ำเสมอได้ถูกพัฒนาต่อยอดจากการค้นหารูปแบบที่ปรากฏบ่อยเช่นกัน วัตถุประสงค์หลักของการค้นหารูปแบบที่ปรากฏบ่อยและสม่ำเสมอจะเป็นการตรวจสอบ/วิเคราะห์พฤติกรรมของการปรากฏขึ้นของรูปแบบภายใต้เงื่อนไขในเชิงความถี่ (จำนวนครั้ง) และความสม่ำเสมอ (ระยะเวลาของการปรากฏซ้ำ) ของการปรากฏ ในการศึกษาเกี่ยวกับพฤติกรรมการปรากฏขึ้นของรูปแบบ ว่ามีพฤติกรรมการปรากฏขึ้นอย่างสม่ำเสมอหรือไม่ เราจะสามารถสังเกตได้จากระยะเวลาในการปรากฏ โดยในการหาระยะเวลาดังกล่าวของรูปแบบ เรา

สามารถคำนวณได้จากการพิจารณาเซตของหมายเลขทรานแซกชันที่มี X ปรากฏขึ้น (T^X) สามารถนิยามได้ ดังนี้

นิยามที่ 13 กำหนดให้ t_p เป็นทรานแซกชันแรกที่มี X ปรากฏ ค่าความสม่ำเสมอของเซตรายการ X ภายใต้การปรากฏขึ้นครั้งแรก สามารถคำนวณและแทนสัญลักษณ์ $fr_{t_p}^X = p$

นิยามที่ 14 กำหนดให้ t_q เป็นทรานแซกชันที่มี X ปรากฏ และมีทรานแซกชัน t_p เป็นทรานแซกชันก่อนหน้าที่มี X ปรากฏ (หมายเหตุ $T^X = \{ \dots, p^X, q^X, \dots \}$) ค่าความสม่ำเสมอของเซตรายการ X ภายใต้การปรากฏขึ้นในทรานแซกชัน t_q จะสามารถคำนวณและแทนสัญลักษณ์ $r_{t_p t_q}^X = q - p$

นิยามที่ 15 กำหนดให้ t_z เป็นทรานแซกชันสุดท้ายที่มี X ปรากฏ และมีทรานแซกชัน t_m เป็นทรานแซกชันสุดท้ายของฐานข้อมูล ค่าความสม่ำเสมอของเซตรายการ X หลังจากปรากฏขึ้นครั้งสุดท้าย จะสามารถคำนวณและแทนสัญลักษณ์ $lr_{t_z}^X = m - z$

จากนิยามที่ 13, 14 และ 15 จะสามารถคำนวณหาค่าความสม่ำเสมอของเซตรายการ X ภายใต้การปรากฏขึ้นครั้งหนึ่งๆ แต่ยังไม่ทราบถึงพฤติกรรมของการปรากฏเกิดขึ้นโดยรวมของเซตรายการ X ที่ซึ่งสามารถนิยามได้ดังนี้

นิยามที่ 16 กำหนดให้ r^X คือช่วงเวลาที่ยาวนานที่สุดที่ไม่มีเซตรายการ X ปรากฏขึ้นในฐานข้อมูลหรือเรียกว่าค่าสม่ำเสมอของเซตรายการ X สามารถคำนวณและแทนสัญลักษณ์ $r^X = \max(fr_{t_p}^X, r_{t_p t_q}^X, \dots, r_{t_y t_z}^X, lr_{t_z}^X)$

ตัวอย่างที่ 3 จากตารางที่ 2-1 ค่าความสม่ำเสมอของเซตรายการ 'a' ที่มีเซต $T^a = \{1^a, 2^a, 3^a, 5^a, 6^a, 7^a, 9^a, 10^a\}$ ในฐานข้อมูล สามารถคำนวณได้ดังนี้ $r^a = \max(fr_{t_1}^a, r_{t_1 t_2}^a, r_{t_2 t_3}^a, r_{t_3 t_5}^a, r_{t_5 t_6}^a, r_{t_6 t_7}^a, r_{t_7 t_9}^a, r_{t_9 t_{10}}^a, lr_{t_{10}}^a) = \max(1, 1, 1, 2, 1, 1, 2, 1, 0) = 2$

ระยะเวลาดังกล่าวสามารถรันตีได้ว่าเซตรายการสินค้า X จะปรากฏขึ้นอย่างน้อย 1 ครั้งในทุก ๆ r^X ทรานแซกชันที่เรียงต่อกัน ซึ่งจากการพิจารณาค่า r^X จะทำให้ทราบถึงพฤติกรรมการปรากฏขึ้นของเซตรายการ โดยจากนิยามที่กล่าวมาข้างต้น การค้นหาเซตรายการที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอสามารถนิยามได้ดังนี้

นิยามที่ 17 เซตรายการ X จะเป็นเซตรายการที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอก็ต่อเมื่อ s^X มีค่ามากกว่าหรือเท่ากับค่าขีดแบ่งสนับสนุน (support/frequency threshold, σ_s) และ r^X มีค่าน้อยกว่าหรือเท่ากับค่าขีดแบ่งความสม่ำเสมอ (regularity threshold, σ_r)

2.2 งานวิจัยที่เกี่ยวข้อง

การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงจะเริ่มจาก (Chan, 2003) และ (Yao, 2004) ได้เห็นถึงปัญหาของการค้นหารูปแบบปรากฏบ่อย (Frequent itemset mining, FIM) ที่จะพิจารณาความน่าสนใจของรูปแบบในแง่มุมมองของความถี่ของการปรากฏเท่านั้น โดยที่รูปแบบปรากฏบ่อยไม่สามารถบ่งบอกได้ถึงคุณประโยชน์ของรูปแบบและความสำคัญของรูปแบบในแง่มุมมองต่าง ๆ อาทิเช่น ผลกำไร ต้นทุน ค่าความผิดพลาด คุณค่าของสิ่งของเหล่านั้น และ อื่น ๆ ได้ ด้วยเหตุนี้จึงทำให้เกิดแนวคิดที่จะทำการวัดความน่าสนใจของรูปแบบด้วยการพิจารณาคุณประโยชน์ของรูปแบบ ซึ่งจากการพิจารณาคุณประโยชน์ของรูปแบบจะทำให้ผู้วิเคราะห์ข้อมูลทราบถึงผลกำไรที่ได้จากรูปแบบเหล่านั้น โดยข้อมูลที่ได้สามารถนำไปประกอบการตัดสินใจเพื่อปรับปรุงคุณภาพ การบริการ หรือวิธีการดำเนินธุรกิจของบริษัท/ห้างร้านต่าง ๆ ได้ นอกเหนือจากการประยุกต์ใช้ในแวดวงธุรกิจ การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงยังสามารถนำไปประยุกต์ใช้กับงานในหลายๆ ด้าน อาทิเช่น การวิเคราะห์ทางพันธุกรรม (Biological gene analysis) การเข้าถึงเว็บแบบมีลำดับ (Web-click sequence analysis) การตรวจสอบการขึ้นลงของดัชนีหุ้น การวัดประสิทธิภาพของการจราจร การตรวจสอบการเข้าใช้งานเซิร์ฟเวอร์ การวิเคราะห์ข้อมูลที่ได้จากเซ็นเซอร์เน็ตเวิร์ค และการวิเคราะห์ข้อมูลการใช้โทรศัพท์ทางไกล เป็นต้น

จากแนวคิดเริ่มต้นเกี่ยวกับการค้นหารูปแบบที่มีค่าคุณประโยชน์สูง (high-utility itemset mining, HUI) ที่ได้กล่าวข้างต้น ได้มีนักวิจัยเป็นจำนวนมากพยายามที่จะพัฒนาขั้นตอนวิธีการค้นหารูปแบบดังกล่าวให้มีประสิทธิภาพมากขึ้น อาทิเช่น (Liu, Y., และคณะ, 2005) ได้พยายามคิดค้นขั้นตอนวิธีที่จะลดทอนปริมาณข้อมูลหรือจำนวนรูปแบบที่ต้องทำการพิจารณา (itemset lattice หรือ search space of itemsets) ด้วยการเสนอค่าประมาณคุณประโยชน์ของรูปแบบ ที่เรียกว่า “*Transaction Weighted Utility, TWU*” จะเป็นผลรวมของค่าคุณประโยชน์ทั้งหมดของทุกทรานแซกชันในฐานข้อมูลที่มีรูปแบบปรากฏ ซึ่งค่าประมาณคุณประโยชน์จะมีค่าสูงกว่าค่าคุณประโยชน์ที่แท้จริงมาก ดังนั้นถ้าค่าประมาณคุณประโยชน์มีค่าน้อยกว่าค่าขีดแบ่งคุณประโยชน์ (utility threshold) แล้วจะทำให้เราสามารถลบรูปแบบออกจากการพิจารณาได้ เนื่องจากรูปแบบดังกล่าวไม่สามารถเป็นผลลัพธ์ได้ แนวคิดเกี่ยวกับการประมาณค่าคุณประโยชน์ได้ถูกประยุกต์ใช้ในหลาย ๆ ขั้นตอนวิธี เช่น การค้นหารูปแบบที่มีค่าประโยชน์สูงแบบล่างขึ้นบน (Erwin, A., และคณะ, 2007), โครงสร้างต้นไม้ที่มีประสิทธิภาพสำหรับการค้นหารายการที่มีค่าคุณประโยชน์สูงในฐานข้อมูลที่เพิ่มขึ้น (Ahmed, C.F., และคณะ, 2009), โครงสร้างต้นไม้ที่มีประสิทธิภาพสำหรับการค้นหารายการที่มีค่าคุณประโยชน์สูงด้วยวิธี HUP-Growth (Lin, C.W., และคณะ, 2011), การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงอย่างมีประสิทธิภาพด้วยการวัดค่าคุณประโยชน์แบบเฉลี่ย (Hong, T.-P., และคณะ, 2011), การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงด้วยวิธี UP-Growth และ UP-Growth+ (Tseng,

V.S., และคณะ, 2013), การทำดัชนีที่มีประสิทธิภาพในการค้นหารายการที่มีค่าคุณประโยชน์สูง (Lan, G.-C., และคณะ, 2014-a)

แต่อย่างไรก็ตาม การประมาณค่าคุณประโยชน์ของรูปแบบด้วยค่า TWU จะเป็นค่าที่เกินจากค่าคุณประโยชน์ไปมาก ที่ซึ่งจะทำให้เราไม่สามารถลดทอนจำนวนรูปแบบที่ต้องการพิจารณาได้อย่างเต็มที่ จึงเป็นเหตุให้ (Liu, M., และ Qu, J.-G., 2012) ได้คิดค้นค่าประมาณคุณประโยชน์แบบกระชับที่เรียกว่า “*tight over-estimate utility, tou*” ที่มีค่าเกินจริงน้อยกว่าค่า TWU ซึ่งจะช่วยเราสามารถลดทอนจำนวนรูปแบบที่ต้องทำการพิจารณาได้มากกว่าเดิม และนับตั้งแต่แนวคิดของ Liu ได้เผยแพร่ออกไปทำให้มีหลาย ๆ งานประยุกต์ใช้แนวคิดดังกล่าวในการลดทอนปริมาณข้อมูลหรือจำนวนรูปแบบที่ต้องทำการพิจารณา อาทิเช่น การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงโดยใช้ค่าคุณประโยชน์โดยประมาณ (Fournier-Viger, P., และคณะ, 2014-a), การตัดแต่งกิ่งสำหรับการค้นหารูปแบบที่มีค่าคุณประโยชน์สูง (Krishnamoorthy, S., 2015)

นอกเหนือจากการพัฒนาประสิทธิภาพในการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงภายใต้การกำหนดค่าขีดแบ่งคุณประโยชน์แล้ว นักวิจัยทางด้านการทำเหมืองข้อมูลได้คิดค้นแนวทาง วิธีการ และขั้นตอนต่าง ๆ ที่ช่วยให้การค้นหารูปแบบดังกล่าวมีความยุ่งยากลดลง รวมถึงเพิ่มความสามารถในแง่มุมต่าง ๆ อีกเป็นจำนวนมาก อาทิเช่น 1) การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงจากฐานข้อมูลที่มีการเปลี่ยนแปลงทั้งการเพิ่ม/ลดจำนวนทรานแซกชัน รวมถึงการเปลี่ยนแปลงข้อมูลในทรานแซกชันในภายหลัง (Lin, M., และ Qu, J.-F., 2012), (Lin, C.-W., และคณะ, 2014), (Yun, U., และ Ryang, H., 2015), 2) การค้นหารูปแบบที่มีคุณประโยชน์สูงจากข้อมูลกระแส (Chu, C.-J., และคณะ, 2008), (Ahmed, C.F., และคณะ, 2012), (Feng, L., และ Jin, B., 2013), 3) การค้นหารูปแบบที่มีคุณประโยชน์สูงโดยการพิจารณาผลกำไรร่วมกับผลขาดทุน (Chu, 2009), (Li, H.-F., และคณะ, 2011), (Fournier-Viger, P., 2014-c), (Lan, G.-C., และคณะ, 2014-b), (Fournier-Viger, P., และ Zida, S., 2015), 4) การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงที่ไม่มีการซ้ำซ้อนของรูปแบบ (Wu, C.W., และคณะ, 2011), (Lin, M.-Y., และคณะ, 2012), (Fournier-Viger, P., และคณะ, 2014-b), (Tseng, V.S., และคณะ, 2015-a), 5) การลดทอนความยุ่งยากของการกำหนดค่าขีดแบ่งคุณประโยชน์ด้วยการกำหนดจำนวนรูปแบบที่ให้ผลกำไรสูงสุด (Wu, C.-W., และคณะ, 2012), (Zihayat, M., และ An A., 2014), (Ryang, H., และ Yun, U., 2015), (Tseng, V.S., และคณะ, 2015-b)

ในส่วนของการพิจารณาพฤติกรรมของการปรากฏขึ้นของรูปแบบ (Tanbeer, S.K., และคณะ, 2009) ชี้ให้เห็นว่าการวัดความน่าสนใจที่ค้นพบจากฐานข้อมูลด้วยการพิจารณาความถี่ (จำนวนครั้ง) ของการปรากฏเพียงอย่างเดียวอาจไม่เพียงพอต่อการค้นหารูปแบบที่น่าสนใจ ด้วยเหตุนี้ Tanbeer และคณะ ได้นำเสนอแนวความคิดในการศึกษาพฤติกรรมของการปรากฏขึ้นของรูปแบบ

ด้วยการค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ ซึ่งจะเป็นการศึกษาพฤติกรรมการปรากฏขึ้นของรูปแบบทั้งในเชิงความถี่และความสม่ำเสมอของการปรากฏ แนวความคิดนี้สามารถนำไปประยุกต์ใช้ได้ในงานหลาย ๆ ด้าน อาทิเช่น ผู้จัดการหรือผู้บริหารของธุรกิจค้าปลีกอาจจะสนใจรายการสินค้าที่ถูกซื้อบ่อย ๆ และถูกซื้ออย่างสม่ำเสมอมากกว่ารายการสินค้าที่ถูกซื้อบ่อย ๆ เพียงอย่างเดียว เพื่อที่จะทำการจัดเตรียมสินค้าให้พอเหมาะกับความต้องการของผู้บริโภค และยังสามารถช่วยในการจัดทำโปรโมชั่นสำหรับสินค้าที่ถูกซื้อบ่อย ๆ ร่วมกับสินค้าที่ไม่ได้ถูกซื้อบ่อยได้อีกด้วย ในส่วนของการพัฒนาการออกแบบเว็บไซต์หรือการดูแลรักษาเว็บไซต์ ผู้ดูแลเว็บไซต์อาจจะสนใจความสม่ำเสมอของการคลิกเพื่อเรียกดูข้อมูลในเว็บเพจที่ต่อเนื่องกันเพื่อนำไปปรับปรุงข้อความหรือเนื้อหาของเว็บไซต์ให้มีความน่าสนใจยิ่งขึ้น ในส่วนของการวิเคราะห์ข้อมูลทางพันธุกรรม กลุ่มของยีนที่ปรากฏบ่อยและสม่ำเสมออาจบ่งบอกถึงข้อมูลที่สำคัญให้แก่นักวิทยาศาสตร์ได้ ในส่วนตลาดหุ้นกลุ่มของหุ้นที่มีดัชนีที่มีการเพิ่มขึ้นอย่างสม่ำเสมออาจจะได้รับความน่าสนใจจากนักลงทุนต่าง ๆ และอื่น ๆ

ในการหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ ผู้ใช้จะต้องทำการกำหนดค่าพารามิเตอร์ 2 ค่าด้วยกันคือ 1) ค่าขีดแบ่งสนับสนุน และ 2) ค่าขีดแบ่งความสม่ำเสมอ เพื่อใช้วัดความน่าสนใจหรือความสำคัญของรูปแบบภายใต้พฤติกรรมการเกิดขึ้นของรูปแบบเหล่านั้น แต่อย่างไรก็ดี เป็นที่ทราบกันดีว่า “ถ้าเราไม่ได้มีความรู้ในข้อมูลมาก่อน การกำหนดค่าขีดแบ่งสนับสนุนเพื่อที่จะได้รับรูปแบบที่น่าสนใจและมีความสำคัญมากที่สุดจะเป็นเรื่องที่ยุ่งยากและลำบาก” โดยถ้าเรากำหนดค่าขีดแบ่งสนับสนุนสูงเกินไป อาจทำให้เราได้ผลลัพธ์เป็นจำนวนน้อยหรืออาจจะไม่ได้ผลลัพธ์เลย ในกรณีนี้ เราจำเป็นต้องคาดเดาค่าขีดแบ่งให้มีความน้อยลง แล้วทำการค้นหาผลลัพธ์ใหม่อีกครั้งซึ่งอาจจะได้รับผลลัพธ์ที่ดีขึ้นหรือไม่ก็ได้ แต่ในกรณีที่ค่าขีดแบ่งถูกกำหนดให้มีความน้อย อาจทำให้เราได้ผลลัพธ์ออกมาเป็นจำนวนมากเกินกว่าที่เราจะทำการพิจารณาองค์ความรู้ได้ และการค้นหาผลลัพธ์จะใช้เวลาค่อนข้างมากอีกด้วย

จากปัญหาข้างต้นดังกล่าว จึงมีงานวิจัยที่ทำการพัฒนาต่อยอดจากงานของ Tanbeer โดยมีวัตถุประสงค์ที่จะหลีกเลี่ยงการกำหนดค่าขีดแบ่งสนับสนุน โดยกำหนดให้ผู้ใช้ทำการกำหนดจำนวนผลลัพธ์ (รูปแบบ) ที่ต้องการแทนด้วยการหารูปแบบทั้งสี่รูปแบบ ซึ่งปรากฏในฐานข้อมูลอย่างสม่ำเสมอและปรากฏบ่อยที่สุด ภายใต้ปัญหานี้ ผู้ที่ต้องการค้นหาหารูปแบบจะต้องทำการกำหนดค่าพารามิเตอร์ 2 ค่าด้วยกันคือ 1) ค่าขีดแบ่งความสม่ำเสมอ และ 2) จำนวนผลลัพธ์ที่ต้องการในการค้นหาหารูปแบบดังกล่าวได้อย่างรวดเร็ว (Amphawan, K., และคณะ, 2009), (Amphawan, K., และคณะ, 2012), (Amphawan, K., และ Lenca, P., 2015)

นอกเหนือจากการหลีกเลี่ยงความยุ่งยากในการกำหนดค่าขีดแบ่งสนับสนุน การค้นหาแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ ได้ถูกพัฒนาอย่างต่อเนื่องในหลาย ๆ แง่มุม อาทิเช่น การค้นหาหารูปแบบที่ปรากฏบ่อยและสม่ำเสมอ จากฐานข้อมูลที่มีการเพิ่มเติมข้อมูลในฐานข้อมูล

(Tanbeer, S.K., และคณะ, 2010-a) และจากฐานข้อมูลแบบกระแส (Kumar, G.V., และ Kumari, V.V., 2012), (Tanbeer, S.K., และคณะ, 2010-b) การหารูปแบบที่เกิดขึ้นอย่างสม่ำเสมอที่ประกอบด้วยรูปแบบที่ปรากฏบ่อยและปรากฏไม่บ่อย (Surana, A., และคณะ, 2012), การค้นหารูปแบบที่ปรากฏบ่อยและสม่ำเสมอด้วยการกำหนดเงื่อนไขเกี่ยวกับค่าสนับสนุน (Kiran, R.U., และ Reddy, P.K., 2010) นอกเหนือจากงานวิจัยข้างต้นแล้วยังมีงานวิจัยที่มุ่งเน้นที่การพัฒนาการค้นหา รูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอในแง่มุมต่าง ๆ อีกมากมาย เช่น การค้นหารูปแบบ เป็นระยะ ๆ ที่เป็นไปได้ในฐานข้อมูลขนาดใหญ่ (Luo, X., และคณะ, 2013), (Kiran, R.U., และ Kitsuregawa, M., 2013), เทคนิคใหม่ในการลดช่องว่างในการค้นหารายการปรากฏบ่อย (Kiran, R.U., และ Kitsuregawa, M., 2014), การค้นหาโรคที่ปรากฏบ่อยเป็นประจำทางการแพทย์จาก ฐานข้อมูล (Khaleel, M.A., และคณะ, 2014)

บทที่ 3

วิธีการดำเนินการวิจัย

จากบทที่กล่าวมาในข้างต้น การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงเป็นการค้นหารูปแบบที่น่าสนใจภายใต้การพิจารณาค่าคุณประโยชน์ของรูปแบบ แต่การค้นหารูปแบบดังกล่าวไม่ได้พิจารณาพฤติกรรมการปรากฏขึ้น อาทิเช่น ความถี่ของการปรากฏ ความสม่ำเสมอของการปรากฏขึ้น เป็นต้น ด้วยเหตุนี้ (Amphawan, K., และ Surarerks, A., 2015) ได้นำเสนอการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงโดยปรากฏขึ้นอย่างสม่ำเสมอ (*High-Utility Regular Itemsets Mining, HURIM*) โดยทำการเพิ่มเติมเงื่อนไขการปรากฏอย่างสม่ำเสมอ ร่วมกับการพิจารณาค่าคุณประโยชน์ของรายการต่างๆ แต่อย่างไรก็ตาม การค้นหารูปแบบที่มีการปรากฏอย่างสม่ำเสมอ ยังอาจไม่เพียงพอสำหรับการค้นหารูปแบบใหม่ๆ ที่มีความหลากหลาย ดังนั้น ในบทนี้จะนำเสนอการค้นหารูปแบบที่ให้ค่าคุณประโยชน์สูงโดยปรากฏอย่างไม่สม่ำเสมอ (*High-Utility Irregular Itemsets Mining, HUIIM*) โดยรูปแบบในลักษณะนี้จะสามารถบ่งบอกถึงพฤติกรรมผู้บริโภคเกี่ยวกับการซื้อสินค้าที่มีผลตอบแทนสูงโดยสินค้าเหล่านั้นถูกซื้ออย่างไม่สม่ำเสมอ ซึ่งจะทำให้ผู้ประกอบการสามารถทราบถึงพฤติกรรมการซื้อสินค้าหรือบริการ และพฤติกรรมในเชิงลึกของผู้บริโภคได้ มีส่วนช่วยเป็นข้อมูลประกอบการตัดสินใจเกี่ยวกับการบริหารจัดการสินค้าคงคลัง การจัดทำโปรโมชั่นเพื่อส่งเสริมการขาย และอื่น ๆ

3.1 นิยามที่เกี่ยวข้องกับการค้นหารูปแบบที่ให้ค่าคุณประโยชน์สูงโดยปรากฏอย่างไม่สม่ำเสมอ

จากนิยามพื้นฐานทั้งหมดในบทที่แล้ว การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงจะเป็นการค้นหารูปแบบ X ใด ๆ ที่มีค่าคุณประโยชน์ $u(X)$ มากกว่าหรือเท่ากับค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด (σ_u) แต่อย่างไรก็ตาม การค้นหารูปแบบดังกล่าวไม่สามารถประยุกต์ใช้ downward closure property ในการลดทอนปริภูมิสถานะ ด้วยเหตุนี้ (Liu, Y., และคณะ, 2005) จึงได้เสนอแนวคิดเกี่ยวกับ “*transaction-weighted utility, TWU*” จะเป็นค่าประมาณของค่าคุณประโยชน์ที่ทำให้สามารถประยุกต์ใช้ downward closure property ในการลดทอนปริภูมิสถานะได้ สามารถนิยามได้ดังนี้

นิยามที่ 20 ค่าคุณประโยชน์ของทรานแซกชัน t_p เป็นผลรวมของค่าคุณประโยชน์ของทุกรายการที่ปรากฏในทรานแซกชัน t_p สามารถคำนวณและแทนด้วยสัญลักษณ์ $tu(t_p) = \sum_{i_j \in t_p} u(i_j, t_p)$

ตัวอย่างที่ 4 กำหนดให้ฐานข้อมูลประกอบด้วย 10 ทรานแซกชัน โดยมีรายการที่แตกต่างกัน 8 รายการ แสดงในตารางที่ 3-1 และค่าคุณประโยชน์ของแต่ละรายการ แสดงในตารางที่ 3-2 ค่าคุณประโยชน์ของทรานแซกชัน $t_1 = \{a(2), b(3), d(13), f(2)\}$ สามารถคำนวณได้เป็น $tu(t_1) = u(a, t_1) + u(b, t_1) + u(d, t_1) + u(f, t_1) = 4 + 9 + 260 + 50 = 323$

ตารางที่ 3-1 ฐานข้อมูลรายการที่ประกอบไปด้วยหมายเลขทรานแซกชันและเซตรายการที่ปรากฏในทรานแซกชันที่มีจำนวนของการปรากฏขึ้นของแต่ละรายการ

หมายเลขทรานแซกชัน (tid)	เซตรายการที่ปรากฏในทรานแซกชัน (a set of items or an itemset)
1	a(2), b(3), d(13), f(2)
2	c(1), e(4), g(1), h(4)
3	a(3), b(2)
4	a(2), g(1)
5	a(1), b(2), d(1)
6	c(20)
7	f(1), h(8)
8	a(1), e(4), g(1)

ตารางที่ 3-2 ตารางค่าคุณประโยชน์ของแต่ละรายการ

รายการ	a	b	c	d	e	f	g	h
ค่าคุณประโยชน์	2	3	4	20	2	25	5	3

นิยามที่ 21 ค่า TWU ของเซตรายการ X ในฐานข้อมูลรายการ TDB จะเป็นค่าประมาณคุณประโยชน์ของ X ที่เกิดจากผลรวมของค่าคุณประโยชน์ของทุกทรานแซกชันในฐานข้อมูลรายการ TDB ที่มี X ปรากฏ สามารถคำนวณและแทนด้วยสัญลักษณ์ $TWU(X) = \sum_{t_p \in D, X \subseteq t_p} tu(t_p)$

ตัวอย่างที่ 5 จากฐานข้อมูลในตารางที่ 3-1 และตารางค่าคุณประโยชน์ที่ 3-2 เราจะสังเกตเห็นได้ว่าเซตรายการ 'ad' ปรากฏในทรานแซกชัน t_1 และ t_5 ดังนั้น ค่าประมาณคุณประโยชน์ของเซตรายการ 'ad' ในฐานข้อมูล TDB สามารถคำนวณได้เป็น $TWU(ad) = tu(t_1) + tu(t_5) = 323 + 28 = 351$

จากนิยามข้างต้นเซตรายการ X ใด ๆ มีค่า $TWU(X)$ น้อยกว่าค่าขีดแบ่งคุณประโยชน์แล้วจะทำให้ทุกซูปเปอร์เซตของ X มีค่าคุณประโยชน์น้อยกว่าค่าขีดแบ่งคุณประโยชน์ด้วยเช่นกัน ดังนั้นเราสามารถตัดการพิจารณาเซตรายการ X และทุกเซตรายการที่เป็นซูปเปอร์เซตของรายการ X ออกจากการพิจารณาได้เนื่องจาก เซตรายการ X และทุกเซตรายการที่เป็นซูปเปอร์เซตของ X จะมีค่าคุณประโยชน์น้อย

แต่อย่างไรก็ตาม TWU ของเซตรายการหนึ่งๆ จะเป็นค่าประมาณคุณประโยชน์ที่มีค่ามากกว่าประมาณคุณประโยชน์จริงค่อนข้างมาก จากเหตุผลข้างต้น (Liu, M., และ Qu, J.F., 2012) ได้คิดค้นแนวความคิดเกี่ยวกับค่าประมาณคุณประโยชน์ที่มีความกระชับ (มีค่าใกล้เคียงกับค่าคุณประโยชน์จริงมากขึ้น)

นิยามที่ 22 กำหนดให้ \succ แสดงถึงลำดับของรายการในเซตรายการ I ที่ซึ่ง $i_1 < i_2 < \dots < i_n$ หมายถึงรายการ i_1 อยู่ในลำดับก่อนหน้ารายการ i_2 และรายการ i_2 อยู่ในลำดับก่อนหน้ารายการ i_3 และรายการ i_n เป็นรายการในลำดับสุดท้าย

นิยามที่ 23 ค่าคุณประโยชน์ส่วนเหลือ (remaining utility) ของเซตรายการ X ในทรานแซกชัน หมายถึงผลรวมของค่าคุณประโยชน์ของทุกรายการที่ปรากฏในทรานแซกชัน t_p และรายการเหล่านั้นมีลำดับหลังจาก X สามารถคำนวณและแทนด้วยสัญลักษณ์ $ru(X, t_p) = \sum_{i_j \in t_p, X < i_j} u(i_j, t_p)$

ตัวอย่างที่ 6 จากฐานข้อมูลในตารางที่ 3-1 และตารางค่าคุณประโยชน์ที่ 3-2 ถ้าลำดับของรายการทั้งหมดเป็น $a < b < c < d < e < f < g < h$ ค่าคุณประโยชน์ส่วนเหลือของเซตรายการ 'ad' ในทรานแซกชัน t_1 สามารถคำนวณได้เป็น $ru(ad, t_1) = u(f, t_1) = 50$

นิยามที่ 24 ค่าคุณประโยชน์ส่วนเหลือของเซตรายการ X ในฐานข้อมูล TDB จะเป็นค่าผลรวมของค่าคุณประโยชน์ส่วนเหลือของเซตรายการ X ในทุกทรานแซกชันที่มี X ปรากฏ สามารถคำนวณและแทนด้วยสัญลักษณ์ $ru(X) = \sum_{t_p \in D, X \subseteq t_p} ru(X, t_p)$

ตัวอย่างที่ 7 จากฐานข้อมูลตารางที่ 3-1 และตารางค่าคุณประโยชน์ที่ 3-2 ค่าคุณประโยชน์ส่วนเหลือของเซตรายการ 'ad' ในฐานข้อมูล TDB สามารถคำนวณได้เป็น $ru(ad) = ru(ad, t_1) + ru(ad, t_5) = 50 + 0 = 50$

นิยามที่ 25 ค่าประมาณคุณประโยชน์แบบกระชับของเซตรายการ X ในฐานข้อมูล TDB จะเป็นค่าผลรวมระหว่างค่าคุณประโยชน์จริงของเซตรายการ X กับค่าคุณประโยชน์ส่วนเหลือของเซตรายการ X ในฐานข้อมูลรายการ TDB สามารถคำนวณและแทนด้วยสัญลักษณ์ $tou(X) = u(X) + ru(X)$

ตัวอย่างที่ 8 จากฐานข้อมูลตารางที่ 3-1 และตารางค่าคุณประโยชน์ที่ 3-2 ค่าประมาณคุณประโยชน์แบบกระชับของเซตรายการ ‘ad’ สามารถคำนวณได้เป็น $tu(ad) = 286 + 50 = 336$

ดังที่กล่าวมาข้างต้น เราสามารถระบุได้ว่า ถ้าเซตรายการ X มีค่าประมาณคุณประโยชน์แบบกระชับน้อยกว่าค่าขีดแบ่งคุณประโยชน์แล้ว เซตรายการ Y ใดที่เกิดจากการรวมกันระหว่างเซตรายการ X และรายการ i_j ใด ๆ ที่มีลำดับหลังจาก X กล่าวคือ $Y = X \cup i_j$ จะมีค่าคุณประโยชน์น้อยกว่าค่าขีดแบ่งคุณประโยชน์เสมอ

จากแนวคิดและนิยามข้างต้น สามารถช่วยให้ลดทอนปริภูมิสถานะของการพิจารณาเซตรายการที่มีค่าคุณประโยชน์สูงได้ แต่อย่างไรก็ตาม เมื่อทำการพิจารณาการปรากฏอย่างไม่สม่ำเสมอร่วมกับค่าคุณประโยชน์ จะทำให้ปัญหาการค้นหาเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏไม่สม่ำเสมอสามารถนิยามได้ดังนี้

นิยามที่ 26 เซตรายการ X จะเป็นเซตรายการที่มีคุณค่าคุณประโยชน์สูงและปรากฏไม่สม่ำเสมอก็ต่อเมื่อ X มีค่าคุณประโยชน์ $u(X)$ ไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด (σ_u) และ X มีความสม่ำเสมอ τ^x มากกว่าค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด (σ_r) ที่ผู้ใช้กำหนด

3.2 ขั้นตอนวิธีการนำเสนอ HUIIM

งานวิจัยนี้ได้นำเสนอขั้นตอนวิธีที่มีชื่อว่า “High-Utility Itemset with Irregular occurrence Miner, HUIIM” ที่สามารถค้นหารูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอจากฐานข้อมูล ภายใต้ค่าขีดคุณประโยชน์ที่ผู้ใช้กำหนด (σ_u) และค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด (σ_r) ที่ซึ่งทำการอ่านข้อมูลจากฐานข้อมูลเพียงครั้งเดียวโดยจะทำการคำนวณและจัดเก็บค่าคุณประโยชน์ของทุกทรานแซกชันในฐานข้อมูลไว้ใน $tuList$ นอกจากนั้น HUIIM ประยุกต์ใช้แนวคิดค่าประมาณคุณประโยชน์ และค่าประมาณคุณประโยชน์แบบกระชับของเซตรายการในการลดทอนปริภูมิสถานะของการพิจารณาเซตรายการ และใช้โครงสร้างข้อมูล “New-modified Utility List, NUL” ที่ได้ปรับปรุงใหม่เพื่อจัดเก็บข้อมูลหมายเลขกำกับทรานแซกชันพร้อมกับค่าคุณประโยชน์ของแต่ละรายการ

3.2.1 โครงสร้างข้อมูล *New-modified Utility List (NUL)*

โครงสร้างข้อมูล *NUL* ของเซตรายการ X หนึ่งๆ จะประกอบไปด้วยลิสต์ของ 4-tuple คือ $\langle p, u(X, t_p), ru(X, t_p), up(X, t_p) \rangle$ ที่ซึ่ง 1) p คือ หมายเลขทรานแซกชันที่มีเซตของรายการ X ปรากฏ 2) $u(X, t_p)$ คือ ค่าคุณประโยชน์ของเซตรายการ X ในทรานแซกชัน t_p 3) $ru(X, t_p)$ คือ ค่าคุณประโยชน์ส่วนเหลือของเซตรายการ X ในทรานแซกชัน t_p และ 4) $up(X, t_p)$ คือ ค่าคุณประโยชน์ของ prefix items ของเซตรายการ X ในทรานแซกชัน t_p (prefix items ของเซตรายการ X คือ เซตของทุกรายการในเซตรายการ X ยกเว้นรายการสุดท้าย ตัวอย่างเช่น กำหนดให้เซตรายการ $X = \{i_p, \dots, i_y, i_z\}$ prefix items ของเซตรายการ X จะเป็น $X - i_z = \{i_p, \dots, i_y\}$)

ตัวอย่างที่ 9 พิจารณาเซตรายการ 'ab' จากตารางฐานข้อมูลที่ 3-1 และตารางค่าคุณประโยชน์ที่ 3-2 เซตรายการ 'ab' ปรากฏในทรานแซกชันฐานข้อมูล คือ t_1, t_3 และ t_5 สามารถสร้าง NUL^{ad} ได้เป็น $\{\langle 1, 13, 310, 4 \rangle, \langle 3, 12, 0, 6 \rangle, \langle 5, 8, 20, 2 \rangle\}$ โดยที่สมาชิกอันดับแรกของ NUL^{ad} ปรากฏในทรานแซกชันที่ t_1 มีค่าคุณประโยชน์ $u(ad, t_1) = 13$ ค่าคุณประโยชน์ส่วนเหลือ $ru(ab, t_1) = 310$ และค่าคุณประโยชน์ของรายการ prefix $up(ab, t_1) = u(a, t_1) = 4$ สำหรับการคำนวณต่อไปตามลำดับ

3.2.2 ขั้นตอนวิธี *HUIIM*

ดังที่กล่าวมาในข้างต้น ขั้นตอนวิธี *HUIIM* ได้ประยุกต์แนวคิดค่าคุณประโยชน์ส่วนเหลือและค่าประมาณคุณประโยชน์แบบกระชับเพื่อทำการลดทอนปริภูมิสถานะ นอกจากนั้นยังทำการประยุกต์ใช้โครงสร้างข้อมูล *New-modified Utility List* เพื่อใช้ในการจัดเก็บข้อมูลเกี่ยวกับการปรากฏขึ้นของเซตรายการหนึ่ง ๆ พร้อมค่าคุณประโยชน์ของเซตรายการนั้น ๆ ที่ปรากฏในทรานแซกชันหนึ่ง ๆ ระหว่างค้นหาผลลัพธ์ นอกจากนั้น *HUIIM* ทำการอ่านฐานข้อมูลเพียงครั้งเดียวด้วยการใช้ลิสต์จัดเก็บค่าคุณประโยชน์ทุกทรานแซกชัน (*tuList*) โดยขั้นตอนวิธี *HUIIM* ประกอบไปด้วย 2 ขั้นตอนหลักในการคำนวณหาผลลัพธ์คือ

1) การระบุถึง single-item ที่มีค่าประมาณคุณประโยชน์สูง โดยเริ่มแรกจะทำการสร้างโครงสร้างต้นไม้เพื่อใช้เก็บรายการและเซตรายการ (เรียกว่า *HUII-tree*) จากนั้นทำการอ่านฐานข้อมูลหนึ่งครั้งเพื่อจัดเก็บข้อมูลรายการที่ปรากฏขึ้นไว้ในโหนดลูกของ R ของ *HUII-tree* โดยเรียกขั้นตอนนี้ว่า *HUIIM-DBscanning*

2) การค้นหาผลลัพธ์จาก *HUII-tree* ที่สร้างขึ้นในขั้นตอนแรก โดยเรียกขั้นตอนนี้ว่า *HUIIM-Mining*

Algorithm 1: HUIIM-DBscanning**Input:** D, σ_u, σ_r **Output:** *HUII-tree, HUII*

```

1: create tuList
2: create and initial root  $R$  of HUII-tree
3: create a node for each item  $i_j \in I$  and set as a child node of  $R$ 
4: create a set of HUII and initial to be  $\phi$ 
5: for each transaction  $t_p$  in  $D$  do
6:    $tu(t_p) \leftarrow \sum_{i_j \in t_p} u(i_j, t_p)$  and collect  $tu(t_p)$  to tuList( $t_p$ )
7:   for each item  $i_j$  in transaction  $t_p$  do
8:     compute  $r(i_j) \leftarrow \max(r(i_j), p - q)$  where  $q$  is the tid of the last
       occurrence of  $i_j$  in the transaction  $t_q$  and  $q$  can be retrieved from
       the tid of the last entry in  $NUL^{i_j}$ 
9:     compute  $u(i_j, t_p) \leftarrow eu(i_j) \times iu(i_j, t_p)$  and then create and
       collect an entry  $\langle p, u(i_j, t_p), 0, 0 \rangle$  at tail of  $NUL^{i_j}$ 
10:    update  $TWU(i_j) \leftarrow TWU(i_j) + tuList(t_p)$ 
11:  for each child node of  $R$  with item  $i_k$  do
12:    if  $TWU(i_k) < \sigma_u$  then
13:      for each entry  $e = \langle p, u(i_k, t_p), 0, 0 \rangle$  in  $NUL^{i_k}$  do
14:        update tuList( $t_p$ )  $\leftarrow tuList(t_p) - u(i_k, t_p)$ 
15:        remove the node of item  $i_k$  out of HUII-tree
16:  create a new tuList called temptuList and copy all entries of tuList to
    temptuList
17:  for each child node of  $R$  with item  $i_k$  do
18:    for each entry  $e = \langle p, u(i_k, t_p), 0, 0 \rangle$  in  $NUL^{i_k}$  do
19:      temptuList( $t_p$ )  $\leftarrow temptuList(t_p) - u(i_k, t_p)$ 
20:       $ru(i_k, t_p) \leftarrow temptuList(t_p)$ 
21:      update entry  $e = \langle p, u(i_k, t_p), 0, 0 \rangle$  to be
         $e = \langle p, u(i_k, t_p), ru(i_k, t_p), 0 \rangle$ 
22:       $u(i_k) \leftarrow u(i_k) + u(i_k, t_p)$ 
23:       $ru(i_k) \leftarrow ru(i_k) + ru(i_k, t_p)$ 
24:    if  $u(i_k) \geq \sigma_u$  and  $r(i_k) > \sigma_r$  then
25:      HUII  $\leftarrow HUII \cup i_k$ 

```

} pruning of
low-utility
items

ภาพที่ 3-1 ขั้นตอนการอ่านฐานข้อมูลและระบุนรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอ

ดังแสดงรายละเอียดในภาพที่ 3-1 “HUIIM-DBscanning” ขั้นตอนการอ่านฐานข้อมูล จะเริ่มจากการสร้างหรือจองพื้นที่หน่วยความจำในการจัดเก็บข้อมูลเบื้องต้นดังนี้ ทำการสร้างลิสต์ *tuList* จะมีขนาดเท่ากับจำนวนทรานแซกชันของฐานข้อมูล เพื่อใช้ในการจัดเก็บค่าคุณประโยชน์ของทุกทรานแซกชันในฐานข้อมูล (การสร้าง *tuList* จะสามารถช่วยลดจำนวนครั้งในการอ่านทรานแซกชันได้) จากนั้นทำการสร้างโครงสร้างต้นไม้ *HUII-tree* และกำหนดโหนดราก R และทำการสร้างโหนดสำหรับจัดเก็บข้อมูลแต่ละรายการ $i_j \in I$ และกำหนดให้เป็นโหนดลูกของ R โดยแต่ละโหนดจะบรรจุไปด้วย 5 ข้อมูล ดังนี้ 1) $u(i_j)$ คือ ผลรวมค่าคุณประโยชน์ของรายการ i_j ในทุกทรานแซกชันที่มี i_j ปรากฏ 2) $ru(i_j)$ คือ ผลรวมค่าคุณประโยชน์ส่วนเหลือของรายการ i_j ในทุกทรานแซกชันที่มี i_j ปรากฏ 3) $TWU(i_j)$ คือ ค่าประมาณคุณประโยชน์ของ i_j เกิดจากผลรวมของค่าคุณประโยชน์ทรานแซกชันที่มี i_j ปรากฏในฐานข้อมูล 4) $r(i_j)$ คือ ค่าความสม่ำเสมอของเซตรายการ i_j 5) NUL^{i_j} คือ โครงสร้างข้อมูลที่ใช้เก็บข้อมูลการปรากฏขึ้นและค่าคุณประโยชน์ของรายการ i_j

เมื่อทำการสร้างหรือจองพื้นที่หน่วยความจำเสร็จ จะเริ่มทำการอ่านข้อมูลแต่ละทรานแซกชัน $t_p \in TDB$ เพื่อทำการคำนวณค่าคุณประโยชน์ทรานแซกชัน $tu(t_p)$ (สามารถคำนวณได้จาก $tu(t_p) = \sum_{i_j \in t_p} u(i_j, t_p)$) และจัดเก็บไว้ใน $tuList(t_p)$ จากนั้นพิจารณาแต่ละรายการ i_j ที่ปรากฏขึ้นในทรานแซกชัน t_p โดยในการพิจารณาจะทำการคำนวณค่าความสม่ำเสมอ $r(i_j)$ (สามารถคำนวณได้จาก $r(i_j) = \max(r(i_j), p - q)$) และทำการคำนวณค่าคุณประโยชน์ $u(i_j, t_p)$ (สามารถคำนวณได้จาก $u(i_j, t_p) = iu(i_j, t_p) \times eu(i_j)$) แล้วจัดเก็บไว้ใน NUL^j ที่อยู่ในรูปแบบ $\langle p, u(i_j, t_p), 0, 0 \rangle$ จากนั้นทำการอัปเดตค่าประมาณคุณประโยชน์ $TWU(i_j)$ (สามารถคำนวณได้จาก $TWU(i_j) = TWU(i_j) + tuList(t_p)$) และจัดเก็บข้อมูลทั้งหมดไว้ในโหนดลูกของ R ของ $HUII-tree$

เมื่อทำการอ่านข้อมูลครบทุกทรานแซกชัน จะทำการตรวจสอบค่าประมาณคุณประโยชน์ของแต่ละ i_j (ที่จัดเก็บไว้ในโหนดลูกของ R ของ $HUII-tree$) โดยดูว่าถ้าค่าประมาณคุณประโยชน์ $TWU(i_j)$ มีค่าน้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด จะทำการลบรายการ i_j ออกจากการพิจารณา (หมายเหตุ รายการ i_j ที่มีค่าประมาณคุณประโยชน์น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด จะทำให้ซูปเปอร์เซตของ i_j มีค่าคุณประโยชน์น้อยกว่าค่าขีดแบ่งคุณประโยชน์ด้วยเช่นกัน เราสามารถตัดการพิจารณารายการ i_j และ เซตรายการที่เป็นซูปเปอร์เซตของ i_j ออกจากการพิจารณาได้) แต่อย่างไรก็ตาม ก่อนที่จะลบรายการ i_j ออกจากการพิจารณาจะทำการลดทอนค่าคุณประโยชน์ของแต่ละทรานแซกชันที่มีรายการ i_j ปรากฏ เพื่อทำการลดทอนค่าประมาณคุณประโยชน์ โดยทำการพิจารณาแต่ละสมาชิกใน NUL^j ที่ซึ่งมีลักษณะเป็น $\langle p, u(i_j, t_p), 0, 0 \rangle$ และทำการลดทอนค่า $tu(t_p)$ ที่ถูกจัดเก็บใน $tuList(t_p)$ ด้วยค่า $u(i_j, t_p)$ และเมื่อทำการพิจารณาทุกสมาชิกใน NUL^j จนครบแล้ว จะทำการลบข้อมูลทั้งหมดของ i_j ที่ถูกจัดเก็บอยู่ในโหนดลูกของ R ของ $HUII-tree$ ออกจากหน่วยความจำและการพิจารณาได้ (บรรทัดที่ 11-15)

หลังจากลบรายการที่มีค่าประมาณคุณประโยชน์ต่ำ ขั้นตอนต่อไปจะเป็นการคำนวณค่าคุณประโยชน์ส่วนเหลือในแต่ละสมาชิกใน NUL^j รวมถึงทำการคำนวณหาค่าคุณประโยชน์ $u(i_j)$ และทำการคำนวณค่าคุณประโยชน์ส่วนเหลือ $ru(i_j)$ (บรรทัดที่ 16-25) การคำนวณหาค่าคุณประโยชน์ส่วนเหลือ จะเริ่มจากสร้าง $temptuList$ และกำหนดให้แต่ละ entries ใน $temptuList(t_p)$ มีค่าเท่ากับ $tuList(t_p)$ จากนั้นทำการพิจารณาทีละรายการ i_j (จากลำดับของรายการที่ทราบก่อนหน้าแล้ว) และทำการพิจารณาแต่ละสมาชิก $\langle p, u(i_j, t_p), 0, 0 \rangle$ ใน NUL^j แล้วทำการอัปเดตค่าคุณประโยชน์ของทรานแซกชัน t_p ที่ถูกจัดเก็บอยู่ใน $temptuList$ ด้วย $temptuList(t_p) = temptuList(t_p) - u(i_j, t_p)$ เพื่อที่จะทราบถึงค่าคุณประโยชน์ของทุกรายการที่อยู่ในลำดับถัดไปจากรายการ i_j ในทรานแซกชัน t_p ซึ่งค่า $temptuList(t_p)$ หลังจากอัปเดตจะหมายถึงค่าคุณประโยชน์

ส่วนเหลือ $ru(i_j, t_p)$ โดยเมื่อทราบถึงค่าดังกล่าวแล้ว จะทำการอัปเดตแต่ละสมาชิกใน NUL^j ที่พิจารณาให้มีค่าเป็น $\langle p, u(i_j, t_p), ru(i_j, t_p), 0 \rangle$ และทำการอัปเดตค่าคุณประโยชน์ $u(i_j)$ ด้วย $u(i_j, t_p)$ และค่าคุณประโยชน์ส่วนเหลือ $ru(i_j)$ ด้วย $ru(i_j, t_p)$ ตามลำดับ หลังจากพิจารณาทุกสมาชิกใน NUL^j จนครบ ถ้าค่าคุณประโยชน์ $u(i_j)$ มีค่าไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด และค่าความสม่ำเสมอ $r(i_j)$ มากกว่าค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด “HUIIM-DBscanning” จะทำการระบุรายการ i_j ว่าเป็นรูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอ และจัดเก็บรายการ i_j ไว้ในเซต $HUII$ (บรรทัดที่ 24-25)

Algorithm 2: HUIIM-Mining

Input: $HUII-tree, \sigma_u, \sigma_r$
Output: $HUII$

```

1: for each node in  $HUII-tree$  with item  $i_j$  do
2:   if  $(tou(i_j) \leftarrow u(i_j) + ru(i_j)) \geq \sigma_u$  then
3:     for each node in  $HUII-tree$  with item  $i_k$  (where  $i_j \prec i_k$ ) do
4:        $NUL^{i_j i_k} \leftarrow \text{intersect}(NUL^{i_j}, NUL^{i_k})$ 
5:       calculate  $r(i_j i_k), u(i_j i_k), ru(i_j i_k), TWU(i_j i_k)$  from  $NUL^{i_j i_k}$ 
6:       if  $TWU(i_j i_k) \geq \sigma_u$  then
7:         create a node of itemset  $i_j i_k$  with  $r(i_j i_k), u(i_j i_k),$ 
            $ru(i_j i_k), TWU(i_j i_k)$  and  $NUL^{i_j i_k}$  and then set the node of
            $i_j i_k$  to be a child of  $i_j$ 
8:       if  $u(i_j i_k) \geq \sigma_u$  and  $r(i_j i_k) > \sigma_r$  then
9:          $HUII \leftarrow HUII \cup i_j i_k$ 
10: for each node in  $HUII-tree$  with item  $i_j$  do
11:   if item  $i_j$  has more than one child then
12:     MiningAllHUII( $HUII-tree$ , node of  $i_j, \sigma_u, \sigma_r$ )
13: Procedure MiningAllHUII( $HUII-tree$ , node of  $X, \sigma_u, \sigma_r$ )
14: for child node of  $X$  with itemset  $Y = X \cdot i_u$  do
15:   if  $(tou(Y) \leftarrow u(Y) + ru(Y)) \geq \sigma_u$  then
16:     for child node of  $X$  with itemset  $Z = X \cdot i_v$  do
17:       if there is a path of itemset ' $i_u i_v$ ' in  $HUII-tree$  then
18:          $NUL^{YZ} \leftarrow \text{intersect}(NUL^Y, NUL^Z)$ 
19:         calculate  $r(YZ), u(YZ), ru(YZ), TWU(YZ)$  from  $NUL^{YZ}$ 
20:         if  $TWU(YZ) \geq \sigma_u$  then
21:           create a node of itemset  $YZ$  with  $r(YZ), u(YZ), ru(YZ), TWU(YZ)$ 
           and  $NUL^{YZ}$  and then set the node of  $YZ$  to be a child of  $Y$ 
22:         if  $u(YZ) \geq \sigma_u$  and  $r(YZ) > \sigma_r$  then
23:            $HUII \leftarrow HUII \cup YZ$ 

```

ภาพที่ 3-2 ขั้นตอนการหารูปแบบทั้งหมดที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอ

หลังจากขั้นตอนการอ่านฐานข้อมูลเสร็จสิ้น เราจะได้ $HUII-tree$ ที่บรรจุไปด้วยแต่ละรายการที่มีค่าประมาณคุณประโยชน์สูง ขั้นตอนต่อไปจะเป็นการค้นหาเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอทั้งหมดจาก $HUII-tree$ ที่สร้างขึ้น (ดังแสดงรายละเอียดในภาพที่ 3-2) โดยในขั้นตอนแรกของ “HUIIM-Mining” จะเป็นการค้นหาเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอที่ประกอบไปด้วย 2 รายการ (บรรทัดที่ 1-9) โดยเริ่มต้นจากการพิจารณาเฉพาะรายการ i_j (ที่เป็นโหนดลูกของ R ของ $HUII-tree$) และทำการตรวจสอบค่า

คุณประโยชน์แบบกระชับ $tou(i_j) = u(i_j) + ru(i_j)$ ของรายการ i_j ถ้า $tou(i_j)$ มีค่ามากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด จะทำการรวมรายการ i_j เข้ากับรายการ i_k ที่อยู่ในลำดับถัดไปจากรายการ i_j เพื่อสร้างการพิจารณาเซตรายการ ' $i_j i_k$ ' จากนั้นทำการอินเทอร์เซกชัน NUL^{i_j} กับ NUL^{i_k} เข้าด้วยกัน เพื่อทำการจัดเก็บข้อมูลการปรากฏขึ้น ค่าคุณประโยชน์ และค่าคุณประโยชน์ส่วนเหลือสำหรับทรานแซกชันหนึ่งๆ ที่เซตรายการ ' $i_j i_k$ ' ปรากฏ หลังจากขั้นตอนอินเทอร์เซกชันเสร็จสิ้น เราจะได้ $NUL^{i_j i_k}$ ของเซตรายการ ' $i_j i_k$ ' ที่จะสามารถใช้ NUL ดังกล่าวในการคำนวณค่าคุณประโยชน์ $u(i_j i_k)$ ค่าคุณประโยชน์ส่วนเหลือ $ru(i_j i_k)$ ค่าประมาณคุณประโยชน์ $TWU(i_j i_k)$ และค่าความสม่ำเสมอ $r(i_j i_k)$

หลังจากทำการคำนวณค่าคุณประโยชน์ ค่าคุณประโยชน์ส่วนเหลือ ค่าประมาณคุณประโยชน์ ค่าความสม่ำเสมอ ของเซตรายการ ' $i_j i_k$ ' แล้ว จะทำการตรวจสอบค่าประมาณคุณประโยชน์ $TWU(i_j i_k)$ ถ้า $TWU(i_j i_k)$ มีค่ามากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด จะทำการสร้างเซตรายการ ' $i_j i_k$ ' พร้อมทั้งข้อมูลทั้งหมดที่เกี่ยวข้องกับเซตรายการ ' $i_j i_k$ ' ไว้ในโหนดลูกของรายการ i_j และทำการตรวจสอบค่าคุณประโยชน์ $u(i_j i_k)$ ถ้ามีค่าไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด และค่าความสม่ำเสมอ $r(i_j i_k)$ มากกว่าค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด ขั้นตอนวิธี "HULL-Mining" จะทำการระบุว่าเซตรายการ ' $i_j i_k$ ' ว่าเป็นรูปแบบที่มีค่าคุณประโยชน์สูงโดยปรากฏอย่างไม่สม่ำเสมอและทำการจัดเก็บเซตรายการ ' $i_j i_k$ ' ในเซต HULL โดยหลังจากทำการรวมรายการ i_j เข้ากับรายการ i_k แล้ว ขั้นตอนต่อไปทำการรวมรายการ i_j เข้ากับรายการ i_l ที่อยู่ในลำดับถัดไปจากรายการ i_k และจะดำเนินการตามกระบวนการต่างๆข้างต้น เมื่อทำการรวมรายการ i_j เข้ากับทุกโหนดลูกของ R ของ HULL-tree ทั้งหมด เราจะได้โหนดลูกของรายการ i_j ที่บรรจุไปด้วยเซตรายการที่ประกอบไปด้วย 2 รายการ และเป็นเซตรายการ i_j เป็นรายการขึ้นต้น และทำการวนซ้ำการทำงานตามลำดับของรายการที่เป็นสมาชิกของเซต I จนครบทั้งหมด เราจะได้ HULL-tree ที่บรรจุไปด้วยเซตรายการ 2 รายการ ที่มีค่าประมาณคุณประโยชน์สูง

หลังจากสร้าง HULL-tree ที่บรรจุไปด้วยเซตรายการ 2 รายการทั้งหมด ขั้นตอนต่อไปจะเป็นการค้นหาเซตรายการที่มากกว่า 2 รายการ "MiningALLHULL" (บรรทัดที่ 10-23) โดยเริ่มแรกทำการพิจารณารายการตามลำดับ i_j ที่มีจำนวนโหนดลูกมากกว่า 1 จากนั้นทำการพิจารณาแบบเรียงตามลำดับโหนดลูกของรายการ i_j คือ $\{ 'i_j i_k', 'i_j i_l', \dots, 'i_j i_m' \}$ และทำการตรวจสอบค่าประมาณคุณประโยชน์แบบกระชับ $tou(i_j i_k)$ ถ้ามากกว่าค่าขีดแบ่งที่ผู้ใช้กำหนด จะทำการรวมเซตรายการ ' $i_j i_k$ ' เข้ากับเซตรายการ ' $i_j i_l$ ' ที่อยู่ลำดับถัดไปจากรายการ ' $i_j i_k$ ' เพื่อสร้างการพิจารณาเซตรายการ ' $i_j i_k i_l$ ' แต่ก่อนขั้นตอนรวมรายการ จะทำการตรวจสอบรายการสุดท้ายของแต่ละเซตรายการ ที่ซึ่งรายการสุดท้ายของเซตรายการ ' $i_j i_k$ ' คือ i_k และรายการสุดท้ายของเซตรายการ ' $i_j i_l$ ' คือ i_l เพื่อสร้างการพิจารณาโหนดของเซตรายการ ' $i_k i_l$ ' และทำการตรวจสอบโหนดของเซตรายการ ' $i_k i_l$ ' ใน

โครงสร้าง *HULL-tree* ถ้าไม่มีโหนดของเซตรายการ $'i_k i_l'$ ก็สามารถกล่าวได้ว่าเซตรายการ $'i_j i_k i_l'$ เป็นเซตรายการที่มีค่าคุณประโยชน์ต่ำ และสามารถตัดเซตรายการ $'i_j i_k i_l'$ ออกจากการพิจารณาได้ ในทางตรงข้ามถ้ามีโหนดของเซตรายการ $'i_k i_l'$ ใน *HULL-tree* จะทำการการอินเทอร์เซกชัน NUL^{ijk} กับ NUL^{jli} เพื่อทำการจัดเก็บข้อมูลการปรากฏขึ้น ค่าคุณประโยชน์ ค่าคุณประโยชน์ส่วนเหลือ สำหรับทรานแซกชันหนึ่งที่เซตรายการ $'i_j i_k i_l'$ ปรากฏ หลังจากขั้นตอนอินเทอร์เซกชันเสร็จสิ้น เราจะได้ NUL^{ijkl} ที่จะสามารถใช้ NUL^{ijkl} ดังกล่าวในการคำนวณค่าคุณประโยชน์ $u(i_j i_k i_l)$ ค่าคุณประโยชน์ส่วนเหลือ $ru(i_j i_k i_l)$ ค่าประมาณคุณประโยชน์ $TWU(i_j i_k i_l)$ และค่าความสม่ำเสมอ $r(i_j i_k i_l)$ หลังจากคำนวณค่าคุณประโยชน์ ค่าคุณประโยชน์ส่วนเหลือ ค่าประมาณคุณประโยชน์ ค่าความสม่ำเสมอ ของเซตรายการ $'i_j i_k i_l'$ แล้วจะทำการตรวจสอบค่าประมาณคุณประโยชน์ $TWU(i_j i_k i_l)$ ถ้ามีค่ามากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด “MiningALLHULL” จะทำการสร้างเซตรายการ $'i_j i_k i_l'$ พร้อมทั้งข้อมูลทั้งหมดที่เกี่ยวข้องกับเซตรายการ $'i_j i_k i_l'$ ไว้ในโหนดลูกของรายการ $'i_j i_k'$ และทำการตรวจสอบค่าคุณประโยชน์ $u(i_j i_k i_l)$ ถ้ามีค่าไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด และค่าความสม่ำเสมอ $r(i_j i_k i_l)$ มีค่ามากกว่าค่าขีดแบ่งความสม่ำเสมอ ขั้นตอนวิธี “MiningALLHULL” จะทำการระบุเซตรายการ $'i_j i_k i_l'$ เป็นรูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอ และจัดเก็บเซตรายการดังกล่าวในเซต *HULL*

เมื่อทำการรวมเซตรายการ $'i_j i_k'$ เข้ากับ $'i_j i_l'$ เสร็จสิ้น ขั้นตอนวิธี “MiningALLHULL” จะดำเนินการรวมเซตรายการ $'i_j i_k'$ เข้ากับ $'i_j i_m'$ ที่ถูกรวบรวมไว้ในโหนดลูกของรายการ i_j ในลำดับหลังของเซตรายการ $'i_j i_l'$ ด้วยวิธีการข้างต้น เมื่อทำการรวมรายการเข้ากับทุกรายการจนครบ เราจะได้โหนดลูกของรายการ $'i_j i_k'$ ที่บรรจุไปด้วยเซตรายการที่ประกอบไปด้วย 3 รายการ และเป็นเซตรายการ $'i_j i_k'$ ขึ้นต้น จากนั้นจะทำการตรวจสอบจำนวนเซตรายการที่บรรจุอยู่ในโหนดลูกของรายการ $'i_j i_k'$ ซึ่งถ้ามีจำนวนเซตรายการมากกว่า 1 จะทำการวนซ้ำการทำงานเพื่อหาเซตรายการที่ประกอบไปด้วย 4 รายการ โดยการดำเนินการจะดำเนินการเช่นเดียวกับการพิจารณาเซตรายการที่มากกว่า 2 รายการ (ส่วนของ MiningALLHULL)

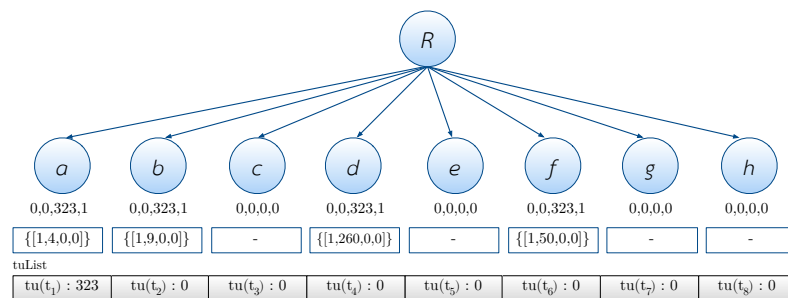
3.3 ตัวอย่างขั้นตอนวิธี HUIIM

กำหนดให้ 1) ฐานข้อมูลรายการประกอบไปด้วย 10 ทรานแซกชันดังแสดงในตารางที่ 3-1, 2) ค่าคุณประโยชน์ของรายการหนึ่ง ๆ ดังแสดงในตารางที่ 3-2, 3) ลำดับของเซตรายการที่อยู่ภายใต้ขอบเขตการพิจารณา คือ $a < b < c < d < e < f < g < h$, 4) ค่าขีดแบ่งคุณประโยชน์ $\sigma_u =$

40 และ 5) ค่าขีดแบ่งความสม่ำเสมอ $\sigma_r = 3$ ตามลำดับ การค้นหาเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏไม่สม่ำเสมอด้วยขั้นตอนวิธี “HUIIM” สามารถแสดงได้ดังนี้

ขั้นตอน “HUIIM-Scanning” (ดังแสดงรายละเอียดในภาพที่ 3-1) จะเริ่มจากการสร้างหรือจองพื้นที่หน่วยความจำในการจัดเก็บข้อมูลเบื้องต้น ที่ซึ่งจะทำการสร้างลิสต์ $tuList$ โดยมีขนาดเท่ากับ 8 (ขนาดจะเท่ากับจำนวนของทรานแซกชันในฐานข้อมูล) นอกจากนี้จะทำการสร้างโครงสร้างต้นไม้ $HUII-tree$ เพื่อจัดเก็บรายการ โดยแต่ละรายการจะถูกจัดเก็บไว้ในโหนดลูกของ R โดยแต่ละโหนดบรรจุไปด้วย 5 ข้อมูล ดังนี้ $u(X)$, $ru(X)$, $TWU(X)$, $r(X)$ และ NUL^x

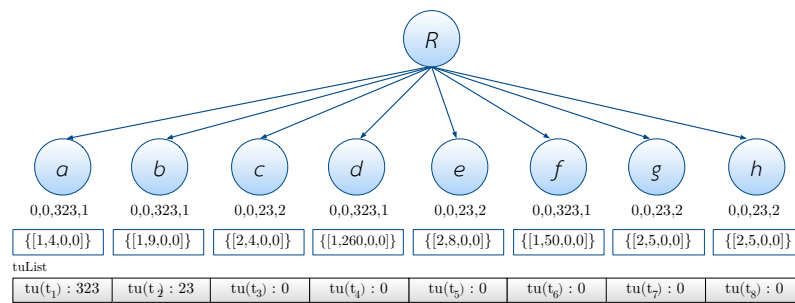
เมื่อทำการสร้างหรือจองพื้นที่หน่วยความจำเสร็จ จะทำการอ่านแต่ละทรานแซกชันในฐานข้อมูลเพื่อคำนวณคุณประโยชน์ โดยเริ่มต้นทรานแซกชัน $t_1 = \{a(2), b(3), d(13), f(2)\}$ เพื่อทำการคำนวณค่าคุณประโยชน์ $tu(t_1) = (iu(a,t_1) \times eu(a)) + (iu(b,t_1) \times eu(b)) + (iu(d,t_1) \times eu(d)) + (iu(f,t_1) \times eu(f)) = (2 \times 2) + (3 \times 3) + (13 \times 20) + (2 \times 25) = 323$ และจัดเก็บ $tu(t_1)$ ใน $tuList(t_1)$ จากนั้นทำการพิจารณาแต่ละรายการที่ปรากฏขึ้นในทรานแซกชัน t_1 และทำการอัปเดต NUL ของแต่ละรายการ ‘a’, ‘b’, ‘d’ และ ‘f’ เริ่มด้วยรายการ ‘a’ ทำการอัปเดต NUL^a ซึ่งจากเดิมจะเป็นเซตว่าง จะถูกอัปเดตเป็น $NUL^a = \{<1, 4, 0, 0>\}$ ซึ่งบ่งบอกได้ว่ารายการ ‘a’ ปรากฏขึ้นในทรานแซกชัน t_1 ที่ซึ่งมีค่าคุณประโยชน์เท่ากับ $u(a, t_1) = iu(a, t_1) \times eu(a) = 2 \times 2 = 4$ (หมายเหตุ ข้อมูลใน tuple ที่อัปเดตใน NUL^a จะยังไม่ได้ทำการคำนวณค่าคุณประโยชน์ที่เหลือของรายการ ‘a’ และรายการ ‘a’ เป็นรายการเดี่ยวที่ซึ่งไม่มี prefix หรือ parent items จึงทำให้ค่าใน tuple ที่ 3 และ 4 มีค่าเป็น 0) และกำหนดค่าความสม่ำเสมอ $r(a) = \max(fr_{t_1}^a) = 1$ สุดท้ายทำการอัปเดตค่าประมาณคุณประโยชน์ $TWU(a) = tu(t_1) = 323$ ดังแสดงในภาพที่ 3-3



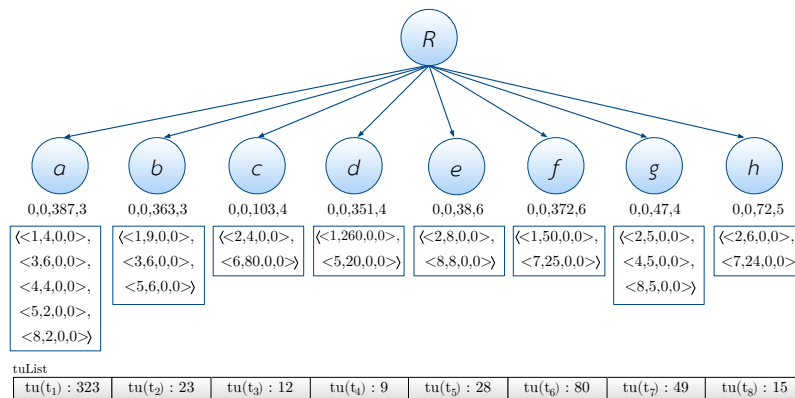
ภาพที่ 3-3 อ่านข้อมูลทรานแซกชันที่ 1 ในฐานข้อมูล

ต่อไปจะทำการอ่านทรานแซกชัน $t_2 = \{c(1), e(4), g(1), h(2)\}$ เพื่อทำการคำนวณค่าคุณประโยชน์ $tu(t_2) = (iu(c,t_2) \times eu(c)) + (iu(e,t_2) \times eu(e)) + (iu(g,t_2) \times eu(g)) + (iu(h,t_2) \times eu(h)) = (1 \times 4) + (2 \times 4) + (1 \times 5) + (3 \times 2) = 23$ และจัดเก็บ $tu(t_2)$ ใน $tuList(t_2)$ จากนั้นทำ

การพิจารณาแต่ละรายการที่ปรากฏขึ้นในทรานแซกชัน t_2 และทำการอัปเดต NUL ของแต่ละรายการ 'c', 'e', 'g' และ 'h' เริ่มด้วยรายการ 'c' ทำการอัปเดต NUL^c ซึ่งจากเดิมจะเป็นเซตว่าง จะถูกอัปเดตเป็น $NUL^c = \{<2, 4, 0, 0>\}$ ที่ซึ่งบ่งบอกได้ว่ารายการ 'c' ปรากฏขึ้นในทรานแซกชัน t_2 ที่มีค่าคุณประโยชน์ $u(c, t_2) = iu(c, t_2) \times eu(c) = 1 \times 4 = 4$ และทำการคำนวณค่าความสม่ำเสมอ $r(c) = \max(fr_t^c) = 2$ สุดท้ายทำการอัปเดตค่าประมาณคุณประโยชน์ $TWU(c) = tu(t_2) = 23$ จากนั้นจะทำการอัปเดตค่าต่างๆข้างต้นของรายการ 'e', 'g' และ 'h' ดังแสดงในภาพที่ 3-4 จากนั้นทำการอ่านทรานแซกชัน $t_3 - t_8$ แบบวิธีการข้างต้นตามลำดับเพื่อที่จะอัปเดต $tuList$ และ $HULL$ -tree ดังแสดงในภาพที่ 3-5



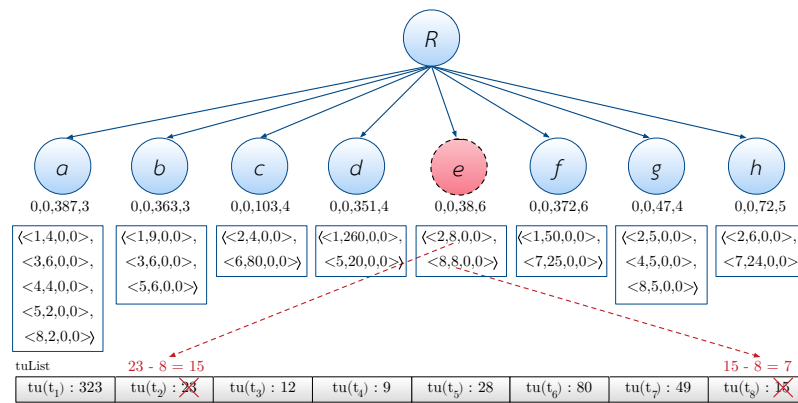
ภาพที่ 3-4 อ่านข้อมูลทรานแซกชันที่ 2 ในฐานข้อมูล



ภาพที่ 3-5 อ่านข้อมูลครบทุกทรานแซกชันในฐานข้อมูล

ขั้นตอนต่อไปทำการตรวจสอบค่าประมาณคุณประโยชน์ของแต่ละรายการ ที่ซึ่งทำการพิจารณารายการแต่ละรายการตามลำดับ $a < b < c < d < e < f < g < h$ เริ่มต้นพิจารณารายการ 'a' แล้วทำการพิจารณา $TWU(a) = 387$ แต่ด้วยเนื่องจาก $TWU(a)$ มากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด จึงทำการเลื่อนการพิจารณาไปยังรายการ 'b' แล้วทำการพิจารณา $TWU(b) = 363$ แต่ด้วยเนื่องจาก $TWU(b)$ มากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด จึงทำการเลื่อนการพิจารณาไปยังรายการ 'c' แล้วทำการพิจารณา $TWU(c) = 103$ แต่ด้วยเนื่องจาก $TWU(c)$

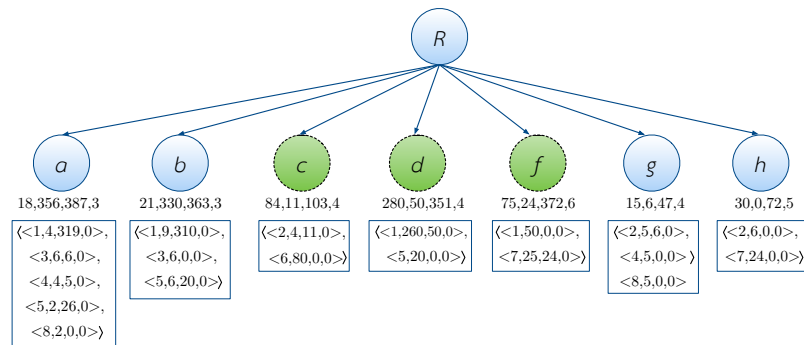
มากกว่าค่าขีดแบ่งคุณสมบัติที่ผู้ใช้กำหนด จึงทำการเลื่อนการพิจารณาไปยังรายการ 'd' แล้วทำการพิจารณา $TWU(d) = 351$ แต่ด้วยเนื่องจาก $TWU(d)$ มากกว่าค่าขีดแบ่งคุณสมบัติที่ผู้ใช้กำหนด จึงทำการเลื่อนการพิจารณาไปยังรายการ 'e' แล้วทำการพิจารณา $TWU(e) = 38$ ที่ซึ่งมีค่าน้อยกว่าค่าขีดแบ่งคุณสมบัติที่ผู้ใช้กำหนด จะทำการลบรายการ 'e' ออกจากการพิจารณา แต่ก่อนที่จะลบรายการ 'e' จะทำการลดทอนค่าคุณสมบัติในทรานแซกชันที่มี 'e' ปรากฏ โดยทำการพิจารณาแต่ละสมาชิกใน $NUL^e = \{<2, 8, 0, 0>, <8, 8, 0, 0>\}$ ซึ่งจะบอกได้ว่ารายการ 'e' ปรากฏในทรานแซกชันที่ t_2 และ t_8 และทำการลดทอนค่า $tuList(t_2)$ ด้วยค่า $u(e, t_2)$ กล่าวคือ $tuList(t_2) = tuList(t_2) - u(e, t_2) = 23 - 8 = 15$ และ $tuList(t_8)$ ด้วยค่า $u(e, t_8)$ กล่าวคือ $tuList(t_8) = tuList(t_8) - u(e, t_8) = 13 - 8 = 7$ แล้วจึงทำการลบข้อมูลของรายการ 'e' ที่ถูกจัดเก็บอยู่ในโหนดลูกของ R ของ HUII-tree ออกจากหน่วยความจำและการพิจารณาได้ (ดังแสดงในภาพที่ 3-6) จากนั้นทำการพิจารณารายการถัดไปตามลำดับด้วยวิธีการแบบเดียวกันจนครบทุกรายการ



ภาพที่ 3-6 ตัดการพิจารณารายการ 'e' ที่มี TWU ไม่ผ่านค่าขีดแบ่งคุณสมบัติ

หลังจากทำการพิจารณารายการที่มีค่าประมาณคุณสมบัติต่ำ ขั้นตอนต่อไปจะเป็นการคำนวณค่าคุณสมบัติส่วนเหลือในแต่ละสมาชิกใน NUL รวมถึงทำการคำนวณค่าคุณสมบัติส่วนเหลือ และค่าคุณสมบัติที่แท้จริงของแต่ละรายการ โดยเริ่มต้นจากการสร้าง temptuList ให้มีขนาดเท่ากับจำนวนทรานแซกชันในฐานข้อมูล และกำหนดให้แต่ละสมาชิกใน temptuList(j) มีค่าเท่ากับ tuList(j) จากนั้นทำการพิจารณาแต่ละรายการที่เป็นโหนดลูกของโหนด R ใน HUII-tree โดยทำการพิจารณารายการตามลำดับ $a < b < c < d < f < g < h$ โดยในการพิจารณาจะเริ่มพิจารณารายการ 'a' เป็นรายการแรก จากนั้นทำการพิจารณาแต่ละสมาชิกใน $NUL^a = \{<1, 4, 0, 0>, <3, 6, 0, 0>, <4, 4, 0, 0>, <5, 2, 0, 0>, <8, 2, 0, 0>\}$ โดยสมาชิกอันดับแรกของ NUL^a คือ $<1, 4, 0, 0>$ ทำการอัปเดตค่าคุณสมบัติทรานแซกชัน $temptuList(t_1) = temptuList(t_1) - u(a, t_1) = 323 - 4 = 319$ เพื่อที่จะทราบถึงค่าคุณสมบัติของทุกรายการที่อยู่ในลำดับถัดไปจาก

รายการ 'a' ในทรานแซกชัน t_1 ซึ่งค่า $temptuList(t_1)$ หลังจากการอัปเดตหมายถึงค่าคุณประโยชน์ส่วนเหลือของรายการ 'a' ในทรานแซกชัน t_1 ซึ่งเท่ากับ $ru(a, t_1) = 319$ โดยเมื่อทราบถึงค่าดังกล่าวจะอัปเดตข้อมูลใน tuple-3 ใน $\langle 1, 4, 319, 0 \rangle$ และอัปเดตค่าคุณประโยชน์ $u(a) = u(a) + u(a, t_1) = 0 + 4 = 4$ ค่าคุณประโยชน์ส่วนเหลือ $ru(a) = ru(a) + ru(a, t_1) = 0 + 319 = 319$ จากนั้นทำการพิจารณาสมาชิกลำดับถัดไปใน NUL^0 คือ $\langle 3, 6, 0, 0 \rangle$ ทำการอัปเดตค่าคุณประโยชน์ทรานแซกชัน $temptuList(t_3) = temptuList(t_3) - u(a, t_3) = 12 - 6 = 6$ หลังจากการอัปเดตคือค่าคุณประโยชน์ส่วนเหลือของรายการ 'a' ในทรานแซกชัน t_3 ซึ่งเท่ากับ $ru(a, t_3) = 6$ โดยเมื่อทราบถึงค่าดังกล่าวจะอัปเดตข้อมูลใน tuple-3 ใน $\langle 3, 6, 6, 0 \rangle$ และอัปเดตค่าคุณประโยชน์ $u(a) = u(a) + u(a, t_3) = 4 + 6 = 10$ ค่าคุณประโยชน์ส่วนเหลือ $ru(a) = ru(a) + ru(a, t_3) = 319 + 6 = 325$ ตามลำดับ และทำการอัปเดตทุกๆสมาชิกใน NUL^0 จนครบ หลังจากการอัปเดตเสร็จเราจะได้ $NUL^0 = \{ \langle 1, 4, 319, 0 \rangle, \langle 3, 6, 6, 0 \rangle, \langle 4, 4, 5, 0 \rangle, \langle 5, 2, 26, 0 \rangle, \langle 8, 2, 5, 0 \rangle \}$ และค่าคุณประโยชน์ $u(a) = 18$ ค่าคุณประโยชน์ส่วนเหลือ $ru(a) = 361$ ท้ายสุดทำการตรวจสอบค่าคุณประโยชน์ $u(a) = 18$ ที่ซึ่งมีค่าน้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด และค่าความสม่ำเสมอ $r(a) = 3$ มีค่าน้อยกว่าค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด ดังนั้นรายการ 'a' จึงไม่เป็นรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอ หลังจากพิจารณารายการ 'a' เสร็จแล้ว จะทำการพิจารณารายการ 'b', 'c', 'd', 'f', 'g' และ 'h' ด้วยวิธีการข้างต้น (ดังที่แสดงในภาพที่ 3-7 รายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอจะมีโหนดเป็นสีเขียว)



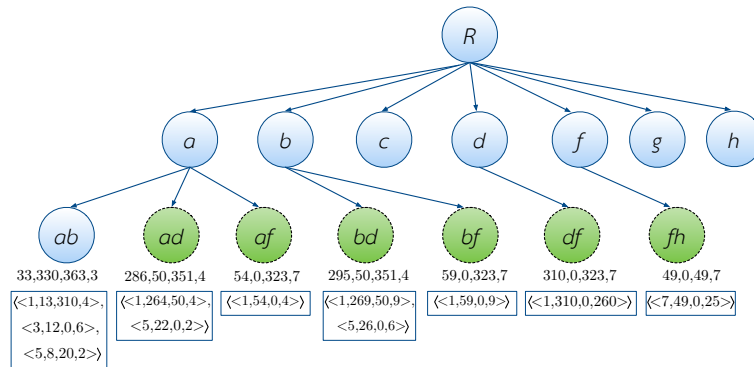
ภาพที่ 3-7 หลังจากอ่านข้อมูลทุกทรานแซกชันในฐานะข้อมูล

หลังจากขั้นตอนการอ่านฐานข้อมูลเสร็จเราจะได้ $HUII$ -tree ที่บรรจุไปด้วย 1 รายการ ขั้นตอนต่อไปจะเป็นการค้นหาเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอทั้งหมดจาก $HUII$ -tree ด้วยวิธี "HUIIM-Mining" โดยในขั้นแรก จะทำการค้นหาเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอที่ประกอบไปด้วย 2 รายการ เริ่มจากพิจารณารายการ 'a' ที่ซึ่งจะทำการรวมรายการ 'a' เข้ากับรายการ 'b', 'c', 'd', 'f', 'g' และ 'h' แต่อย่างไรก็ตามก่อนที่จะทำการรวมรายการ 'a' เข้ากับรายการอื่นๆ เราจะต้องทำการตรวจสอบค่าประมาณ

คุณประโยชน์แบบกระชับ $tou(a)$ ว่ามีค่าไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้กำหนดหรือไม่ ถ้า $tou(a)$ มีค่าไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้กำหนด จะทำการรวมรายการ 'a' เข้ากับรายการอื่น โดยเริ่มจากรวมรายการ 'a' เข้ากับรายการ 'b' เพื่อสร้างการพิจารณาเซตรายการ 'ab' จากนั้นทำการอินเทอร์เซกชัน $NUL^a = \{<1, 4, 319, 0>, <3, 6, 6, 0>, <4, 4, 5, 0>, <5, 2, 26, 0>, <8, 2, 5, 0>\}$ และ $NUL^b = \{<1, 9, 310, 0>, <3, 6, 0, 0>, <5, 6, 20, 0>\}$ เข้าด้วยกัน (หมายเหตุ รายการ 'a' และ 'b' เกิดร่วมกันในทรานแซกชันที่ t_1, t_3 และ t_5) เพื่อทำการจัดเก็บข้อมูลการปรากฏขึ้น ค่าคุณประโยชน์ $u(ab) = (u(a, t_1) + u(b, t_1)) + (u(a, t_3) + u(b, t_3)) + (u(a, t_5) + u(b, t_5)) = (4 + 9) + (6 + 6) + (2 + 6) = 33$ ค่าคุณประโยชน์ส่วนเหลือ $ru(ab) = ru(b, t_1) + ru(b, t_3) + ru(b, t_5) = 310 + 0 + 20 = 330$ ค่าประมาณคุณประโยชน์ $TWU(ab) = tuList(t_1) + tuList(t_3) + tuList(t_5) = 323 + 12 + 28 = 363$ ค่าความสม่ำเสมอ $r(ab) = \max(fr_{t_1}^{ab}, r_{t_1 t_3}^{ab}, r_{t_3 t_5}^{ab}, r_{t_5}^{ab}) = \max(1, 2, 2, 3) = 3$ ตามลำดับ โดยหลังจากการขั้นตอนอินเทอร์เซกชันเสร็จสิ้น เราจะได้ $NUL^{ab} = \{<1, 13, 310, 4>, <3, 12, 0, 6>, <5, 8, 20, 2>\}$ (หมายเหตุ ข้อมูลในอันดับที่ 4 ในแต่ละสมาชิกของ NUL^{ab} จะจัดเก็บค่าคุณประโยชน์ของรายการ 'a' ที่ซึ่งเป็น prefix หรือ parent item ของเซตรายการ 'ab' เพื่อทำการคำนวณในอนาคต) จากนั้นทำการตรวจสอบค่าประมาณคุณประโยชน์ $TWU(ab) = 363$ ว่ามีค่ามากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้กำหนดหรือไม่ ถ้าค่าประมาณคุณประโยชน์ $TWU(ab)$ มากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้กำหนด จะทำการสร้างเซตรายการ 'ab' และจัดเก็บ ค่าคุณประโยชน์ ค่าคุณประโยชน์ส่วนเหลือ ค่าประมาณคุณประโยชน์ ค่าความสม่ำเสมอ และ NUL^{ab} ไว้ในโหนดลูกของรายการ 'a' จากนั้นทำการตรวจสอบค่าคุณประโยชน์ $u(ab) = 33$ ว่ามีค่าไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้กำหนดหรือไม่ และค่าความสม่ำเสมอ $r(ab) = 3$ ว่ามีค่ามากกว่าค่าขีดแบ่งความสม่ำเสมอที่ผู้กำหนดหรือไม่ ถ้าผ่าน 2 เงื่อนไขข้างต้น $HUIIM$ จะทำการระบุและจัดเก็บรายการ 'ab' ว่าเป็นเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอซึ่งจะถูกจัดเก็บไว้ในเซต $HUII$ (หมายเหตุ รายการ 'ab' มีค่าคุณประโยชน์ $u(ab)$ น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้กำหนด และค่าความสม่ำเสมอ $r(ab)$ น้อยกว่าค่าขีดแบ่งความสม่ำเสมอที่ผู้กำหนด จึงไม่เป็นเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอ) ต่อไปทำการรวมรายการ 'a' เข้ากับรายการ 'c' เพื่อสร้างการพิจารณาเซตรายการ 'ac' และทำการอินเทอร์เซกชัน $NUL^a = \{<1, 4, 319, 0>, <3, 6, 6, 0>, <4, 4, 5, 0>, <5, 2, 26, 0>, <8, 2, 5, 0>\}$ กับ $NUL^c = \{<2, 4, 11, 0>, <6, 80, 0, 0>\}$ แต่ในฐานข้อมูลรายการ 'a' และ 'c' ไม่เกิดขึ้นร่วมกัน ดังนั้น $TWU(ac) = 0$ จากนั้นจะทำการรวมรายการ 'a' เข้ากับรายการ 'd', 'f', 'g' และ 'h' ตามลำดับ ด้วยวิธีการข้างต้น

หลังจากทำการรวมรายการ 'a' เข้ากับทุกโหนดลูกของ R ของ $HUII$ -tree ทั้งหมด เราจะได้โหนดลูกของรายการ 'a' ที่บรรจุไปด้วยเซตรายการที่ประกอบไปด้วย 2 รายการ และเป็นเซต

รายการ 'a' เป็นรายการขึ้นต้น จากนั้นทำการวนซ้ำการทำงานตามลำดับของรายการที่จัดเก็บไว้ในโหนดลูกของ R จนครบทั้งหมดด้วยวิธีการข้างต้น เราจะได้ *HUII-tree* ที่บรรจุไปด้วย 2 รายการทั้งหมด (ดังแสดงในภาพที่ 3-8)

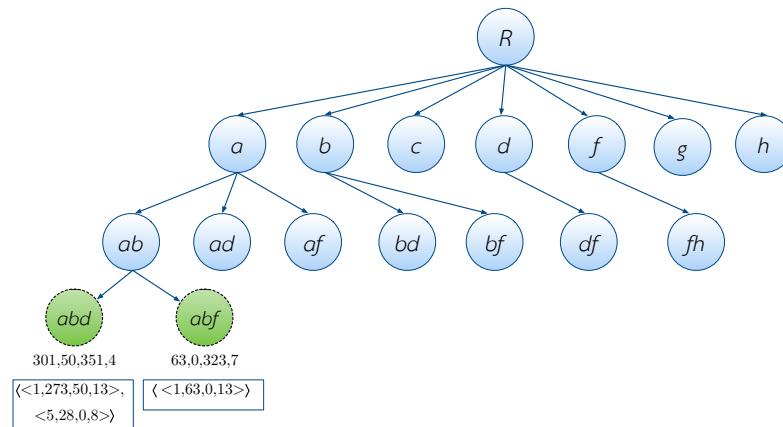


ภาพที่ 3-8 *HUII-tree* ที่บรรจุ 2-*HUIIs*

หลังจากได้ *HUII-tree* ที่บรรจุไปด้วย 2 รายการทั้งหมด จะดำเนินการในขั้นตอนสุดท้าย “MiningAll” ซึ่งก็คือการสร้างเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอที่ประกอบไปด้วย 3 รายการ และมากกว่า 3 รายการ โดยเริ่มจากการพิจารณารายการตามลำดับ 'a', 'b', 'c', 'd', 'f', 'g' และ 'h' ที่มีโหนดลูกมากกว่า 1 เริ่มจากพิจารณารายการ 'a' ซึ่งรายการ 'a' มีโหนดลูกทั้งหมด 3 รายการ คือ 'ab', 'ad' และ 'af' และพิจารณารายการ 'ab' ที่ซึ่งจะทำการรวมเซตรายการ 'ab' เข้ากับเซตรายการ 'ad' และ 'af' แต่อย่างไรก็ตามก่อนที่จะทำการรวมเซตรายการ 'ab' เข้ากับเซตรายการอื่นๆ เราจะต้องทำการตรวจสอบค่าประมาณคุณประโยชน์แบบกระชับ $tou(ab) = 360$ ว่ามีค่ามากกว่าค่าขีดแบ่งที่ผู้ใช้กำหนดหรือไม่ ถ้าค่าประมาณคุณประโยชน์แบบกระชับ $tou(ab)$ มากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด จะทำการรวมเซตรายการ 'ab' เข้ากับเซตรายการอื่นๆ โดยเริ่มจากรวมเซตรายการ 'ab' เข้ากับเซตรายการ 'ad' ที่อยู่ลำดับถัดไปจากรายการ 'ab' เพื่อสร้างการพิจารณาเซตรายการ 'abd' แต่ก่อนที่จะทำการอินเทอร์เซก NUL^{ab} และ NUL^{ad} เข้าด้วยกัน จะทำการตรวจสอบรายการสุดท้ายของเซตรายการ 'ab' และรายการสุดท้ายของเซตรายการ 'ad' ที่ซึ่งรายการสุดท้ายของเซตรายการ 'ab' คือรายการ 'b' และรายการสุดท้ายของเซตรายการ 'ad' คือรายการ 'd' เพื่อสร้างการพิจารณาเซตรายการ 'bd' จากนั้นทำการตรวจสอบโหนดของเซตรายการ 'bd' โดยโครงสร้างต้นไม้ *HUII-tree* จะทำให้เราสามารถตรวจสอบเส้นทางของโหนดเซตรายการ 'bd' (หมายเหตุ ถ้าไม่มีโหนดของเซตรายการ 'bd' ที่ซึ่งจะกล่าวได้ว่าเซตรายการ 'abd' เป็นเซตรายการที่มีค่าคุณประโยชน์ต่ำ) เนื่องจากมีโหนดของเซตรายการ 'bd' ในโครงสร้างต้นไม้ *HUII-tree* จึงทำการการอินเทอร์เซกชัน $NUL^{ab} = \{<1, 13, 310, 4>, <3, 12, 0, 6>, <5, 8, 20, 2>\}$ และ $NUL^{ad} = \{<1, 264, 50, 4>, <5, 22, 0, 2>\}$ เข้าด้วยกัน (หมายเหตุ เซตรายการ 'ab' และ 'ad' เกิดร่วมกันในทรานแซกชันที่ t_1 และ t_5) เพื่อทำการจัดเก็บข้อมูลการ

ปรากฏขึ้น ค่าคุณประโยชน์ $u(abd) = (u(ab, t_1) + u(ad, t_1) - up(ab, t_1)) + (u(ab, t_5) + u(ad, t_5) - up(ab, t_5)) = (13 + 264) - 4 + (8 + 22) - 2 = 301$ ค่าคุณประโยชน์ส่วนเหลือ $ru(abd) = ru(ad, t_1) + ru(ad, t_5) = 50 + 0 = 50$ ค่าประมาณคุณประโยชน์ $TWU(abd) = tuList(t_1) + tuList(t_5) = 321 + 28 = 351$ ค่าความสม่ำเสมอ $r(abd) = \max(fr_{t_1}^{abd}, r_{t_1 t_5}^{abd}, lr_{t_5}^{abd}) = \max(1, 4, 3) = 4$ โดยหลังจากการขั้นตอนอินเทอร์เซกชันเสร็จสิ้น เราจะได้ $NUL^{abd} = \{<1, 273, 50, 13>, <5, 28, 0, 8>\}$ จากนั้นทำตรวจสอบค่าประมาณคุณประโยชน์ $TWU(abd) = 351$ ว่ามีค่ามากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนดหรือไม่ เมื่อค่า $TWU(abd)$ มากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด จะทำการสร้างเซทรายการ 'abd' และจัดเก็บค่าคุณประโยชน์ ค่าคุณประโยชน์ส่วนที่เหลือ ค่าความสม่ำเสมอ ค่าประมาณคุณประโยชน์ และ NUL^{abd} ไว้ในโหนดลูกของรายการ 'ab' จากนั้นทำการตรวจสอบค่าคุณประโยชน์ $u(abd) = 301$ ว่ามีค่าไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนดหรือไม่ และค่าความสม่ำเสมอ $r(abd) = 4$ ว่ามีค่ามากกว่าค่าขีดแบ่งความสม่ำเสมอหรือไม่ ถ้าผ่าน 2 เงื่อนไขขั้นต้น $HUIIM$ จะทำการระบุและจัดเก็บรายการ 'abd' ว่าเป็นรูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอและจัดเก็บไว้ในเซต $HUII$ (หมายเหตุ รายการ 'abd' มีค่าคุณประโยชน์ $u(abd)$ มากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด และค่าความสม่ำเสมอ $r(abd)$ มากกว่าค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด จึงเป็นเซทรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอและจัดเก็บไว้ในเซต $HUII$) ต่อไปทำการรวมเซทรายการ 'ab' เข้ากับเซทรายการที่อยู่ในลำดับถัดไปจากเซทรายการ 'ad' ด้วยวิธีการข้างต้น (ดังแสดงในภาพที่ 3-9)

หลังจากทำการรวมรายการ 'ab' เข้ากับทุกโหนดลูกของรายการ 'a' ทั้งหมด เราจะได้โหนดลูกของรายการ 'ab' ที่บรรจุไปด้วยเซทรายการที่ประกอบไปด้วย 3 รายการ และเป็นเซทรายการ 'ab' เป็นรายการขั้นต้น จากนั้นจะทำการวนซ้ำการทำงานเพื่อหาเซทรายการที่ประกอบไปด้วย 4 รายการ โดยการดำเนินการจะดำเนินเช่นเดียวกับการพิจารณาเซทรายการที่มากกว่า 2 รายการ "MiningAll" เมื่อขั้นตอน "HUIIM-Mining" เสร็จสิ้น เราจะได้เซทรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอทั้งหมดที่จัดเก็บไว้ในเซต $HUII$ (ดังแสดงในภาพที่ 3-10)



ภาพที่ 3-9 HUII-tree ที่บรรจุ 3-HUI

itemsets	Utility & regularity
c	84, 4
d	280, 4
f	75, 6
ad	286, 4
af	54, 7
bd	295, 4
bf	59, 7
df	310, 7
fh	49, 7
abd	301, 4
abf	63, 7
adf	314, 7
bdf	319, 7
abdf	323, 7

ภาพที่ 3-10 เซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอจากขั้นตอน HUIIM

บทที่ 4

ผลการทดลอง

ในบทนี้จะนำเสนอการทดสอบประสิทธิภาพการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอจากขั้นตอนวิธี *HUIIM* เนื่องจากผลลัพธ์ที่ทำการพิจารณาในงานวิจัยนี้มีความแตกต่างกับผลลัพธ์ที่ทำการค้นหาจากงานวิจัยอื่นๆ การดำเนินการทดลองจึงทำการเปรียบเทียบกับวิธีการที่ใกล้เคียงคือขั้นตอนวิธี *MHUIR-NUL* ที่เป็นอัลกอริทึมสำหรับการค้นหารูปแบบที่มีค่าคุณประโยชน์และปรากฏอย่างสม่ำเสมอ

ในการทำการทดลองผู้วิจัยได้กำหนดค่าขีดแบ่งให้มีความใกล้เคียงกับงานวิจัยที่เกี่ยวข้อง กล่าวคือ ค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด (σ_r) จะกำหนดให้มีความอยู่ระหว่าง 1–30% ของจำนวนทรานแซกชันทั้งหมดในฐานข้อมูล (กล่าวคือ รูปแบบจะเป็นรูปแบบที่ปรากฏอย่างไม่สม่ำเสมอจะต้องมีค่าความสม่ำเสมอมากกว่า 1-30% ของจำนวนทรานแซกชันทั้งหมดในแต่ละฐานข้อมูล) และค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด (σ_u) จะกำหนดให้มีความอยู่ระหว่าง 0.001–45% ของค่าคุณประโยชน์ทั้งหมดของรูปแบบที่ปรากฏในแต่ละฐานข้อมูล (กล่าวคือ รูปแบบจะเป็นรูปแบบที่มีค่าคุณประโยชน์สูง จะต้องมีความคุณประโยชน์ไม่น้อยกว่า 0.001–45% ของค่าคุณประโยชน์ทั้งหมดของรูปแบบที่ปรากฏในฐานข้อมูล) ตามลำดับ โดยในการทดสอบประสิทธิภาพของขั้นตอนวิธี *HUIIM* ผู้วิจัยได้ทำการเขียนโปรแกรมการคำนวณตามขั้นตอนด้วยภาษา C และทำการทดสอบประสิทธิภาพในเครื่อง Mac mini Core i5 2.6GHz macOS Sierra หน่วยความจำ 8 GB ในการทดสอบประสิทธิภาพจะดำเนินการทดสอบกับชุดข้อมูลจริง 10 (ที่ซึ่งสามารถดาวน์โหลดได้จาก P. F. Viger, “SPMF : An Open-Source Data Mining Library⁴”) แต่ละชุดข้อมูลจะมีรายละเอียดดังแสดงในตารางที่ 4-1

⁴ <http://www.philippe-fournier-viger.com>

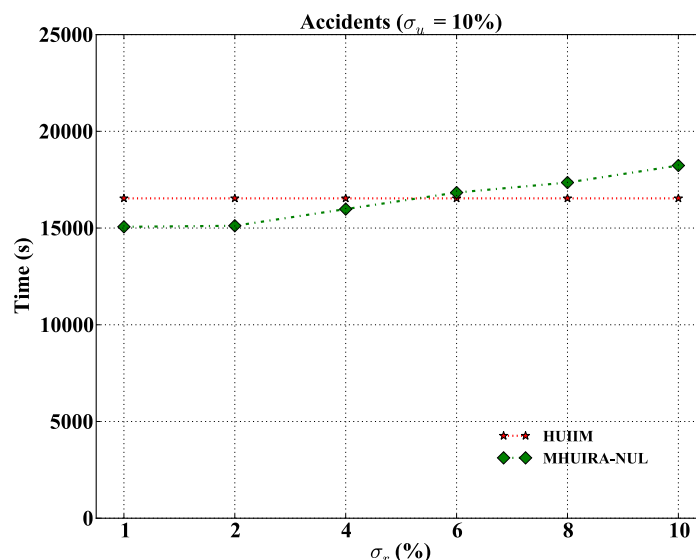
ตารางที่ 4-1 คุณลักษณะของชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพ

ชื่อฐานข้อมูล	จำนวนรายการที่ปรากฏ	จำนวนทรานแซกชัน	ความยาวเฉลี่ยของทรานแซกชัน	ชนิดของฐานข้อมูล
Accidents	468	340,183	33.8	หนาแน่น
BMS	497	59,601	4.8	เบาบาง
Chainstore	46,086	1,112,949	7.2	เบาบาง
Chess	75	3,196	37	หนาแน่น
Connect	129	67,557	43	หนาแน่น
Foodmart2000	1,559	36,869	11	เบาบาง
Kosarak	41,270	990,002	7.3	เบาบาง
Mushroom	119	8,124	23	หนาแน่น
Pumsb	2,113	49,046	74	หนาแน่น
Retail	16,469	88,162	10.3	เบาบาง

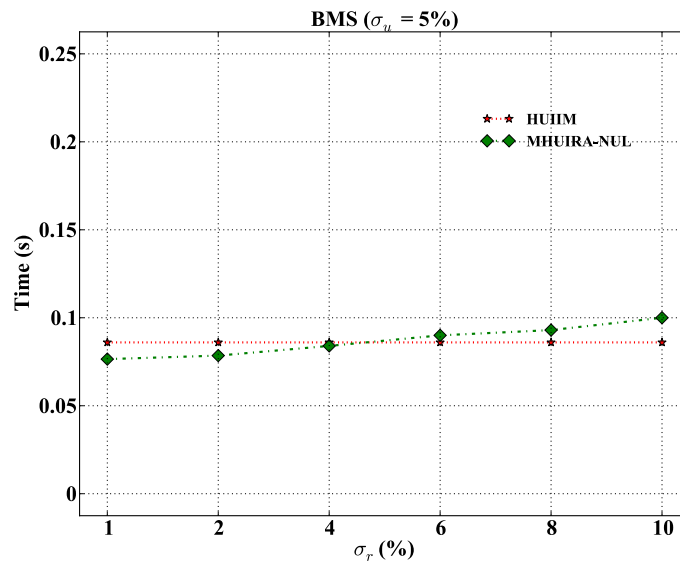
การทดลองที่ผู้วิจัยได้ทำการทดสอบประสิทธิภาพของขั้นตอนวิธี *HUIIM* เปรียบเทียบกับขั้นตอนวิธี *MHUIRA-NUL* สามารถแบ่งได้เป็น 6 กรณี คือ 1) การทดสอบเวลาในการคำนวณเมื่อทำการกำหนดค่าขีดแบ่งคุณประโยชน์แบบตายตัวและกำหนดค่าขีดแบ่งความสม่ำเสมอแบบแปรปรวน 2) การทดสอบเวลาในการคำนวณเมื่อทำการกำหนดค่าขีดแบ่งสม่ำเสมอแบบตายตัวและกำหนดค่าขีดแบ่งคุณประโยชน์แบบแปรปรวน 3) หน่วยความจำที่ใช้ในการประมวลผลเมื่อทำการกำหนดค่าขีดแบ่งคุณประโยชน์แบบตายตัวและกำหนดค่าขีดแบ่งความสม่ำเสมอแบบแปรปรวน 4) หน่วยความจำที่ใช้ในการประมวลผลเมื่อทำการกำหนดค่าความสม่ำเสมอแบบตายตัวและกำหนดค่าขีดแบ่งคุณประโยชน์แบบแปรปรวน 5) การพิจารณาจำนวนผลลัพธ์ที่สามารถค้นหาได้เมื่อทำการกำหนดค่าขีดแบ่งคุณประโยชน์แบบตายตัวและกำหนดค่าขีดแบ่งความสม่ำเสมอแบบแปรปรวน 6) การพิจารณาจำนวนผลลัพธ์ที่สามารถค้นหาได้เมื่อทำการกำหนดค่าขีดแบ่งสม่ำเสมอแบบตายตัวและกำหนดค่าขีดแบ่งคุณประโยชน์แบบแปรปรวนตามลำดับ

4.1 เวลาที่ใช้ในการคำนวณ

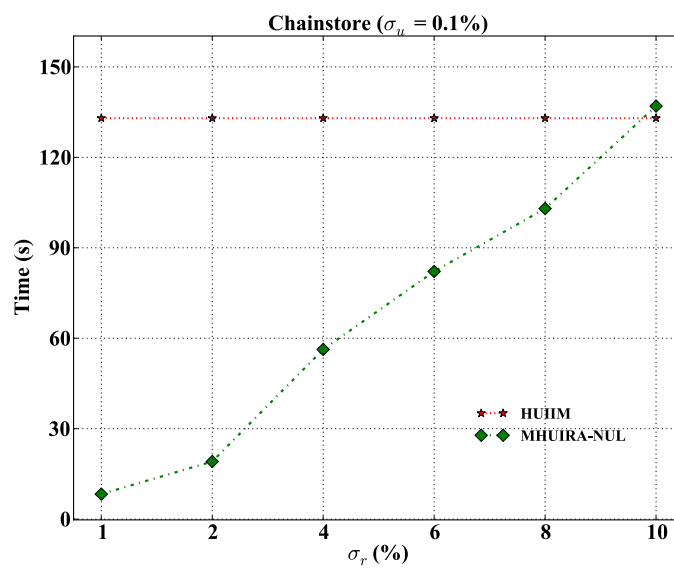
ภาพที่ 4-1 ถึง 4-10 แสดงให้เห็นถึงเวลาที่ใช้ในการประมวลผลแต่ละชุดข้อมูลด้วยขั้นตอนวิธี *HUIIM* เปรียบเทียบกับขั้นตอนวิธี *MHURIA-NUL* ภายใต้การกำหนดค่าขีดแบ่งความสม่าเสมอแบบแปรปรวนและค่าขีดแบ่งคุณสมบัติแบบตายตัวที่ต่ำสุด ซึ่งจากการกำหนดค่าให้กับพารามิเตอร์ดังกล่าวจะทำให้มีเซตรายการเป็นจำนวนมากมีค่าคุณสมบัติมากกว่าหรือเท่ากับค่าขีดแบ่งคุณสมบัติ และจะทำให้การค้นหาเซตรายการใช้เวลามาก ที่ซึ่งสะท้อนให้เห็นถึงประสิทธิภาพของขั้นตอนวิธี *HUIIM* เมื่อต้องพิจารณาเซตรายการเป็นจำนวนมาก โดยจากภาพจะสามารถสังเกตได้ว่าขั้นตอนวิธี *MHURIA-NUL* ใช้เวลาในการประมวลผลน้อยกว่าขั้นตอนวิธี *HUIIM* ในหลายกรณี เนื่องจากขั้นตอนวิธี *MHURIA-NUL* สามารถใช้ประโยชน์จากค่าขีดแบ่งความสม่าเสมอ ในการตัดเซตรายการที่ปรากฏอย่างไม่สม่าเสมอออกจากการพิจารณา ซึ่งส่งผลให้พื้นที่การค้นหาลดลงอย่างมีนัยสำคัญ โดยเมื่อค่าขีดแบ่งความสม่าเสมอมีค่าที่ต่ำจะทำให้ *MHURIA-NUL* ใช้เวลาในการพิจารณาเซตรายการน้อยลง แต่อย่างไรก็ตาม ในชุดข้อมูล *Kosarak* และ *PUMSB* ขั้นตอนวิธี *HUIIM* ใช้เวลาประมวลผลน้อยกว่าขั้นตอนวิธี *MHURIA-NUL* เนื่องจาก *MHURIA-NUL* ไม่สามารถใช้ประโยชน์จากค่าขีดแบ่งความสม่าเสมอ ในการลดทอนเซตรายการที่ปรากฏไม่สม่าเสมอออกจากการพิจารณาได้มากเหมือนกับฐานข้อมูลอื่นๆ แต่ขั้นตอนวิธี *HUIIM* สามารถใช้ประโยชน์จากค่าประมาณคุณสมบัติ *TWU* เพื่อระบุเซตรายการที่มีค่าประมาณคุณสมบัติต่ำ และทำการลบเซตรายการเหล่านั้นออกจากการพิจารณาได้อย่างรวดเร็ว



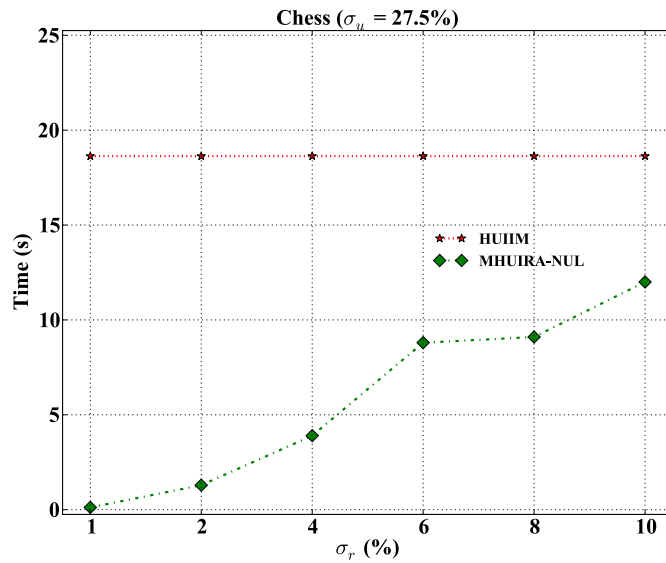
ภาพที่ 4-1 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่าเสมอ ของฐานข้อมูล *Accidents*



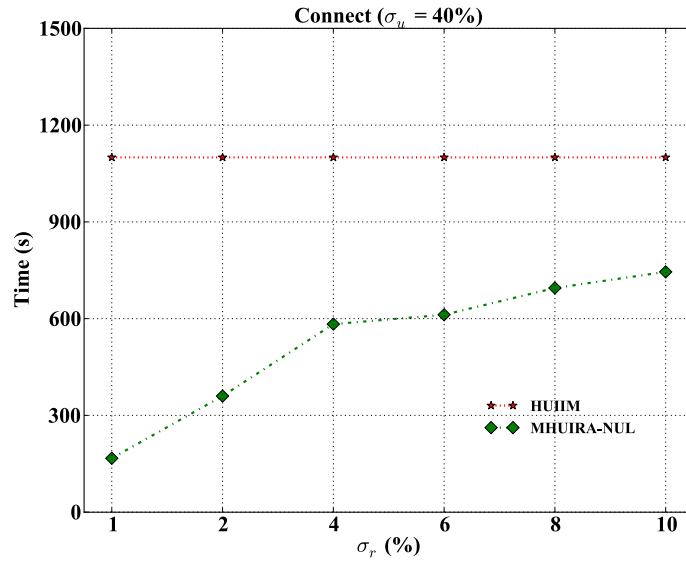
ภาพที่ 4-2 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHUIRA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอของฐานข้อมูล BMS



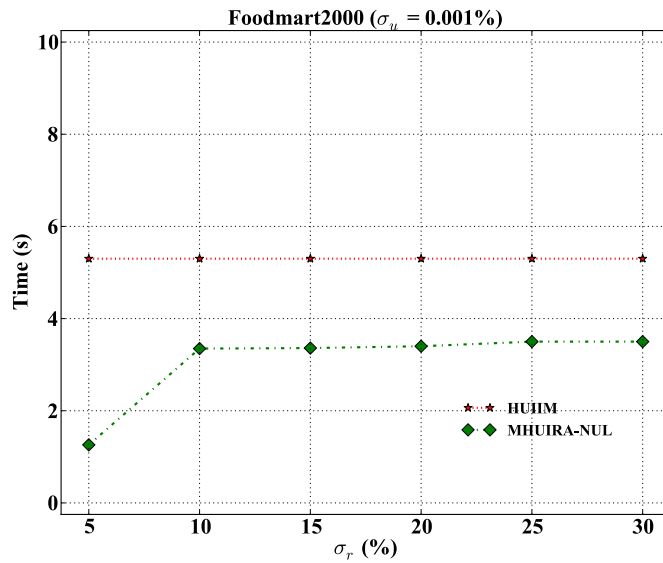
ภาพที่ 4-3 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHUIRA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอของฐานข้อมูล Chainstore



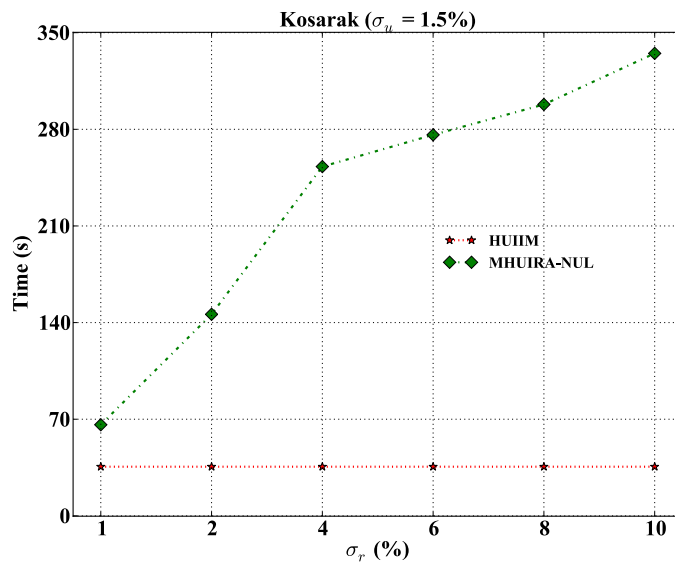
ภาพที่ 4-4 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอของฐานข้อมูล Chess



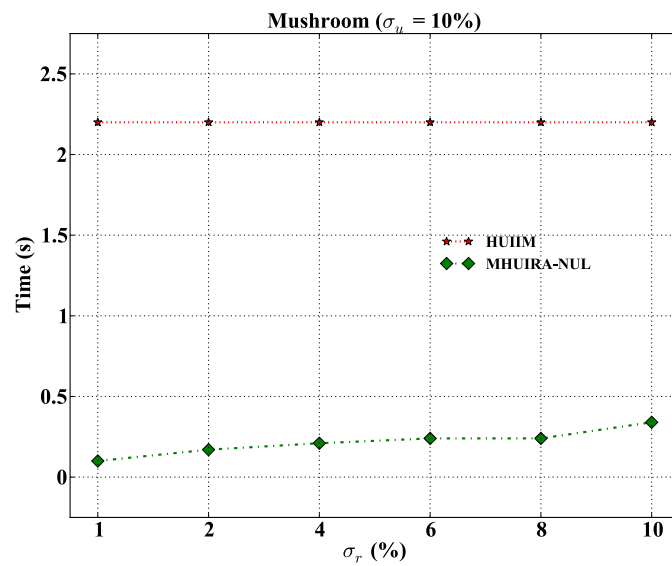
ภาพที่ 4-5 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอของฐานข้อมูล Connect



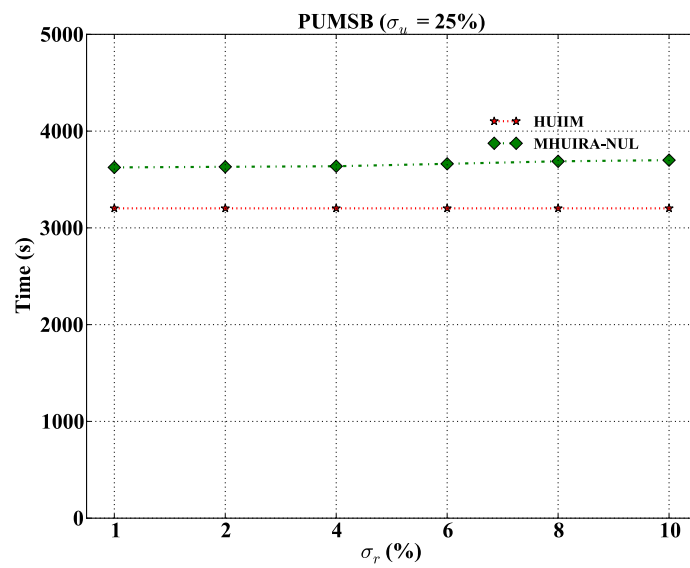
ภาพที่ 4-6 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอของฐานข้อมูล Foodmart2000



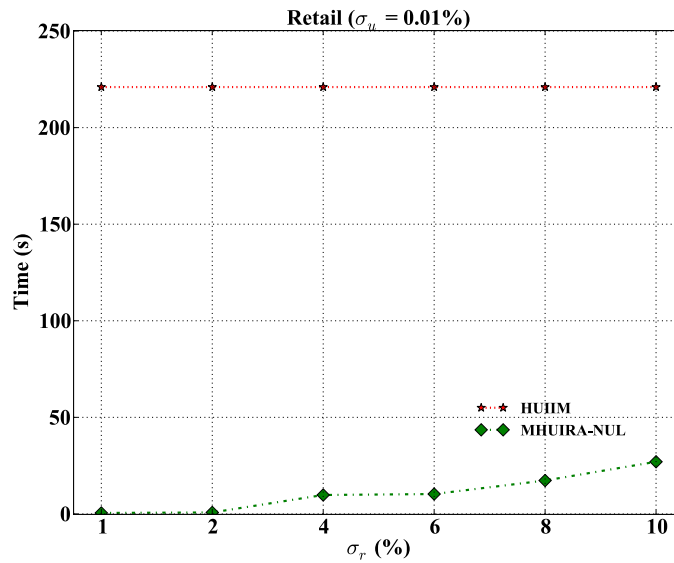
ภาพที่ 4-7 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอของฐานข้อมูล Kosarak



ภาพที่ 4-8 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอของฐานข้อมูล Mushroom

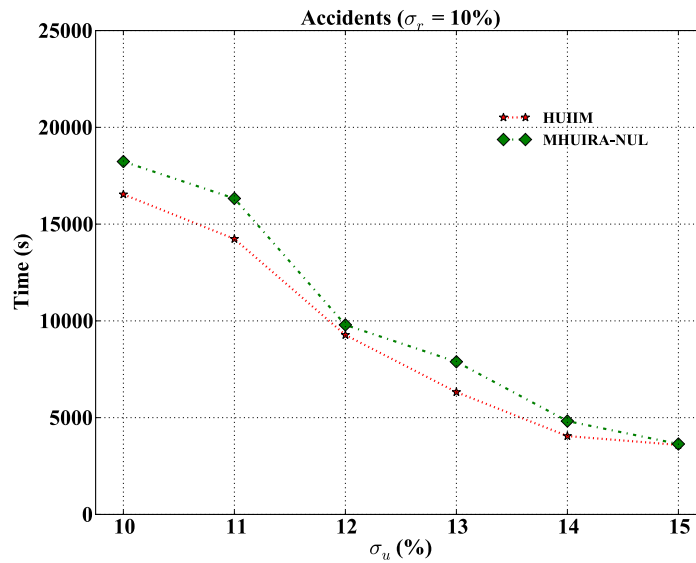


ภาพที่ 4-9 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอของฐานข้อมูล PUMSB

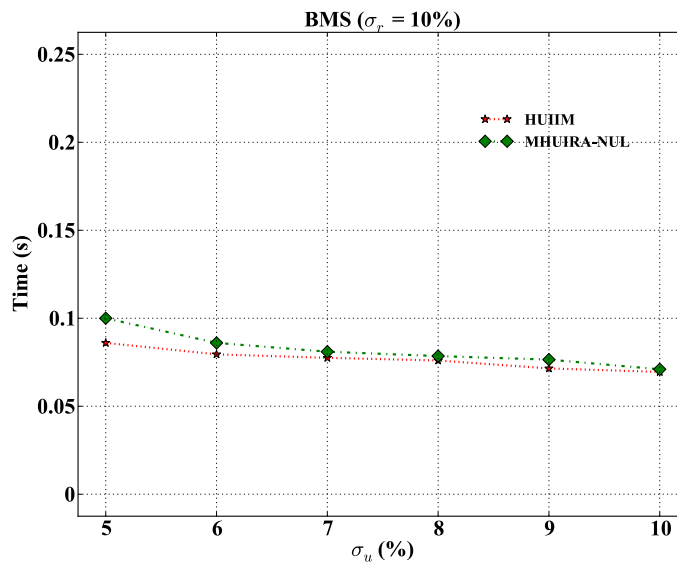


ภาพที่ 4-10 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอของฐานข้อมูล Retail

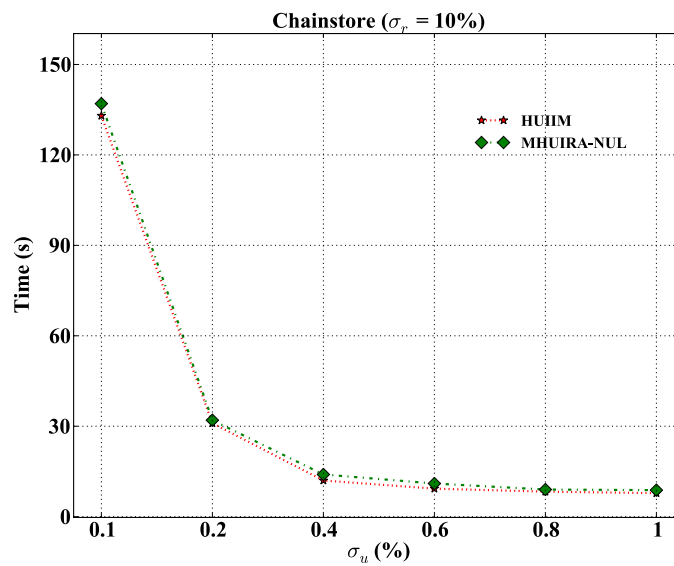
ภาพที่ 4-11 ถึง 4-20 แสดงให้เห็นถึงเวลาที่ใช้ในการประมวลผลแต่ละชุดข้อมูลด้วยขั้นตอนวิธี HUIIM เปรียบเทียบกับวิธี MHURIA-NUL ภายใต้การกำหนดค่าขีดคุณสมบัติแบบแปรปรวน และค่าขีดแบ่งความสม่ำเสมอแบบตายตัวที่ต่ำสุด จากภาพสังเกตเห็นได้ว่าเวลาในการประมวลผลของทั้งสองวิธีจะลดลงเมื่อเพิ่มค่าขีดแบ่งคุณสมบัติ เนื่องจากสามารถใช้ประโยชน์จากค่าขีดแบ่งคุณสมบัติในการตัดรูปแบบที่มีค่าคุณสมบัติต่ำ นอกจากนี้เรายังสังเกตเห็นได้ว่าเวลาในการประมวลผลของขั้นตอนวิธี HUIIM มากกว่า MHURIA-NUL ในชุดข้อมูล Chess, Connect, Mushroom, Foodmart2000 และ Retail เนื่องจากขั้นตอนวิธี MHURIA-NUL สามารถใช้ประโยชน์จากค่าขีดแบ่งความสม่ำเสมอในการลดทอนเซตรายการที่ปรากฏอย่างไม่สม่ำเสมอ แต่ในส่วนของชุดข้อมูล Accidents, Chainstore, Kosarak และ PUMSB ขั้นตอนวิธี HUIIM มีประสิทธิภาพที่ดีกว่าขั้นตอนวิธี MHURIA-NUL เนื่องจากขั้นตอนวิธี MHURIA-NUL ไม่สามารถใช้ประโยชน์จากค่าขีดแบ่งความสม่ำเสมอในการตัดเซตรายการที่ปรากฏอย่างไม่สม่ำเสมอ



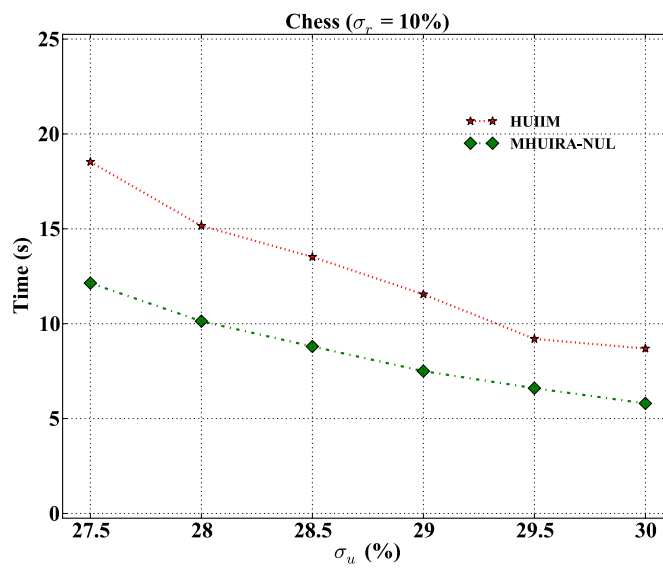
ภาพที่ 4-11 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ของฐานข้อมูล Accidents



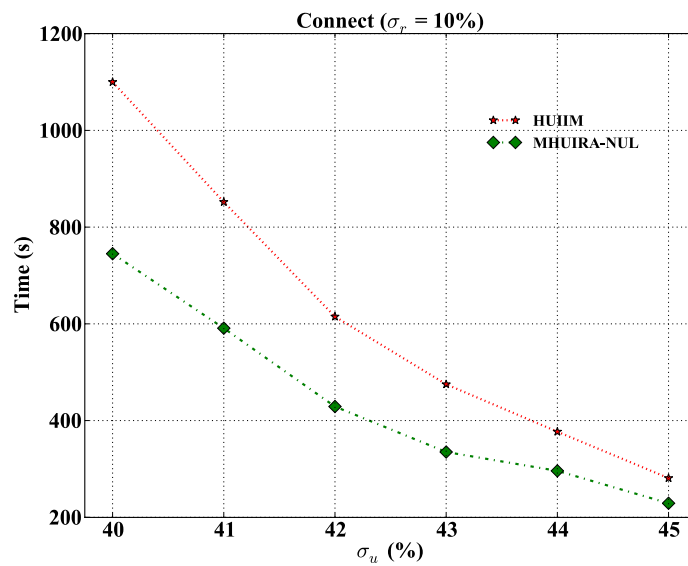
ภาพที่ 4-12 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ของฐานข้อมูล BMS



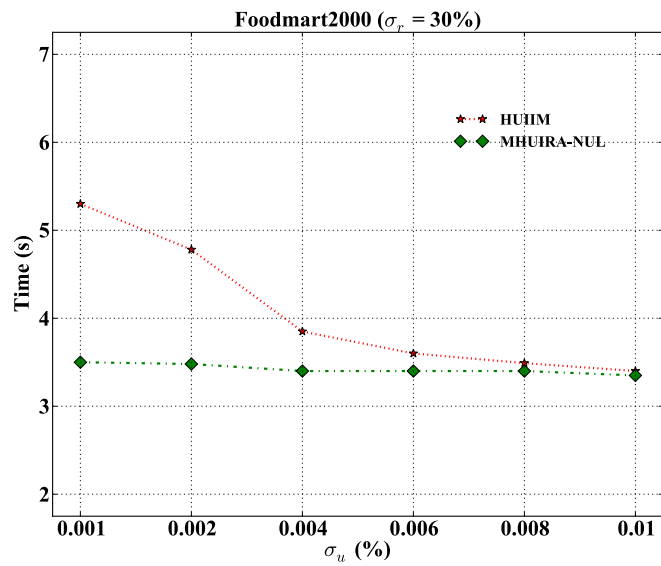
ภาพที่ 4-13 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ของฐานข้อมูล Chainstore



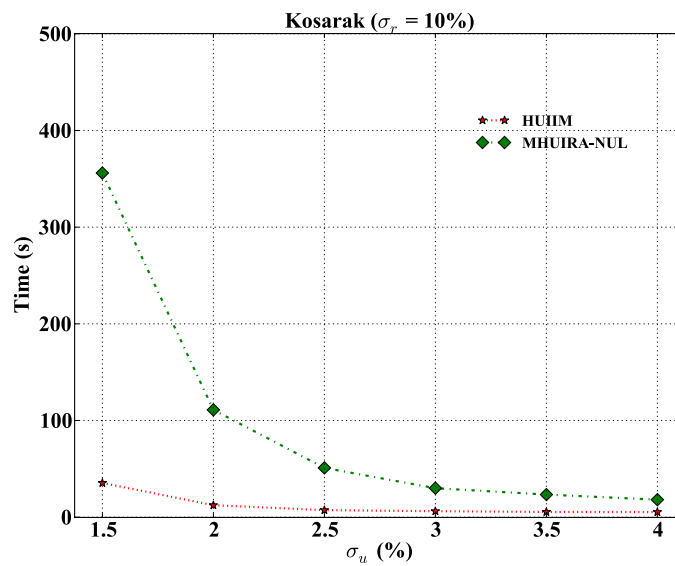
ภาพที่ 4-14 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ของฐานข้อมูล Chess



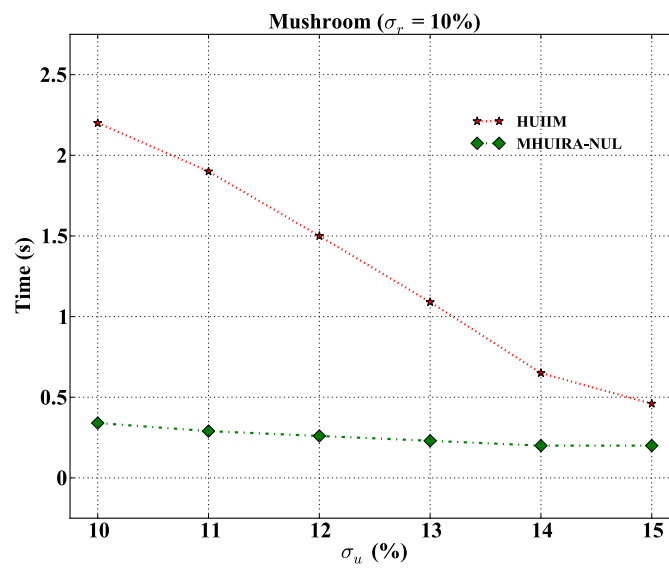
ภาพที่ 4-15 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ของฐานข้อมูล Connect



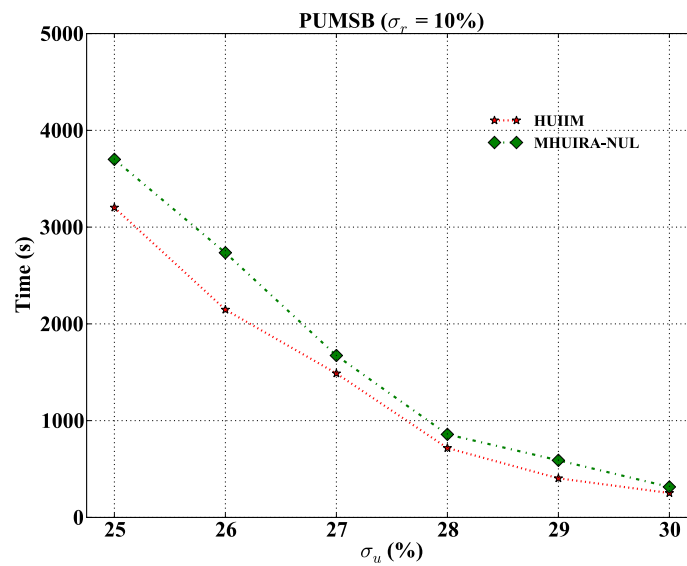
ภาพที่ 4-16 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ของฐานข้อมูล Foodmart2000



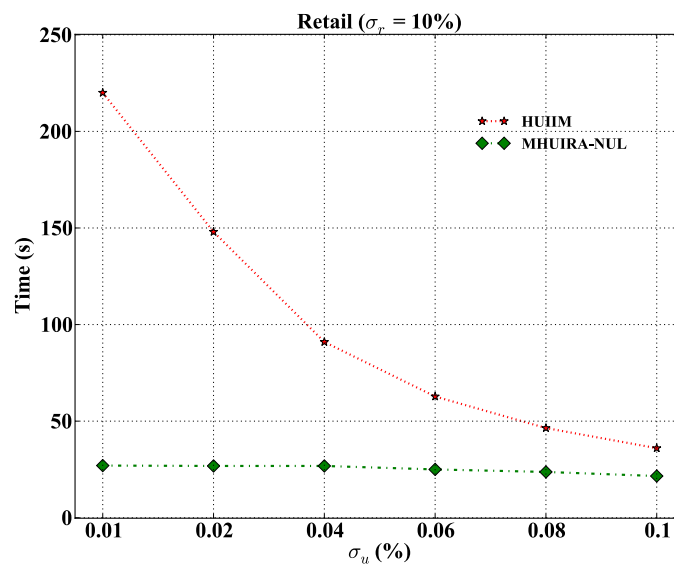
ภาพที่ 4-17 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ของฐานข้อมูล Kosarak



ภาพที่ 4-18 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดคุณประโยชน์ของฐานข้อมูล Mushroom



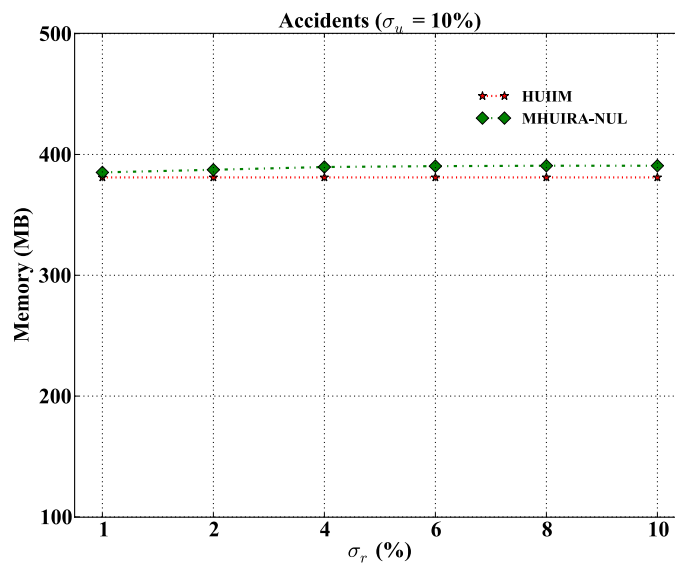
ภาพที่ 4-19 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดคูณประโยชน์ของฐานข้อมูล PUMSB



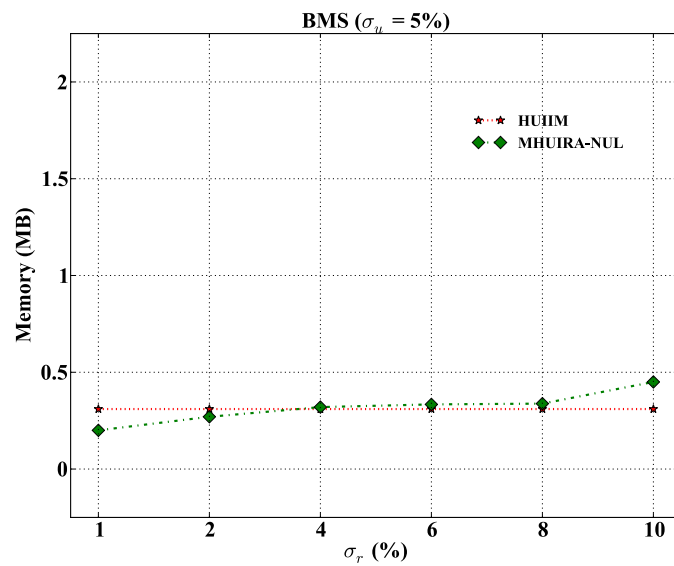
ภาพที่ 4-20 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าขีดคูณประโยชน์ของฐานข้อมูล Retail

4.2 หน่วยความจำที่ใช้ในการคำนวณ

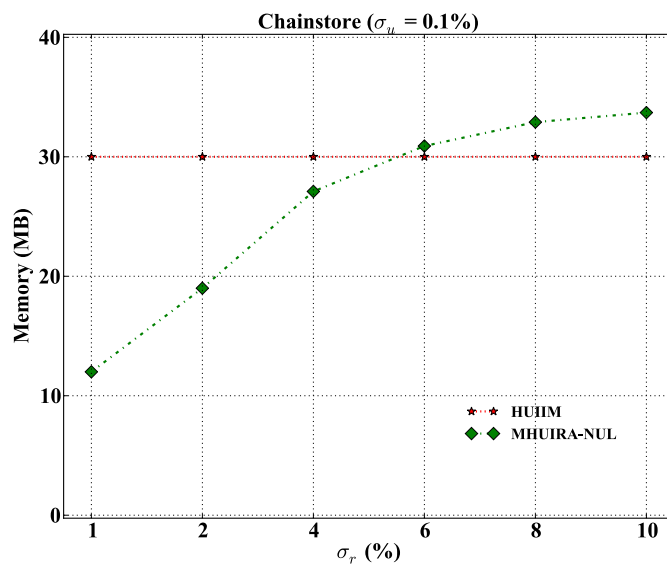
จากภาพที่ 4-21 ถึง 4-30 แสดงการใช้หน่วยความจำสูงสุดในการประมวลผลของขั้นตอนวิธี *HUIIM* และ *MHURIA-NUL* เมื่อทำการกำหนดค่าขีดแบ่งคุณประโยชน์แบบตายตัวและกำหนดค่าขีดแบ่งความสม่ำเสมอแบบแปรปรวน จากภาพสามารถสังเกตได้ว่าขั้นตอนวิธี *HUIIM* จะใช้หน่วยความจำคงที่ในทุกกรณี เนื่องจากขั้นตอนวิธี *HUIIM* ไม่ได้ใช้ประโยชน์จากค่าขีดแบ่งความสม่ำเสมอ ในการตัดเซตรายการที่ต้องเก็บในหน่วยความจำ ในส่วนของขั้นตอนวิธี *MHURIA-NUL* จะสังเกตได้ว่าเมื่อค่าขีดแบ่งความสม่ำเสมอมีค่าต่ำ จะทำให้ขั้นตอนวิธี *MHURIA-NUL* ใช้หน่วยความจำน้อย เนื่องจากขั้นตอนวิธี *MHURIA-NUL* สามารถใช้ประโยชน์จากค่าขีดแบ่งความสม่ำเสมอในการลดทอนเซตรายการที่ปรากฏอย่างไม่สม่ำเสมอ ซึ่งเป็นสาเหตุให้ขั้นตอนวิธี *MHURIA-NUL* เก็บเซตรายการในหน่วยความจำเพียงไม่กี่เซตรายการ



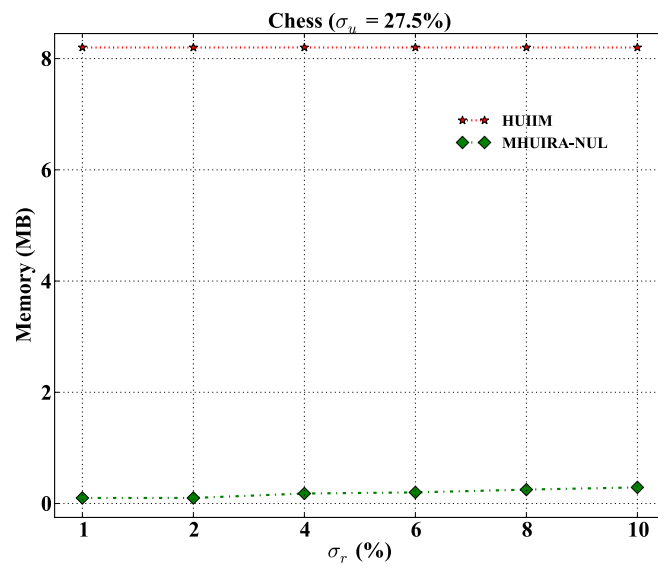
ภาพที่ 4-21 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าความสม่ำเสมอของฐานข้อมูล *Accidents*



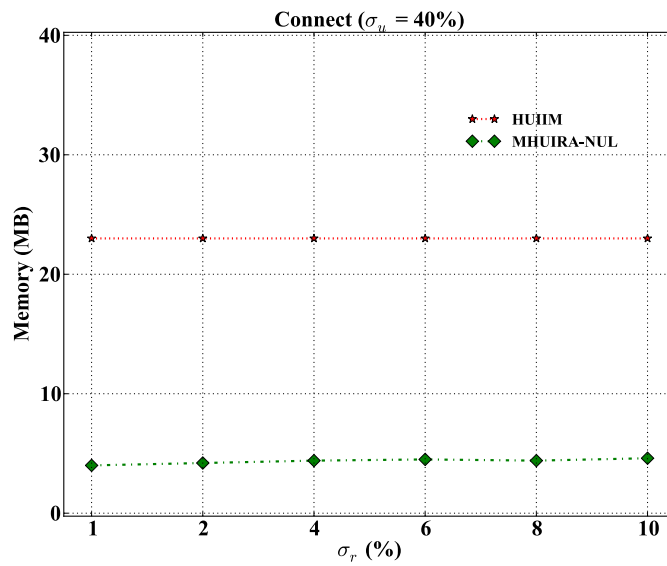
ภาพที่ 4-22 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHUIRA-NUL เมื่อทำการเปลี่ยนแปลงค่าความสม่ำเสมอของฐานข้อมูล BMS



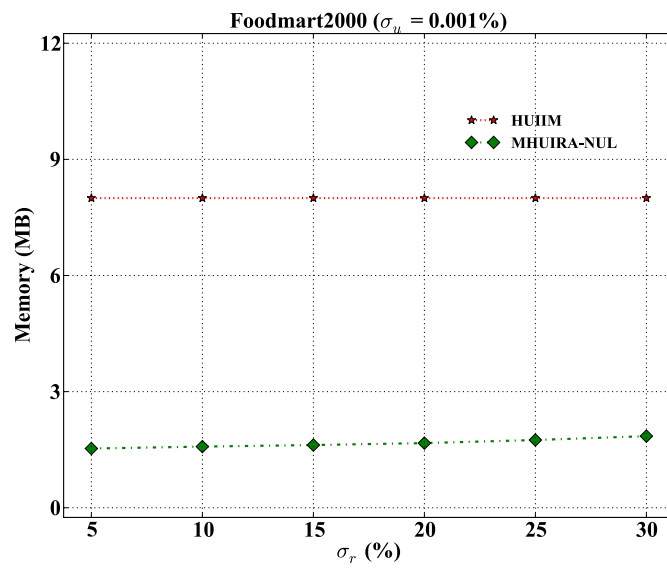
ภาพที่ 4-23 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHUIRA-NUL เมื่อทำการเปลี่ยนแปลงค่าความสม่ำเสมอของฐานข้อมูล Chainstore



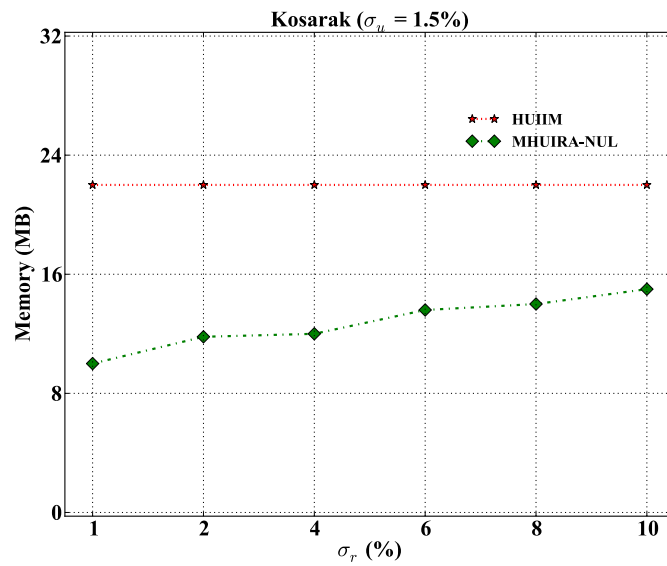
ภาพที่ 4-24 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHUIRA-NUL เมื่อทำการเปลี่ยนแปลงค่าความสม่ำเสมอของฐานข้อมูล Chess



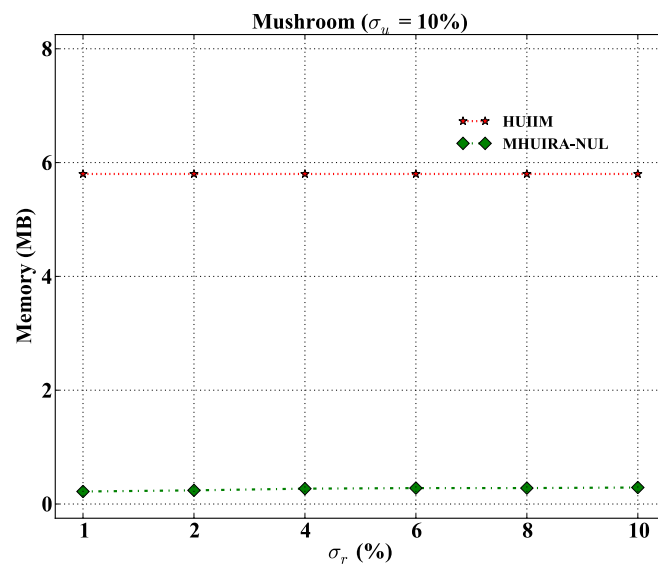
ภาพที่ 4-25 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHUIRA-NUL เมื่อทำการเปลี่ยนแปลงค่าความสม่ำเสมอของฐานข้อมูล Connect



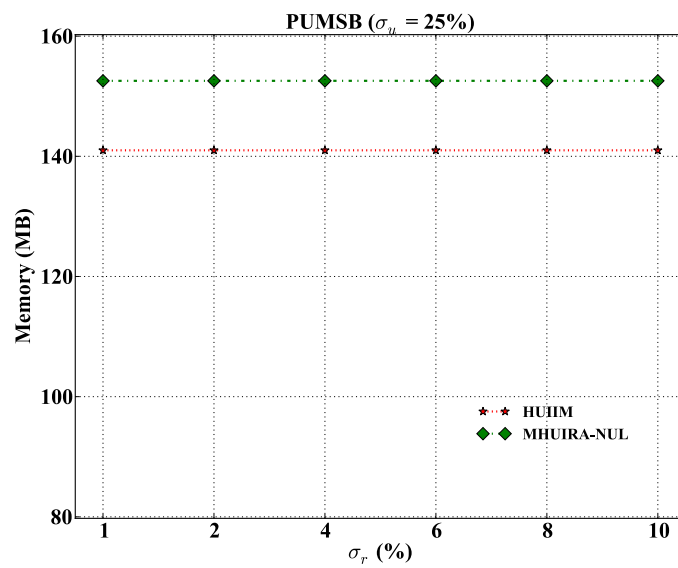
ภาพที่ 4-26 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHUIRA-NUL เมื่อทำการเปลี่ยนแปลงค่าความสม่ำเสมอของฐานข้อมูล Foodmart2000



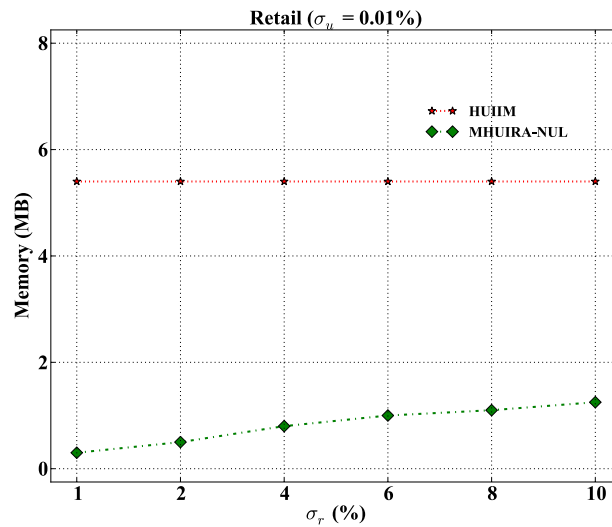
ภาพที่ 4-27 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHUIRA-NUL เมื่อทำการเปลี่ยนแปลงค่าความสม่ำเสมอของฐานข้อมูล Kosarak



ภาพที่ 4-28 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าความสม่ำเสมอของฐานข้อมูล Mushroom

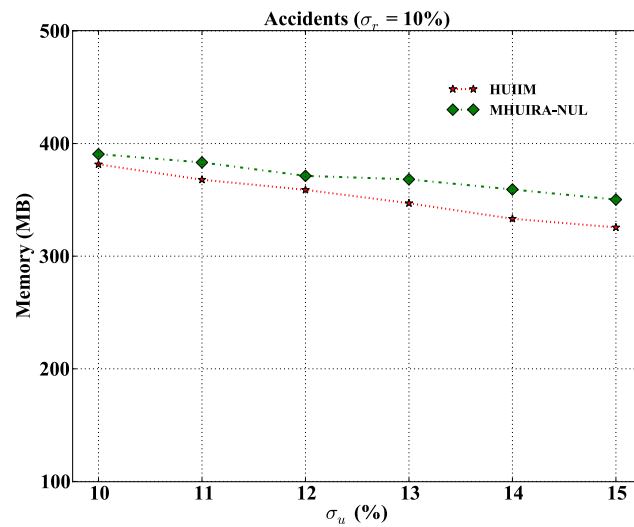


ภาพที่ 4-29 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าความสม่ำเสมอของฐานข้อมูล PUMSB

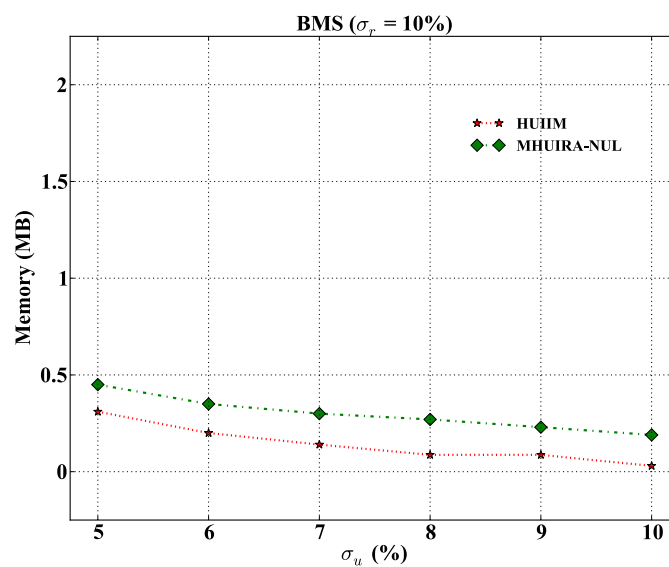


ภาพที่ 4-30 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าความสม่ำเสมอของฐานข้อมูล *Retail*

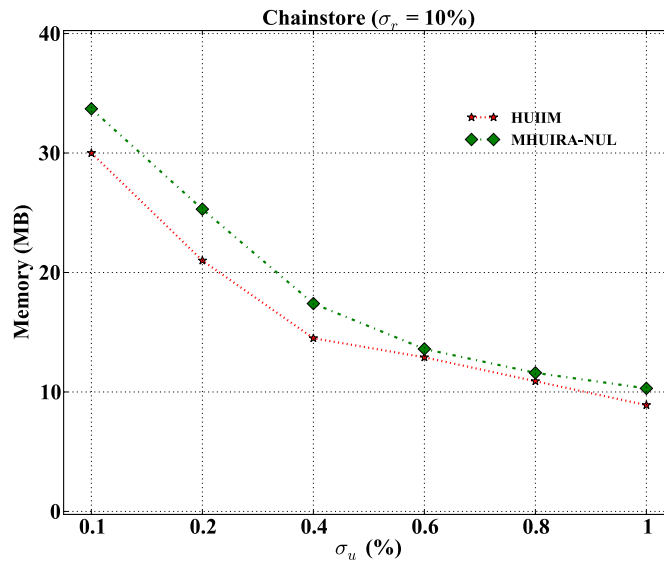
จากภาพที่ 4-31 ถึง 4-40 แสดงการใช้หน่วยความจำสูงสุดในการประมวลผลของขั้นตอนวิธี *HUIIM* และ *MHURIA-NUL* เมื่อทำการกำหนดค่าขีดแบ่งความสม่ำเสมอแบบตายตัวและกำหนดค่าขีดแบ่งคุณสมบัติแบบแปรปรวน จากภาพสังเกตเห็นว่าการใช้หน่วยความจำของทั้งสองวิธีจะลดลงเมื่อค่าขีดแบ่งคุณสมบัติเพิ่มขึ้น เนื่องจากทั้งสองวิธีสามารถใช้ประโยชน์จากค่าขีดแบ่งคุณสมบัติในการตัดเซตรายการที่มีค่าคุณสมบัติต่ำ ซึ่งเป็นสาเหตุให้ทั้งสองวิธีเก็บเซตรายการในหน่วยความจำเพียงไม่กี่เซตรายการ



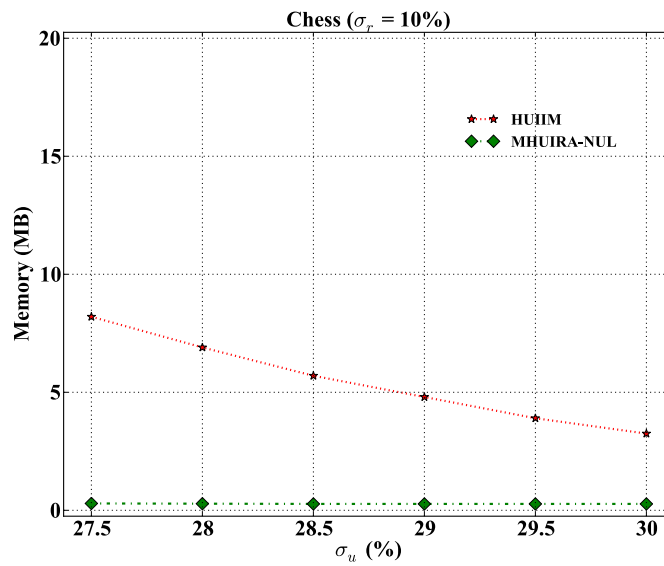
ภาพที่ 4-31 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHUIRA-NUL เมื่อทำการเปลี่ยนแปลงค่าคุณประโยชน์ของฐานข้อมูล Accidents



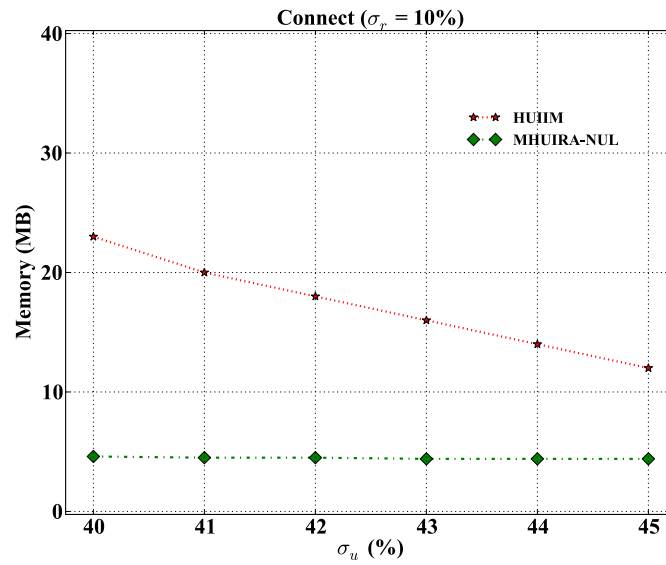
ภาพที่ 4-32 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHUIRA-NUL เมื่อทำการเปลี่ยนแปลงค่าคุณประโยชน์ของฐานข้อมูล BMS



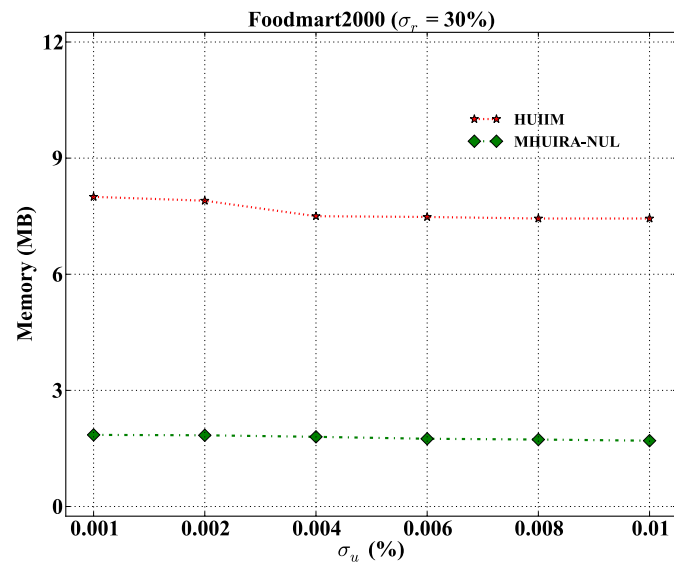
ภาพที่ 4-33 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHUIRA-NUL เมื่อทำการเปลี่ยนแปลงค่าคุณประโยชน์ของฐานข้อมูล Chainstore



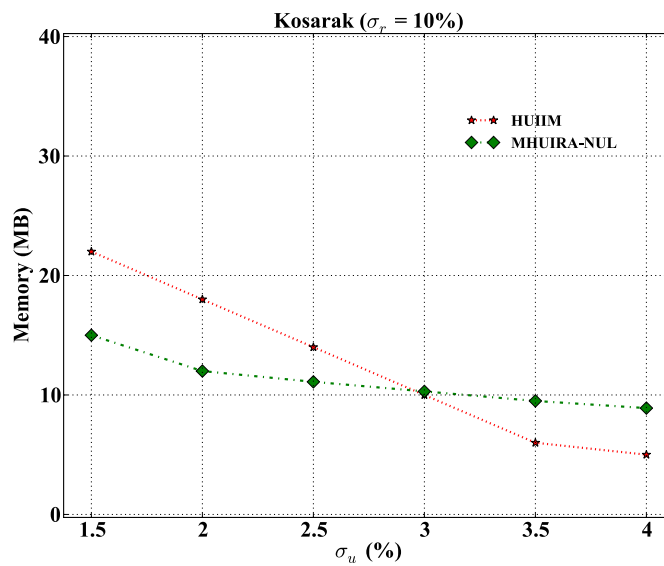
ภาพที่ 4-34 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHUIRA-NUL เมื่อทำการเปลี่ยนแปลงค่าคุณประโยชน์ของฐานข้อมูล Chess



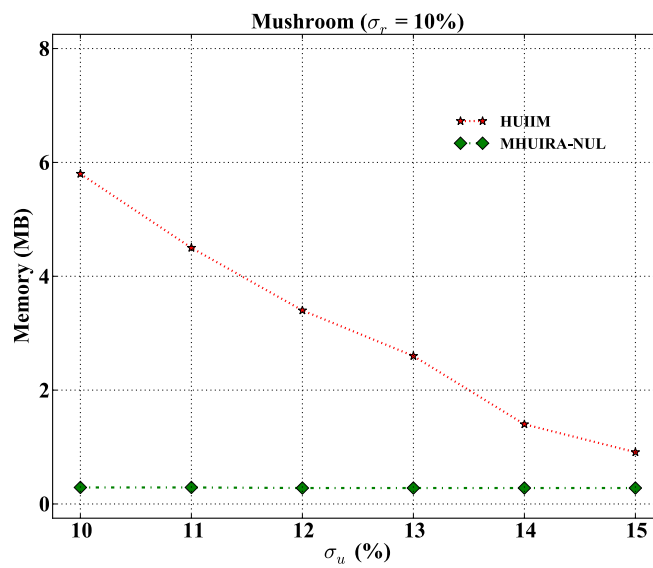
ภาพที่ 4-35 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าคุณสมบัติของฐานข้อมูล Connect



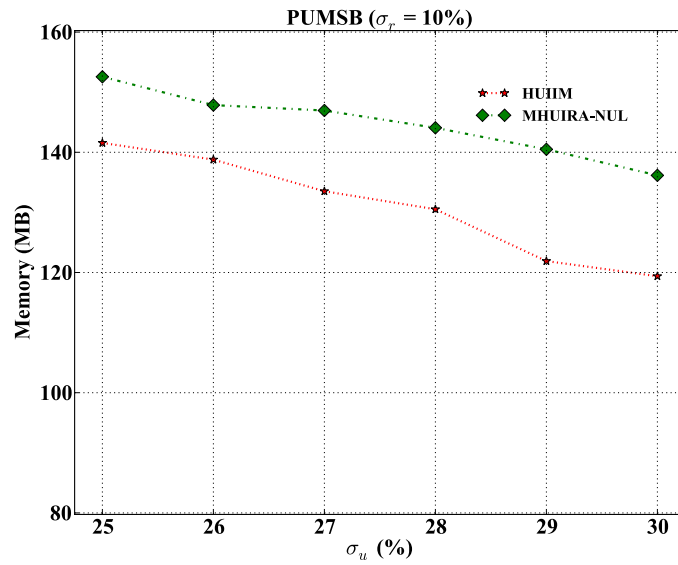
ภาพที่ 4-36 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHURIA-NUL เมื่อทำการเปลี่ยนแปลงค่าคุณสมบัติของฐานข้อมูล Foodmart2000



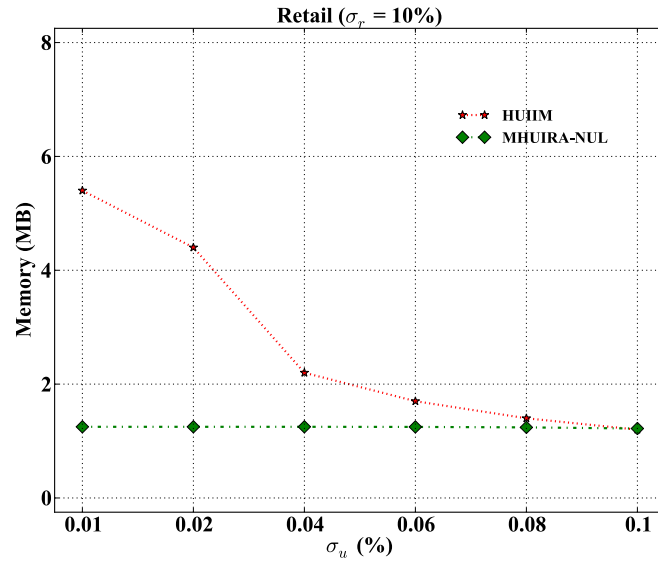
ภาพที่ 4-37 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHUIRA-NUL เมื่อทำการเปลี่ยนแปลงค่าคุณประโยชน์ของฐานข้อมูล Kosarak



ภาพที่ 4-38 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHUIRA-NUL เมื่อทำการเปลี่ยนแปลงค่าคุณประโยชน์ของฐานข้อมูล Mushroom



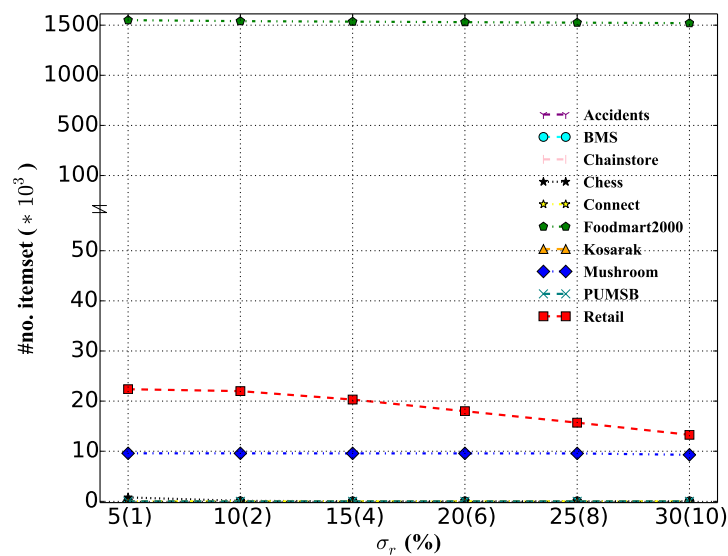
ภาพที่ 4-39 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHUIRA-NUL เมื่อทำการเปลี่ยนแปลงค่าคุณประโยชน์ของฐานข้อมูล PUMSB



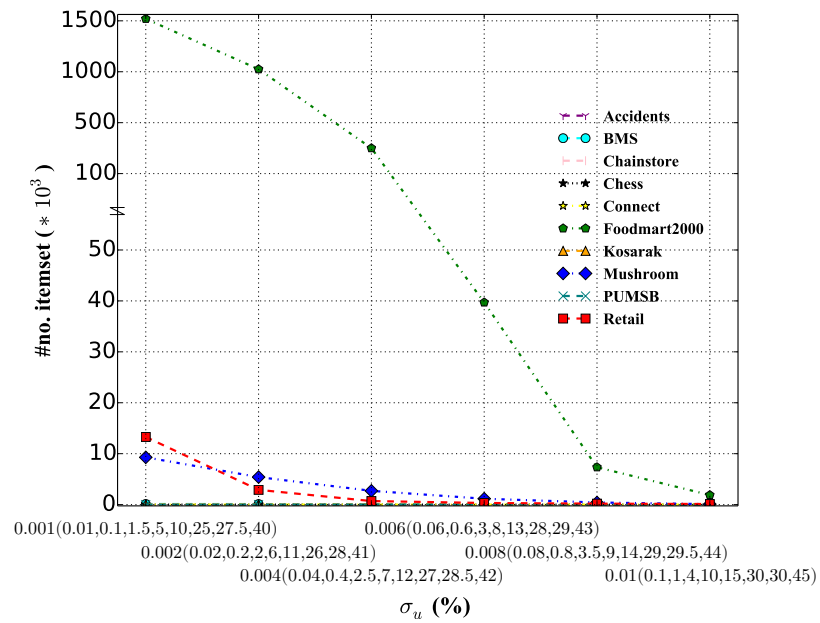
ภาพที่ 4-40 หน่วยความจำที่ใช้ในการคำนวณของขั้นตอนวิธี HUIIM และ MHUIRA-NUL เมื่อทำการเปลี่ยนแปลงค่าคุณประโยชน์ของฐานข้อมูล Retail

4.3 ผลลัพธ์

ในส่วนสุดท้าย แสดงให้เห็นถึงจำนวนของเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอที่ค้นหาได้จากชุดข้อมูลทั้งหมด ภายใต้การกำหนดค่าขีดแบ่งคุณประโยชน์แบบตายตัว และกำหนดค่าขีดแบ่งความสม่ำเสมอแบบแปรปรวน (แสดงในภาพที่ 4-41) และภายใต้การกำหนดค่าขีดแบ่งความสม่ำเสมอแบบตายตัว และกำหนดค่าขีดแบ่งคุณประโยชน์แบบแปรปรวน (แสดงในภาพที่ 4-42) จากภาพทั้งสองจะสังเกตได้ว่าจำนวนของเซตรายการจะลดลงอย่างมีนัยสำคัญเมื่อทำการกำหนดค่าขีดแบ่งความสม่ำเสมอหรือค่าขีดแบ่งคุณประโยชน์เพิ่มขึ้น นอกจากนี้เรายังสังเกตได้ว่า ภาพที่ 4-41 เมื่อค่าขีดแบ่งความสม่ำเสมอมีค่าที่ต่ำ จะทำให้มีเซตรายการจำนวนมากที่มีค่าความสม่ำเสมอมากกว่าค่าขีดแบ่งความสม่ำเสมอ ในทำนองเดียวกัน ภาพที่ 4-42 เมื่อค่าขีดแบ่งคุณประโยชน์มีค่าที่ต่ำ จะทำให้มีเซตรายการจำนวนมากที่มีค่าคุณประโยชน์ไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์



ภาพที่ 4-41 จำนวนผลลัพธ์ที่ค้นหาได้จากขั้นตอนวิธี *HUIIM* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ



ภาพที่ 4-42 จำนวนผลลัพธ์ที่ค้นหาได้จากขั้นตอนวิธี *HUIIM* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์

บทที่ 5

สรุปผลวิจัย

งานวิจัยนี้นำเสนอแนวความคิดในการค้นหารูปแบบภายใต้การพิจารณาความไม่สม่ำเสมอของการปรากฏและค่าคุณประโยชน์ของรูปแบบ (โดยค่าคุณประโยชน์อาจหมายถึง ผลกำไร ต้นทุน หรือ อื่น ๆ) ซึ่งการค้นหารูปแบบภายใต้การพิจารณาข้างต้นจะมีส่วนช่วยในการสังเกตพฤติกรรม การซื้อของลูกค้าที่ทำการซื้อสินค้าที่ให้ผลตอบแทนสูง ที่ซึ่งเป็นข้อมูลที่ใช้ในการจัดการบริหาร สินค้าคงคลัง วางแผนการผลิต และทำให้ผู้บริหารสามารถวิเคราะห์ความต้องการลูกค้า สามารถวางแผนการตลาดได้ดียิ่งขึ้น จากกรอบความคิดข้างต้น รูปแบบหนึ่งๆจะเป็นรูปแบบที่น่าสนใจก็ต่อเมื่อเป็นรูปแบบที่ปรากฏอย่างไม่สม่ำเสมอ (กล่าวคือ รูปแบบมีค่าความสม่ำเสมอต้องมากกว่าค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด) และเป็นรูปแบบที่ให้ผลตอบแทนสูง (กล่าวคือ รูปแบบมีค่าคุณประโยชน์ไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด)

ในการค้นหาแบบดังกล่าวอย่างมีประสิทธิภาพ ผู้วิจัยได้นำเสนอขั้นตอนวิธีที่เรียกว่า “*Mining High-Utility Itemsets with Irregular Occurrence, HUIIM*” ที่ทำการอ่านข้อมูลจากฐานข้อมูลเพียงครั้งเดียว และทำการปรับปรุงโครงสร้างการเก็บข้อมูล (*New modified utility-list, NUL*) เพื่อทำการจัดเก็บข้อมูลการปรากฏขึ้นและค่าคุณประโยชน์ของรูปแบบหนึ่งๆ ให้มีประสิทธิภาพ และได้ประยุกต์ใช้แนวความคิดเกี่ยวกับค่าคุณประโยชน์คงเหลือ (*remaining utility*) ค่าประมาณคุณประโยชน์แบบกระชับ (*tight over-estimated utility, tou*) และค่าประมาณคุณประโยชน์ (*transaction-weighted utility, TWU*) เพื่อทำการลดทอนปริภูมิสถานะของการค้นหาแบบ

ในการศึกษาเกี่ยวกับประสิทธิภาพของขั้นตอนที่นำเสนอ ผู้วิจัยได้ทำการทดลองเพื่อทดสอบประสิทธิภาพของขั้นตอนวิธีที่เสนอกับชุดข้อมูลจริง 10 ชุดข้อมูล โดยทำการทดลองในสามแง่มุมด้วยกันคือ เวลาที่ใช้ในการหาผลลัพธ์ หน่วยความจำที่ใช้ในการหาผลลัพธ์ และจำนวนผลลัพธ์ที่สามารถค้นหาได้ โดยการทดลองชี้ให้เห็นว่าขั้นตอนวิธี *HUIIM* สามารถค้นหาแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอได้อย่างมีประสิทธิภาพ

บรรณานุกรม

- Ahmed, C.F., Tanbeer, S.K., Jeong, B.S., and Lee Y.K. (2009), Efficient tree structures for high utility pattern mining in incremental databases, *IEEE Trans. Knowl. Data Eng.*, 21, pp. 1708–1721.
- Ahmed, C.F., Tanbeer, S.K., Jeong, B.-S., and Choi, H.-J. (2012), Interactive mining of high utility patterns over data streams. *Expert Systems with Applications*, 39(15), 2012, pp. 11979–11991.
- Amphawan, K., Lenca, P., and Surarerks, A. (2009), Mining top-k periodic-frequent pattern from transactional databases without support threshold, *Proceeding of the 3rd International Conference on Advances in Information Technology (IAIT-09)*, 55, 2009, pp. 18-29.
- Amphawan, K., Lenca, P., and Surarerks, A. (2012), Mining top-k regular-frequent itemsets using database partitioning and support estimation, *Expert Systems with Applications*, 39(2), 2012, pp. 1924-1936
- Amphawan, K., Lenca, P. (2015), Mining top-k frequent-regular closed patterns, *Expert Systems with Applications*, Elsevier, 42(21), 2015, pp. 7882 - 7894.
- Amphawan, K., Lenca, P., Jitpattanakul, A., and Surarerks, A. (2016), Mining high utility itemsets with regular occurrence, *Journal of ICT Research and Applications*, vol. 10, no. 2, pp. 153–176,
- Amphawan, K., Surarerks, A. (2015-b), Pushing regularity constraint on high utility itemsets mining, *Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*.
- Chan, R., Yang, Q., Shen, Y.D. (2003), Mining high utility itemsets, in: *IEEE International Conference on Data Mining*, pp. 19–26.
- Chu, C.-J., Tseng, V.S., and Liang, T. (2008), An Efficient mining for mining temporal high utility itemsets from data streams. *Journal of Systems and Software*, 81, 2008, pp. 1105–1117.
- Chu, C.-J., Tseng, V. S., and Liang, T. (2009), An Efficient Algorithm for Mining High utility Itemsets with Negative Values in Large Databases. *In Applied Mathematics and Computation*, 215(2), 2009, pp. 767-778.

- Deng, Zhi-Hong. "An efficient structure for fast mining high utility itemsets." *Applied Intelligence* (2018): 1-17.
- Erwin, A., Gopalan, R.P., and Achuthan, N. (2007), A bottom-up projection based algorithm for mining high utility itemsets, *Proceedings of the 2nd international workshop on Integrating artificial intelligence and data mining*, 84, pp. 3 – 11.
- Fournier-Viger, P., Wu, C.-W. (2016), and V.S. Tseng, Novel Concise Representations of High Utility Itemsets using Generator Patterns. *In: Proc. 10th International Conference on Advanced Data Mining and Applications, Springer LNAI*, 2014.
- Fournier-Viger, P., and Zida, S. (2015), FOSHU: Faster On-Shelf High Utility Itemset Mining—with or without Negative Unit Profit, *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, 2015, pp. 857-864.
- Fournier-Viger, P. (2014), FHN: Efficient Mining of High-Utility Itemsets with Negative Unit Profits. *In Proc. 10th International Conference on Advanced Data Mining and Applications*, Springer.
- Fournier-Viger, P., Wu, C.-W., Zida, S., and Tseng V.S. (2014), FHM: Faster High-Utility Itemset Mining using Estimated Utility Co-occurrence Pruning. *In: Proc. 21st Intern. Symp. Methodologies Intell. Systems, Springer*, pp. 83-92.
- Feng, L., Wang, L., and Jin, B. (2013), UT-tree: efficient mining of high utility itemsets from data streams, *Intelligent Data Analysis*, 17 (4), 2013, pp. 585 – 602.
- Fournier-Viger, P., Lin, J. C.-W., Duong, Q.-H., and Dam, T.-L., (2016), *PHM: Mining Periodic High-Utility Itemsets*, pp. 64–79.
- Han, J., Pei, J., and Yin, Y. (2000), Mining Frequent Patterns without Candidate Generation, *SIGMOD '00 Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, Pages 1-12.
- Hong, T.-P., Lee, C.-H., and Wang, S.-L. (2011), Effective utility mining with the measure of average utility. *Expert Systems with Applications*, 38(7), pp. 8259–8265.
- Kumar, G.V., Kumari, V.V. (2012), Sliding window technique to mine regular frequent patterns in data streams using vertical format, *IEEE International Conference on Computational Intelligence & Computing Research (ICIC)*, 2012, pp. 1–4.

- Kiran, R.U., and Kitsuregawa, M. (2013), Discovering Quasi-Periodic-Frequent Patterns in Transactional Databases, *International Conference on Big Data Analytics*, 2013, pp. 97 – 115.
- Kiran, R.U., and Kitsuregawa, M. (2014), Novel Techniques to Reduce Search Space in Periodic-Frequent Pattern Mining, *Database Systems for Advanced Application*, vol. 8422, 2014, pp. 377 – 391.
- Khaleel, M.A., Dash, G.N., Choudhury, K.S., and Khan, M.A. (2014), Medical Data Mining for Discovering Periodically Frequent Diseases from Transactional Databases, *Proceedings of the International Conference Computational Intelligence in Data Mining*, 2014, pp. 87 – 96.
- Kiran, R.U., and Kitsuregawa, M. (2015), Discovering Chronic-Frequent Patterns in Transactional Databases, *International Workshop on Databases in Networked Information Systems*, 2015, pp. 12 – 26.
- 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*, 2010.
- Krishnamoorthy, S. (2015), Pruning strategies for mining high utility itemsets, *Expert Systems with Applications*, 42, 2015, pp. 2371 – 2381.
- Lan, G.-C., Hong, T.-P., Tseng, V.S. (2014), An efficient projection-based indexing approach for mining high utility itemsets, *Knowledge and Information Systems*, pp. 85 – 107.
- Liu, M., and Qu, J.-F. (2012), Mining high utility itemsets without candidate generation, *International Conference on Information and Knowledge Management (CIKM 2012)*, pp. 55–64.
- Lin, C.-W., Lan, G.-C., and Hong, T.-P. (2012), An incremental mining algorithm for high utility itemsets. *Expert Systems with Applications*, (8)39, pp. 7173–7180
- Lin, C.-W., Gan, W., Hong, T.-P., and Pan, J.-S. (2014), Incrementally Updating High-Utility Itemsets with Transaction Insertion, *Advanced Data Mining and Applications*, Vol. 8933, 2014, pp. 44-56.

- Li, H.-F., Huang, H.-Y., and Lee, S.-Y. (2011), Fast and memory efficient mining of high-utility itemsets from data streams: with and without negative item profits. *Knowledge Information and Systems*, 28(3), pp. 495–522.
- Lin, M.-Y., Tu, T.-F., and S.-C. (2012). Hsueh, High utility pattern mining using the maximal itemset property and lexicographic tree structures. *Information Sciences*, 215, 2012, pp. 1–14.
- Luo, X., Yuan, H., and Luo, Q. (2013), On Discovering Feasible Periodic Patterns in Large Database, *IEEE International Conference Dependable, Autonomic and Secure Computing*, 2013, pp. 344 – 351.
- Liu, Y., Liao, W.K., and Choudhary, A. (2005), A two-phase algorithm for fast discovery of high utility itemsets, *Advance Knowledge Discovery Data Mining*, pp. 689–695.
- Lin, C.W., Hong, T.P., and Lu, W.H. (2011), An effective tree structure for mining high utility itemsets, *Expert Syst. Appl.*, 38, 2011, pp. 7419–7424.
- Lan, G.-C., Hong, J.-P. (2014). Huang and V.S. Tseng, On-shelf utility mining with negative item values. In *Expert Systems with Applications*, 41, 2014, pp. 3450–3459.
- MacKinnon, R.K., Strauss, T.D., and Leung, C.K. (2014), DISC: efficient uncertain frequent pattern mining with tightened upper bounds. In *Proc. IEEE ICDM Workshops*, pp. 1038 – 1045.
- Mai, Thang, Bay Vo, and Loan TT Nguyen. "A lattice-based approach for mining high utility association rules." *Information Sciences* 399 (2017): 81-97.
- Ryang, H., and Yun, U. (2015), Top-K High Utility Pattern Mining with Effective Threshold Raising Strategies, *Knowledge-Based Systems*, Vol. 76, 2015, pp. 109-126.
- 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*, 2012.
- Tseng, V.S., Shie, B.-E., Wu, C.-W., and Yu, P.S. (2013), Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases. *IEEE Trans. Knowl. Data Eng.*, 25(8), 2013, pp. 1772–1786.
- Tseng, V.S., Wu, C., Fournier-Viger, P. (2015), and P. S. Yu, Efficient Algorithms for Mining the Concise and Lossless Representation of High Utility Itemsets, *IEEE*

- Transactions on Knowledge and Data Engineering*, Vol. 27(3), 2015, pp. 726 - 739.
- Tseng, V.S., Fournier-Viger, C.-W. Wu. P., and Yu, P.S. (2015-b), Efficient Algorithms for Mining Top-K High Utility Itemsets, *IEEE Transactions on Knowledge and Data Engineering*, 99, 2015.
- Tanbeer, S.K., Ahmed, C.F., Jeong, B.S., and Lee, Y.K. (2009), Discovering periodic-frequent patterns in transactional databases. *Proceedings of 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD)*, 2009, pp. 242 – 253.
- Tanbeer, S.K., Ahmed, C.F., and Jeong, B.S. (2010-a), Mining regular patterns in incremental transactional databases. *Proceedings of Int. Asia-Pacific Web Conference, IEEE Computer Society*, 2010.
- Tanbeer, S.K., Ahmed, C.F., and Jeong, B.S. (2010-b), Mining regular patterns in data streams. *Proceedings of DASFAA, Volume 5981 of LNCS.*, Springer, 2010.
- Wu, C.W., Fournier-Viger, P., Yu, P.S. and V. S. Tseng (2011), Efficient mining of a concise and lossless representation of high utility itemsets. In *Proc. IEEE Int'l Conf. Data Mining, 2011*, pp. 824 –833.
- Wu, C.-W., Shie, B.-E., Tseng, V.S., and Yu, P.S. (2012), Mining top-K high utility itemsets, *Knowledge Discovery and Data Mining (KDD 2012)*, 2012, pp. 78–86.
- Yun, U., Leggett, J. (2005), Wfim: weighted frequent itemset mining with a weight range and a minimum weight. In: *Proceeding of the 2005 SIAM international conference on data mining (SDM'05)*, Newport Beach, CA, pp 636–640
- Yao, H., Hamilton, H.J., Butz, C.J. (2004), A foundational approach to mining itemset utilities from databases, in: *SIAM International Conference on Data Mining*, pp. 482–486.
- Yun, U., and Ryang H. (2015), Incremental High Utility Pattern Mining with Static and Dynamic Databases, *Applied Intelligence*, Vol. 42(2), 2015, pp.323-352.
- Zihayat, M., and An, A. (2014), Mining top-k high utility patterns over data streams, *Information Science*, 285, 2014, pp. 138 - 161.

ภาคผนวก

Mining High-Utility Itemsets with Irregular Occurrence

Supachai Laoviboon*, Komate Amphawan†

Computational Innovation Laboratory, Faculty of Informatics, Burapha University, Chonburi, 20131, Thailand

Email: *supachai.lwb@gmail.com, †komate@gmail.com

Abstract—High-utility itemsets mining (*HUIM*) is proposed to discover itemsets giving high utilities (such as high profit, low cost/risk and other factors). This can help to extract hidden-knowledge from buying behavior of customers. However, *HUIM* may not sufficiently give hidden-knowledge and observe occurrence behavior of itemsets in some applications, since it only considers utilities of items/itemsets. Thus, we propose to mine high utility itemsets with irregular occurrence (also called *High Utility-Irregular Itemsets*, *HUIIs*). *HUIIs* can help to gain knowledge about “products giving high profits even if customers do not regularly purchase them together” and to improve marketing strategies and sale profit. To mine *HUIIs*, an efficient single-pass algorithm based on the use of new modified utility-list structure, called *HUIIM (HUIIs-Miner)*, is designed. Experiments on real and synthetic datasets were done to investigate computational time and memory consumption of *HUIIM*.

Keywords—data mining; association rules; high utility mining; regular/irregular itemsets; irregular occurrence

I. INTRODUCTION

Association rule mining (*ARM*) and frequent itemsets mining (*FIM*) are fundamental concepts in data mining and widely applied in several applications. For example, finding set of frequent-sold products in retails can help to create new promotions, advertisement, manage layout of shelves and warehouse, etc. Moreover, *ARM* and *FIM* can be applied in medical analysis, weblog analysis, mobile commerce, elderly habit monitoring, and so on. However, traditional *FIM* only considers occurrence of an item in a transaction in binary manner (*i.e.* consider only whether an item occurs in a transaction or not) and it does not consider importance of items which can be different in specific applications that desire to discover interesting itemsets based on these factors.

To address above issues, Chan et al. [1] proposed to discover interesting itemsets based on considering of unit utility of each item (*e.g.* profit, cost, risk, and other user-defined factors) and the number of occurrences of each item occurring in a transaction. Then, the problem of *high utility itemsets mining (HUIM)* is introduced. *HUIM* can tell about “products that give high utility value based on observation of their unit-utilities and amounts of occurrence”. To mine *HUIIs*, there is a main challenge that is *downward closure property* [2] cannot be held (*i.e.* an itemset X may have utility lower or higher than its superset Y (where $Y = X \cup Z$ and $\forall i_k \in Z | i_k \notin X$). Thus, if X has low utility value, we then cannot disregard X and its supersets from our consideration due to we cannot guarantee that its supersets will have low utility value or not. This causes overwhelm of itemsets to

be considered. To alleviate this problem, two techniques used for estimating utility value (in upper bound manner) of each item/itemset, called *transaction weighted utility (TWU)* [3] and *tight-overestimated utility* [4] are thus proposed. These, technique can help to keep *downward closure property* in which if X has low estimated utility value, all of X 's supersets cannot have high utility value. Thus, X and its supersets can be disregarded from the computation based on estimated utility value of X . From these two techniques, there are approaches for improving performance of *HUIM* such as UP-Growth and UP-growth+ [5], FHM [6], EFIM [7], d²HUP [8], etc.

Recently, there are efforts to discover high utility itemsets with occurrence behaviors investigation. Then, the problem of high utility-regular (periodic) itemsets mining was thus proposed in [9], [10], [11] to mine high utility itemset with regular (periodic) occurrence. However, these approaches consider only regular occurrence behavior which may not sufficient in some applications. Thus, in this paper, we propose to discover a different kind of itemsets under consideration of their utility values and occurrence behavior (in the term of irregular of occurrence). Then, the task of mining high utility itemsets with irregular itemsets (called *High Utility-Irregular Itemsets*, *HUIIs*) is introduced. With *HUIIs*, it can help to know about “sets of products that give high profit even if customers do not regularly purchase them together” and to improve marketing strategy and management for increasing sale amount and profit. For example, “LCD TV and smart digital box” can give high profit even if there are only few customers buy them together. This can help to know behavior of customers and help to manage warehouse in order to avoid depreciation of these product. To mine *HUIIs*, an efficient single-pass algorithm based on the use of new modified utility-list structure, called *HUIIM (HUIIs-Miner)*, is designed. Experimental study on real and synthetic datasets were done and show efficiency of the proposed *HUIIM* in the terms of computational time and memory consumption.

II. PROBLEM DEFINITIONS

Let $I = \{i_1, i_2, \dots, i_n\}$ be a set of items. Each item $i_j \in I$ has its own unit utility expressing profit, cost, risk and other user-defined factors, called *external utility* (denoted as $eu(i_j)$). A set $X = \{i_j, \dots, i_k\} \subseteq I$ is called a k -itemset, if X contains k items. A transactional database $D = \{t_1, t_2, \dots, t_m\}$ contains a set of m transactions in which each transaction $t_p \in D$ has (i) a unique transaction identifier p (called *tid* in short) and (ii) a set of items $Y \subseteq I$. Each $i_j \in Y$ is associated with a positive integer expressing quantity of its own occurrence in

t_p , called *internal utility* (denoted as $iu(i_j, t_p)$). For an itemset X , if $X \subseteq Y$ of t_p , it can be said that t_p contains X or X occurs in transaction t_p , denoted as p^X . From the occurrence of X , the set $T^X = \{p^X, \dots, q^X\}$ is the ordered set w.r.t. *tids* of transactions that contain X .

TABLE I: External utility of items.

item	a	b	c	d	e	f	g	h
external utility	2	3	4	20	2	25	5	3

TABLE II: A transaction database.

<i>tid</i>	items(internal utility)
t_1	a(2), b(3), d(13), f(2)
t_2	c(2), e(3), h(4)
t_3	a(3), b(2)
t_4	a(3), f(1), g(1)
t_5	a(1), b(2), d(1),
t_6	c(20)
t_7	f(1), g(2), h(8)
t_8	a(1), b(1), f(1), g(1)

A. Utility of an item/itemset

Definition 1: The utility value of an item i_j in a transaction t_p is the multiplication of unit utility and quantity of i_j 's occurrence in t_p , denoted as $u(i_j, t_p) = iu(i_j, t_p) \times eu(i_j)$.

Definition 2: The utility value of an itemset $X = \{i_j, \dots, i_k\}$ in a transaction t_p is the summation of all utility value of all items in X occurring in transaction t_p , denoted as $u(X, t_p) = \sum_{i_j \in X \in t_p} eu(i_j) \times iu(i_j, t_p)$.

Definition 3: The utility value of an itemset X in a database D is the summation of utility of X occurring in transactions of D , denoted as $u(X) = \sum_{X \in t_p \in D} u(X, t_p)$.

Definition 4: The transaction utility of a transaction t_p is the summation of utility values of all items occurring in transaction t_p , denoted as $tu(t_p) = \sum_{i_j \in t_p} u(i_j, t_p)$.

Definition 5: The transaction-weighted utility of an itemset X in a database D is the summation of utility of transactions in D containing X , defined as $TWU(X) = \sum_{t_p, t_p \in D} tu(t_p)$.

Property 1: As in [3], if $TWU(X)$ is less than a give utility threshold (σ_u), all superset of X are not high utility.

Definition 6: Let \prec be the order of all items $\in I$. The remaining utility of an itemset X in a transaction t_p is the summation of utility values of all items ordered after X , defined as $ru(X, t_p) = \sum_{i_j \in t_p, X \prec i_j} u(i_j, t_p)$.

Definition 7: The remaining utility of an itemset X in database D is the summation of all remaining utility value of X in all transactions containing X , defined as $ru(X) = \sum_{X \in t_p \in D} ru(X, t_p)$.

Definition 8: The tight over-estimated utility of an itemset X in database D is the summation between the utility (actual) and the remaining utility of X in database D , defined as $tou(X) = u(X) + ru(X)$.

Property 2: As in [4], if $tou(X)$ is less than a give utility threshold (σ_u), all itemsets with $X.Y$ (the extension of X with other itemsets ordered after X) are not high utility.

B. Regularity of an item/itemset

Definition 9: The regularity of an itemset X before its first occurrence in a transaction t_p is the gap of absence of

X between the first transaction t_1 in database and the first occurrence of X in transaction t_p , defined as $fr(X, t_p) = p$.

Definition 10: The regularity of an itemset X between two consecutive occurrence of X in transactions t_p and t_q (where $p < q$) is the gap of occurrence of X between t_p and t_q , defined as $r(X, t_p, t_q) = q - p$.

Definition 11: The regularity of an itemset X after its last occurrence in a transaction t_z is the gap of absence from the last occurrence of X in t_z to the last transaction t_m of database, defined as $lr(X, t_z) = m - z$.

Definition 12: The regularity of an itemset X in a database D is the maximal gap of absence based on its own occurrence in database D , defined as $r(X) = \max(fr(X, t_p), r(X, t_p, t_q), \dots, r(X, t_y, t_z), lr(X, t_z))$.

Definition 13: An itemset X is called a regular-itemset, if its regularity ($r(X)$) is no greater than a user-given regularity threshold (σ_r). Otherwise, it is called an irregular itemset.

Definition 14: An itemset X is called a high utility-irregular itemset, if i) its utility value ($u(X)$) is no less than a user-specified utility threshold (σ_u) and ii) its regularity ($r(X)$) is greater than a user-given regularity threshold (σ_r).

Problem statement. Given a transactional database D with external utilities of items, a regularity threshold (σ_r) and a utility threshold (σ_u). The problem of mining high utility-irregular itemsets to discover a complete set of high utility-irregular itemsets having utilities no less than a user-specified utility threshold, and regularities greater than a user-given regularity threshold, respectively.

III. PROPOSED METHOD : HUIIM

In this section, we here present an efficient single-pass algorithm named *HUIIM* for mining high utility-irregular itemsets. *HUIIM* scan database once to capture essential information of each item occurring in each transaction (*i.e.* transaction-id and utility value). A new modified utility list structure (NUL) [11] is utilized to efficiently maintain essential information and compute utility and regularity of an item/itemset. The concepts of transaction weighted utility (TWU) [3], remaining utility [4], tight overestimated utility [4], and analysis of item co-occurrences technique [6] are applied to efficiently prune search space. *HUIIM* consists of two main steps: i) *HUIIM-ScanningDatabase*—scanning of database to create a list of single item and collect transaction-ids (also simultaneously with utility value) containing each item, and ii) *HUIIM-MiningHUIIs*—mining complete set of high utility-irregular itemsets from informations collected in the first step.

A. New-modified Utility List Structure

A new-modified utility list structure (NUL) [11] is an extension of utility list structure [4] used for maintaining occurrence information (simultaneously with utility values) of each item/itemset. For an itemset X , its NUL can be represented as an ordered set of 4-tuples, denoted as $NUL^X = \{e_1, e_2, \dots, e_k\}$ where each entry $e_j = \langle p, u(X, t_p), ru(X, t_p), up(X, t_p) \rangle$ contains i) the transaction id of transaction t_p containing X , ii) utility of X in transaction t_p , iii) remaining utility of items ordered after X in t_p , and iv) utility of prefix items of X in t_p (Notice: if $X = \{i_p, \dots, i_q, i_r\}$,

the prefix of X in this context is all items in X except the last item i_r , respectively.

Example 1: From external utilities in Table I and the transactional database in Table II, item ‘a’ occurs in transactions t_1, t_3, t_4, t_5 and t_8 , respectively. Then, NUL^a of ‘a’ will have 5 entries as its number occurrence in database i.e. $\{ \langle 1, 4, 319, 0 \rangle, \langle 3, 6, 6, 0 \rangle, \langle 4, 6, 30, 0 \rangle, \langle 5, 2, 26, 0 \rangle, \langle 8, 2, 33, 0 \rangle \}$. The first entry $\langle 1, 4, 319, 0 \rangle$ collects information of the first occurrence of ‘a’ in transaction t_1 which contains 4 elements i.e. 1 is *tid* of transaction t_1 containing ‘a’, 4 is utility of ‘a’ in t_1 , 319 is remaining utility of ‘a’ in t_1 , and 0 is utility of prefix items of ‘a’ in t_1 (for now, ‘a’ is single item and does not have any prefix items), respectively.

B. HUIIM-ScanningDatabase

As detailed in Algo. 1, HUIIM creates a simple-list called *tuList* for maintaining transaction utility of all transactions in database. With *tuList*, HUIIM can avoid repeatedly scan of database which causes HUIIM scans database once. A tree structure, called *HUII-tree*, is initial with child nodes of single items. Each child node is used for maintaining a single item with their essential information (i.e. its utility, remaining utility, transaction weighted utility, regularity, and *NUL*).

Next, each transaction t_p of database is sequentially scanned to compute and collect transaction utility $tu(t_p)$ into *tuList*. Then, each item i_j occurring in transaction t_p is considered and then its regularity i_j is calculated and updated. The utility of i_j in transaction t_p is computed and a new entry $e = \langle p, u(i_j, t_p), 0, 0 \rangle$ is inserted to NUL^{i_j} . Last, twu^{i_j} of i is updated by $tu(t_p)$, respectively.

After scanning all transactions in database, each item $i_j \in I$ having twu^{i_j} less than the user-given utility threshold is eliminated from *HUII-tree* and our consideration (Notice : this item and all of its supersets cannot give high utility [3]). To remove each item i_j , each entry $e = \langle p, u(i_j, t_p), 0, 0 \rangle$ in NUL^{i_j} is considered and the transaction utility of $tu(t_p)$ (corresponding to *tid* p of the entry e) is decreased by $u(j, t_p)$. This process can help to minimize transaction-weighted utility of other items occurring in the same transaction with item i_j .

After pruning low utility items as above, the remaining utility of each item i_j in each transaction t_p containing i is then calculate and updated in each entry of NUL^{i_j} . The total remaining and actual utilities are then calculated. Last, each item i_j in the *HUII-tree* is identified and collect as a HUII if its regularity is greater than the regularity threshold and its utility is no less than the utility threshold, respectively.

Example 2: Based on the setting of regularity and utility thresholds to be 3 and 40 (i.e. $\sigma_r = 3$ and $\sigma_u = 40$), the task of mining all HUIIs from database of Table I and II is to find all itemsets having utility at least 40 and usually occur at least once in every three consecutive transactions.

Initially, *tuList* and *HUII-tree* with nodes of single items are created and initialized. Next, each transaction in database is sequentially scanned. For example, for the transaction $t_1 = \{a(2), b(3), d(13), f(2)\}$, its transaction utility $tu(t_1)$ is computed by $tu(t_1) = (2 \times 2) + (3 \times 3) + (13 \times 20) + (2 \times 25) = 323$ and then collected in *tuList*. In addition, transaction weighted

utilities and *NULs* of items a, b, d, f and g are updated and their regularities set is to be 1 (as shown in Fig. 1(a) the information contained in each node is ordered as *i*) utility, *ii*) remaining utility, *iii*) *TWU*, *iv*) regularity and *v*) *NUL*, respectively). Next, the transaction $t_2 = \{c(2), e(3), h(4)\}$ is scanned which leads to updating on entries of items c, e and f and $tu(t_2) = 26$ (as in Fig. 1(b)). The transactions t_3-t_8 are also scanned in order to update *tuList* and *HUII-tree*. After scanning all transactions, the entry of item ‘e’ (with $TWU(e) = 26$ which is less than the utility threshold σ_u) is thus eliminated from *HUII-tree*, since item ‘e’ and all of its superset cannot give high utility value. As shown in Fig. 1(c), utility and remaining utility of each item in *HUII-tree* are calculated for further consideration of longer itemsets. Last, the item is thus identify as a HUII, if its utility is no less than σ_u and its regularity is greater than σ_r (it is labeled as green).

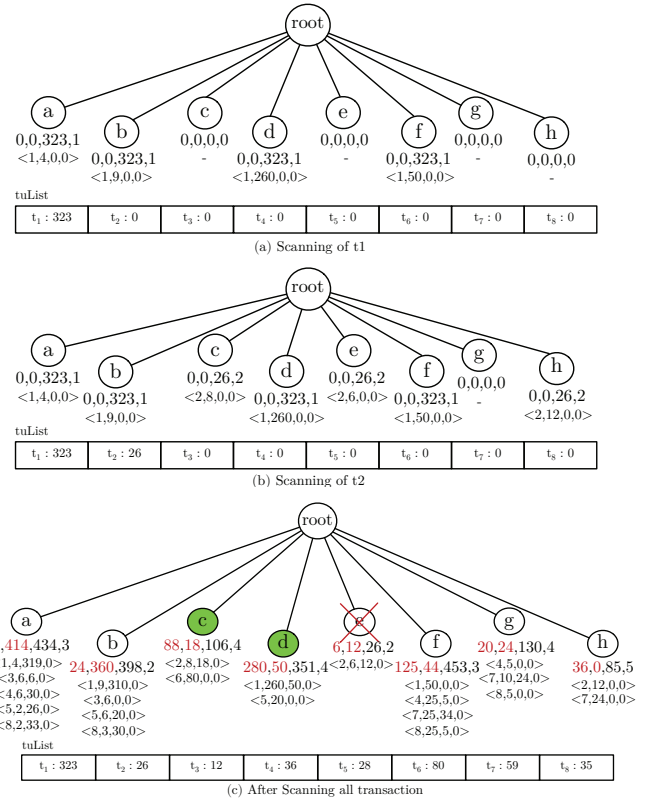


Fig. 1: HUIIM-ScanningDatabase : *tuList* and *HUII-tree*

C. HUIIM-MiningHUIIs

From the HUIIM-ScanningDatabase, *HUII-tree* contains single items in which their transaction-weighted utilities (twu) are not less than a user-given utility threshold. Then, in HUIIM-MiningHUIIs, pairs of items/itemsets in *HUII-tree* are recursively considered to mine a complete set of HUIIs. A breadth-first search strategy is applied to firstly generate all 2-itemsets. These itemsets can use for pruning search space, i.e. if X is a 2-itemset with $twu(X) < \sigma_u$, then all supersets of X cannot have high utility and they can be eliminated from consideration. To generate each 2-itemset, each item i in *HUII-tree* having tight-overestimate utility $tou(i)$ not less than utility threshold is considered (based on property 2) and then merged with another item j in *HUII-tree*. Then, NUL^i and NUL^j are intersected to collected NUL^{ij} and to calculate $u(ij)$, $twu(ij)$,

Algorithm 1 *HUIIM-ScanningDatabase*

Input: D : Transactional database, σ_u : a utility threshold,
 σ_r : a regularity threshold

Output: *HUII-tree*, *HUIIs* contains single HUIIs

```

1: create tuList to hold transaction-utility of all transactions
2: initial root of HUII-tree and create child nodes in which each child node is
   for a single item  $i \in I$ 
3: initial HUIIs to be empty
4: for each trasaction  $t_p$  in  $D$  do
5:   compute  $tu(t_p) \leftarrow \sum_{i \in t_p} u(i, t_p)$  and collect  $tu(t_p)$  at tail of tuList
6:   for each item  $i$  in transaction  $t_p$  do
7:     compute  $r(i) = \max(r(i), p-q)$  where  $q$  is the tid of the last occurrence
       of  $i$  ( $q$  is collected in the last entry in  $NUL^i$ )
8:     compute  $u(i, t_p) \leftarrow eu(i) \times iu(i, t_p)$  and collect an entry  $(p, u(i, t_p), 0, 0)$ 
       at tail of  $NUL^i$ 
9:     update  $twu(i) \leftarrow twu(i) + tu(t_p)$ 
10:  for each child node of root with item  $i$  do
11:    if  $twu(i) < \sigma_u$  then
12:      for each entry  $e = (p, u(i, t_p), 0, 0)$  in  $NUL^i$  do
13:        update  $tu(t_p) \leftarrow tu(t_p) - u(i, t_p)$ 
14:        remove entry of item  $i$  out of siList
15:  for each child node of root with item  $i$  do
16:    for each entry  $e = (p, u(i, t_p), 0, 0)$  in  $NUL^i$  do
17:      update  $tu(t_p) \leftarrow tu(t_p) - u(i, t_p)$ 
18:      set  $ru(i, t_p) \leftarrow tu(t_p)$  and update entry  $e = (p, u(i, t_p), ru(i, t_p), 0)$ 
19:      update  $u(i) \leftarrow u(i) + u(i, t_p)$  and  $ru(i) \leftarrow ru(i) + ru(i, t_p)$ 
20:  if  $u(i) \geq \sigma_u$  and  $r^i < \sigma_r$  then
21:    HUIIs = HUIIs  $\cup$   $i$ 

```

$ru(ij)$, $tou(ij)$ and $r(ij)$, respectively. If $twu(ij)$ is greater than the utility threshold, HUIIM identifies itemset ij as a candidate itemset. Then, its entry is created with its information and assigned to be a child node of item i (Notice it can said that itemset ij has item i as a prefix itemset). Last, if the utility $u(ij)$ is not less than the utility threshold and the regularity $r(ij)$ is greater than the regularity threshold, itemset ij is then identified as HUII and collected in HUIIs. After generating all 2-itemsets, HUIIM-MiningHUIIs continues to generate longer itemsets with size 3 or more by recursively regarding pairs of itemsets contained in HUII-tree.

To mine long itemsets (with size 3 or more), each 2-itemset $X = \{i_j, \dots, i_k\}$ in HUII-Tree is considered. If X has tight overestimated utility ($tou(X)$) not less than the utility threshold, it then merge with another itemset $Y = \{i_j, i_l\}$ having the same prefix as X (i.e. i_j). However, before merging these two itemsets, the last items from each itemset, i.e. i_k from X and i_l from Y , are considered and merged together to be $i_k i_l$. Then, HUII-tree is traversed to find a path of $i_k i_l$ and $twu(i_k i_l)$. If there is no path of $i_k i_l$ in the HUII-tree (means that $twu(i_k i_l) < \sigma_u$), it can say that $i) i_k i_l$ and all of its superset will have twu less than the utility threshold, and $ii) X \cup Y$ and all of its supersets are low-utility itemsets which can disregard from mining process. Otherwise, $X \cup Y$ is candidate itemset which may have high utility value. Then, NUL^{XY} and NUL^{XY} are intersected to collected NUL^{XY} and to calculate $u(XY)$, $twu(XY)$, $ru(XY)$, $tou(XY)$ and $r(XY)$, respectively. If $twu(XY)$ is greater than the utility threshold, XY is then identified as a candidate itemset and its entry is also created with its information and assigned to be a child node of item X . Last, if the utility $u(XY)$ is not less than the utility threshold and the regularity $r(XY)$ is greater than the regularity threshold, itemset XY is then identified as HUII and collected in HUIIs. The process of merging X is repeated until all itemsets having the same prefix as X is considered. Then, HUIIM then recursively mine longer itemsets under consideration of X 's children. After consider considering X , HUIIM moves consideration to another 2-itemset Y and process in the same manner as X . At the end of HUIIM, HUIIs contains a complete set of high utility-irregular itemsets.

Example 3: To mine n -HUIIs (where $n \geq 2$), HUIIM-

MiningHUIIs firstly generates all of 2-HUIIs. To do this, item 'a' is first considered and merged together with items 'b', 'c', 'd', 'f', 'g' and 'h', respectively. For each merging likes a merging of 'a' with 'b', $NUL^a = \{ \langle 1, 4, 319, 0 \rangle, \langle 3, 6, 6, 0 \rangle, \langle 4, 6, 30, 0 \rangle, \langle 5, 2, 26, 0 \rangle, \langle 8, 2, 33, 0 \rangle \}$ and $NUL^b = \{ \langle 1, 9, 310, 0 \rangle, \langle 3, 6, 0, 0 \rangle, \langle 5, 6, 20, 0 \rangle, \langle 8, 3, 30, 0 \rangle \}$ are intersected together in order to compute utility, remaining utility, tight over-estimated utility, transaction weighted utility and regularity value of itemset 'ab' and to collect NUL^{ab} for the further computation i.e. $u(ab) = 13 + 12 + 8 + 5 = 38$, $ru(ab) = 310 + 0 + 20 + 30 = 360$, $tou(ab) = 38 + 360 = 398$, $TWU(ab) = 323 + 12 + 28 + 35 = 398$, $r(ab) = 3$, and $NUL^{ab} = \{ \langle 1, 13, 310, 4 \rangle, \langle 3, 12, 0, 6 \rangle, \langle 5, 8, 20, 2 \rangle, \langle 8, 5, 30, 2 \rangle \}$, respectively. Then, a node of 'a,b' is then created and linked to be a child node of 'a', since its TWU is greater than σ_u (as shown in Fig. 2). Next, NUL^a is intersected with $NUL^c = \{ \langle 2, 8, 12, 0 \rangle, \langle 6, 80, 0, 0 \rangle \}$ in order to form itemset 'a,c'. Unfortunately, items 'a' and 'c' never occur together in database. Then, its TWU ($TWU(a, c) = 0$) is less than σ_u and 'a,c' with all of its supersets are then removed from consideration. The merging process is repeated for item 'a' with items 'd', 'f', 'g', 'h' and also items 'b', 'c', 'd', 'd' and 'g' with items ordered after them. At the end of generating all 2-HUIIs, we gain *HUII-tree* as shown in Fig. 2.

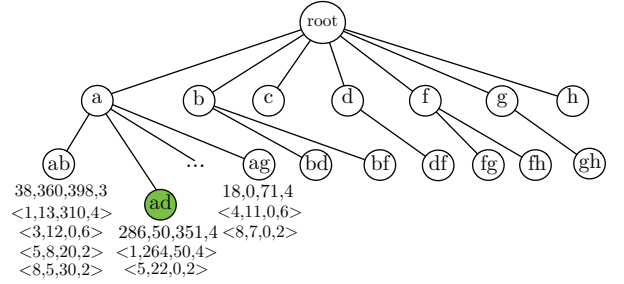


Fig. 2: HUII-tree containing 2-HUIIs

Next, 3(or more)-HUIIs are considered and generated. To do that, item 'a' and the number of its children are considered. Since, 'a' has four children (more than one), then each child is sequentially merged with other children of 'a'. For example, itemset 'a,b' is merged with itemset 'a,d' to generate itemset 'a,b,d'. However, before merging, a path of 'b,d' (the itemset generated from considering on last item of 'a,b' and 'a,d') in the HUII-tree is investigated. Since, there exists a path of 'b,d', then HUIIM-MiningHUIIs cannot identify and eliminate itemset 'a,b,d' from consideration (based on downward closure property [2], [6]). Thus, NUL^{ab} and NUL^{ad} are then intersect to calculate $u(abd) = 301$, $ru(abd) = 50$, $tou(abd) = 351$, $twu(abd) = 351$, $r(abd) = 4$ and to collect $NUL^{abd} = \{ \langle 1, 273, 50, 13 \rangle, \langle 5, 28, 0, 8 \rangle, \}$. Since $TWU(abd) > \sigma_u$, a node of 'a,b,d' is thus created and linked to be a child node of 'a,b'. The itemset 'a,b,d' is identified and collected in HUIIs, due to $u(abd)$ and $r(abd)$ are greater than σ_u and σ_r , respectively.

Next, itemset 'a,b' is thus merge with itemset 'a,f' and then 'a,g' to generate 3-HUIIs with itemset 'a,b' as a prefix. Then, the merging process is recursively performed, if itemset 'a,b' has more than one child. Moreover, this process also repeats for child of item 'b' and 'f', since they have more than one child. In the end, all HUIIs are contained in HUIIs.

Algorithm 2: HUIIM-MiningHUIIs

Input: *HUII-tree*, σ_u : a utility threshold, σ_r : a regularity threshold
Output: *HUIIs* contains a complete set of *HUIIs*

```

1: for each item  $i$  in HUII-tree do
2:   if  $\text{tou}(i) \geq \sigma_u$  then
3:     for item  $j$  in HUII-tree (where  $i < j$ ) do
4:        $NUL^{ij} \leftarrow \emptyset$ 
5:       IntersectNUL ( $NUL^{ij}$ ,  $NUL^i$ ,  $NUL^j$ )
6:       calculate  $r(ij)$ ,  $\text{twu}(ij)$ ,  $u(ij)$ ,  $ru(ij)$ ,  $up(ij)$  from  $NUL^{ij}$ 
7:       calculate  $\text{tou}(ij) \leftarrow u(ij) + ru(ij) - up(ij)$ 
8:       if  $\text{twu}(ij) \geq \sigma_u$  then
9:         create a node of itemset  $ij$  with  $ps(ij)$ ,  $\text{twu}(ij)$ ,  $u(ij)$ ,  $ru(ij)$ ,  $up(ij)$ 
          and  $NUL^{ij}$  and set node of  $ij$  to be a child of item  $i$ 
10:        if  $u(ij) \geq \sigma_u$  and  $r(ij) > \sigma_r$  then
11:           $HUIIs \leftarrow HUIIs \cup ij$ 
12: for each item  $i$  in HUII-tree do
13:   if item  $i$  has more than one child then
14:     MiningAllHUII(HUII-tree, node of  $i$ ,  $\sigma_u$ ,  $\sigma_r$ )
15: Procedure MiningAllHUII(HUII-tree, node of  $X$ ,  $\sigma_u$ ,  $\sigma_r$ )
16: for child node of  $X$  with itemset  $Y$  do
17:   if  $\text{tou}(Y) \geq \sigma_u$  then
18:      $l_Y \leftarrow$  the last item in itemset  $Y$ 
19:     for child node of  $X$  with itemset  $Z$  do
20:        $l_Z \leftarrow$  the last item in itemset  $Z$ 
21:       if there is a path of itemset  $l_Y, l_Z$  in HUII-tree then
22:         IntersectNUL ( $NUL^{YZ}$ ,  $NUL^Y$ ,  $NUL^Z$ )
23:         calculate  $r(YZ)$ ,  $\text{twu}(YZ)$ ,  $u(YZ)$ ,  $ru(YZ)$ ,  $up(YZ)$  from  $NUL^{YZ}$ 
24:         calculate  $\text{tou}(YZ) \leftarrow u(YZ) + ru(YZ) - up(YZ)$ 
25:         if  $\text{twu}(YZ) \geq \sigma_u$  then
26:           create a node of itemset  $YZ$  with  $ps(YZ)$ ,  $\text{twu}(YZ)$ ,  $u(YZ)$ ,
             $ru(YZ)$ ,  $up(YZ)$  and  $NUL^{YZ}$  and set node of  $YZ$  to be a child
            of itemset  $X$ 
27:           if  $u(YZ) \geq \sigma_u$  and  $r(YZ) > \sigma_r$  then
28:              $HUIIs \leftarrow HUIIs \cup YZ$ 
29:         else
30:           initial  $tuList$  with all entries to be 0
31:           for each entry  $e = (p, u(YZ, t_p), ru(YZ, t_p), up(YZ, t_p))$  in
             $NUL^{YZ}$  do
32:             set  $tu(t_p) \leftarrow u(YZ, t_p)$ 
33:           for each child node of  $X$  with itemset  $Q$  do
34:             set  $\text{twu}(Q) \leftarrow 0$  and set  $ru(Q) \leftarrow 0$ 
35:             for each entry  $e = (p, u(Q, t_p), ru(Q, t_p), up(Q, t_p))$  in
             $NUL^Q$  do
36:               update  $ru(Q, t_p)$  in the entry  $e$ ,
                 $ru(Q, t_p) \leftarrow ru(Q, t_p) - tu(t_p)$ 
37:                $\text{twu}(Q) \leftarrow \text{twu}(Q) + u(Q, t_p) + ru(Q, t_p) + up(Q, t_p)$ 
38:                $ru(Q) \leftarrow ru(Q) + ru(Q, t_p)$ 
39:               update  $\text{tou}(Q) \leftarrow u(Q) + ru(Q)$ 

```

IV. EXPERIMENTAL STUDY

In this section, we performed experimental study on HUIIM for mining HUIIs. From best of our knowledge, there is no effort to mine high utility-irregular itemsets. We then only make a comparative study between *MHUII*, *MHUIRA-UL* and *MHUIRA-NUL* (for mining high utility-regular itemsets) and *HUI-Miner-reg* (for mining high utility itemsets with considering regularity of occurrence) in order to show performance of algorithms on two approaches.

As in Table III, four benchmark datasets from [12] are considered. *HUIIM* is implemented in C and experiments were performed run on a mac mini with OS X Sierra, CPU speed at 2.4 GHz and 8 GB of memory. Two experiments setting, *i.e.* *i*) fixing utility threshold on a variation of regularity threshold and *ii*) fixing regularity threshold on a variation of utility threshold, are designed to investigate computational time and memory usage under variation of regularity and/or utility thresholds. The utility and regularity thresholds are set in the same manner as in [13], [14], [11] which ranging between 0.1 – 1.0% and 1 – 10%, respectively.

TABLE III: Datasets characteristics

Dataset	No. of items	Avg. transactions size	No. of transactions
Chess	75	37	3,196
Foodmart2000	1,559	11	36,869
Mushroom	119	23	8,124
Retail	16,469	10.3	88,162

The computational time of *HUIIM* on the variation of regularity (with a fixed utility threshold) are illustrated in Fig. 3. With the variation of regularity threshold, we can observe runtime slightly increases as the threshold increases. With high regularity threshold, items/itemsets have more chance to meet the threshold. Then, *HUIIM* have to take more time to consider more items/itemsets. From Fig. 3, the running time of *HUIIM* on Chess is quite stable and much less than other algorithms for mining high utility-regular itemsets. It is because Chess is a dense dataset and *HUIIM* can take advantage from analysis of item co-occurrences technique (as in [6]) to early prune low utility itemsets. Otherwise, the runtime of *HUIIM* on Foodmart2000, Mushroom, and Retail dataset (which are sparse datasets) are higher than other algorithms. The reason is that on sparse datasets, items/itemsets usually have low utility but have high *TWU* then there is a large amount of itemsets having *TWU* satisfies the utility threshold and having regularity higher than the regularity threshold.

Meanwhile, the variation of utility threshold (with a fixed regularity threshold) causing the fluctuation on computational time (see Fig. 4). With low utility threshold, there is a large amount of candidate itemsets having transaction weighted utility greater than the utility threshold. Then, *HUIIM* have to take time to consider these candidate itemsets which causing high computational time. On the other hand, with high utility threshold, candidate itemsets can be pruned since their *TWU* and/or *tou* are not meet the utility threshold (Thanks to *downward closure property* from [3], [4]).

For memory usage, we also observe on the highest peak memory usage during mining process. Similarly with the computational time, the memory on variation of regularity threshold is quite constant, but it is unstable on variation of utility threshold (as shown in Fig. 5 and 6).

V. CONCLUSION

In this paper, we have introduced to discover itemsets having high utility and irregular occurrence. This kind of itemsets can let us know about “sets of products that give high profit even if customers are not regularly purchase”. It also can help to create marketing strategy, manage inventory and so on. To mine these itemsets, an efficient single pass named *HUIIM* is proposed. *HUIIM* applied concepts of transaction weighted utility, remaining utility and tight overestimated utility to early filter uninteresting itemsets. It also applied *NUL* (New modified Utility List structure) to maintain occurrence information simultaneously with utility value of each itemset. An extensive experimental study was done on synthetic and real dataset and shew that our proposed *HUIIM* is runtime and memory efficient on mining high utility-irregular itemsets.

VI. ACKNOWLEDGEMENTS

This work was financially supported by a research grant of Burapha University through the National Research Council of Thailand (Grant No. 65/2560).

REFERENCES

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

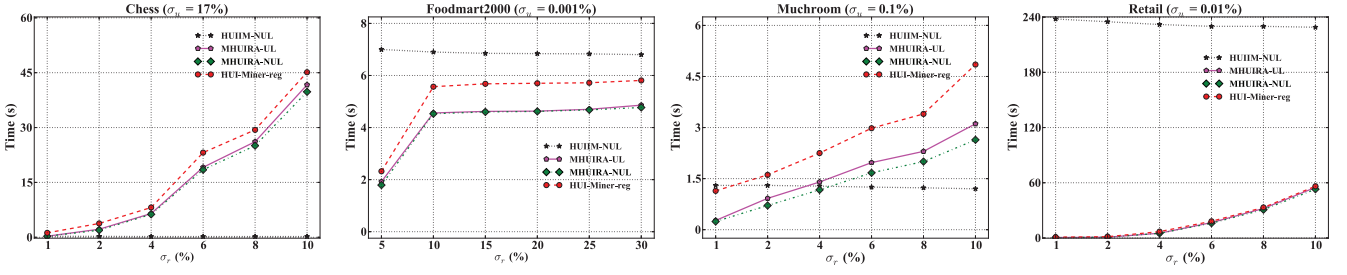


Fig. 3: Runtime with variation of regularity threshold

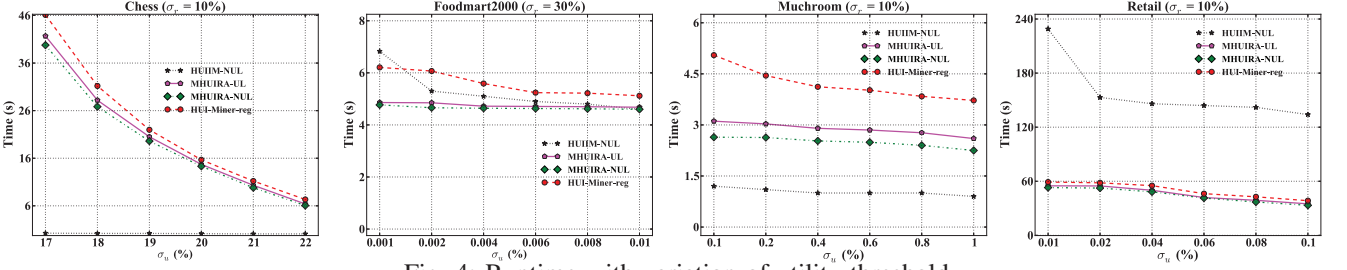


Fig. 4: Runtime with variation of utility threshold

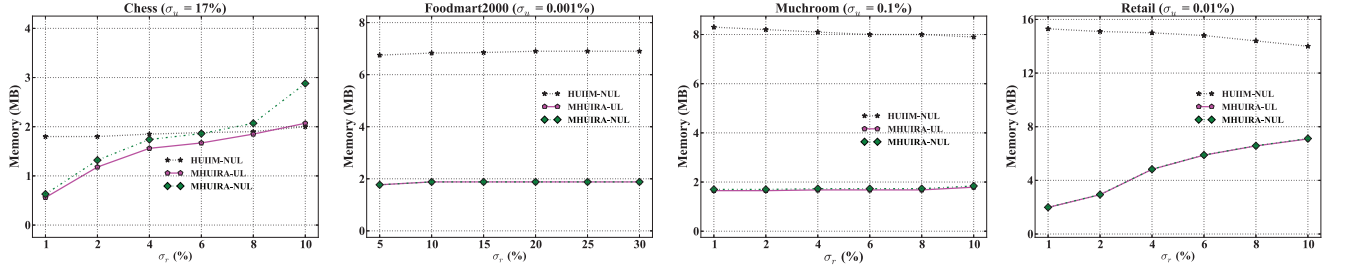


Fig. 5: Memory usage with variation of regularity threshold

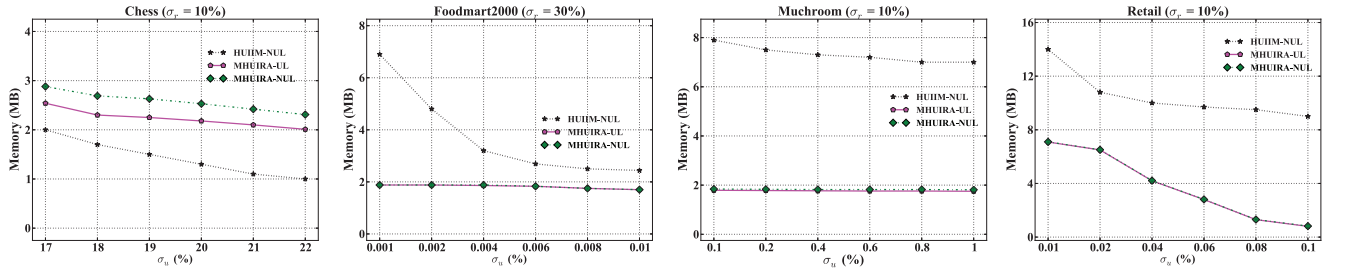


Fig. 6: Memory usage with variation of utility threshold

- [3] Y. Liu, W.-k. Liao, and A. Choudhary, "A two-phase algorithm for fast discovery of high utility itemsets," in *Advances in Knowledge Discovery and Data Mining*, 2005, vol. 3518, pp. 689–695.
- [4] M. Liu and J. Qu, "Mining high utility itemsets without candidate generation," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 2012, pp. 55–64.
- [5] V. S. Tseng, B. E. Shie, C. W. Wu, and P. S. Yu, "Efficient algorithms for mining high utility itemsets from transactional databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 8, pp. 1772–1786, 2013.
- [6] P. Fournier-Viger, C.-W. Wu, S. Zida, and V. S. Tseng, *FHM: Faster High-Utility Itemset Mining Using Estimated Utility Co-occurrence Pruning*, 2014, pp. 83–92.
- [7] S. Zida, P. Fournier-Viger, J. C.-W. Lin, C.-W. Wu, and V. S. Tseng, "Efim: a fast and memory efficient algorithm for high-utility itemset mining," *Knowledge and Information Systems*, pp. 1–31, 2016.
- [8] J. Liu, K. Wang, and B. C. M. Fung, "Mining high utility patterns in one phase without generating candidates," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 5, pp. 1245–1257, 2016.
- [9] K. Amphawan and A. Surarerks, "Pushing regularity constraint on high utility itemsets mining," in *Advanced Informatics: Concepts, Theory and Applications, 2015 2nd International Conference on*, 2015, pp. 1–6.
- [10] P. Fournier-Viger, J. C.-W. Lin, Q.-H. Duong, and T.-L. Dam, *PHM: Mining Periodic High-Utility Itemsets*, 2016, pp. 64–79.
- [11] K. Amphawan, P. Lenca, A. Jitpattanakul, and A. Surarerks, "Mining high utility itemsets with regular occurrence," *Journal of ICT Research and Applications*, vol. 10, no. 2, pp. 153–176, 2016.
- [12] P. F. Viger, "SPMF: An Open-Source Data Mining Library," <http://www.philippe-fournier-viger.com/spmf/>, 2015.
- [13] 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.
- [14] 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.