การค้นหารูปแบบที่ให้ค่าคุณประโยชน์สูงโดยปรากฏอย่างไม่สม่ำเสมอ

ศุภชัย เล้าวิบูลย์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการสารสนเทศ
คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา
มิถุนายน 2561

Mining High-Utility Itemsets with Irregular Occurrence

SUPACHAI LAOVIBOON

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

คณะกรรมการควบคุมวิทยานิพนธ์และคณะกรรมการสอบวิทยานิพนธ์ ได้พิจารณาวิทยานิพนธ์ของ ศุภชัย เล้าวิบูลย์ ฉบับนี้แล้ว เห็นสมควรรับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการสารสนเทศ ของมหาวิทยาลัยบูรพาได้

คณะกรรมการควบคุมวิทยานิพนธ์

..... อาจารย์ที่ปรึกษาหลัก
(ผู้ช่วยศาสตราจารย์ ดร.โกเมศ อัมพวัน)

คณะกรรมการสอบวิทยานิพนธ์

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

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

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

คณะวิทยาการสารสนเทศ อนุมัติให้รับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาการสารสนเทศ สาขาวิชาวิทยาการสารสนเทศของมหาวิทยาลัยบูรพา

..... คณะบดีคณะวิทยาการสารสนเทศ
(ผู้ช่วยศาสตราจารย์ ดร.กฤษณะ ชินสาร)
วันที่ 23 เดือน มิถุนายน พ.ศ.2561

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดีด้วยความกรุณาจาก ผู้ช่วยศาสตราจารย์ ดร. โกเมศ อัมพวัน อาจารย์ที่ปรึกษาที่ได้ให้คำเสนอแนะ แนวคิด ตลอดจนแก้ไขข้อบกพร่องต่างๆ มาโดยตลอด และคณาจารย์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

ขอขอบคุณ คุณพ่อ คุณแม่ คุณครูศุกร์เกษม ปรุงผล และเพื่อนๆ น้องๆ ที่ช่วยเหลือในการให้คำแนะนำและให้กำลังใจในการดำเนินงานวิทยานิพนธ์ตลอดสามปีที่ผ่านมาทำให้ผู้วิจัยประสบความสำเร็จในการศึกษาด้วยดี

วิทยานิพนธ์นี้ได้รับทุนอุดหนุนวิทยานิพนธ์จากสำนักงานคณะกรรมการวิจัยแห่งชาติ ปีงบประมาณ ๒๕๕๙

ศุภชัย เล้าวิบูลย์

58910113: สาขาวิชาวิทยาการสารสนเทศ; วท.ม.(วิทยาการสารสนเทศ)
คำสำคัญ: การวิเคราะห์พฤติกรรมผู้บริโภค การวิเคราะห์ข้อมูลการซื้อสินค้า การวิเคราะห์ข้อมูลเพื่อสนับสนุนการตัดสินใจ การทำเหมืองข้อมูล การค้นหารูปแบบการพิจารณาคุณประโยชน์เงื่อนไขความสม่ำเสมอ

ศุภชัย เล้าวิบูลย์: การค้นหารูปแบบที่ให้ค่าคุณประโยชน์สูงโดยปรากฏอย่างไม่สม่ำเสมอ คณะกรรมการควบคุมวิทยานิพนธ์: ผู้ช่วยศาสตราจารย์ ดร.โกเมศ อัมพวัน, Ph.D., 95 หน้า. ปี พ.ศ. 2561

การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงเป็นหัวข้องานวิจัยหนึ่ง ภายใต้การทำเหมืองข้อมูลที่น่าสนใจ การค้นหารูปแบบดังกล่าวสามารถประยุกต์ใช้อย่างแพร่หลาย ตัวอย่างเช่น การประยุกต์ใช้ในธุรกิจค้าปลีก เพื่อทำการค้นหารูปแบบของสินค้าที่ถูกซื้อจากลูกค้า โดยรูปแบบของสินค้าดังกล่าวจะเป็นรายการสินค้าต่างๆ ที่ถูกซื้อร่วมกันที่จะให้ผลตอบแทนที่สูง เป็นต้น แต่อย่างไรก็ตาม การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงจะทำการพิจารณาเพียงแค่ค่าคุณประโยชน์ของรายการต่างๆ เท่านั้น ที่ซึ่งการค้นหารูปแบบดังกล่าวอาจไม่เพียงพอต่อการสังเกตถึงพฤติกรรมการซื้อสินค้าของผู้บริโภค ด้วยเหตุนี้ งานวิจัยจึงมุ่งเน้นที่จะทำการเพิ่มเติมเงื่อนไขการพิจารณารูปแบบโดยจะทำการเพิ่มเติมเงื่อนไขของการปรากฏอย่างไม่สม่ำเสมอร่วมกับการพิจารณาค่าคุณประโยชน์ของรูปแบบต่างๆ ภายใต้แนวคิดใหม่ข้างต้น รูปแบบที่น่าสนใจจะเป็นรูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏขึ้นในชุดข้อมูลอย่างไม่สม่ำเสมอ

ในการค้นหารูปแบบดังกล่าว ผู้วิจัยได้เสนอขั้นตอนวิธีที่มีประสิทธิภาพที่ชื่อว่า “*Mining High-Utility Itemsets with Irregular Occurrence*, *HUIIM*” ซึ่งจะทำการอ่านข้อมูลจากฐานข้อมูลเพียงครั้งเดียว และทำการปรับปรุงโครงสร้างการเก็บข้อมูล *“New modified utility-list, NUL”* เพื่อทำการจัดเก็บข้อมูลการปรากฏขึ้นและค่าคุณประโยชน์ของเซตรายการหนึ่งๆ ให้มีประสิทธิภาพและได้ประยุกต์ใช้แนวความคิดเกี่ยวกับค่าประมาณคุณประโยชน์ ค่าคุณประโยชน์คงเหลือ ค่าประมาณคุณประโยชน์แบบกระชับ เพื่อทำการลดทอนปริภูมิสถานะของการค้นหาเซตรายการ นอกจากนี้เทคนิคใหม่ที่เรียกว่า “*Efficient Pruning technique”* ถูกออกแบบมาใช้กับ *HUIIM* เพื่อเพิ่มประสิทธิภาพในการคำนวณและเรียกขั้นตอนวิธี *HUIIM* ที่มีการประยุกต์ใช้ “*Efficient Pruning technique”* ว่า “*Efficient High Utility Itemsets with Irregular occurrence Miner, EHUIIM”* โดยในการทดสอบประสิทธิภาพของขั้นตอนวิธีที่นำเสนอ เราจะสังเกตได้ว่าขั้นตอนวิธีที่นำเสนอสามารถค้นหารูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอได้อย่างมีประสิทธิภาพ

58910113: MAJOR: INFORMATICS; M.Sc. (Informatics)
KEYWORD: Customers' behavior analysis, Buying behavior analysis,
Decision support data analysis, Data mining, Pattern mining,
Utility mining, Regular constraint
SUPACHAI LAOVIBOON: Mining High-Utility Itemsets with Irregular Occurrence
THESIS ADVISSOR: KOMATE AMPHAWAN, Ph.D, 95 P. 2018

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.

In this task, itemsets having high utility and irregular occurrence are identified as interesting. To mine such itemsets, we have introduced an efficient algorithm named *"Mining High-Utility Itemsets with Irregular Occurrence*, *HUIIM"* scans database once to capture occurrence information and utility value of single items into the "*new-modified utility list* structure, *NUL"*. The concept *"transaction weighted utility, TWU"* and *remaining utility* and "*tight over- estimated utility, tou"* of an item/itemset are utilized to prune search space. Moreover, a new pruning technique based on the above concepts is designed and applied to *HUIIM* in order to identify low utility items leading to quickly cutting down of search space (also called *EHUIIM* "*Efficient High Utility Itemset with Irregular occurrence Miner"*). Experimental studies are conducted to investigate performance of the proposed methods and the results show that with the new pruning technique, *HUIIM* can effectively mine high utility itemsets with irregular occurrence.

สารบัญ

หน้า

สารบัญ(ต่อ)

หน้า

สารบัญภาพ

หน้า

ภาพที่

สารบัญภาพ(ต่อ)

หน้า

ภาพที่

สารบัญภาพ(ต่อ)

หน้า

ภาพที่

สารบัญภาพ(ต่อ)

หน้า

ภาพที่

สารบัญภาพ(ต่อ)

หน้า

ภาพที่

สารบัญตาราง

หน้า

ตารางที่

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในยุคปัจจุบันเป็นยุคของข้อมูลข่าวสารและสารสนเทศ ที่มีการเปิดเผยอย่างแพร่หลาย ประกอบกับปัจจุบันมีแอพพลิเคชันหรือโปรแกรมที่ทำการจัดเก็บข้อมูลได้หลากหลาย ดังนั้นจึงเป็นเหตุให้ธุรกิจต่างๆ ทั้งธุรกิจ SME ธุรกิจขนาดกลาง และธุรกิจขนาดใหญ่มีการประยุกต์ใช้ข้อมูลข่าวสาร สารสนเทศ และองค์ความรู้ต่างๆ ในการประกอบการตัดสินใจเกี่ยวกับการดำเนินธุรกิจมากยิ่งขึ้น จากการประยุกต์ใช้งานข้อมูลดังกล่าว วิธีการวิเคราะห์ทั้งในเชิงปริมาณและคุณภาพที่เหมาะสมกับธุรกิจต่างๆ ได้ถูกคิดค้นโดยนักวิจัยในหลายแขนงสาขา แต่อย่างไรก็ตามมีแนวคิดหนึ่งที่ถูกประยุกต์ใช้อย่างแพร่หลายคือ การวิเคราะห์พฤติกรรมผู้บริโภคด้วยการค้นหารูปแบบที่ปรากฏบ่อย (Frequent Itemset Mining, FIM) ที่สามารถบ่งบอกถึงสิ่งของหรือเหตุการณ์ที่ปรากฏขึ้นพร้อมกันบ่อยๆ ตัวอย่างเช่น ในธุรกิจห้างสรรพสินค้าหรือธุรกิจค้าปลีกจะทำการหาความสัมพันธ์ของรายการสินค้าที่ถูกซื้อร่วมกันบ่อยๆ เพื่อช่วยในการจัดทำโปรโมชันสินค้า ช่วยในการจัดชั้นวางสินค้าให้สินค้าที่ถูกซื้อร่วมกันบ่อยๆ อยู่ในพื้นที่ใกล้ๆกัน เพื่ออำนวยความสะดวกให้แก่ลูกค้าและยังช่วยกระตุ้นการจับจ่ายใช้สอยของลูกค้า นอกจากนี้รูปแบบที่ปรากฏบ่อยยังช่วยในการจัดทำแคตตาล็อกสินค้าให้สินค้าที่ถูกซื้อร่วมกันบ่อยๆ อยู่ใกล้ๆ กัน ซึ่งการดำเนินการทั้งหมดนี้จะช่วยให้ห้างสรรพสินค้ารักษาฐานลูกค้าให้ยังคงซื้อสินค้ากับห้างสรรพสินค้าต่อไปได้

แนวความคิดเบื้องต้นของการค้นหารูปแบบที่ปรากฏบ่อยจะประยุกต์ใช้ค่าสนับสนุน (ค่าความถี่หรือจำนวนครั้งในการเกิดขึ้นของรูปแบบนั้นๆ) เป็นตัวชี้วัดความสำคัญหรือความน่าสนใจของรูปแบบ แต่อย่างไรก็ตามการใช้เพียงค่าสนับสนุนอาจจะไม่เพียงพอต่อการค้นหารูปแบบที่มีความหลากหลาย โดยแนวความคิดนี้ถูกพัฒนาอย่างต่อเนื่องในหลายๆ แง่มุม อาทิเช่น การค้นหารูปแบบที่ปรากฏบ่อยแบบเรียงลำดับ (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) และอื่นๆ

จากงานวิจัยข้างต้นจะมีหัวข้องานวิจัยหนึ่งที่ทำการพิจารณาค่าคุณประโยชน์ของรูปแบบ (Utility of patterns) ซึ่งค่าคุณประโยชน์ของรูปแบบอาจหมายถึง ผลกำไรที่ได้รับจากการขายสินค้าชิ้นหนึ่งๆ ของรายการสินค้าหนึ่งๆ หรือการบริการหนึ่งๆ เมื่อเราทำการคำนวณค่าคุณประโยชน์ทั้งหมดของรายการสินค้าจะทำให้เราสามารถทราบได้ถึงจำนวนผลกำไรหรือขาดทุนที่ได้จากรายการสินค้านั้นๆ และยังทราบถึงรายการสินค้าที่ให้ค่าคุณประโยชน์สูง แต่อย่างไรก็ดีการค้นหา

รูปแบบที่มีค่าคุณประโยชน์สูงไม่ได้ทำการพิจารณาถึงพฤติกรรมการปรากฏขึ้นของรูปแบบหรือรายการนั้นๆ ว่ามีพฤติกรรมการปรากฏขึ้นอย่างสม่ำเสมอ ไม่สม่ำเสมอ สม่ำเสมอในบางช่วงเวลาหรือไม่

จากปัญหาข้างต้นจึงได้มีนักวิจัยคิดค้นวิธีการค้นหารูปแบบที่มีคุณประโยชน์สูงที่ปรากฏขึ้นอย่างสม่ำเสมอ (Amphawan, K. และ Surarerks, A., 2015-b) ได้เพิ่มเติมเงื่อนไขการพิจารณารูปแบบ โดยทำการเพิ่มการพิจารณาการปรากฏอย่างสม่ำเสมอร่วมกับการพิจารณาค่าคุณประโยชน์ของรายการต่างๆ ซึ่งจะทำให้เราทราบถึงรูปแบบที่มีค่าคุณประโยชน์สูง และมีพฤติกรรมการปรากฏขึ้นอย่างสม่ำเสมอ ที่ซึ่งอาจช่วยให้เราทราบถึงช่วงเวลาที่รายการสินค้าได้รับความนิยม นำมาซึ่งการจัดทำโปรโมชันให้กับรายการสินค้าที่พิจารณาเพื่อช่วยกระตุ้นยอดขายและอื่นๆ เป็นต้น แต่อย่างไรก็ตามเรายังสามารถเพิ่มหรือปรับเปลี่ยนมุมมองของการพิจารณาความน่าสนใจของรูปแบบหนึ่งๆ เพื่อที่จะได้รับสารสนเทศ หรือองค์ความรู้ในรูปแบบใหม่ ๆ ที่แตกต่างจากเดิมได้

ดังนั้น ในงานวิทยานิพนธ์นี้จึงเสนอปัญหาการค้นหารูปแบบที่ให้ค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอ (ยกตัวอย่างจากข้อมูลห้างสรรพสินค้าแห่งหนึ่งในช่วงเทศกาล มีพฤติกรรมการซื้อสินค้าของผู้บริโภคจะซื้อ {กล้อง, เลนส์} มีค่าคุณประโยชน์ที่สูงแต่พฤติกรรมการปรากฏจะไม่สม่ำเสมอ) ซึ่งจากการทราบถึงรูปแบบที่มีค่าคุณประโยชน์ที่สูงดังกล่าวจะทำให้ผู้บริหารสามารถคิดกลยุทธ์เพื่อสามารถกระตุ้นการซื้อสินค้าของผู้บริโภคอันนำมาซึ่งผลประกอบการของบริษัทที่เพิ่มขึ้นได้และการวางแผนจัดการสินค้าคงคลัง ถ้าธุรกิจมีสินค้าคงคลังน้อยเกินไป ก็อาจประสบปัญหาสินค้าขาดแคลนไม่เพียงพอ (out of stock) สูญเสียโอกาสในการขายสินค้าให้แก่ลูกค้า อีกทั้งจะเป็นการเพิ่มโอกาสในการขายสินค้าให้กับบริษัทคู่แข่ง และอาจต้องสูญเสียลูกค้าในที่สุด นอกจากนี้ ถ้าสิ่งที่ขาดแคลนนั้นเป็นวัตถุดิบที่สำคัญ การดำเนินงานทั้งการผลิตและการขายก็อาจต้องหยุดชะงัก ซึ่งอาจส่งผลต่อภาพลักษณ์ของธุรกิจในอนาคตได้ ดังนั้น นี้จึงเป็นหน้าที่ของผู้ประกอบการในการจัดการสินค้าคงคลังของตนให้อยู่ในระดับที่เหมาะสมไม่มากหรือน้อยจนเกินไป เพราะการลงทุนในสินค้าคงคลังต้องใช้เงินจำนวนมากและมีโอกาสที่วัตถุดิบจะเสียหาย เช่น อุปกรณ์อิเล็กทรอนิกส์ เครื่องใช้ไฟฟ้าจะเสื่อมสภาพอาจส่งผลกระทบถึงสภาพคล่องของธุรกิจได้ โดยการวิเคราะห์พฤติกรรมการบริโภคที่นำเสนอจะเหมาะกับทุกธุรกิจการค้าที่มีรายการสินค้าที่หลากหลาย และมีการขายสินค้าต่อเนื่อง อาทิเช่น ธุรกิจค้าปลีก ธุรกิจค้าส่ง ธุรกิจร้านอาหาร ธุรกิจเกี่ยวกับเฟอร์นิเจอร์ ธุรกิจเกี่ยวกับอุปกรณ์ไฟฟ้า ห้างสรรพสินค้า ธุรกิจขายตรง ธุรกิจขายอะไหล่รถจักรยานยนต์หรือรถยนต์ ธุรกิจเกี่ยวกับอาหารทะเล ธุรกิจเกี่ยวกับวัสดุอุปกรณ์การเกษตร ธุรกิจเกี่ยวกับวัสดุอุปกรณ์ก่อสร้าง ธุรกิจเกี่ยวกับผลิตภัณฑ์ทารกและเด็ก ธุรกิจเกี่ยวกับอุปกรณ์เครื่องเขียนและสิ่งพิมพ์ ธุรกิจเกี่ยวกับอุปกรณ์คอมพิวเตอร์และโทรศัพท์เคลื่อนที่ ธุรกิจเกี่ยวกับดอกไม้และผลิตภัณฑ์ทางการเกษตร ธุรกิจเกี่ยวกับเบเกอรี่ ธุรกิจเกี่ยวกับเวชกรรมและอุปกรณ์ทางการแพทย์ ธุรกิจเกี่ยวกับเครื่องประดับ ธุรกิจสิ่งทอและเครื่องแต่งกาย ธุรกิจเกี่ยวกับเว็บไซต์ต่างๆ ธุรกิจ/องกรทางการแพทย์ และอื่นๆ

ตัวอย่างการประยุกต์ใช้งานด้านธุรกิจ เช่น บริษัท Amazon.com[1] เป็นศูนย์กลางของผู้บริโภคทั่วโลกที่จะหาซื้อสินค้าใดๆ ที่ต้องการผ่านทางระบบออนไลน์ในราคาที่สมเหตุสมผลเท่าที่จะเป็นไปได้ จุดขายคือความสะดวก สินค้ามากมายให้เลือกซื้อ ฤดูกาลจำหน่ายที่สำคัญของบริษัท Amazon.com Inc. คือ ช่วงเทศกาล Black Friday (Slice Intelligence[2] บริษัทเก็บข้อมูลด้านอีคอมเมิร์ซได้เปิดเผยว่าจากเทศกาล Black Friday ในปี 2015 ยอดขายเฉพาะของ Amazon คิดเป็น 36% ของยอดขายออนไลน์ทั้งหมด) และเทศกาลต่างๆ ในแต่ละปี ถ้าเราทราบถึงรูปแบบที่ให้ค่าคุณประโยชน์ที่สูง และปรากฏอย่างไม่สม่ำเสมอหรือสินค้าบางชนิดที่ถูกซื้ออย่างมากเฉพาะช่วงเทศกาล ซึ่งรูปแบบรายการสินค้าดังกล่าวจะสามารถบ่งบอกถึงช่วงเวลาที่มีการซื้อสินค้าและช่วงเวลาที่สินค้าไม่ถูกซื้อ จากการทราบถึงรูปแบบรายการสินค้าดังกล่าว อาจช่วยให้เราทราบถึงช่วงเวลาที่รายการสินค้าได้รับความนิยม อันนำมาซึ่งการตัดสินใจในการเพิ่ม ลดการผลิต การเตรียมวัตถุดิบเพื่อให้สอดคล้องกับกำลังการผลิต ลดปัญหาในการสต๊อกสินค้า และให้ผู้บริหารสามารถคิดกลยุทธ์เพื่อสามารถกระตุ้นการซื้อสินค้าของผู้บริโภคนำมาซึ่งผลประกอบการของบริษัทที่เพิ่มขึ้นได้

การประยุกต์ใช้งานในด้านการแพทย์ ปัจจุบันพบว่าโรคหลายชนิดมีสาเหตุสืบเนื่องมาจากความผิดปกติของยีน เช่น โรคมะเร็ง การวิเคราะห์ข้อมูลทางพันธุกรรมของยีน DNA Microarray Technology[3] ที่มีการปรากฏร่วมกันอย่างสม่ำเสมอและมีความผิดปกติในลำดับ DNA จะทำให้นักวิทยาศาสตร์ทราบถึงกลไกการเกิดความผิดปกตินำมาซึ่งสาเหตุการเกิดโรคที่ร้ายแรงได้และ อาจทราบถึงวิธีการรักษาโรคต่างๆ แต่อย่างไรก็ตามรูปแบบยีนที่ปรากฏร่วมกันอย่างสม่ำเสมอไม่สามารถตอบสนองต่อการวิเคราะห์ความผิดปกติของยีนที่ทำให้เกิดโรคได้ ซึ่งถ้าเราทราบถึงยีนที่ปรากฏร่วมกันอย่างไม่สม่ำเสมอและมีความผิดปกติ อาจช่วยให้นักวิทยาศาสตร์หรือแพทย์ทราบถึงแนวโน้มที่เป็นสาเหตุที่ทำให้เกิดโรคต่างๆเพิ่มมากขึ้นและการรักษาโรคเพิ่มขึ้นได้

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อสร้างแนวคิดใหม่ในการตรวจสอบพฤติกรรมของการบริโภคที่สามารถนำไปประยุกต์ใช้ในธุรกิจต่างๆ ได้จริง โดยแนวคิดดังกล่าวสามารถประยุกต์ใช้ได้กับ ธุรกิจค้าปลีก ธุรกิจค้าส่ง ธุรกิจร้านอาหาร ธุรกิจเกี่ยวกับเฟอร์นิเจอร์ ธุรกิจเกี่ยวกับอุปกรณ์ไฟฟ้า ห้างสรรพสินค้า ธุรกิจขายตรง ธุรกิจขายอะไหล่รถจักรยานยนต์หรือรถยนต์ ธุรกิจเกี่ยวกับอาหารทะเล ธุรกิจเกี่ยวกับวัสดุอุปกรณ์การเกษตร ธุรกิจเกี่ยวกับวัสดุอุปกรณ์ก่อสร้าง ธุรกิจเกี่ยวกับผลิตภัณฑ์ทารกและเด็ก ธุรกิจเกี่ยวกับอุปกรณ์เครื่องเขียนและสิ่งพิมพ์ ธุรกิจเกี่ยวกับอุปกรณ์คอมพิวเตอร์และโทรศัพท์เคลื่อนที่ ธุรกิจเกี่ยวกับดอกไม้และผลิตภัณฑ์ทางการเกษตร ธุรกิจเกี่ยวกับเบเกอรี่ ธุรกิจเกี่ยวกับเวชกรรมและอุปกรณ์ทางการแพทย์ ธุรกิจเกี่ยวกับเครื่องประดับ ธุรกิจสิ่งทอและเครื่องแต่งกาย ธุรกิจเกี่ยวกับเว็บไซต์ต่างๆ ธุรกิจ/องค์กรทางการแพทย์ และอื่นๆ

---

[1] https://www.amazon.com

[2] https://intelligence.slice.com

[3] http://ejournals.swu.ac.th/index.php/pharm

2. เพื่อศึกษาการวิเคราะห์พฤติกรรมหรือรูปแบบการบริโภคของผู้บริโภคภายใต้กรอบแนวคิดเกี่ยวกับการค้นหารูปแบบที่ให้ค่าคุณประโยชน์สูง และการค้นหารูปแบบที่ปรากฏแบบไม่สม่ำเสมอ

3. เพื่อทราบถึงกลุ่มของรายการ (สินค้า) ที่ปรากฏไม่สม่ำเสมอแต่ให้ค่าคุณประโยชน์สูง ซึ่งจากข้อมูลดังกล่าวจะนำไปสู่การค้นหาสาเหตุของการเกิดขึ้นของพฤติกรรมหรือรูปแบบที่ซึ่งจะสามารถนำข้อมูลดังกล่าวไปประกอบการตัดสินใจเพื่อทำการพัฒนาผลิตภัณฑ์และขั้นตอนการดำเนินธุรกิจได้

1.3 ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย

1. ได้ทราบถึงข้อมูลที่ได้จากการวิเคราะห์พฤติกรรมการบริโภคของลูกค้า ซึ่งเป็นข้อมูลเพื่อสนับสนุนการตัดสินใจที่จะดำเนินการกระตุ้นพฤติกรรมการจับจ่ายใช้สอยของผู้บริโภคได้ อาทิเช่น การจัดทำโปรโมชัน การนำเสนอรายการสินค้าใหม่ๆ และอื่นๆ โดยการวิเคราะห์พฤติกรรมการบริโภคที่นำเสนอจะเหมาะกับทุกธุรกิจการค้าที่มีรายการสินค้าที่หลากหลาย และมีการขายสินค้าต่อเนื่อง อาทิเช่น ธุรกิจค้าปลีก ธุรกิจค้าส่ง ธุรกิจร้านอาหาร ธุรกิจเกี่ยวกับเฟอร์นิเจอร์ ธุรกิจเกี่ยวกับอุปกรณ์ไฟฟ้า ห้างสรรพสินค้า ธุรกิจขายอะไหล่รถจักรยานยนต์หรือรถยนต์ ธุรกิจเกี่ยวกับอาหารทะเล ธุรกิจเกี่ยวกับวัสดุอุปกรณ์การเกษตร ธุรกิจเกี่ยวกับวัสดุอุปกรณ์ก่อสร้าง ธุรกิจเกี่ยวกับผลิตภัณฑ์ทารกและเด็ก ธุรกิจเกี่ยวกับอุปกรณ์เครื่องเขียนและสิ่งพิมพ์ ธุรกิจเกี่ยวกับอุปกรณ์คอมพิวเตอร์และโทรศัพท์เคลื่อนที่ ธุรกิจเกี่ยวกับดอกไม้และผลิตภัณฑ์ทางการเกษตร ธุรกิจเกี่ยวกับเบเกอรี่ ธุรกิจเกี่ยวกับเวชกรรมและอุปกรณ์ทางการแพทย์ ธุรกิจเกี่ยวกับเครื่องประดับ ธุรกิจสิ่งทอและเครื่องแต่งกาย ธุรกิจเกี่ยวกับเว็บไซต์ต่างๆ ธุรกิจ/องค์กรทางการแพทย์ และอื่นๆ

2. ได้ขั้นตอนและวิธีสำหรับการวิเคราะห์พฤติกรรมการบริโภคที่ให้ค่าคุณประโยชน์สูงโดยปรากฏอย่างไม่สม่ำเสมอที่มีประสิทธิภาพในแง่มุมของความถูกต้อง เวลาที่ใช้ในการประมวลผลและหน่วยความจำที่ใช้สำหรับการคำนวณ

3. ได้ผลงานวิจัยที่จะสามารถตีพิมพ์ในงานประชุมวิชาการ “*Knowledge and Smart Technology*” (*KST), 2017 9th International Conference on*. IEEE, 2017.

4. สามารถนำแนวคิดการวิเคราะห์พฤติกรรมการบริโภค และขั้นตอนวิธีที่นำเสนอไปต่อยอดเพื่อการดำเนินการวิจัยขั้นสูงต่อไป

1.4 ขอบเขตของการวิจัย

1. ข้อมูลการซื้อสินค้าที่จะทำการพิจารณาจะต้องมีลักษณะเป็นแบบทรานแซกชันที่ประกอบไปด้วยรายการสินค้าและจำนวนชิ้นของแต่ละสินค้าที่ถูกซื้อในแต่ละทรานแซกชัน โดยข้อมูล

ที่นำมาใช้ทดสอบเป็นชุดข้อมูลจริง สามารถดาว์นโหลดได้จาก P.F. Viger, “SPMF: An Open-Source Data Mining Library[4]”

2. การวิเคราะห์พฤติกรรมผู้บริโภคจะเป็นการวิเคราะห์ภายใต้การค้นหารูปแบบที่ให้ค่าคุณประโยชน์สูงโดยปรากฏอย่างไม่สม่ำเสมอ ซึ่งผู้ที่ต้องการผลลัพธ์จะต้องทำการกำหนดค่าขีดแบ่งคุณประโยชน์ (utility threshold) และค่าขีดแบ่งความสม่ำเสมอ (regularity threshold) เพื่อใช้เป็นเกณฑ์สำหรับวัดความน่าสนใจของรูปแบบที่จะทำการค้นหาจากข้อมูลที่ต้องการวิเคราะห์

3. การวัดผลของการวิเคราะห์จะสามารถดำเนินการได้ใน 3 แง่มุม คือ 1) เวลาที่ใช้ในการประมวลผล 2) หน่วยความจำที่ใช้ในการประมวลผล และ 3) จำนวนผลลัพธ์ที่สามารถค้นหาได้ตามลำดับ โดยผลลัพธ์ที่ได้จากการวิเคราะห์สามารถนำไปประกอบการตัดสินใจได้ แต่การตัดสินใจและการติดตามผลการตัดสินใจไม่สามารถดำเนินการได้

1.5 แผนการดำเนินงาน

การดำเนินงานวิจัยนี้ได้ถูกดำเนินการภายใต้แผนการดำเนินงานและระยะเวลาที่ระบุไว้ในตารางที่ 1-1

ตารางที่ 1-1 ระยะเวลาในการดำเนินการวิจัย

แผนการดำเนินงานวิจัย	ปีการศึกษา 2559				ปีการศึกษา 2560			
	เดือน				เดือน			
	1-3	4-6	7-9	10-12	1-3	4-6	7-9	10-12
จัดเตรียมข้อมูลที่จะใช้เป็นต้นแบบในการวิเคราะห์ข้อมูลผู้บริโภค	↔							
ศึกษางานวิจัยที่เกี่ยวข้องโดยมุ้งเน้นการค้นหาเซตรายการสินค้าที่ให้ค่าคุณประโยชน์/ต่ำการซื้อสินค้าแบบสม่ำเสมอ/ไม่สม่ำเสมอ	↔	↔						

[4] http://www.philippe-fournier-viger.com/

ตารางที่ 1-1 ระยะเวลาในการดำเนินการวิจัย(ต่อ)

แผนการดำเนินงานวิจัย	ปีการศึกษา 2559				ปีการศึกษา 2560			
	เดือน				เดือน			
	1-3	4-6	7-9	10-12	1-3	4-6	7-9	10-12
กำหนดขอบเขตเพื่อวิเคราะห์พฤติกรรมผู้บริโภครวมถึงการกำหนดนิยามพื้นฐานที่เกี่ยวกับการวิเคราะห์ข้อมูลพฤติกรรมผู้บริโภค		←	→					
ออกแบบขั้นตอนการวิเคราะห์ข้อมูลผู้บริโภค พัฒนาโปรแกรม ทดสอบประสิทธิภาพ ในแง่มุมต่างๆ		←	→					
เขียนบทความเพื่อตีพิมพ์งานประชุมวิชาการ			←	→				
จัดทำเอกสารและทำการสอบบท 3			←	→				
ศึกษาและปรับปรุงข้อบกพร่องในแง่มุมต่างๆ					←		→	
จัดทำเอกสารและทำการสอบ 5 บท							←	→

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

งานวิจัยนี้จะเป็นการค้นหารูปแบบ ภายใต้การพิจารณารูปแบบแบบใหม่ ที่ซึ่งเกิดจากการนำแนวคิดเกี่ยวกับการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงมาผสมผสานกับการค้นหารูปแบบที่ปรากฏบ่อยและสม่ำเสมอ โดยจากการผสมผสานลักษณะของรูปแบบข้างต้นจะทำให้ผู้วิจัยสามารถกำหนดนิยามเกี่ยวกับการค้นหารูปแบบที่มีประโยชน์สูงและปรากฏขึ้นอย่างไม่สม่ำเสมอ ภายใต้การค้นหารูปแบบ 3 แง่มุม ดังนี้

1. การค้นหารูปแบบที่ปรากฏบ่อย (*Frequent Itemsets Mining, FIM*)
2. การค้นหารูปแบบที่มีค่าคุณประโยชน์สูง (*High-Utility Itemsets Mining, HUIM*)
3. การค้นหารูปแบบที่ปรากฏบ่อยและสม่ำเสมอ (*Frequent-Regular Itemsets Mining, FRIM*)

2.1.1 การค้นหารูปแบบที่ปรากฏบ่อย (*Frequent Itemsets Mining, FIM*)

การค้นหารูปแบบที่ปรากฏบ่อยเป็นการค้นหารูปแบบที่น่าสนใจภายใต้การพิจารณาจำนวนครั้ง (หรือ ความบ่อย หรือ ความถี่) ในการปรากฏขึ้นของรูปแบบเหล่านั้น โดยปัญหาการค้นหารูปแบบที่ปรากฏบ่อยจะมุ่งเน้นที่การค้นหาเซตรายการ (รายการสินค้า) ที่ถูกซื้อร่วมกันบ่อยๆ ซึ่งจะทำให้บริษัท/ห้างร้าน/สถานประกอบการสามารถทราบถึงปริมาณการซื้อที่ซึ่งสามารถนำข้อมูลดังกล่าวไปเป็นส่วนประกอบในการจัดทำโปรโมชั่น การจัดการคลังสินค้า การทำแคตตาล็อกสินค้า การจัดวางชั้นสินค้า และอื่นๆ 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) หมายเลขกำกับทรานแซกชัน (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 $1 \le j < k \le |TDB|$ 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^{α}, 6^{α}, 7^{α}, 9^{α}, 10^{α}} ดังนั้นค่าสนับสนุนของเซตรายการ *'α'* สามารถคำนวณได้ดังนี้ *s^{α} = |1^{α}, 2^{α}, 3^{α}, 5^{α}, 6^{α}, 7^{α}, 9^{α}, 10^{α}| = 8*

จากนิยามข้างต้นการค้นหาเซตรายการที่ปรากฏที่ปรากฏบ่อยสามารถนิยามได้ดังนี้

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

2.1.2 การค้นหารูปแบบที่มีค่าคุณประโยชน์สูง (*High-Utility Itemset Mining, HUIM*)

การค้นหารูปแบบที่มีค่าคุณประโยชน์สูง ถูกพัฒนามาจากการค้นหารูปแบบที่ปรากฏบ่อย โดยการค้นหารูปแบบดังกล่าวจะทำการพิจารณาเกี่ยวกับค่าคุณประโยชน์ของรูปแบบที่อยู่ในรูปของผลกำไร ต้นทุน ค่าความเสี่ยง ค่าความผิดพลาด และอื่นๆ การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงจะสามารถบอกได้ถึงรายการสินค้าที่ให้ผลกำไรสูงหรือต้นทุนต่ำ นอกจากนั้นยังสามารถบอกได้ถึงปริมาณ/จำนวนสินค้าที่ถูกซื้อได้ ซึ่งจะทำให้บริษัท/ห้างร้าน/สถานประกอบการสามารถทราบถึงปริมาณการซื้อสินค้า แล้วสามารถนำข้อมูลดังกล่าวไปเป็นส่วนประกอบในการจัดทำโปรโมชัน การจัดการคลังสินค้า การทำแค็ตตาล็อกสินค้า การจัดวางชั้นสินค้า โดยปัญหาการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงสามารถนิยามได้ ดังนี้

นิยามที่ 7 *กำหนดให้แต่ละรายการ $i_j \in I$ (ดังนิยามที่ 1) จะมีค่าคุณประโยชน์ต่อการปรากฏขึ้นหนึ่งครั้งของรายการ i_j เรียกว่า external utility อาทิเช่น ผลกำไรจากสินค้าชิ้นหนึ่งๆ ต้นทุนของสินค้าชิ้นหนึ่งๆ หรืออื่นๆ 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 $t_p = \{i_j, ..., i_k\}$ ประกอบไปด้วยเซตของรายการ $Y = = \{i_j, ..., i_k\}$ ที่ถูกบรรจุอยู่ในทรานแซกชัน t_p โดยแต่ละ $i_j \in Y$ จะมีจำนวนครั้งของการปรากฏขึ้นของรายการ i_j ในทรานแซกชัน $t_p = \{i_j, ..., i_k\}$ เรียกว่า internal utility สามารถแทนได้ด้วยสัญลักษณ์ $iu(i_j, t_p)$*

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

ตารางที่ 2-3 ตัวอย่างข้อมูลการซื้อสินค้าของลูกค้า

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

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 จะประกอบไปด้วยค่าคุณประโยชน์ของแต่ละรายการและฐานข้อมูลรายการจำนวน 10 ทรานแซกชัน ค่าคุณประโยชน์ของเซตรายการ *'ab'* ในแทรานแซกชัน

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_1, t_3, t_5, t_7, t_9) สามารถคำนวณได้ดังนี้ $u(ab) = u(ab,t_1) + u(ab,t_3) + u(ab,t_5) + u(ab,t_7) + u(ab,t_9) = 60 + 75 + 15 + 40 + 40 = 230$

จากนิยามข้างต้นการค้นหาเซตรายการที่มีค่าคุณประโยชน์สูงสามารถนิยามได้ ดังนี้

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

2.1.3 การค้นหารูปแบบที่ปรากฏบ่อยและสม่ำเสมอ (*Frequent-Regular Itemsets Mining, FRIM*)

การค้นหารูปแบบที่ปรากฏบ่อยและสม่ำเสมอได้ถูกพัฒนาต่อยอดจากการค้นหารูปแบบที่ปรากฏบ่อยเช่นกัน วัตถุประสงค์หลักของการค้นหารูปแบบที่ปรากฏบ่อยและสม่ำเสมอจะเป็นการตรวจสอบ/วิเคราะห์พฤติกรรมของการปรากฏขึ้นของรูปแบบภายใต้เงื่อนไขในเชิงความถี่ (จำนวนครั้ง) และความสม่ำเสมอ (ระยะเวลาของการปรากฏซ้ำ) ของการปรากฏ ในการศึกษาเกี่ยวกับพฤติกรรมการปรากฏขึ้นของรูปแบบ ว่ามีพฤติกรรมการปรากฏขึ้นอย่างสม่ำเสมอหรือไม่ เราจะสามารถสังเกตได้จากระยะเวลาในการปรากฏ โดยในการหาระยะเวลาดังกล่าวของรูปแบบ เราสามารถคำนวณได้จากการพิจารณาเซตของหมายเลขทรานแซกชันที่มี *X* ปรากฏขึ้น (T^X) สามารถนิยามได้ดังนี้

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

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

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

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

นิยามที่ 16 *กำหนดให้ r^X คือช่วงเวลาที่ยาวนานที่สุดที่ไม่มีเซตรายการ X ปรากฏขึ้นในฐานข้อมูลหรือเรียกว่าค่าสม่ำเสมอของเซตรายการ X สามารถคำนวณและแทนสัญลักษณ์ $r^X = max(fr^X_{t_p}, r^X_{t_pt_q}, ..., r^X_{t_yt_z}, lr^X_{t_z})$*

ตัวอย่างที่ 3 จากตารางที่ 2-1 ค่าความสม่ำเสมอของเซตรายการ *'a'* หรือที่มีเซต $T^a = \{1^a, 2^a, 3^a, 5^a, 6^a, 7^a, 9^a, 10^a\}$ ในฐานข้อมูล สามารถคำนวณได้ดังนี้ $r^a = max(fr^a_{t_1}, r^a_{t_1t_2}, r^a_{t_2t_3}, r^a_{t_3t_5}, r^a_{t_5t_6}, r^a_{t_6t_7}, r^a_{t_7t_9}, r^a_{t_9t_{10}}, lr^a_{t_{10}}) = max(1, 1, 1, 2, 1, 1, 2, 1, 0) = 2$

ระยะเวลาดังกล่าวสามารถการันตีได้ว่าเซตรายการสินค้า X จะปรากฏขึ้นอย่างน้อย 1 ครั้งในทุกๆ r^X ทรานแซกชันที่เรียงต่อกัน ซึ่งจากการพิจารณาค่า r^X จะทำให้ทราบถึงพฤติกรรมการปรากฏขึ้นของเซตรายการ โดยจากนิยามที่กล่าวมาข้างต้น การค้นหาเซตรายการที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอนิยามได้ดังนี้

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

2.2 งานวิจัยที่เกี่ยวข้อง

(Agrawal, Srikant, และคณะ, 1994) ได้ชี้ให้เห็นถึงแนวคิดการค้นหารูปแบบที่ปรากฏบ่อย (Frequent itemset mining, FIM) ที่จะพิจารณาความน่าสนใจของรูปแบบในแง่มุมของความถี่ของการปรากฏ ที่สามารถบ่งบอกได้ถึงสิ่งของหรือเหตุการณ์ที่ปรากฏขึ้นพร้อมกันบ่อยๆ ตัวอย่างเช่น ในธุรกิจห้างสรรพสินค้าหรือธุรกิจค้าปลีกจะทำการหาความสัมพันธ์ของรายการสินค้าที่ถูกซื้อร่วมกันบ่อยๆ เพื่อช่วยในการจัดทำโปรโมชันสินค้า ช่วยในการจัดชั้นวางสินค้าให้สินค้าที่ถูกซื้อร่วมกันบ่อยๆ ให้อยู่ในพื้นที่ใกล้ๆ กันเพื่ออำนวยความสะดวกให้แก่ลูกค้า ซึ่งการดำเนินการทั้งหมดนี้จะช่วยให้ห้างสรรพสินค้ารักษาฐานลูกค้าให้ยังคงซื้อสินค้ากับห้างสรรพสินค้าต่อไป โดยการค้นหารูปแบบที่ปรากฏบ่อยในแง่มุมความถี่ของการปรากฏมีหลายๆ งานวิจัย อาทิเช่น การค้นหารูปแบบที่ปรากฏบ่อยเคอันดับแรก (Han, Wang, Lu, และคณะ 2002) การค้นหาเซตรายการจากเซตรายการที่มีความยาวมากที่สุด (Hu, Sung, Xiong, และคณะ 2008) แต่เซตรายการปรากฏบ่อยไม่สามารถบ่งบอกได้ถึงค่าคุณประโยชน์ของเซตรายการและความสำคัญของเซตรายการในแง่มุมต่างๆ อาทิเช่น ผลกำไร ต้นทุน ค่าความผิดพลาด คุณค่าของสิ่งของเหล่านั้น และอื่นๆ ด้วยเหตุนี้ (Chan, R., และคณะ, 2003) และ (Yao, H., และคณะ, 2004) ชี้ให้เห็นถึงแนวคิดการวัดความน่าสนใจของรูปแบบ ด้วยการพิจารณาค่าคุณประโยชน์ของรูปแบบ (high-utility itemset mining) ซึ่งจากการพิจารณาค่าคุณประโยชน์ของรูปแบบจะทำให้ผู้วิเคราะห์ข้อมูลทราบถึงผลกำไรที่ได้จากเซตรายการเหล่านั้น โดยข้อมูลที่ได้สามารถนำไปประกอบการตัดสินใจเพื่อปรับปรุงคุณภาพ การบริการ หรือวิธีการดำเนินธุรกิจของบริษัท/ห้างร้านต่างๆ ได้ นอกเหนือจากการประยุกต์ใช้ในแวดวงธุรกิจ การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงยังสามารถนำไปประยุกต์ใช้กับงานในหลายๆ ด้าน อาทิเช่น การวิเคราะห์ทางพันธุกรรม (Biological gene analysis) การเข้าถึงเว็บแบบมีลำดับ (Web-click sequence

analysis) การตรวจสอบการขึ้นลงของดัชนีหุ้น การวัดประสิทธิภาพของการจราจร การตรวจสอบการเข้าใช้งานเซิร์ฟเวอร์ การวิเคราะห์ข้อมูลที่ได้จากเซ็นเซอร์เน็ตเวิร์ค และการวิเคราะห์ข้อมูลการใช้โทรศัพท์ทางไกล เป็นต้น

จากแนวคิดเริ่มต้นเกี่ยวกับการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงที่ได้กล่าวข้างต้น ได้มีนักวิจัยเป็นจำนวนมากพยายามที่จะพัฒนาขั้นตอนวิธีการค้นหารูปแบบดังกล่าวให้มีประสิทธิภาพมากขึ้น อาทิเช่น (Liu, Y., และคณะ, 2005) ได้พยายามคิดค้นขั้นตอนวิธีที่จะลดทอนปริมาณข้อมูลหรือจำนวนเซตรายการที่ต้องทำการพิจารณา (itemset lattice หรือ search space of itemsets) ด้วยการเสนอค่าประมาณคุณประโยชน์ของเซตรายการ ที่เรียกว่า “Transaction Weighted Utility, TWU” จะเป็นผลรวมของค่าคุณประโยชน์ทั้งหมดของทุกทรานแซกชันในฐานข้อมูลที่มีเซตรายการปรากฏ ซึ่งค่าประมาณคุณประโยชน์จะมีค่าสูงกว่าค่าคุณประโยชน์ที่แท้จริงมาก ดังนั้นถ้าค่าประมาณคุณประโยชน์มีค่าน้อยกว่าค่าขีดแบ่งคุณประโยชน์ (utility threshold) แล้วจะทำให้เราสามารถลบเซตรายการออกจากการพิจารณาได้ เนื่องจากเซตรายการดังกล่าวไม่สามารถเป็นเซตรายการที่เป็นผลลัพธ์ได้ แนวคิดเกี่ยวกับการประมาณค่าคุณประโยชน์ได้ถูกประยุกต์ใช้ในหลายๆ 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)

ถึงแม้ว่าค่าประมาณคุณประโยชน์ TWU จะทำให้เราสามารถลดทอนปริมาณข้อมูลหรือจำนวนเซตรายการที่ต้องทำการพิจารณาได้ แต่อย่างไรก็ตามค่าประมาณคุณประโยชน์ของเซตรายการ จะเป็นค่าที่เกินจากค่าคุณประโยชน์จริงค่อนข้างมาก ทำให้เราอาจไม่สามารถลดทอนจำนวนเซตรายการที่ต้องทำการพิจารณาได้อย่างเต็มที่ จึงเป็นเหตุให้ (Liu, M., และ Qu, J.-G., 2012) ได้คิดค้นค่าประมาณคุณประโยชน์แบบกระชับที่เรียกว่า “Tight Over-estimate Utility, tou” ที่มีค่าเกินจริงน้อยกว่าค่า TWU ซึ่งจะช่วยให้สามารถลดทอนจำนวนเซตรายการที่ต้องทำการพิจารณาได้มากกว่าเดิม และนับตั้งแต่แนวคิดของ Liu ได้เผยแพร่ออกไปทำให้มีหลายๆ งานประยุกต์ใช้แนวคิดดังกล่าวในการลดทอนปริมาณข้อมูลหรือจำนวนเซตรายการที่ต้องทำการพิจารณา อาทิเช่น การค้นหาเซตรายการที่มีค่าคุณประโยชน์สูงโดยใช้ค่าคุณประโยชน์โดยประมาณ FHM (Fournier-Viger, P., และคณะ, 2014-a), การตัดแต่งกิ่งสำหรับการค้นหาเซตรายการที่มีค่าคุณประโยชน์สูง (Krishnamoorthy, S., 2015) นอกเหนือจากการพัฒนาประสิทธิภาพในการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงภายใต้การกำหนดค่าขีดแบ่งคุณประโยชน์แล้ว นักวิจัยทางด้านการทำเหมืองข้อมูลได้คิดค้นแนวทาง วิธีการ และขั้นตอนต่างๆ ที่ช่วยให้การค้นหาเซตรายการดังกล่าวมีความยุ่งยากลดลง รวมถึงเพิ่มความสามารถในแง่มุมต่างๆ อีกเป็นจำนวนมาก อาทิเช่น 1) การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงจากฐานข้อมูลที่มีการเปลี่ยนแปลงทั้งการเพิ่ม/ลดจำนวนทรานแซกชัน รวมถึงการ

เปลี่ยนแปลงข้อมูลในทรานแซกชันในภายหลัง (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)[21], (Founier-Viger, P., 2014-c), (Lan, G.-C., และคณะ, 2014-b), (Founier-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) การลดทอนความยุ่งยากของการกำหนดค่าขีดแบ่งคุณประโยชน์ด้วยการกำหนดจำนวนเซตรายการที่ให้ผลกำไรสูงสุด (Wu, C.-W., และคณะ, 2012), (Zihayat, M., และ An A., 2014), (Ryang, H., และ Yun, U., 2015), (Tseng, V.S., และคณะ, 2015-b)

นอกเหนือจากการพัฒนา Frequent itemset mining ด้วยการพิจารณาค่าคุณประโยชน์สูงของรูปแบบแล้ว Tanbeer, S.K., และคณะ, 2009 ชี้ให้เห็นว่าการค้นหารูปแบบปรากฏบ่อยจากฐานข้อมูลโดยใช้ค่าสนับสนุน (จำนวนครั้งของการเกิดขึ้นของรูปแบบในฐานข้อมูล) อาจไม่เพียงพอต่อการค้นหารูปแบบที่น่าสนใจในแง่มุมอื่นๆ ด้วยเหตุนี้ Tanbeer และคณะ จึงได้นำเสนอแนวความคิดในการศึกษาพฤติกรรมของการปรากฏขึ้นของรูปแบบด้วยการค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ ซึ่งจะเป็นการศึกษาพฤติกรรมการปรากฏขึ้นของรูปแบบทั้งในเชิงความถี่และความสม่ำเสมอของการปรากฏ แนวความคิดนี้สามารถนำไปประยุกต์ใช้ได้ในงานหลายๆ ด้าน อาทิเช่น ผู้จัดการหรือผู้บริหารของธุรกิจค้าปลีก อาจจะสนใจรายการสินค้าที่ถูกซื้อบ่อยๆ และถูกซื้ออย่างสม่ำเสมอมากกว่ารายการสินค้าที่ถูกซื้อบ่อยๆ เพียงอย่างเดียว เพื่อที่จะทำการจัดเตรียมสินค้าให้พอเหมาะกับความต้องการของผู้บริโภค และยังสามารถช่วยในการจัดทำโปรโมชั่นสำหรับสินค้าที่ถูกซื้อบ่อยๆ ร่วมกับสินค้าที่ไม่ได้ถูกซื้อบ่อยได้อีกด้วย ในส่วนของการพัฒนาการออกแบบเว็บไซต์หรือการดูแลรักษาเว็บไซต์ ผู้ดูแลเว็บไซต์อาจจะสนใจความสม่ำเสมอของการคลิกเพื่อเรียกดูข้อมูลในเว็บเพจที่ต่อเนื่องกันเพื่อนำไปปรับปรุงข้อความหรือเนื้อหาของเว็บไซต์ให้มีความน่าสนใจยิ่งขึ้น ในส่วนของการวิเคราะห์ข้อมูลทางพันธุกรรม กลุ่มของยีนที่ปรากฏบ่อยและสม่ำเสมออาจบ่งบอกถึงข้อมูลที่สำคัญให้แก่นักวิทยาศาสตร์ได้ ในส่วนตลาดหุ้นกลุ่มของหุ้นที่มีดัชนีที่มีการเพิ่มขึ้นอย่างสม่ำเสมออาจจะได้รับความน่าสนใจจากนักลงทุนต่างๆ และอื่นๆ

ในการหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ ผู้ใช้จะต้องทำการกำหนดค่าพารามิเตอร์ 2 ค่าด้วยกันคือ 1) ค่าขีดแบ่งสนับสนุน และ 2) ค่าขีดแบ่งความสม่ำเสมอ เพื่อใช้วัดความน่าสนใจหรือความสำคัญของรูปแบบภายใต้พฤติกรรมการเกิดขึ้นของรูปแบบเหล่านั้น แต่อย่างไรก็ดี เป็นที่ทราบกันดีว่า “ถ้าเราไม่ได้มีความรู้ในข้อมูลมาก่อน การกำหนดค่าขีดแบ่งสนับสนุนเพื่อที่จะได้รับรูปแบบที่น่าสนใจและมีความสำคัญมากที่สุดจะเป็นเรื่องที่ยุ่งยากและลำบาก” โดยถ้าเรากำหนดค่าขีดแบ่งสนับสนุนสูงเกินไป อาจทำให้เราได้ผลลัพธ์เป็นจำนวนน้อยหรืออาจจะไม่ได้ผลลัพธ์เลย ในกรณีนี้ เราจำเป็นต้องคาดเดาค่าขีดแบ่งให้มีค่าน้อยลงแล้วทำการค้นหาผลลัพธ์ใหม่อีกครั้งซึ่งอาจจะได้รับผลลัพธ์ที่ดีขึ้นหรือไม่ก็ได้ แต่ในกรณีที่ค่าขีดแบ่งถูกกำหนดให้มีค่าน้อย อาจทำให้

เราได้ผลลัพธ์ออกมาเป็นจำนวนมากเกินกว่าที่เราจะทำการพิจารณาองค์ความรู้ได้ และการค้นหาผลลัพธ์จะใช้เวลาค่อนข้างมากอีกด้วย

จากปัญหาข้างต้นดังกล่าว จึงมีงานวิจัยที่ทำการพัฒนาต่อยอดจากงานของ Tanbeer โดยมีวัตถุประสงค์ที่จะหลีกเลี่ยงการกำหนดค่าขีดแบ่งสนับสนุน โดยกำหนดให้ผู้ใช้ทำการกำหนดจำนวนผลลัพธ์ (เซตรายการ) ที่ต้องการแทนด้วยการหารูปแบบทั้งสิ้นเครูปแบบ ซึ่งปรากฏในฐานข้อมูลอย่างสม่ำเสมอและปรากฏบ่อยที่สุด ภายใต้ปัญหานี้ ผู้ที่ต้องการค้นหารูปแบบจะต้องทำการกำหนดค่าพารามิเตอร์ 2 ค่าด้วยกันคือ 1) ค่าขีดแบ่งความสม่ำเสมอ และ 2) จำนวนผลลัพธ์ที่ต้องการในการค้นหารูปแบบดังกล่าวได้อย่างรวดเร็ว (Amphawan, K., และคณะ, 2009), (Amphawan, K., และคณะ, 2012), (Amphawan, K., และ Lenca, P., 2015)

นอกเหนือจากการหลีกเลี่ยงความยุ่งยากในการกำหนดค่าขีดแบ่งสนับสนุน การค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอ ได้ถูกพัฒนาอย่างต่อเนื่องในหลายๆ แง่มุม อาทิเช่น การค้นหารูปแบบที่ปรากฏบ่อยและสม่ำเสมอ 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), การค้นหารูปรูปแบบที่ปรากฏบ่อยและสม่ำเสมอด้วยการกำหนดเงื่อนไขเกี่ยวกับค่าสนับสนุน (Kiran, R.U., และ Reddy, P.K., 2010) นอกเหนือจากงานวิจัยข้างต้นแล้วยังมีงานวิจัยต่างๆ ที่มุ่งเน้นที่การพัฒนาการค้นหารูปแบบที่ปรากฏบ่อยและปรากฏอย่างสม่ำเสมอในแง่มุมต่างๆ อีกมากมาย เช่น การค้นหารูปแบบเป็นระยะๆ ที่เป็นไปได้ในฐานข้อมูลขนาดใหญ่ (Luo, X., และคณะ, 2013), (Kiran, R.U., และ Kitsuregawa, M., 2013), เทคนิคใหม่ในการลดช่องว่างในการค้นหารายการปรากฏบ่อย (Kiran, R.U., และ Kitsuregawa, M., 2014), การค้นหาโรคที่ปรากฏบ่อยเป็นประจำทางการแพทย์จากฐานข้อมูล (Khaleel, M.A., และคณะ, 2014) ตามลำดับ

ในการค้นหารูปแบบที่ปรากฏบ่อยและปรากฏสม่ำเสมอจะไม่สามารถบ่งบอกถึงความสำคัญหรือคุณประโยชน์ของรูปแบบได้ แต่สำหรับการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงจะไม่สามารถบ่งบอกถึงพฤติกรรมการปรากฏขึ้นของรูปแบบได้ ด้วยเหตุนี้ Amphawan, K., และ Surarerks, A., 2015 ได้นำเสนอการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงโดยปรากฏขึ้นอย่างสม่ำเสมอ ชี้ให้เห็นว่าการวัดความน่าสนใจที่ค้นพบจากฐานข้อมูลด้วยการพิจารณาค่าคุณประโยชน์เพียงอย่างเดียวอาจไม่เพียงพอต่อการค้นหารูปแบบที่น่าสนใจ จึงได้เพิ่มเติมเงื่อนไขการพิจารณารูปแบบโดยจะทำการเพิ่มเติมเงื่อนไขของการปรากฏอย่างสม่ำเสมอร่วมกับการพิจารณาค่าคุณประโยชน์ของรายการต่างๆ โดยในการพิจารณาความน่าสนใจของรายการ/เซตรายการ ผู้ใช้จะต้องทำการกำหนดค่าพารามิเตอร์ 2 ค่าด้วยกันคือ 1) ค่าขีดแบ่งคุณประโยชน์ และ 2) ค่าขีดแบ่งความสม่ำเสมอ ซึ่งจะทำให้เราทราบถึงรูปแบบที่มีค่าคุณประโยชน์สูงและมีพฤติกรรมการปรากฏขึ้นอย่างสม่ำเสมออาจช่วยให้เราทราบช่วงเวลาที่รายการสินค้าได้รับความนิยม และนำมาซึ่งการการจัดทำโปรโมชั่นให้กับรายการสินค้าที่พิจารณาเพื่อช่วยกระตุ้นยอดขาย และอื่นๆ นอกจากนั้นได้ทำการพัฒนาประสิทธิภาพด้วยการปรับปรุงโครงสร้างการเก็บข้อมูลคือ “New modified utility-list, NUL” เพื่อลดเวลาการคำนวณค่าคุณประโยชน์ (Amphawan, K., และคณะ, 2016) และTight

Over-estimate Utility (tou) ค่าประมาณคุณประโยชน์แบบกระชับ (Liu, M., และ Qu, J.-G., 2012) และอื่นๆ

2.3 ชุดข้อมูลจาก SPMF

ในการทดสอบประสิทธิภาพของอัลกอริทึม จะดำเนินการทดสอบกับชุดข้อมูล Benchmark ที่ซึ่งเป็นข้อมูลที่ได้รับความน่าเชื่อถือ สามารถดาว์นโหลดได้จาก P. F. Viger, “SPMF : An Open-Source Data Mining Library[1]” โดยแบ่งชนิดของข้อมูลเป็น 2 ชนิดคือ 1) ชนิดข้อมูลแบบเบาบางมีทั้งหมด 5 ชุดข้อมูล คือ BMS, Chainstore, Foodmart2000, Kosatak และ Retail 2) ชนิดข้อมูลแบบหนาแน่นมีทั้งหมด 5 ชุดข้อมูล คือ Accidents, Chess, Connect, Mushroom, และ PUMSB แต่ละชุดข้อมูลจะมีรายละเอียดดังแสดงในตารางที่ 2-4

- ชุดข้อมูล Accidents เป็นข้อมูลอุบัติเหตุจราจรจากสถาบันสถิติแห่งชาติ “National Institute of Statistics, NIS” จากภูมิภาคของฟลามส์สประเทศเบลเยี่ยมในระยะเวลา 1991-2000
- ชุดข้อมูล BMS คือ ข้อมูลการขายสินค้าจากร้านค้าปลีกอิเล็กทรอนิกส์ขนาดใหญ่ ชุดข้อมูล Chainstore คือข้อมูลการทำธุรกรรมของลูกค้าจากร้านค้าปลีก
- ชุดข้อมูล Chess คือ ข้อมูลการเล่นหมากรุกที่ได้จากเครื่อง “UCI Machine Learning Repository”
- ชุดข้อมูล Connect คือ ข้อมูล ที่ได้จากเครื่อง “UCI Machine Learning Repository”
- ชุดข้อมูล Foodmart2000 คือ ข้อมูลที่ได้จากร้านอาหาร
- ชุดข้อมูล Kosarak คือ ข้อมูลจาก click-stream ของ “Hungarian on-line news portal”
- ชุดข้อมูล Mushroom ประกอบไปด้วยบันทึกการอธิบายลักษณะของสายพันธุ์เห็ดต่างๆ
- ชุดข้อมูล PUMSB คือ ข้อมูลการสำรวจสำมะโนประชากร
- ชุดข้อมูล Retail คือ ข้อมูลรายการค้าปลีกที่ได้รับจากร้านค้าซูปเปอร์มาร์เก็ตในเบลเยี่ยม ตั้งแต่เดือนธันวาคม 2542 ถึงพฤศจิกายน 2543

จากชุดข้อมูลทั้งหมด ขั้นตอนเริ่มแรกทำการพิจารณาคุณลักษณะของข้อมูล เพื่อที่จะได้รู้ว่าขั้นตอนวิธีที่นำเสนอสามารถทำงานได้อย่างมีประสิทธิภาพกับข้อมูลในลักษณะใด โดยทำการพิจารณาใน 2 แง่มุมดังนี้ 1) จำนวนเปอร์เซ็นรายการต่อเปอร์เซ็นค่าคุณประโยชน์ (ดังแสดงในภาพที่ 2-1 ถึง 2-10) และ 2) จำนวนเปอร์เซ็นรายการต่อเปอร์เซ็นค่าความสม่ำเสมอ (ดังแสดงในภาพที่ 2-11 ถึง 2-20)

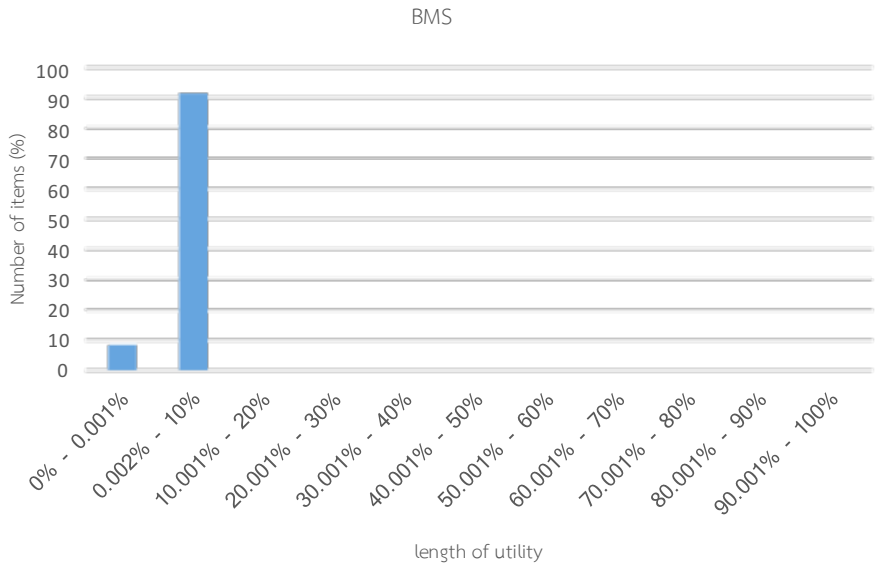
[1] http://www.philippe-fournier-viger.com

ตารางที่ 2-4 คุณลักษณะของชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพ

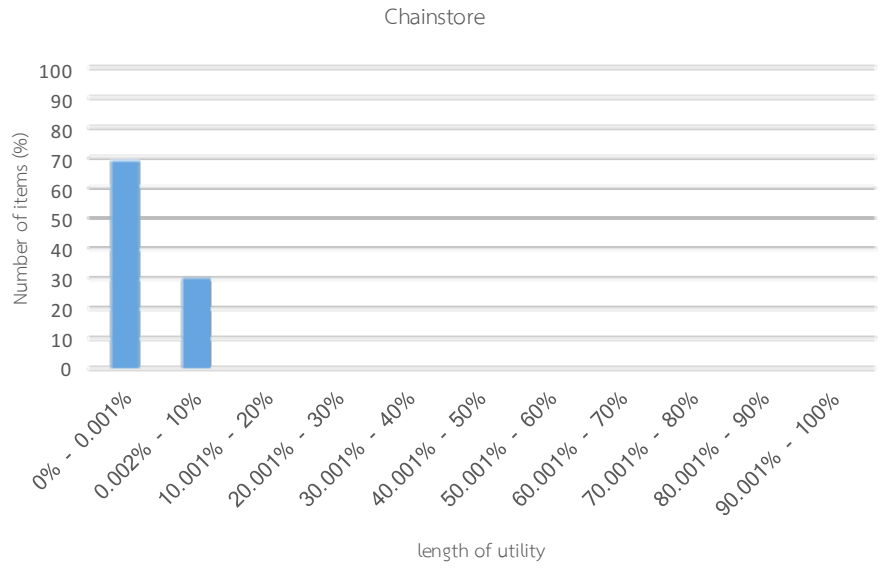
ชื่อฐานข้อมูล	จำนวนรายการที่ปรากฏ	จำนวนทรานแซกชัน	ความยาวเฉลี่ยของทรานแซกชัน	ชนิดของฐานข้อมูล
Accidents	468	340,183	33.8	หนาแน่น
BMS	497	59,601	4.8	เบาบาง
Chainstore	46,086	1,112,949	7.2	เบาบาง
Chess	75	3,196	37	หนาแน่น
Connect	129	67,557	43	หนาแน่น
Foodmart2000	1,559	36,869	11	เบาบาง
Kosarak	41,270	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

จากภาพที่ 2-1 ถึง 2-5 แสดงชุดข้อมูลชนิดเบาบาง 5 ชุดข้อมูล ประกอบด้วย BMS, Chainstore, Foodmart2000, Kosarak และ Retail ตามลำดับ โดยแบ่งค่าคุณประโยชน์ห่างกัน 11 ช่วง ดังนี้ 0%–0.001%, 0.002%–10%, 10.001%–20%, 20.001%–30%, 30.001%–40%, 40.001%–50%, 50.001%–60%, 60.001%–70%, 70.001%–80%, 80.001%–90% และ 90.001%–100% ซึ่งมีช่วงค่าคุณประโยชน์ต่ำสุด คือ 0%-0.001% การแบ่งช่วงค่าคุณประโยชน์ที่ต่ำจะทำให้เห็นลักษณะการปรากฏขึ้นของรายการแบบมีนัยสำคัญ

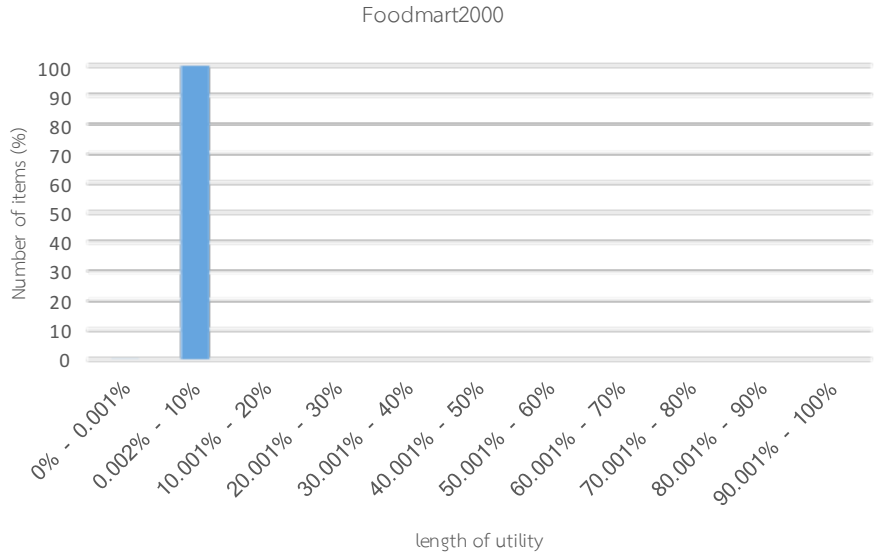
จากการพิจารณาฐานข้อมูล จะสังเกตได้ว่าฐานข้อมูล Chainstore และ Kosarak มีจำนวนรายการที่ปรากฏขึ้นในช่วงค่าคุณประโยชน์ 0%-0.001% มากกว่าช่วงค่าคุณประโยชน์ 0.002%-10% และไม่มีจำนวนรายการปรากฏขึ้นในช่วงค่าคุณประโยชน์อื่นๆเลย ในส่วนของฐานข้อมูล BMS และ Foodmart2000 แนวโน้มส่วนใหญ่ของจำนวนรายการจะปรากฏขึ้นในช่วงค่าคุณประโยชน์ 0.002%-10% และไม่มีการปรากฏขึ้นของรายการในช่วงค่าคุณประโยชน์ที่เกินกว่า 10% ในส่วนฐานข้อมูล Retail มีจำนวนรายการ 49% ปรากฏขึ้นในช่วงค่าคุณประโยชน์ 0%-0.001% และมีจำนวนรายการ 51% ปรากฏขึ้นในช่วงค่าคุณประโยชน์ 0.002%-1 จะเห็นได้ว่าทั้งสองช่วงค่าคุณประโยชน์มีจำนวนรายการปรากฏขึ้นใกล้เคียงกัน



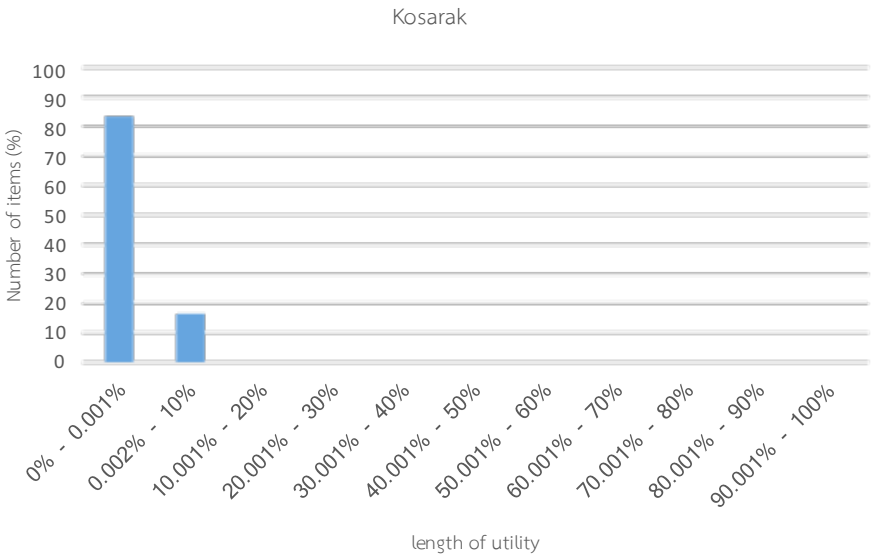
ภาพที่ 2-1 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าคุณประโยชน์ ของฐานข้อมูล BMS



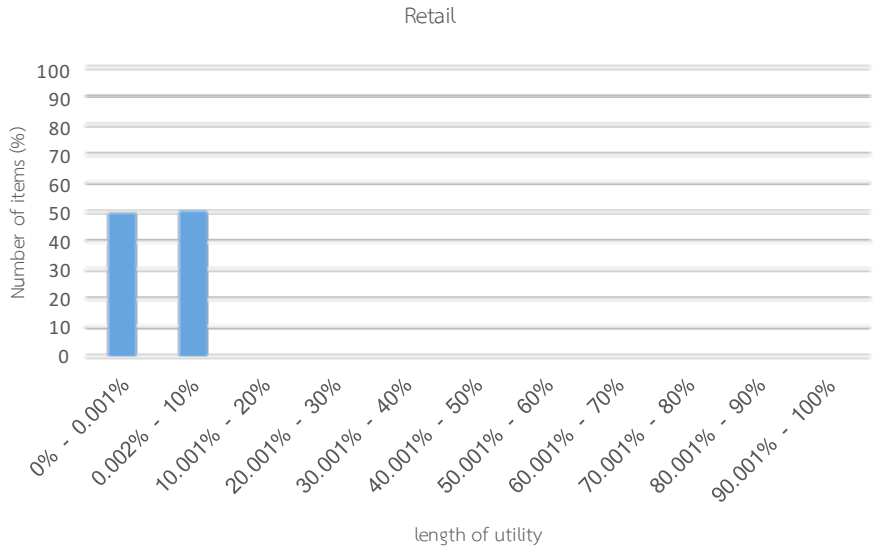
ภาพที่ 2-2 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าคุณประโยชน์ ของฐานข้อมูล Chainstore



ภาพที่ 2-3 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าคุณประโยชน์ ของฐานข้อมูล Foodmart2000



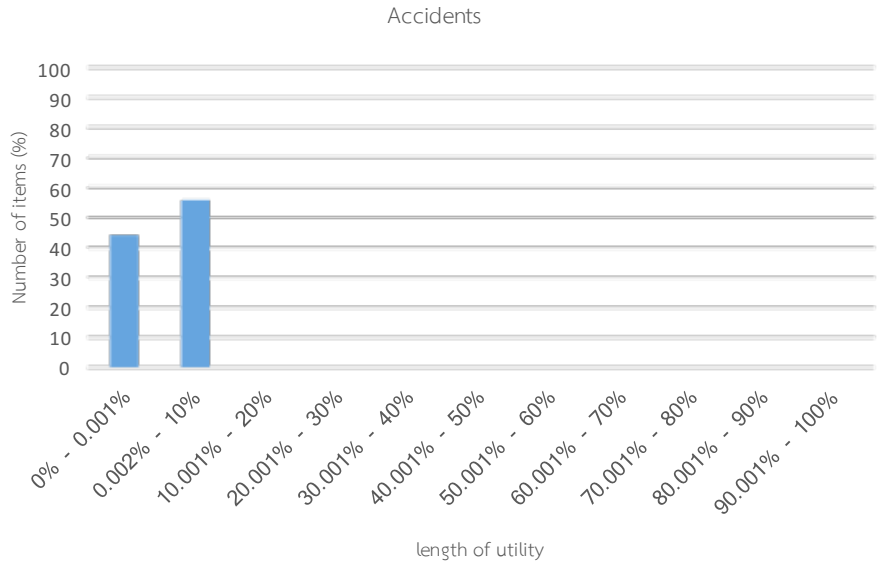
ภาพที่ 2-4 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าคุณประโยชน์ ของฐานข้อมูล Kosarak



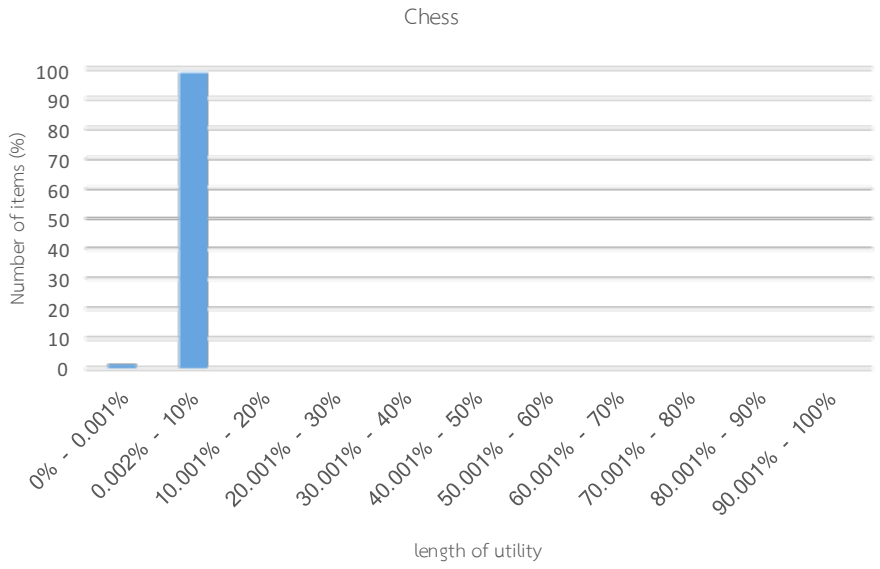
ภาพที่ 2-5 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าคุณประโยชน์ ของฐานข้อมูล Retail

จากภาพที่ 2-6 ถึง 2-10 แสดงชุดข้อมูลชนิดหนาแน่น 5 ชุดข้อมูล ประกอบด้วย Accidents, Chess, Connect, Mushroom, และ PUMSB ตามลำดับ โดยแบ่งช่วงค่าคุณประโยชน์ห่างกัน 11 ช่วง ได้แก่ 0%–0.001%, 0.002%–10%, 10.001%–20%, 20.001%–30%, 30.001%–40%, 40.001%–50%, 50.001%–60%, 60.001%–70%, 70.001%–80%, 80.001%–90% และ 90.001%–100% ซึ่งมีช่วงค่าคุณประโยชน์ต่ำสุด คือ 0%-0.001% การแบ่งช่วงค่าคุณประโยชน์ที่ต่ำจะทำให้เห็นลักษณะการปรากฏขึ้นของรายการแบบมีนัยสำคัญ

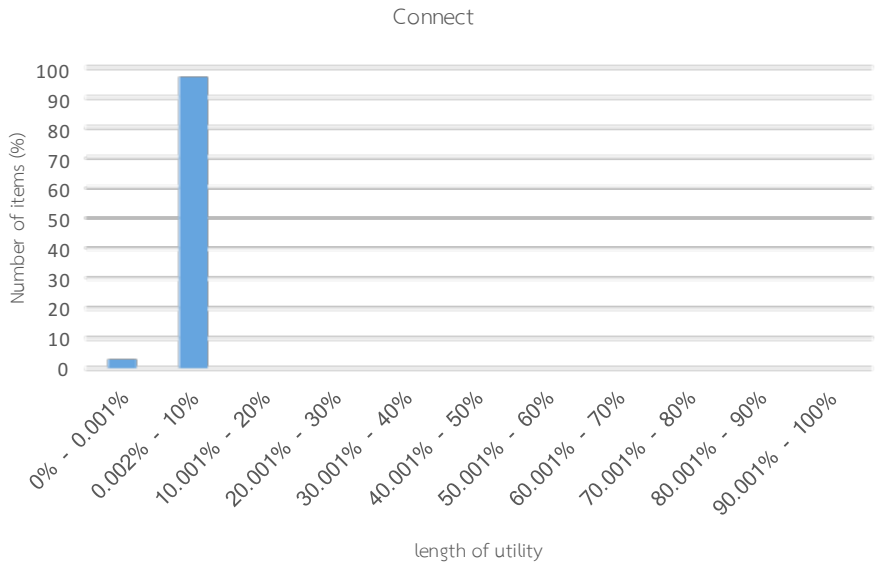
จากการพิจารณาฐานข้อมูล จะสังเกตได้ว่าฐานข้อมูล PUMSB มีจำนวนรายการถึง 86% ปรากฏขึ้นในช่วงค่าคุณประโยชน์ 0%-0.001% ซึ่งมีจำนวนรายการมากกว่าช่วงค่าคุณประโยชน์ 0.002%-10% และไม่มีรายการปรากฏขึ้นในช่วงค่าคุณประโยชน์อื่นๆ เลย ในส่วนของฐานข้อมูล Accidents, Chess, Connect และ Mushroom แนวโน้มส่วนใหญ่ของจำนวนรายการจะปรากฏขึ้นในช่วงค่าคุณประโยชน์ 0.002%-10% และไม่มีการปรากฏขึ้น ของรายการในช่วงค่าคุณประโยชน์ที่เกินกว่า 10%



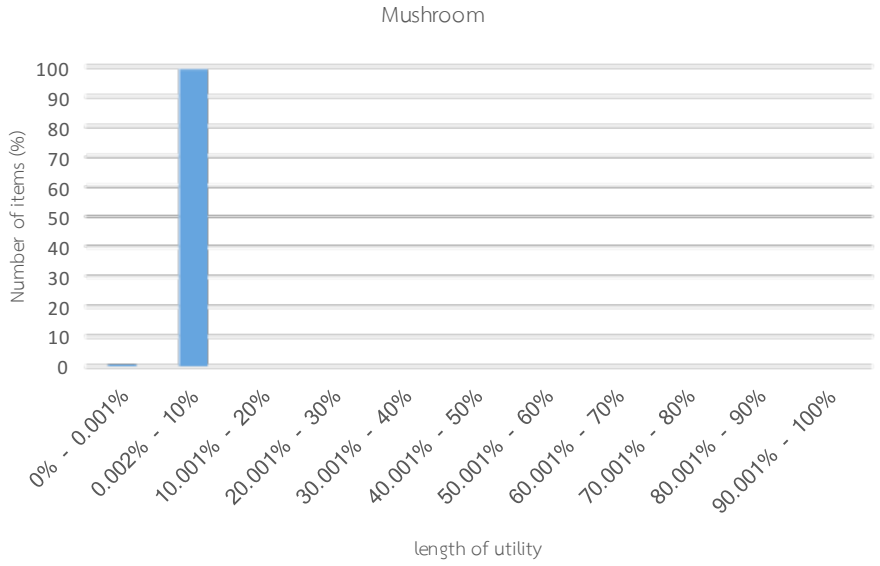
ภาพที่ 2-6 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าคุณประโยชน์ ของฐานข้อมูล Accidents



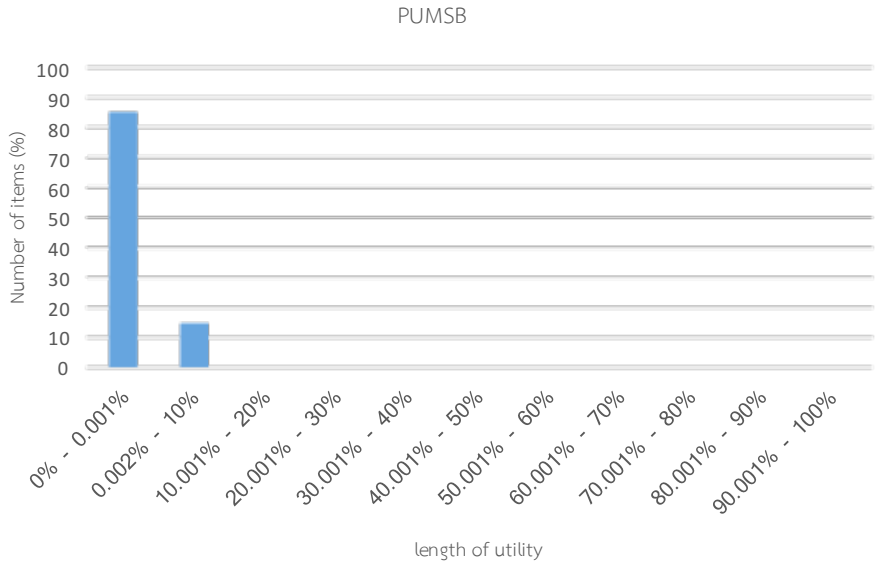
ภาพที่ 2-7 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าคุณประโยชน์ ของฐานข้อมูล Chess



ภาพที่ 2-8 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าคุณประโยชน์ ของฐานข้อมูล Connect



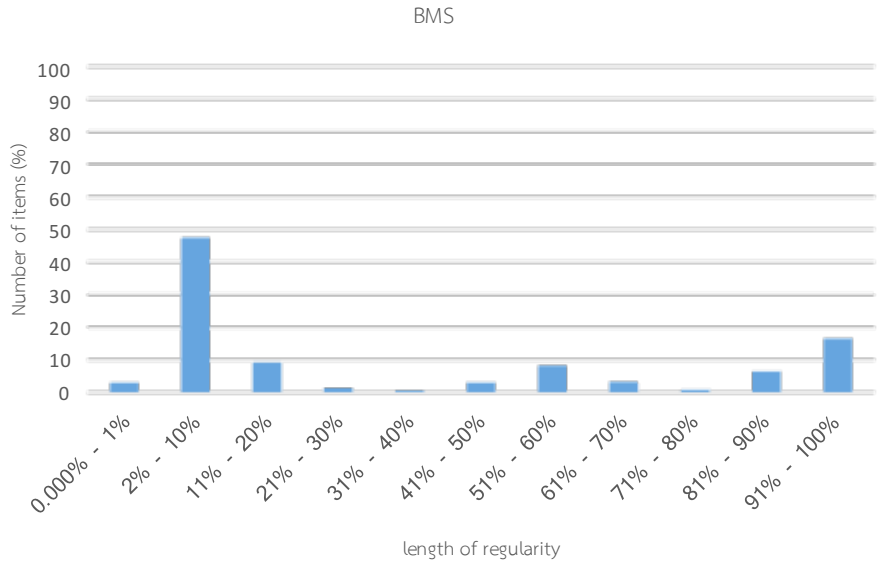
ภาพที่ 2-9 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าคุณประโยชน์ ของฐานข้อมูล Mushroom



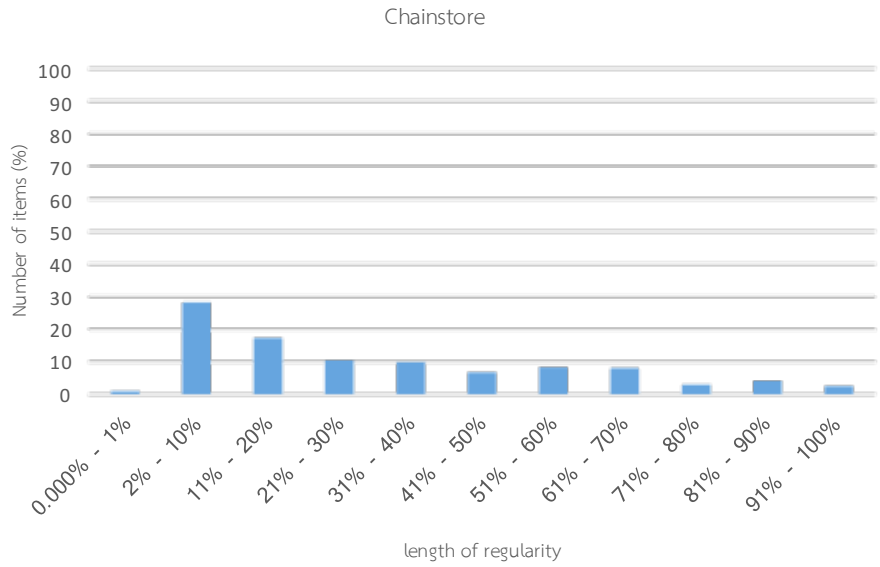
ภาพที่ 2-10 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าคุณประโยชน์ ของฐานข้อมูล PUMSB

จากภาพที่ 2-11 ถึง 2-15 แสดงชุดข้อมูลชนิดเบาบาง 5 ชุดข้อมูล ประกอบด้วย BMS, Chainstore, Foodmart2000, Kosarak และ Retail ตามลำดับ โดยแบ่งช่วงค่าความสม่ำเสมอห่างกัน 11 ช่วง ได้แก่ 0%–1%, 2%–10%, 11%–20%, 21%–30%, 31%–40%, 41%–50%, 51%–60%, 61%–70%, 71%–80%, 81%–90% และ 91%–100% ซึ่งมีช่วงค่าความสม่ำเสมอต่ำสุด คือ 0%-1% การแบ่งช่วงค่าความสม่ำเสมอที่ต่ำ จะทำให้เห็นลักษณะการปรากฏขึ้นของรายการแบบมีนัยสำคัญ

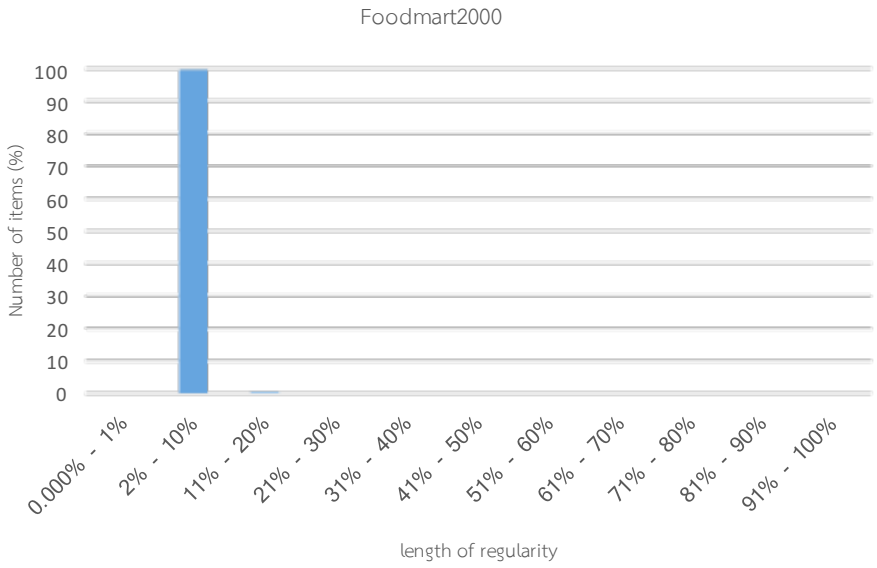
จากการพิจารณาฐานข้อมูล จะสังเกตได้ว่าฐานข้อมูล Foodmart2000 และ BMS แนวโน้มส่วนใหญ่ของรายการจะปรากฏขึ้นในช่วงค่าความสม่ำเสมอ 0%-1% เกือบทั้งหมด และฐานข้อมูล Chainstore, Kosarak และ Retail จะมีรายการปรากฏขึ้นทุกๆ ช่วงค่าความสม่ำเสมอ โดยแต่ละช่วงค่าความสม่ำเสมอจะมีจำนวนรายการปรากฏขึ้นใกล้เคียงกัน



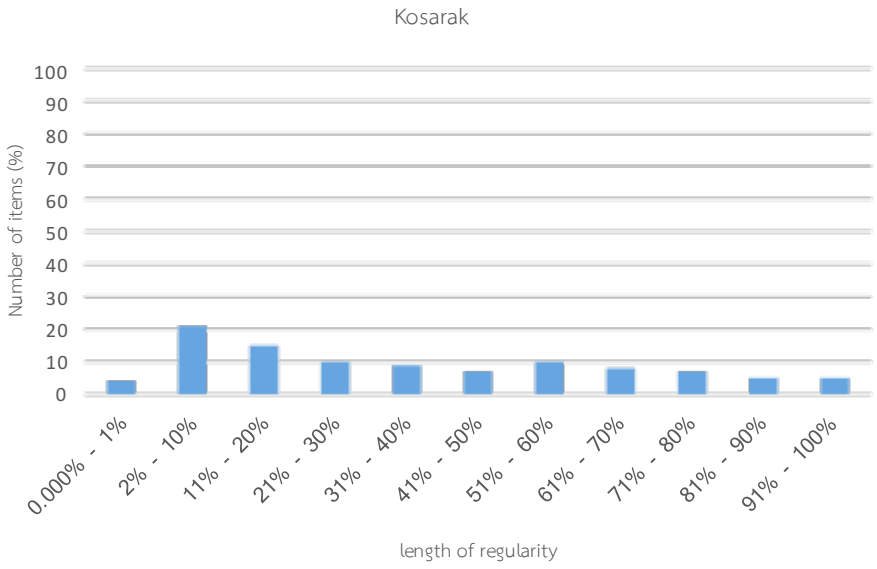
ภาพที่ 2-11 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าความสม่ำเสมอ ของฐานข้อมูล BMS



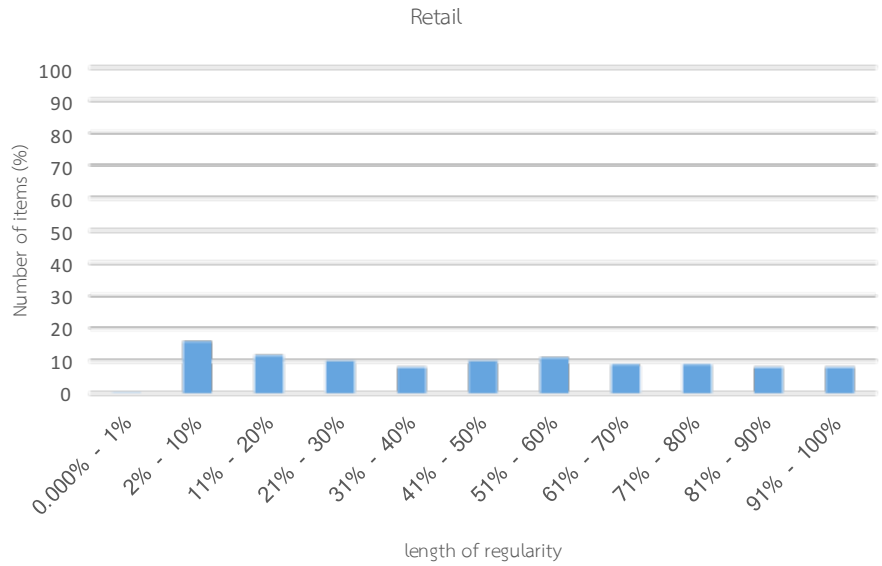
ภาพที่ 2-12 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าความสม่ำเสมอ ของฐานข้อมูล Chainstore



ภาพที่ 2-13 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าความสม่ำเสมอ ของฐานข้อมูล Foodmart2000



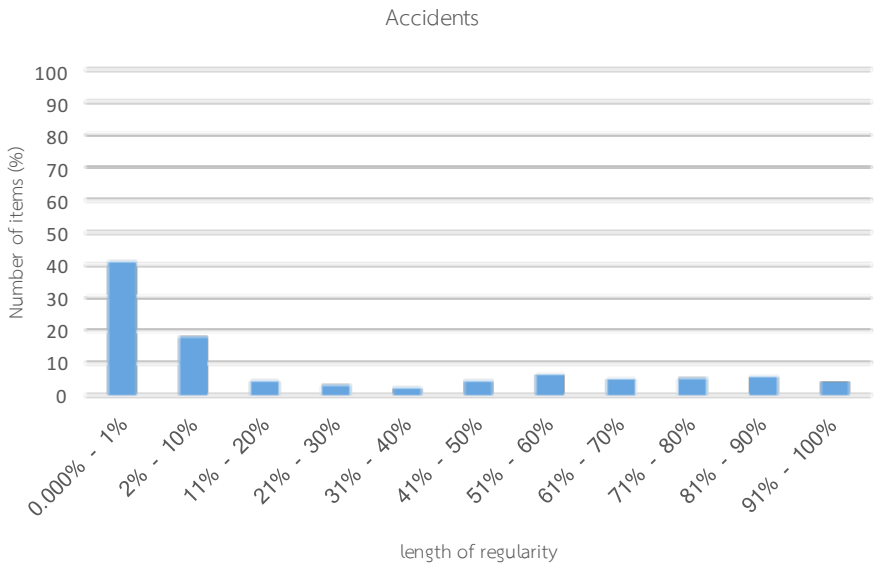
ภาพที่ 2-14 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าความสม่ำเสมอ ของฐานข้อมูล Kosarak



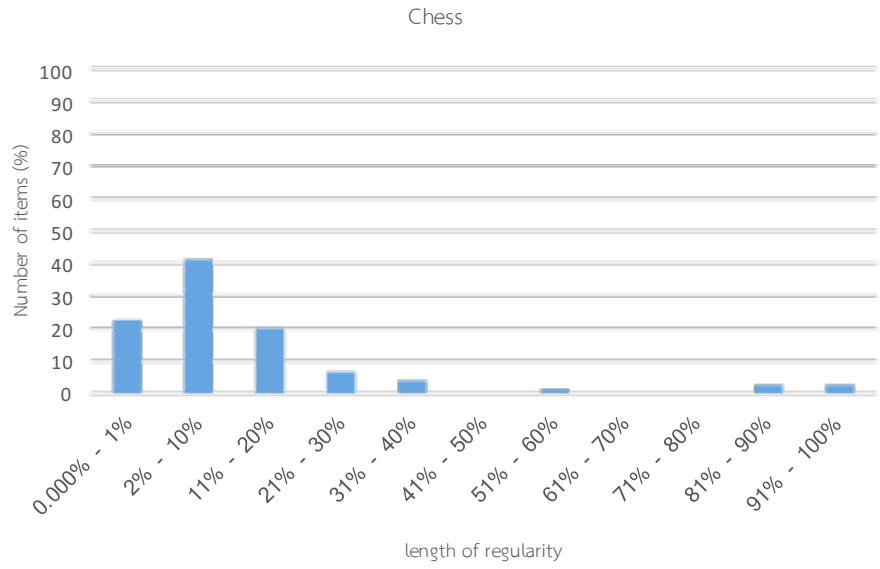
ภาพที่ 2-15 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าความสม่ำเสมอ ของฐานข้อมูล Retail

จากภาพที่ 2-16 ถึง 2-20 แสดงชุดข้อมูลชนิดหนาแน่น 5 ชุดข้อมูล ประกอบด้วย Accidents, Chess, Connect, Mushroom, และ PUMSB ตามลำดับ โดยแบ่งช่วงค่าความสม่ำเสมอห่างกัน 11 ช่วง ได้แก่ 0%–1%, 2%–10%, 11%–20%, 21%–30%, 31%– 40%, 41%–50%, 51%–60%, 61%–70%, 71%–80%, 81%– 90% และ 91%–100% ซึ่งมีช่วงค่าความสม่ำเสมอต่ำสุด คือ 0%-1% การแบ่งช่วงค่าความสม่ำเสมอที่ต่ำ จะทำให้เห็นลักษณะการปรากฏขึ้นของรายการแบบมีนัยสำคัญ

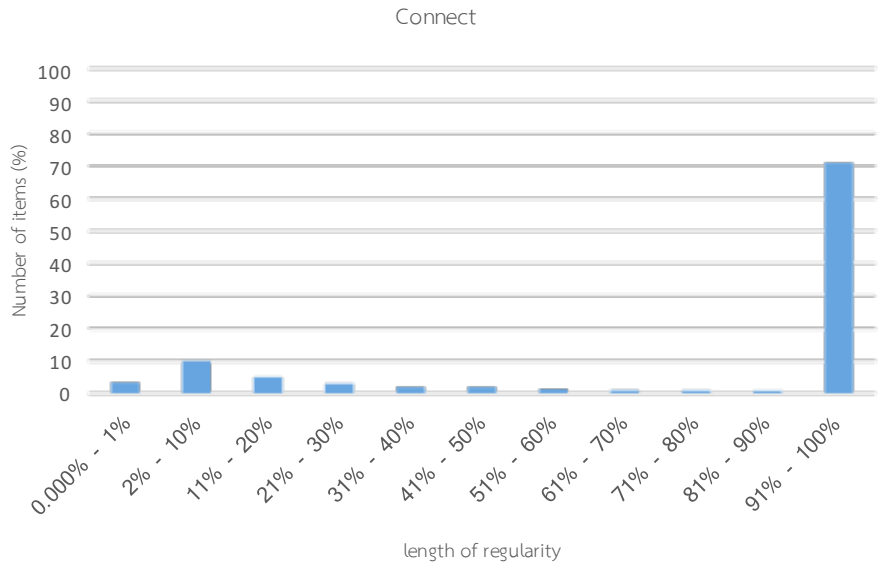
จากการพิจารณาฐานข้อมูล จะสังเกตได้ว่าฐานข้อมูล Connect มีจำนวนรายการ 72% ปรากฏขึ้นในช่วงค่าความสม่ำเสมอ 91%-100% และฐานข้อมูล Accidents, Chess, Mushroom, และ PUMSB จะมีรายการปรากฏขึ้นเกือบทุกๆช่วงค่าความสม่ำเสมอ โดยแต่ละช่วงจะมีจำนวนรายการปรากฏขึ้นใกล้เคียงกัน



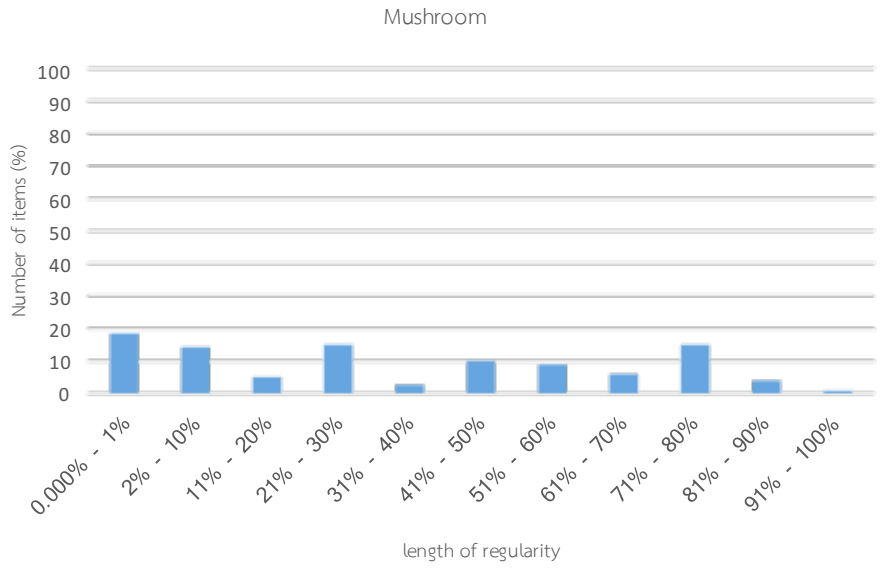
ภาพที่ 2-16 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าความสม่ำเสมอ ของฐานข้อมูล Accidents



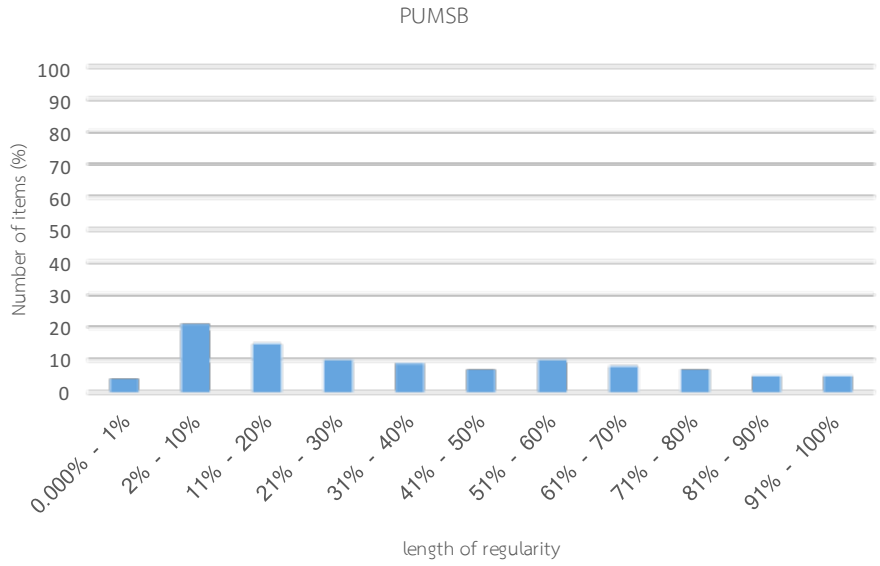
ภาพที่ 2-17 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าความสม่ำเสมอ ของฐานข้อมูล Chess



ภาพที่ 2-18 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าความสม่ำเสมอ ของฐานข้อมูล Connect



ภาพที่ 2-19 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าความสม่ำเสมอ ของฐานข้อมูล Mushroom



ภาพที่ 2-20 แผนภูมิแท่งแสดงถึงจำนวนรายการในแต่ละช่วงค่าความสม่ำเสมอ ของฐานข้อมูล PUMSB

บทที่ 3

การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงโดยปรากฏอย่างไม่สม่ำเสมอ

จากบทที่กล่าวมาในข้างต้น การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงเป็นการค้นหารูปแบบที่น่าสนใจภายใต้การพิจารณาค่าคุณประโยชน์ของเซตรายการ อาทิเช่น ผลกำไร-ขาดทุน ความเสี่ยง และ อื่นๆ โดยรูปแบบที่น่าสนใจจะเป็นรูปแบบที่มีค่าคุณประโยชน์สูงกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด แต่อย่างไรก็ตามการค้นหารูปแบบดังกล่าวไม่ได้พิจารณาพฤติกรรมการปรากฏขึ้น อาทิเช่น ความถี่ของการปรากฏ ความสม่ำเสมอของการปรากฏขึ้น เป็นต้น ด้วยเหตุนี้จึงได้มีนักวิจัยนำเสนอการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงโดยปรากฏขึ้นอย่างสม่ำเสมอ (*Mining High-Utility Itemsets with Regular Occurrence*, *MHUIR*) โดยทำการเพิ่มเติมเงื่อนไขของการปรากฏอย่างสม่ำเสมอ ร่วมกับการพิจารณาค่าคุณประโยชน์ของรายการต่างๆ แต่อย่างไรก็ตามการค้นหารูปแบบที่มีการปรากฏอย่างสม่ำเสมอ ยังอาจไม่เพียงพอสำหรับการค้นหารูปแบบใหม่ๆที่มีความหลากหลาย ด้วยเหตุนี้ในบทนี้จะนำเสนอการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงโดยปรากฏอย่างไม่สม่ำเสมอ *(Mining High-Utility Itemsets with Irregular Occurrence*, *MHUII*) โดยรูปแบบในลักษณะนี้จะสามารถบ่งบอกถึงพฤติกรรมผู้บริโภคเกี่ยวเนื่องกับการซื้อสินค้าที่มีค่าคุณประโยชน์สูงโดยสินค้าเหล่านั้นถูกซื้ออย่างไม่สม่ำเสมอ ตัวอย่างเช่น กล้อง, เลนส์ ซึ่งมักเป็นสินค้าที่ถูกซื้อร่วมกัน โดยถูกซื้อไม่สม่ำเสมอแต่ยังให้ผลกำไรที่สูง ซึ่งจะทำให้ผู้ประกอบการสามารถทราบถึงพฤติกรรมการซื้อสินค้าหรือบริการ และพฤติกรรมในเชิงลึกของผู้บริโภคได้ มีส่วนช่วยเป็นข้อมูลประกอบการตัดสินใจเกี่ยวกับการบริหารจัดการสินค้าคงคลัง การจัดทำโปรโมชั่นเพื่อส่งเสริมการขายและอื่นๆ

3.1 การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอ

จากนิยามพื้นฐานทั้งหมดในบทที่แล้ว การค้นหารูปแบบที่มีค่าคุณประโยชน์สูงจะเป็นการค้นหารูปแบบ X ใดๆ ที่มีค่าคุณประโยชน์ $u(X)$ มากกว่าหรือเท่ากับค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด (σ_u) แต่อย่างไรก็ตามในการลดทอนปริภูมิสถานะ (Liu, Y., และคณะ, 2005) จึงได้เสนอแนวคิดเกี่ยวกับ “*transaction-weighted utility*” *(TWU)* จะเป็นค่าประมาณของค่าคุณประโยชน์ที่สามารถประยุกต์ใช้ downward closure property ที่ซึ่งสามารถนิยามได้ดังนี้

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 $tu(t_1) = u(a,t_1) + u(b,t_1) + u(d,t_1) + u(f,t_1) = 4 + 9 +260 + 50 = 323$

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

ตัวอย่างที่ 5 จากฐานข้อมูลในตารางที่ 3-1 และตารางค่าคุณประโยชน์ที่ 3-2 เราจะสังเกตได้ว่าเซตรายการ *'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* ใดๆ มีค่า *TWU(X)* น้อยกว่าค่าขีดแบ่งคุณประโยชน์แล้ว จะทำให้ทุกซุปเปอร์เซตของ *X* มีค่าคุณประโยชน์น้อยกว่าค่าขีดแบ่งคุณประโยชน์ด้วยเช่นกัน ดังนั้น เราสามารถตัดการพิจารณาเซตรายการ *X* และทุกเซตรายการที่เป็นซุปเปอร์เซตของรายการ *X* ออกจากการพิจารณาได้เนื่องจาก เซตรายการ *X* และทุกเซตรายการที่เป็นซุปเปอร์เซตของ *X* จะมีค่าคุณประโยชน์น้อย

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

นิยามที่ 22 *กำหนดให้* $\succ$ *แสดงถึงลำดับของรายการในเซตรายการ I ที่ซึ่ง* $i_1 \prec i_2 \prec \dots i_n$ หมายถึง รายการ i_1 อยู่ในลำดับก่อนหน้ารายการ i_2 และรายการ i_2 อยู่ในลำดับก่อนหน้ารายการ i_3 และรายการ i_n เป็นรายการในลำดับสุดท้าย

นิยามที่ 23 *ค่าคุณประโยชน์ส่วนเหลือ (remaining utility) ของเซตรายการ X ในทรานแซกชันหมายถึงผลรวมของค่าคุณประโยชน์ของทุกรายการที่ปรากฏในทรานแซกชัน* t_p *และรายการเหล่านั้นมีลำดับหลังจาก X สามารถคำนวณและแทนด้วยสัญลักษณ์* $ru(X,t_p)=\sum_{i_j \in t_p, X \prec i_j} u(i_j,t_p)$

ตัวอย่างที่ 6 จากฐานข้อมูลในตารางที่ 3-1 และตารางค่าคุณประโยชน์ที่ 3-2 ถ้าลำดับของรายการทั้งหมดเป็น $a \prec b \prec c \prec d \prec e \prec f \prec g \prec h$ 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 กับค่าคุณประโยชน์ส่วนที่เหลือของเซตรายการ X ในฐานข้อมูลรายการ TDB สามารถคำนวณและแทนด้วยสัญลักษณ์ tou(X) = u(X) + ru(X)*

ตัวอย่างที่ 8 จากฐานข้อมูลตารางที่ 3-1 และตารางค่าคุณประโยชน์ที่ 3-2 ค่าประมาณคุณประโยชน์แบบกระชับของเซตรายการ *'ad'* สามารถคำนวณได้เป็น *tou(ad) = 286 +50 = 336*

ดังที่กล่าวมาข้างต้นเราสามารถระบุได้ว่า ถ้าเซตรายการ *X* มีค่าประมาณคุณประโยชน์แบบกระชับน้อยกว่าค่าขีดแบ่งคุณประโยชน์แล้ว เซตรายการ *Y* ใดที่เกิดจากการรวมกันระหว่างเซตรายการ *X* และรายการ i_j ใดๆ ที่มีลำดับหลังจาก *X* กล่าวคือ $Y = X \cup i_j$ จะมีค่าคุณประโยชน์น้อยกว่าค่าขีดแบ่งคุณประโยชน์เสมอ

จากแนวคิดและนิยามข้างต้น สามารถช่วยให้ลดทอนปริภูมิสถานะของการพิจารณาเซตรายการที่มีค่าคุณประโยชน์สูงได้ แต่อย่างไรก็ตามเมื่อทำการพิจารณาการปรากฏอย่างไม่สม่ำเสมอร่วมกับค่าคุณประโยชน์ จะทำให้ปัญหาการค้นหาเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏไม่สม่ำเสมอสามารถนิยามได้ดังนี้

นิยามที่ 26 *เซตรายการ X จะเป็นเซตรายการที่มีคุณค่าคุณประโยชน์สูงและปรากฏไม่สม่ำเสมอก็ต่อเมื่อ X มีค่าคุณประโยชน์ u(X) ไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด (σ_u) และ X มีค่าความสม่ำเสมอ r^X มากกว่าค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด (σ_r) ที่ผู้ใช้กำหนด*

3.2 ขั้นตอนวิธีการนำเสนอ *HUIIM* และ *Efficient Pruning technique*

งานวิจัยนี้ได้นำเสนอขั้นตอนที่มีชื่อว่า *"High Utility Itemset with Irregular occurrence Miner, HUIIM"* ที่สามารถค้นหารูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอจากฐานข้อมูล ภายใต้ค่าขีดคุณประโยชน์ที่ผู้ใช้กำหนด *(σ_u)* และค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด *(σ_r)* ที่ซึ่งวิธีนี้จะทำการอ่านข้อมูลจากฐานข้อมูลเพียงครั้งเดียว โดยจะทำการคำนวณและจัดเก็บค่าคุณประโยชน์ของทุก ทรานแซกชันในฐานข้อมูลไว้ใน *tuList* นอกจากนั้น *HUIIM* ประยุกต์ใช้แนวคิดค่าประมาณคุณประโยชน์ และค่าประมาณคุณประโยชน์แบบกระชับของรูปแบบในการลดทอนปริภูมิสถานะของการพิจารณารูปแบบ และการใช้โครงสร้างข้อมูล *"New-modified Utility List, NUL"* ที่ได้ปรับปรุงใหม่เพื่อจัดเก็บข้อมูลหมายเลขกำกับทรานแซกชันพร้อมกับค่าคุณประโยชน์ของแต่ละรายการ นอกจากนี้เทคนิคใหม่ที่เรียกว่า *"Efficient Pruning technique"* ถูกออกแบบมาใช้กับ *HUIIM* เพื่อเพิ่มประสิทธิภาพในการคำนวณและเรียกขั้นตอนวิธี *HUIIM* ที่มีการประยุกต์ใช้ *"Efficient Pruning technique"* ว่า *"Efficient High Utility Itemset with Irregular occurrence Miner, EHUIIM"* สามารถค้นหารูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอจากฐานข้อมูลรายการได้อย่างมีประสิทธิภาพ

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

โครงสร้างข้อมูล *NUL* ของเซตรายการ *X* หนึ่งๆ จะประกอบไปด้วยลิสต์ของ 4-tuple คือ $<p, u(X, t_p), ru(X, t_p), up(X, t_p)>$ 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 $X = \{i_p, ..., i_y, i_z\}$ prefix items ของเซตรายการ X จะเป็น $X - i_z = \{i_p, ..., i_y\}$)

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 *{<1, 13, 310, 4>, <3, 12, 0, 6>, <5, 8, 20, 2>}* โดยที่สมาชิกอันดับแรกของ NUL^{ad} ปรากฏในทรานแซกชันที่ t_1 เซตรายการ *'ab'* มีค่าคุณประโยชน์ $u(ad,t_1) = 13$ ค่าคุณประโยชน์ส่วนเหลือ $ru(ab,t_1) = 310$ และค่าคุณประโยชน์ของรายการ prefix $up(ab,t_1) = u(a,t_1) = 4$ สำหรับการคำนวณต่อไปตามลำดับ

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

3.3 ขั้นตอนวิธี *HUIIM*

ดังที่กล่าวมาในข้างต้น ขั้นตอนวิธี *HUIIM* ได้ประยุกต์แนวคิดค่าคุณประโยชน์ส่วนเหลือและค่าประมาณคุณประโยชน์แบบกระชับเพื่อทำการลดทอนปริภูมิสถานะ นอกจากนั้นยังทำการประยุกต์ใช้โครงสร้างข้อมูล *New-modified Utility List* เพื่อใช้ในการจัดเก็บข้อมูลเกี่ยวกับการปรากฏขึ้นของรูปแบบหนึ่งๆ พร้อมค่าคุณประโยชน์ของรูปแบบนั้นๆ ที่ปรากฏในทรานแซกชันหนึ่งๆ ระหว่างค้นหาผลลัพธ์ *HUIIM* เพื่อลดการอ่านฐานข้อมูลให้เหลือเพียงครั้งเดียว โดยขั้นตอนวิธี *HUIIM* มี 2 ขั้นตอนหลักในการคำนวณหาผลลัพธ์คือ

1) การระบุถึง single-item ที่ปรากฏอย่างไม่สม่ำเสมอและมีค่าประมาณคุณประโยชน์สูง จะทำการสร้างต้นไม้เพื่อใช้เก็บรายการและเซตของรายการที่มีคุณสมบัติข้างต้น (เรียกว่า *HUII-tree*) โดยการระบุถึงรายการที่ปรากฏไม่สม่ำเสมอและค่าประมาณคุณประโยชน์สูง จากการอ่านฐานข้อมูลหนึ่งครั้งจะถูกจัดเก็บไว้ในโหนดลูกของ *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 ϕ
5: **for** each trasaction 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*

pruning of low-utility items

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: $HUII \leftarrow HUII \cup i_k$

ภาพที่ 3-1 ขั้นตอนการอ่านฐานข้อมูลและระบุรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอ

ดังแสดงรายละเอียดในภาพที่ 3-1 “*HUIIM-DBscanning*” ขั้นตอนการอ่านฐานข้อมูล จะเริ่มจากการสร้างหรือจองพื้นที่หน่วยความจำในการจัดเก็บข้อมูลเบื้องต้นดังนี้ ทำการสร้าง “*transaction-utility list*” *(tuList)* ลิสต์จะมีขนาดเท่ากับจำนวนทรานแซกชันของฐานข้อมูล 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^{i_j} ที่อยู่ในรูปแบบ $<p, u(i_j,t_p), 0, 0>$ จากนั้นทำการอัพเดตค่าประมาณคุณประโยชน์ $TWU(i_j)$ (สามารถคำนวณได้จาก $TWU(i_j) = TWU(i_j) + tuList(t_p)$) และจัดเก็บข้อมูลทั้งหมดไว้ในโหนดลูกของ *R* ของ *HUII-tree*

เมื่อทำการอ่านข้อมูลครบทุกทรานแซกชัน จะทำการตรวจสอบค่าประมาณคุณประโยชน์ของแต่ละ i_j (ที่เป็นโหนดลูกของ *R* ของ *HUII-tree*) โดยดูว่าถ้าค่าประมาณคุณประโยชน์ $TWU(i_j)$ มีค่าน้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด จะทำการลบรายการ i_j ออกจากการพิจารณา (<u>หมายเหตุ</u> รายการ i_j ที่มีค่าประมาณคุณประโยชน์น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด จะทำให้ซุปเปอร์เซตของ i_j มีค่าคุณประโยชน์น้อยกว่าค่าขีดแบ่งคุณประโยชน์ด้วยเช่นกัน เราสามารถตัดการพิจารณารายการ i_j และ เซตรายการที่เป็นซุปเปอร์เซตของ i_j ออกจากการพิจารณาได้) แต่อย่างไรก็ตาม ก่อนที่จะลบรายการ i_j ออกจากการพิจารณาจะทำการลดทอนค่าคุณประโยชน์ของแต่ละทรานแซกชันที่มีรายการ i_j ปรากฏ เพื่อทำการลดทอนค่าประมาณคุณประโยชน์ โดยทำการพิจารณาแต่ละสมาชิกใน NUL^{i_j} ที่ซึ่งมีลักษณะเป็น $<p, u(i_j,t_p), 0, 0>$ และทำการลดทอนค่า $tu(t_p)$ ที่ถูกจัดเก็บใน $tuList(t_p)$ ด้วยค่า $u(i_j,t_p)$ และเมื่อทำการพิจารณาทุกสมาชิกใน NUL^{i_j} จนครบแล้วจะทำการลบข้อมูลทั้งหมดของ i_j ที่ถูกจัดเก็บอยู่ในโหนดลูกของ *R* ของ *HUII-tree* ออกจากหน่วยความจำและการพิจารณาได้ (บรรทัดที่ 11-15)

หลังจากลบรายการที่มีค่าประมาณคุณประโยชน์ต่ำ ขั้นตอนต่อไปจะเป็นการคำนวณค่าคุณประโยชน์ส่วนเหลือในแต่ละสมาชิกใน NUL^{i_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 $<p, u(i_j, t_p), 0, 0>$ ใน NUL^{i_j} แล้วทำการอัพเดตค่าคุณประโยชน์ของทรานแซกชัน t_p ที่ถูกจัดเก็บอยู่ใน *temptuList* ด้วย $temptuList(t_p) = temptuList(t_p) - u(i_j,t_p)$ เพื่อที่จะทราบถึงค่าคุณประโยชน์ของทุกรายการที่อยู่ในลำดับถัดไปจากรายการ i_j ในทรานแซกชัน t_p ซึ่งค่า $temptuList(t_p)$ หลังจากอัพเดทจะหมายถึงค่าคุณประโยชน์ส่วนเหลือ $ru(i_j,t_p)$ โดยเมื่อทราบถึงค่าดังกล่าวแล้ว จะทำการอัพเดทแต่ละสมาชิกใน NUL^{i_j} ที่พิจารณาให้มีค่าเป็น $<p, u(i_j,t_p), ru(i_j,t_p), 0>$ และทำการอัพเดทค่าคุณประโยชน์ $u(i_j)$ ด้วย $u(i_j,t_p)$ และค่าคุณประโยชน์ส่วนเหลือ $ru(i_j)$ ด้วย $ru(i_j,t_p)$ ตามลำดับ หลังจากพิจารณาทุกสมาชิกใน NUL^{i_j} จนครบ ถ้าค่าคุณประโยชน์ $u(i_j)$ มีค่าไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด และค่าความสม่ำเสมอ $r(i_j)$ มากกว่าค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด "*HUIIM-DBscanning*" จะทำการระบุและจัดเก็บรายการ i_j ว่าเป็นรูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอ และจัดเก็บรายการดังกล่าวไว้ในเซต *HUII* (บรรทัดที่ 24-25)

Algorithm 2: *HUIIM-Mining*

Input: *HUII-tree*, σ_u, σ_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: $NUL^{i_j i_k} \leftarrow$ 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, σ_u, σ_r)
13: **Procedure** MiningAllHUII(*HUII-tree*, node of X, σ_u, σ_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: $NUL^{YZ} \leftarrow$ 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) โดยในขั้นแรกของ "*HUIIM-Mining*" จะเป็นการค้นหาเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอที่ประกอบไปด้วย 2 รายการ (บรรทัดที่ 1-9) โดยเริ่มต้นจากการพิจารณาเฉพาะรายการ i_j (ที่เป็นโหนดลูกของ *R* ของ *HUII-tree*) และทำการตรวจสอบค่าคุณประโยชน์แบบกระชับ $tou(i_j) = u(i_j) + ru(i_j)$ 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} เข้าด้วยกันเพื่อทำการจัดเก็บข้อมูลการปรากฏขึ้น 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 $r(i_j\, i_k)$

หลังจากคำนวณค่าคุณประโยชน์ ค่าคุณประโยชน์ส่วนเหลือ ค่าประมาณคุณประโยชน์ ค่าความสม่ำเสมอ ของเซตรายการ '$i_j\, i_k$' แล้วจะทำการตรวจสอบค่าประมาณคุณประโยชน์ $TWU(i_j\, i_k)$ ถ้า $TWU(i_j\, i_k)$ มีค่ามากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด จะทำการสร้างเซตรายการ '$i_j\, i_k$' พร้อมทั้งข้อมูลทั้งหมดที่เกี่ยวข้องกับเซตรายการ '$i_j\, i_k$' ไว้ในโหนดลูกของรายการ i_j และทำการตรวจสอบค่าคุณประโยชน์ $u(i_j\, i_k)$ ที่ซึ่งมีค่าไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด และค่าความสม่ำเสมอ $r(i_j\, i_k)$ ที่ซึ่งมากกว่าค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด ขั้นตอนวิธี "*HUIIM-Mining*" จะทำการระบุว่าเซตรายการ '$i_j\, i_k$' ว่าเป็นรูปแบบที่มีค่าคุณประโยชน์สูงโดยปรากฏอย่างไม่สม่ำเสมอและทำการจัดเก็บเซตรายการ '$i_j\, i_k$' ใน *HUII* โดยหลังจากทำการรวมรายการ i_j เข้ากับรายการ i_k แล้ว ขั้นตอนต่อไปทำการรวมรายการ i_j เข้ากับรายการ i_l ที่อยู่ในลำดับถัดไปจากรายการ i_k และจะดำเนินการตามกระบวนการต่างๆข้างต้น เมื่อทำการรวมรายการ i_j เข้ากับทุกโหนดลูกของ *R* ของ *HUII-tree* ทั้งหมด เราจะได้โหนดลูกของรายการ i_j ที่บรรจุไปด้วยเซตรายการที่ประกอบไปด้วย 2 รายการ และเป็นเซตรายการ i_j เป็นรายการขึ้นต้น และทำการวนซ้ำการทำงานตามลำดับของรายการที่เป็นสมาชิกของเซต *I* จนครบทั้งหมด เราจะได้ *HUII-tree* ที่บรรจุไปด้วยเซตรายการ 2 รายการ ที่มีค่าประมาณคุณประโยชน์สูง

หลังจากสร้าง *HUII-tree* ที่บรรจุไปด้วยเซตรายการ 2 รายการทั้งหมด ขั้นตอนต่อไปจะเป็นการค้นหาเซตรายการที่มากกว่า 2 รายการ "*MiningALLHUII*" (บรรทัดที่ 10-23) โดยเริ่มแรกทำการพิจารณารายการตามลำดับ i_j ที่มีจำนวนโหนดลูกมากกว่า 1 ต่อไปจะทำการพิจารณาแบบเรียงตามลำดับโหนดลูกของรายการ i_j คือ { '$i_j\, i_k$', '$i_j\, i_l$', ..., '$i_j\, i_m$'} 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 รายการสุดท้ายของเซตรายการ ‘$i_j\, i_k$’ คือ i_k และรายการสุดท้ายของเซตรายการ ‘$i_j\, i_l$’ คือ i_l เพื่อสร้างการพิจารณาโหนดของเซตรายการ ‘$i_k\, i_l$’ และทำการตรวจสอบโหนดของเซตรายการ ‘$i_k\, i_l$’ ในโครงสร้าง *HUII-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 *HUII-tree* จะทำการการอินเทอร์เซกชัน $NUL^{i_j i_k}$ กับ $NUL^{i_j i_l}$ เพื่อทำการจัดเก็บข้อมูลการปรากฏขึ้น ค่าคุณประโยชน์ ค่าคุณประโยชน์ส่วนเหลือสำหรับทรานแซกชันหนึ่งๆที่เซตรายการ ‘$i_j\, i_k\, i_l$’ ปรากฏ หลังจากขั้นตอนอินเทอร์เซกชันเสร็จสิ้น เราจะได้ $NUL^{i_j i_k i_l}$ ที่จะสามารถใช้ $NUL^{i_j i_k i_l}$ 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 $TWU(i_j\, i_k\, i_l)$ ที่ซึ่งถ้ามีค่ามากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด “*MiningALLHUII*” จะทำการสร้างเซตรายการ ‘$i_j\, i_k\, i_l$’ พร้อมทั้งข้อมูลทั้งหมดที่เกี่ยวข้องกับเซตรายการ ‘$i_j\, i_k\, i_l$’ ไว้ในโหนดลูกของรายการ ‘$i_j\, i_k$’ และทำการตรวจสอบค่าคุณประโยชน์ $u(i_j\, i_k\, i_l)$ ที่ซึ่งถ้ามีค่าไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด และค่าความสม่ำเสมอ $r(i_j\, i_k\, i_l)$ ที่ซึ่งมีค่ามากกว่าค่าขีดแบ่งความสม่ำเสมอ ขั้นตอนวิธี “*MiningALLHUII*” จะทำการระบุ 2 เซตรายการ ‘$i_j\, i_k\, i_l$’ เป็นรูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอ และจัดเก็บเซตรายการดังกล่าวในเซต *HUII*

เมื่อทำการรวมเซตรายการ ‘$i_j\, i_k$’ เข้ากับ ‘$i_j\, i_l$’ เสร็จสิ้น ขั้นตอนวิธี “*MiningALLHUII*” จะดำเนินการรวมเซตรายการ ‘$i_j\, i_k$’ เข้ากับ ‘$i_j\, i_m$’ ที่ถูกบรรจุไว้ในโหนดลูกของรายการ i_j ในลำดับหลังของเซตรายการ ‘$i_j\, i_l$’ เมื่อทำการรวมรายการเข้ากับทุกรายการ เราจะได้โหนดลูกของรายการ ‘$i_j\, i_k$’ ที่บรรจุไปด้วยเซตรายการที่ประกอบไปด้วย 3 รายการ และเป็นเซตรายการ ‘$i_j\, i_k$’ ขึ้นต้น จากนั้นจะทำการตรวจสอบจำนวนเซตรายการที่บรรจุอยู่ในโหนดลูกของรายการ ‘$i_j\, i_k$’ ซึ่งถ้ามีจำนวนเซตรายการมากกว่า 1 จะทำการวนซ้ำการทำงานเพื่อหาเซตรายการที่ประกอบไปด้วย 4 รายการ โดยการดำเนินการจะดำเนินการเช่นเดียวกับการพิจารณาเซตรายการที่ มากกว่า 2 รายการ (ส่วนของ *MiningALLHUII*)

3.4 เทคนิค *efficient pruning technique*

จากรายละเอียดในภาพขั้นตอนที่ 3-1 วิธีการ *HUIIM* สามารถลดทอนจำนวนรายการที่ต้องทำการพิจารณาได้ จากการตรวจสอบค่าประมาณคุณประโยชน์ของแต่ละรายการ $TWU(i_j)$ ที่ซึ่งถ้าค่าประมาณคุณประโยชน์มีค่าน้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด จะทำการลบรายการ i_j ออกจากการพิจารณา (บรรทัดที่ 11-15) (<u>หมายเหตุ</u> รายการ i_j ใดๆมีค่า $TWU(i_j)$ น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนดแล้ว จะทำให้ซุปเปอร์เซตของ i_j มีค่าคุณประโยชน์น้อยกว่าค่าขีดแบ่ง

คุณประโยชน์ด้วยเช่นกัน เราสามารถตัดการพิจารณารายการ i_j และเซตรายการที่เป็นซุปเปอร์เซตของรายการ i_j ออกจากการพิจารณาได้ เนื่องจากรายการ i_j และทุกเซตรายการที่เป็นซุปเปอร์เซตของ i_j จะมีคุณประโยชน์น้อย) อย่างไรก็ตามจากการลบรายการ i_j ออกจากการพิจารณาข้างต้น ก็ยังคงมีรายการที่มีค่าประมาณคุณประโยชน์ต่ำที่ยังไม่ได้ทำการลบออกจากการพิจารณา ที่ซึ่งเป็นสาเหตุจึงทำให้ *HUIIM* ต้องใช้เวลาค่อนข้างสูงในการคำนวณเซตรายการที่มีค่าคุณประโยชน์ต่ำ ด้วยเหตุผลนี้เราจึงได้ทำการเพิ่มประสิทธิภาพการคำนวณของ *HUIIM* โดยการเพิ่มเทคนิคใหม่ที่เรียกว่า "*Efficient Pruning technique*" ถูกออกแบบมาและประยุกต์ใช้กับ *HUIIM* โดยเรียกว่า *"Efficient High Utility Itemsets with Irregular occurrence Miner, EHUIIM"* โดยในขั้นตอนการลบรายการ i_j ที่มีค่าประมาณคุณประโยชน์ต่ำแบบเดิมของวิธี *HUIIM* (ภาพขั้นตอนที่ 3-1 บรรทัดที่ 11-15) จะถูกแทนที่ด้วยวิธีการใหม่ "*Efficient Pruning technique*" ขั้นตอนโดยรายละเอียดจะกล่าวในส่วนต่อไป

Algorithm 3: *Efficient Pruning technique*

Input: *HUII-tree*, σ_u, σ_r
Output: *HUII-tree* , *HUII*

1: create a list named *TUL* and initial the first entry to be *0*
2: create a list named *LU* with $|I|$ entries and initial all entries to be *0*
3: initial *NLI* $\leftarrow$ *0*
4: **repeat**
5: *numChild* $\leftarrow$ the current number of children of *R*
6: **for** each child node of root *R* with item i_k **do**
7: **if** $TWU(i_k) < \sigma_u$ **then**
8: $NLI \leftarrow NLI+1$
9: $TUL(NLI) \leftarrow TUL(NLI-1)$
10: **for** each entry $e = \langle p, u(i_k, t_p), 0, 0\rangle$ in NUL^{i_k} **do**
11: $tuList(t_p) \leftarrow tuList(t_p) - u(i_k, t_p)$
12: $TUL(NLI) \leftarrow TUL(NLI) + u(i_k, t_p)$
13: remove entry of item i_k out of child node of *R*
14: **else**
15: **if** $(TWU(i_k) - (TUL(NLI) - TUL(LU(i_k))) < \sigma_u$ **then**
16: $TWU(i_k) \leftarrow 0$
17: **for** each entry $e = \langle p, u(i_k, t_p), 0, 0\rangle$ in NUL^{i_k} **do**
18: $TWU(i_k) \leftarrow TWU(i_k) + tuList(t_p)$
19: **if** $TWU(i_k) < \sigma_u$ **then**
20: $NLI \leftarrow NLI+1$
21: $TUL(NLI) \leftarrow TUL(NLI-1)$
22: **for** each entry $e = \langle p, u(i_k, t_p), 0, 0\rangle$ in NUL^{i_k} **do**
23: $tuList(t_p) \leftarrow tuList(t_p) - u(i_k, t_p)$
24: $TUL(NLI) \leftarrow TUL(NLI) + u(i_k, t_p)$
25: remove entry of item i_k out of child node of *R*
26: **else**
27: $LU(i_k) \leftarrow NLI$
28: **until** *numChild* $\neq$ the current number of children of root *R*

ภาพที่ 3-3 ขั้นตอน "*Efficient Pruning technique*"

ดังแสดงในภาพที่ 3-3 ขั้นตอนวิธี *"Efficient Pruning technique"* จะเริ่มจากการสร้างหรือจองพื้นที่หน่วยความจำในการจัดเก็บข้อมูลเบื้องต้นคือ 1) ลิสต์ *"Total Utility of Low-utility itemsets, TUL"* ที่ใช้ในการจัดเก็บผลรวมของรายการที่มีค่าคุณประโยชน์ต่ำ และ 2) ลิสต์ *"Last Update, LU"* ที่ใช้ในการจัดเก็บการอัพเดตจำนวนรอบครั้งสุดท้ายของแต่ละรายการ โดยกำหนดให้ขนาดของลิสต์มีขนาดเท่ากับจำนวนรายการในฐานข้อมูล และ 3) ตัวแปรอินเด็กซ์ *"Number of low-utility items, NLI"* ที่ใช้นับจำนวนรายการที่มีค่าคุณประโยชน์ต่ำ เริ่มต้นมีค่าเท่ากับ *0* (บรรทัดที่ 1-3)

ขั้นตอนต่อไปทำการพิจารณาเฉพาะรายการ i_j (ที่เป็นโหนดลูกของ *R* ของ *HUII-tree*) และทำการตรวจสอบค่าประมาณคุณประโยชน์ $TWU(i_j)$ ที่ซึ่งถ้ามีค่าน้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด (บรรทัดที่ 7-13) ค่า *NLI* จะถูกเพิ่มค่าขึ้น *1* และทำการกำหนดค่า *TUL*(*NLI*) ให้มีค่าเท่ากับค่า *TUL*(*NLI – 1*) จากนั้นจะทำการลบรายการ i_j ออกจากการพิจารณา แต่ก่อนที่จะทำการลบรายการ i_j จะทำการลดทอนค่าคุณประโยชน์แต่ละทรานแซกชันที่มีรายการ i_j ปรากฏ โดยทำการพิจารณาแต่ละสมาชิกใน NUL^{i_j} ที่ซึ่งมีลักษณะเป็น $<p, u(i_j,t_p), 0, 0>$ ทำการลดทอนค่า $tuList(t_p)$ ด้วยค่า $u(i_j,t_p)$ (สามารถคำนวณได้เป็น $tuList(t_p) = tuList(t_p) - u(i_j,t_p)$) และทำการอัพเดทค่า *TUL*(*NLI*) ด้วยค่า $u(i_j,t_p)$ (สามารถคำนวณได้จาก $TUL(NLI) = TUL(NLI) + u(i_j,t_p)$) เมื่อทำการพิจารณาทุกสมาชิกใน NUL^{i_j} จนครบแล้ว จะทำการลบข้อมูลทั้งหมดของรายการที่ถูกจัดเก็บอยู่ในโหนดลูกของ *R* ของ *HUII-tree* ออกจากหน่วยความจำและการพิจารณาได้

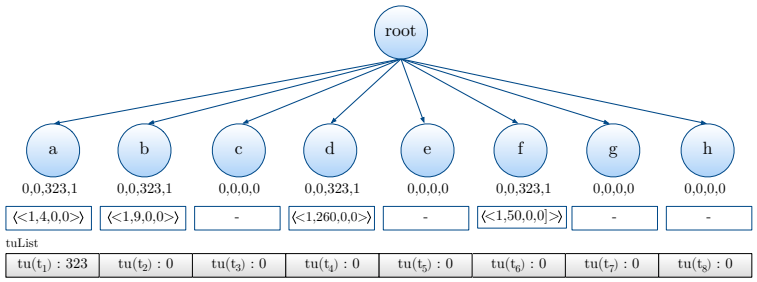
แต่อย่างไรก็ตามถ้าค่าประมาณคุณประโยชน์ $TWU(i_j)$ มีค่ามากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด (บรรทัดที่ 14-26) จะทำการประมาณค่าคร่าวๆโดยลบ $TWU(i_j)$ ด้วยผลรวมของค่าคุณประโยชน์ของรายการที่มีค่าคุณประโยชน์ต่ำ *TUL* (สามารถคำนวณได้จาก $ETWU(i_j) = TWU(i_j) - (TUL(NLI) – TUL(LU(i_j)))$) (ถ้าค่า $ETWU(i_j)$ มีค่ามากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด เราสามารถสรุปได้ว่ารายการ i_j ณ ปัจจุบันเป็นรายการที่มีค่าคุณประโยชน์สูง ถ้าไม่เช่นนั้น $TWU(i_j)$ จะต้องถูกคำนวณใหม่ โดยเริ่มจากกำหนดให้ค่า $TWU(i_j) = 0$ จากนั้นทำการพิจารณาแต่สมาชิกใน NUL^{i_j} ที่ซึ่งมี ลักษณะเป็น $<p, u(i_j, t_p), 0, 0>$ และทำการคำนวณค่า $TWU(i_j)$ ด้วย $tuList(t_p)$ (สามารถคำนวณได้จาก $TWU(i_j) = TWU(i_j) + tuList(t_p)$) โดยหลังจากทำการคำนวณค่า $TWU(i_j)$ เสร็จสิ้น จะทำการตรวจสอบ $TWU(i_j)$ ถ้ามีค่ามากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด จะทำการอัพเดต $LU(i_j)$ ให้เท่ากับ *NLI* กล่าวคือ $(LU(i_j) = NLI)$ เพื่อบันทึกรอบการพิจารณาครั้งสุดท้ายของรายการ i_j แต่ถ้า $TWU(i_j)$ น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนดค่า *NLI* จะถูกเพิ่มค่าขึ้น *1* และค่า *TUL* จะได้รับการอัพเดท และทำการลดทอนค่าคุณประโยชน์แต่ละทรานแซกชันที่มีรายการ i_j ปรากฏ ก่อนที่รายการ i_j จะถูกลบข้อมูลทั้งหมดออกจากหน่วยความจำและการพิจารณาได้ การพิจารณาแต่ละรายการจะถูกทำซ้ำ และขั้นตอนนี้จะสิ้นสุดลงหากไม่มีรายการใดที่ถูกลบออกจาก *HUII-tree*

3.5 ตัวอย่างขั้นตอนวิธี *EHUIIM*

กำหนดให้ 1) ฐานข้อมูลรายการประกอบไปด้วย 10 ทรานแซกชันดังแสดงในตารางที่ 3-1 2) ค่าคุณประโยชน์ของรายการหนึ่งๆดังแสดงในตารางที่ 3-2 3) ลำดับของเซตรายการที่อยู่ภายใต้ขอบเขตการพิจารณา คือ $a \prec b \prec c \prec d \prec e \prec f \prec g \prec h$ 4) ค่าขีดแบ่งคุณประโยชน์ $\boldsymbol{\sigma_u}$ = 40 และ 5) ค่าขีดแบ่งความสม่ำเสมอ $\boldsymbol{\sigma_r}$ = 3 ตามลำดับ การค้นหาเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏไม่สม่ำเสมอด้วยขั้นตอนวิธี *"HUIIM"* สามารถแสดงได้ดังนี้

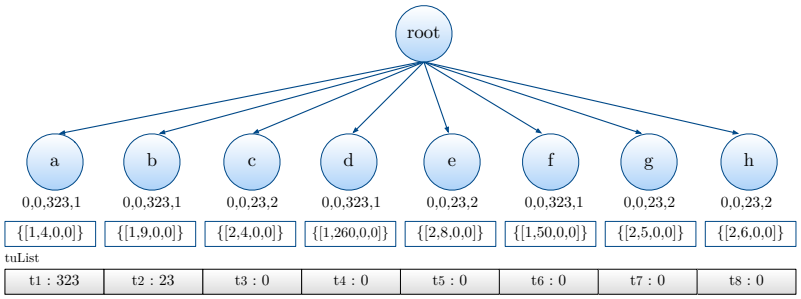
ขั้นตอน *"EHUIIM-Scanning"* (ดังแสดงรายละเอียดในภาพที่ 3-1 และ 3-3) จะเริ่มจากการสร้างหรือจองพื้นที่หน่วยความจำในการจัดเก็บข้อมูลเบื้องต้น ที่ซึ่งจะทำการสร้างลิสต์ *tuList* โดยมีขนาดเท่ากับ *8* (ขนาดจะเท่ากับจำนวนของทรานแซกชันในฐานข้อมูล) นอกจากนี้จะทำการสร้างโครงสร้างต้นไม้ *HUII-tree* เพื่อจัดเก็บรายการ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

เมื่อทำการสร้างหรือจองพื้นที่หน่วยความจำเสร็จ ขั้นตอนวิธี *"EHUIIM-Scanning"* 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 และทำการอัพเดต *NUL* ของแต่ละรายการ *'a'*, *'b'*, *'d'* และ *'f'* เริ่มด้วยรายการ *'a'* ทำการอัพเดต NUL^a ซึ่งจากเดิมจะเป็นเซตว่าง จะถูกอัพเดทเป็น $NUL^a = \{<1, 4, 0, 0>\}$ ซึ่งบ่งบอกได้ว่ารายการ *'a'* ปรากฏขึ้นในทรานแซกชัน t_1 ที่ซึ่งมีค่าคุณประโยชน์เท่ากับ $u(a, t_1) = iu(a, t_1) \times eu(a) = 2 \times 2 = 4$ (หมายเหตุ ข้อมูลใน tuple ที่อัพเดทใน NUL^a จะยังไม่ได้ทำการคำนวณค่าคุณประโยชน์ที่เหลือของรายการ *'a'* และรายการ *'a'* เป็นรายการเดี่ยวที่ซึ่งไม่มี prefix หรือ parent items จึงทำให้ค่าใน tuple ที่ 3 และ 4 มีค่าเป็น *0*) และกำหนดค่าความสม่ำเสมอ $r(a) = max(fr^a_{t_1}) = 1$ สุดท้ายทำการอัพเดตค่าประมาณคุณประโยชน์ $TWU(a) = tu(t_1) = 323$ ดังแสดงในภาพที่ 3-4

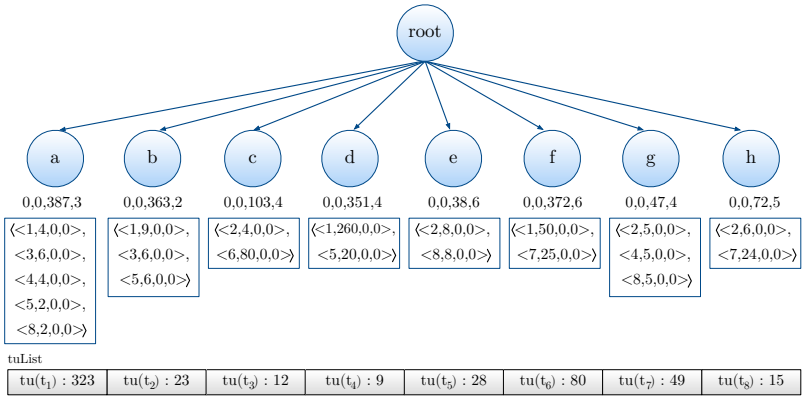


ภาพที่ 3-4 อ่านข้อมูลทรานแซกชันที่ 1 ในฐานข้อมูล

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 $tuList(t_2)$ จากนั้นทำการพิจารณาแต่ละรายการที่ปรากฏขึ้นในทรานแซกชัน t_2 และทำการอัพเดต *NUL* ของแต่ละรายการ *'c', 'e', 'g'* และ *'h'* ที่ซึ่งเริ่มด้วยรายการ *'c'* ทำการอัพเดต NUL^c ซึ่งจากเดิมจะเป็นเซตว่าง จะถูกอัพเดทเป็น $NUL^c = \{<2, 4, 0, 0>\}$ ซึ่งบ่งบอกได้ว่ารายการ *'c'* ปรากฏขึ้นในทรานแซกชัน t_2 ที่มีค่าคุณประโยชน์ $u(c,t_2) = iu(c,t_2) \times eu(c) = 1 \times 4 = 4$ และทำการคำนวณค่าความสม่ำเสมอ $r(c) = max(fr^c_{t_2}) = 2$ สุดท้ายทำการอัพเดตค่าประมาณคุณประโยชน์ $TWU(c) = tu(t_2) = 23$ จากนั้นจะทำการอัพเดตค่าต่างๆข้างต้นของรายการ *'e', 'g'* และ *'h'* ดังแสดงในภาพที่ 3-5 จากนั้นทำการอ่านทรานแซกชัน t_3- t_8 แบบวิธีการข้างต้นตามลำดับเพื่อที่จะอัพเดต *tuList* และ *HUll-tree* ดังแสดงในภาพที่ 3-6

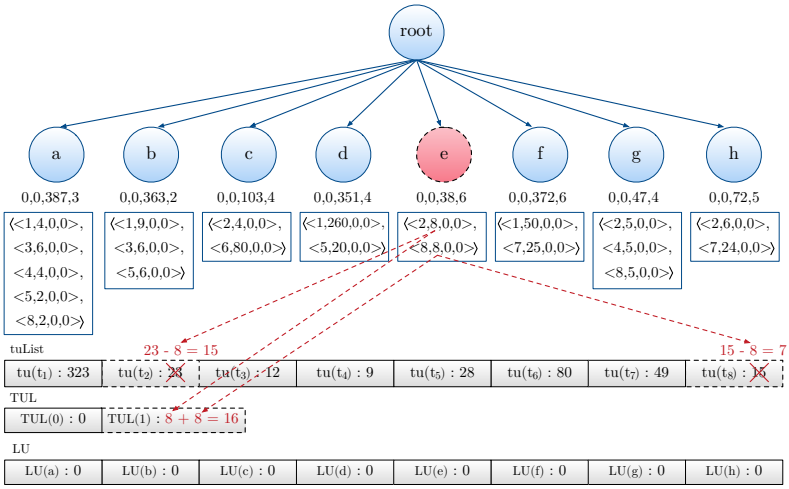


ภาพที่ 3-5 อ่านข้อมูลทรานแซกชันที่ 2 ในฐานข้อมูล



ภาพที่ 3-6 อ่านข้อมูลครบทุกทรานแซกชันในฐานข้อมูล

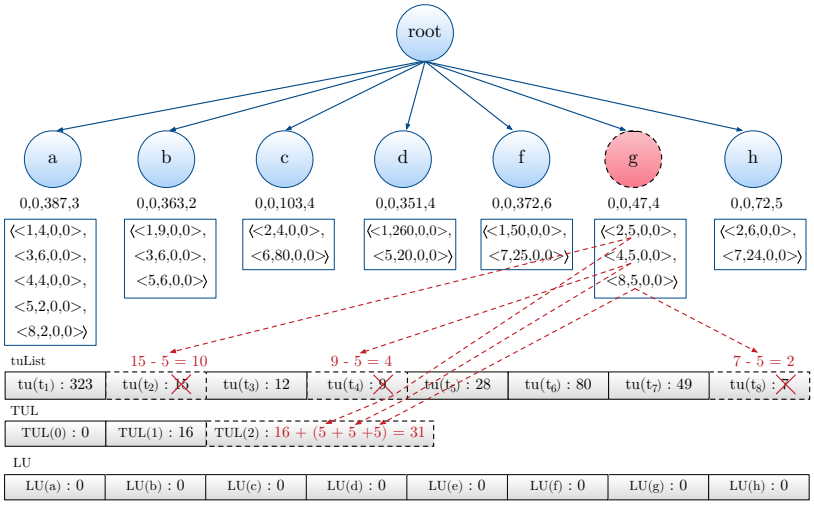
ขั้นตอนต่อไปทำการตรวจสอบค่าประมาณคุณประโยชน์ของแต่ละรายการ ด้วยขั้นตอนวิธี “*Efficient Pruning technique*” (ดังแสดงรายละเอียดในภาพที่ 3-3) ที่ซึ่งจะเริ่มจากการสร้างหรือจองพื้นที่หน่วยความจำในการจัดเก็บข้อมูลเบื้องต้น โดยทำการสร้างลิสต์ *TUL* ให้มีจำนวนสมาชิกเพียง 1 ตัวเท่านั้น โดยสามชิกตัวแรกมีค่าเท่ากับ 0 กำหนดให้ (*TUL(0) = 0)* จากนั้นทำการสร้างลิสต์ *LU* ให้มีจำนวนสมาชิกทั้งสิ้น 8 ตัว(มีขนาดเท่ากับจำนวนรายการในฐานข้อมูล) และสร้างตัวแปร *NLI* เริ่มต้นมีค่าเท่ากับ *0* จากนั้นจะทำการพิจารณารายการแต่ละรายการตามลำดับ $a \prec b \prec c \prec d \prec e \prec f \prec g \prec h$ โดยในตอนเริ่มต้นจะพิจารณารายการ *‘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)* มากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด จึงทำการเลื่อนการพิจารณาไปยังรายการ *‘d’* แล้วทำการพิจารณา *TWU(d) = 351* แต่ด้วยเนื่องจาก *TWU(d)* มากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด จึงทำการเลื่อนการพิจารณาไปยังรายการ *‘e’* แล้วทำการพิจารณา *TWU(e) = 38* ที่ซึ่งมีค่าน้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด ดังนั้น *NLI* ถูกเพิ่มค่าขึ้น *1* และทำการสร้างลิสต์ *TUL(1)* และทำการลบรายการ *‘e’* ออกจากการพิจารณา แต่ก่อนที่จะลบรายการ *‘e’* จะทำการลดทอนค่าคุณประโยชน์ในทรานแซกชันที่มี *‘e’* ปรากฏ โดยทำการพิจารณาแต่ละสมาชิกใน NUL^{e} *= {<2, 8, 0, 0>, {<8, 8, 0, 0>}* ซึ่งจะบอกได้ว่ารายการ *‘e’* ปรากฏในทรานแซคชันที่ 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 และทำการอัพเดตค่า *TUL(1)* ด้วยค่า *u*(*e*,t_2) และ *u*(*e*,t_8) กล่าวคือ *TUL(1)* = *TUL(1)* + (*u*(*e*,t_2) + *u*(*e*,t_8)) = (8 + 8) = 16 แล้วจึงทำการลบข้อมูลของรายการ *‘e’* ที่ถูกจัดเก็บอยู่ในโหนดลูกของ *R* ของ *HUII-tree* ออกจากหน่วยความจำและการพิจารณาได้ (ดังแสดงในภาพที่ 3-7)



ภาพที่ 3-7 ตัดการพิจารณารายการ *‘e’* ที่มี *TWU* ไม่ผ่านค่าขีดแบ่งคุณประโยชน์

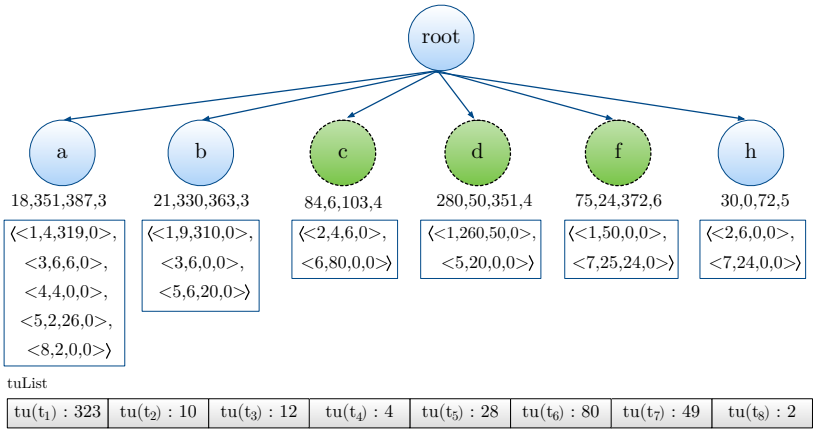
จากนั้นจะทำการพิจารณารายการ *'f'* แล้วทำการพิจารณา *TWU(f) = 372* ที่ซึ่งมากกว่าค่าประมาณคุณประโยชน์ที่ผู้ใช้กำหนด จากนั้น *EHUIIM* จะทำการประมาณลบค่า *TWU(f)* คร่าวๆ ด้วยค่า *TUL* กล่าวคือ *ETWU(f) = TWU(f) - (TUL(1) – TUL(LU(f)) = TWU(f) – (TUL(1) – TUL(0)) = 372 – (16 - 0) = 356* ที่ซึ่ง *ETWU(f)* มากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด ดังนั้นเราสามารถกล่าวได้ว่ารายการ *'f'* เป็นรายการที่น่าจะมีค่าคุณประโยชน์สูง ณ ปัจจุบัน

หลังจากการพิจารณารายการ *'f'* แล้วจะทำการเลื่อนการพิจารณาไปยังรายการ *'g'* และทำการพิจารณา *TWU (g) = 47* ที่ซึ่งมากกว่าค่าประมาณคุณประโยชน์ที่ผู้ใช้กำหนด จากนั้น *EHUIIM* จะทำการประมาณลบค่า *TWU(g)* คร่าวๆ ด้วยค่า *TUL* กล่าวคือ *ETWU(g) = TWU(g) - (TUL(1) – TUL(LU(g)) = TWU(g) – (TUL(1) – TUL(0)) = 47 – (16 - 0) = 31* ที่ซึ่งมีค่าน้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด ในกรณีเช่นนี้ *EHUIIM* จำเป็นต้องทำการคำนวณค่า *TWU(g)* ใหม่ เพื่อให้ได้ค่าที่แท้จริงหลังจากมีการลบรายการออกจาก *HUII-tree* โดยการคำนวณ *TWU(g)* ใหม่สามารถทำได้ด้วยการตรวจสอบแต่ละสมาชิกใน NUL^g กล่าวคือ *{<2, 5, 0, 0>, {<4, 5, 0, 0>, {<8, 5, 0, 0>}* ซึ่งจะทราบได้ว่ารายการ *'g'* ปรากฏในทรานแซคชันที่ t_2, t_4 และ t_8 ซึ่งค่าประมาณคุณประโยชน์ *TWU(g)* สามารถคำนวณได้เป็น $TWU(g) = tuList(t_2) + tuList(t_4) + tuList(t_8) = 15 + 9 + 7 = 31$ ที่ซึ่งมีค่าน้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด เมื่อค่า *TWU(g)* น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด รายการ *'g'* จึงถูกระบุว่าเป็นรายการที่ค่าคุณประโยชน์ต่ำ ดังนั้นค่า *NLI* จะถูกเพิ่มค่าขึ้นเป็น *2* และทำการสร้าง *TUL(2)* แล้วทำการอัพเดตค่า *TUL(2)* เท่ากับ *31* กล่าวคือ $TUL(2) = TUL(1) + u(g,t_2) + u(g,t_4)) + u(g,t_8) = 16 + 5 + 5 + 5 = 31$ และทำการลดทอนค่าคุณประโยชน์ทรานแซกชันที่มี *'g'* ปรากฏ ดังนี้ ทรานแซคชันที่ t_2, t_4 และ t_8 ลดทอนค่า $tuList(t_2) =$ ด้วยค่า $u(g,t_2)$ กล่าวคือ $tuList(t_2) – u(g,t_2) = 15 – 5 = 10$ และลดทอนค่า $tuList(t_4)$ ด้วยค่า $u(g,t_4)$ กล่าวคือ $tuList(t_4) = tuList(t_4) – u(g,t_4) = 9 – 5 = 4$ และลดทอนค่า $tuList(t_8)$ ด้วยค่า $u(g,t_8)$ กล่าวคือ $tuList(t_8) = tuList(t_8) – u(g, t_8) = 7 – 5 = 2$ แล้วจึงทำการลบข้อมูลของรายการ *'g'* ที่ถูกจัดเก็บอยู่ในโหนดลูกของ *R* ของ *HUII-tree* ออกจากหน่วยความจำและการพิจารณาได้ต่อไปทำการเลื่อนการพิจารณาไปยังรายการ *'h'* และพิจารณา *TWU(h) = 72* ที่ซึ่งมากกว่าค่าประมาณคุณประโยชน์ที่ผู้ใช้กำหนด *EHUIIM* ทำการลดทอนค่าประมาณคุณประโยชน์ของรายการ *ETWU(h)* ด้วยค่า *TUL* กล่าวคือ *ETWU(h) = TWU(h) - (TUL(2) – TUL(LU(h)) = TWU(h) – (TUL(2) – TUL(0)) = 72 – (31 - 0) = 41* ที่ซึ่งมากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด ดังนั้นเราสามารถกล่าวได้ว่ารายการ *'h'* เป็นรายการที่น่าจะมีค่าคุณประโยชน์สูง ณ ปัจจุบัน ขั้นตอน *"Efficient Pruning technique"* จะทำซ้ำโดยเริ่มพิจารณารายการ *'a'* ถึง *'h'* และขั้นตอนนี้จะสิ้นสุดลงหากไม่มีรายการใดที่ถูกลบออกจาก *HUII-tree*



ภาพที่ 3-8 ตัดการพิจารณารายการ *'g'* ที่มี *TWU* ไม่ผ่านค่าขีดแบ่งคุณประโยชน์

หลังจากทำการพิจารณาลบรายการที่มีค่าประมาณคุณประโยชน์ต่ำ ขั้นตอนต่อไปจะเป็นการคำนวณค่าคุณประโยชน์ส่วนเหลือในแต่ละสมาชิกใน *NUL* รวมถึงทำการคำนวณค่าคุณประโยชน์ส่วนเหลือและค่าคุณประโยชน์ที่แท้จริงของแต่ละรายการ โดยเริ่มต้นจากทำการสร้าง *temptuList* ให้มีขนาดเท่ากับจำนวนทรานแซกชันในฐานข้อมูล และกำหนดให้แต่ละสมาชิกใน *temptuList(j)* มีค่าเท่ากับ *tuList(j)* จากนั้นทำการพิจารณาแต่ละรายการที่เป็นโหนดลูกของโหนด *R* ใน *HUII-tree* โดยทำการพิจารณารายการตามลำดับ $a < b < c < d < f < h$ 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> ทำการอัพเดตค่าคุณประโยชน์ทรานแซกชัน $temptuList(t_1) = temptuList(t_1) - u(a,t_1) = 323 - 4 = 319$ เพื่อที่จะทราบถึงค่าคุณประโยชน์ของทุกรายการในทรานแซกชัน ที่อยู่ในลำดับถัดไปจากรายการ *'a'* ซึ่งค่า $temptuList(t_1)$ หลังจากการอัพเดตหมายถึงค่าคุณประโยชน์ส่วนเหลือในแต่ละทรานแซกชัน $ru(a,t_1) = 319$ และทำการอัพเดตข้อมูลอันดับที่ 3 ใน <1, 4, 319, 0> และอัพเดตค่าคุณประโยชน์ $u(a) = u(a) + u(a,t_1) = 0 + 4 = 4$ ค่าคุณประโยชน์ส่วนเหลือ $ru(a) = ru(a) + ru(a,t_1) = 0 + 319 = 319$ จากนั้นทำการพิจารณาสมาชิกลำดับถัดไปใน NUL^a คือ <3, 6, 0, 0> ทำการอัพเดตค่าคุณประโยชน์ทรานแซกชัน $temptuList(t_3) = temptuList(t_3) - u(a,t_3) = 12 - 6 = 6$ หลังจากการอัพเดตคือค่าคุณประโยชน์ส่วนเหลือในแต่ละทรานแซกชัน $ru(a,t_3) = 6$ และทำการอัพเดตใน tuple-3 ใน <3, 6, 6, 0> และอัพเดตค่าคุณประโยชน์ $u(a) = u(a) + u(a,t_3) = 4 + 6 = 10$ ค่าคุณประโยชน์ส่วนเหลือ $ru(a) = ru(a) + ru(a,t_3) = 319 + 6 = 325$ และทำการอัพเดตทุกๆสมาชิกใน NUL^a จนครบ หลังจากการอัพเดตเสร็จเราจะได้ NUL^a={<1, 4, 319, 0>, <3, 6, 6, 0>, <4, 4, 0, 0>, <5, 2, 26, 0>, <8, 2, 0, 0>} และค่าคุณประโยชน์ $u(a) = 18$ ค่าคุณประโยชน์ส่วนเหลือ $ru(a) = 351$ หลังจากพิจารณารายการ *'a'* เสร็จแล้ว จะทำการพิจารณารายการ *'b'*, *'c'*, *'d'*, *'f'* และ *'h'* ด้วยวิธีการแบบเดียวกัน (ดังที่แสดงในภาพที่ 3-9 เซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอมีโหนดเป็นสีเขียว)

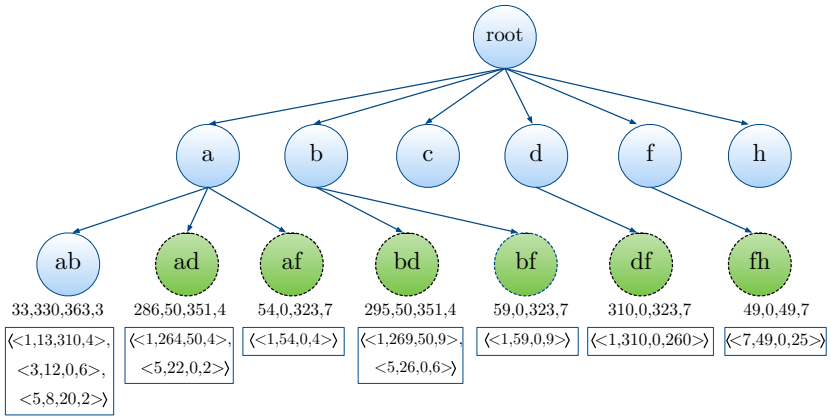


ภาพที่ 3-9 หลังจากอ่านข้อมูลทุกทรานแซกชันในฐานข้อมูล

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’* และ *‘h’* แต่อย่างไรก็ตามก่อนที่จะทำการรวมรายการ *‘a’* เข้ากับรายการอื่นๆ เราจะต้องทำการตรวจสอบค่าประมาณคุณประโยชน์แบบกระชับ *tou(a)* ว่ามีค่าไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนดหรือไม่ ถ้า *tou(a)* มีค่าไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด จะทำการรวมรายการ *‘a’* เข้ากับรายการอื่น โดยเริ่มจากรวมรายการ *‘a’* เข้ากับรายการ *‘b’* เพื่อสร้างการพิจารณาเซตรายการ *‘ab’* จากนั้นทำการอินเทอร์เซกชัน NUL^a = *{<1, 4, 319, 0>, <3, 6, 6, 0>, <4, 4, 0, 0>, <5, 2, 26, 0>, <8, 2, 0, 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 $r(ab) = max(fr^{ab}_{t_1}, r^{ab}_{t_1t_3}, r^{ab}_{t_3t_5}, lr^{ab}_{t_5}) = max(1, 2, 2, 3) = 3$ โดยหลังจากการขั้นตอนอินเทอร์เซกชันเสร็จสิ้น เราจะได้ NUL^{ab} = *{<1, 13, 310, 4>, <3, 12, 0, 6>, <5, 8, 20, 2>}* (หมายเหตุ ข้อมูลในอันดับที่ 4 ในแต่ละสมาชิกของ NUL^{ab} จะจัดเก็บค่าคุณประโยชน์ของรายการ *‘a’* ที่ซึ่งคือ prefix หรือ parent item ของเซตรายการ *‘ab’* เพื่อทำการคำนวณในอนาคต) จากนั้นทำการตรวจสอบค่าประมาณคุณประโยชน์ *TWU(ab) = 363* ว่ามีค่ามากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนดหรือไม่ ถ้าค่าประมาณคุณประโยชน์ *TWU(ab)* มากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด จะทำการสร้างเซตรายการ *‘ab’* และจัดเก็บ ค่าคุณประโยชน์ ค่าคุณประโยชน์ส่วนเหลือ ค่าประมาณคุณประโยชน์ ค่าความสม่ำเสมอ และ NUL^{ab} ไว้ในโหนดลูก

ของรายการ *'a'* จากนั้นทำการตรวจสอบค่าคุณประโยชน์ *u(ab) = 33* ว่ามีค่าไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนดหรือไม่ และค่าความสม่ำเสมอ *r(ab) = 3* ว่ามีค่ามากกว่าค่าขีดแบ่งความสม่ำเสมอหรือไม่ ถ้าผ่าน 2 เงื่อนข้างต้น *HUIIM* จะทำการระบุและจัดเก็บรายการ *'ab'* ว่าเป็นเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอซึ่งถูกจัดเก็บไว้ใน *HUII* (หมายเหตุ รายการ *'ab'* มีค่าคุณประโยชน์ *u(ab)* น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด และค่าความสม่ำเสมอ *r(ab)* น้อยกว่าค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด จึงไม่เป็นเซทรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอ) ต่อไปทำการรวมรายการ *'a'* เข้ากับรายการ*'c'* เพื่อสร้างการพิจารณาเซตรายการ *'ac'* และทำการอินเทอร์เซกชัน NUL^a *= {<1, 4, 319, 0>, <3, 6, 6, 0>, <4, 4, 0, 0>, <5, 2, 26, 0>, <8, 2, 0, 0>}* กับ NUL^c *= {<2, 4, 6, 0>, <6, 80, 0, 0>}* แต่ในฐานข้อมูลรายการ *'a'* และ *'c'* ไม่เกิดขึ้นร่วมกัน ดังนั้น *TWU(ac) = 0* จากนั้นจะทำการรวมรายการ *'a'* เข้ากับรายการ *'d'*, *'f'* และ *'h'* ตามลำดับ ด้วยวิธีการข้างต้น

หลังจากทำการรวมรายการ *'a'* เข้ากับทุกโหนดลูกของ *R* ของ *HUII-tree* ทั้งหมด เราจะได้โหนดลูกของรายการ *'a'* ที่บรรจุไปด้วยเซตรายการที่ประกอบไปด้วย 2 รายการ และเป็นเซตรายการ *'a'* เป็นรายการขึ้นต้น ที่ซึ่งจะทำการวนซ้ำการทำงานตามลำดับของรายการที่จัดเก็บไว้ในโหนดลูกของ *R* จนครบทั้งหมด เราจะได้ *HUII-tree* ที่บรรจุไปด้วย 2 รายการ (ดังแสดงในภาพที่ 3-10)

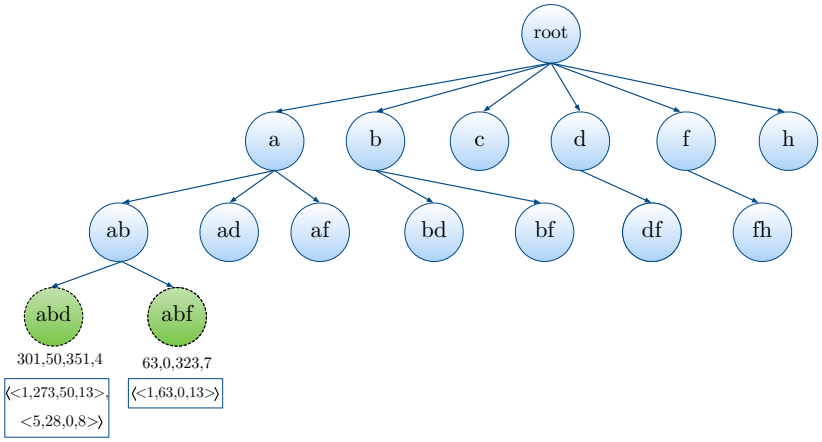


ภาพที่ 3-10 *HUII-tree* ที่บรรจุ *2-HUIIs*

หลังจากได้ *HUII-tree* ที่บรรจุไปด้วย 2 รายการ จะดำเนินการในขั้นตอนสุดท้าย *"MiningAll"* ซึ่งก็คือการสร้างเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอที่ประกอบไปด้วย 3 รายการ และมากกว่า 3 รายการ โดยเริ่มจากทำการพิจารณารายการตามลำดับ *'a'*, *'b'*, *'c'*, *'d'*, *'f'* 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 เราจะต้องทำการตรวจสอบค่าประมาณคุณประโยชน์แบบ

กระชับ *tou(ab) = 360* ว่ามีค่ามากกว่าค่าขีดแบ่งที่ผู้ใช้กำหนดหรือไม่ ถ้าค่าประมาณคุณประโยชน์แบบกระชับ *tou(ab)* มากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด จะทำการรวมเซตรายการ *'ab'* เข้ากับเซตรายการอื่น 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) เพื่อทำการจัดเก็บข้อมูลการปรากฏขึ้น 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 $r(abd) = max(fr^{abd}_{t_1}, r^{abd}_{t_1t_5}, lr^{abb}_{t_5}) = max(1, 4, 3) = 4$ 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 เงื่อนข้างต้น *HUIIM* จะทำการระบุและจัดเก็บรายการ *'abd'* ว่าเป็นรูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอและจัดเก็บไว้ใน *HUII* (หมายเหตุ รายการ *'abd'* มีค่าคุณประโยชน์ *u(abd)* มากกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด และค่าความสม่ำเสมอ *r(abd)* มากกว่าค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด จึงเป็นเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอและจัดเก็บไว้ในเซต *HUII*) ต่อไปทำการรวมเซตรายการ *'ab'* เข้ากับเซตรายการอื่นๆที่อยู่ในลำดับถัดไปด้วยวิธีการข้างต้น (ดังแสดงในภาพที่ 3-10)

หลังจากทำการรวมรายการ *'ab'* เข้ากับทุกโหนดลูกของรายการ *'a'* ทั้งหมด เราจะได้โหนดลูกของรายการ *'ab'* ที่บรรจุไปด้วยเซตรายการที่ประกอบไปด้วย 3 รายการ และเป็นเซตรายการ *'ab'* เป็นรายการขึ้นต้น จากนั้นจะทำการวนซ้ำการทำงานเพื่อหาเซตรายการที่ประกอบไปด้วย 4 รายการ โดยการดำเนินการจะดำเนินเช่นเดียวกับการพิจารณาเซตรายการที่มากกว่า 2 รายการ *"MiningAll"* เมื่อขั้นตอน *"HUIIM-Mining"* เสร็จสิ้น เราจะได้เซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอทั้งหมดที่จัดเก็บไว้ในเซต *HUII* (ดังแสดงในภาพที่ 3-11)



ภาพที่ 3-11 *HUII-tree* ที่บรรจุ *3-HUII*

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 3-12 เซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอจากขั้นตอน *EHUIIM*

บทที่ 4

ผลการทดลอง

ในบทนี้จะนำเสนอการทดสอบประสิทธิภาพการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอจาขั้นตอนวิธี *HUIIM* และ *EHUIIM* เนื่องจากผลลัพธ์ที่ทำการพิจารณาในงานวิจัยนี้มีความแตกต่างกับผลลัพธ์ที่ทำการค้นหาจากงานวิจัยอื่นๆ การดำเนินการทดลองจึงทำการเปรียบเทียบกับวิธีการที่ใกล้เคียง คือ ขั้นตอนวิธี *MHUIR-NUL* ที่เป็นอัลกอริทึมสำหรับการค้นหารูปแบบที่มีค่าคุณประโยชน์และปรากฏอย่างสม่ำเสมอ

ในการทำการทดลองผู้วิจัยได้กำหนดค่าขีดแบ่งให้มีความใกล้เคียงกับงานวิจัยที่เกี่ยวข้อง กล่าวคือ ค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด (σ_r) จะกำหนดให้มีค่าอยู่ระหว่าง 1–30% ของจำนวนทรานแซกชันทั้งหมดในฐานข้อมูล (กล่าวคือ รูปแบบจะเป็นรูปแบบที่ปรากฏอย่างไม่สม่ำเสมอจะต้องมีค่าความสม่ำเสมอมากกว่า 1-30% ของจำนวนทรานแซกชันทั้งหมดในแต่ละฐานข้อมูล) และค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด (σ_u) จะกำหนดให้มีค่าระหว่าง 0.001–45% ของค่าคุณประโยชน์ทั้งหมดของรูปแบบที่ปรากฏในฐานข้อมูล (กล่าวคือ รูปแบบจะเป็นรูปแบบที่มีค่าคุณประโยชน์สูง จะต้องมีค่าคุณประโยชน์ไม่น้อยกว่า 0.001–45% ของค่าคุณประโยชน์ทั้งหมดของรูปแบบที่ปรากฏในแต่ละฐานข้อมูล) ตามลำดับ โดยในการทดสอบประสิทธิภาพของขั้นตอนวิธี *HUIIM* และ *EHUIIM* ผู้วิจัยได้ทำการเขียนโปรแกรมการคำนวณตามขั้นตอนด้วยภาษาC และทำการทดสอบประสิทธิภาพในเครื่อง Mac mini Core i5 2.6GHz macOS Sierra หน่วยความจำ 8 GB โดยในการทดสอบประสิทธิภาพจะดำเนินการทดสอบกับชุดข้อมูลจริง 10 ชุดข้อมูล (ที่ซึ่งสามารถดาว์นโหลดได้จาก P. F. Viger, “SPMF : An Open-Source Data Mining Library[1]”) แต่ละชุดข้อมูลจะมีรายละเอียดดังแสดงในตารางที่ 4-1

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

[1] http://www.philippe-fournier-viger.com

ตารางที่ 4-1 คุณลักษณะของชุดข้อมูลที่ใช้ในการทดสอบประสิทธิภาพ(ต่อ)

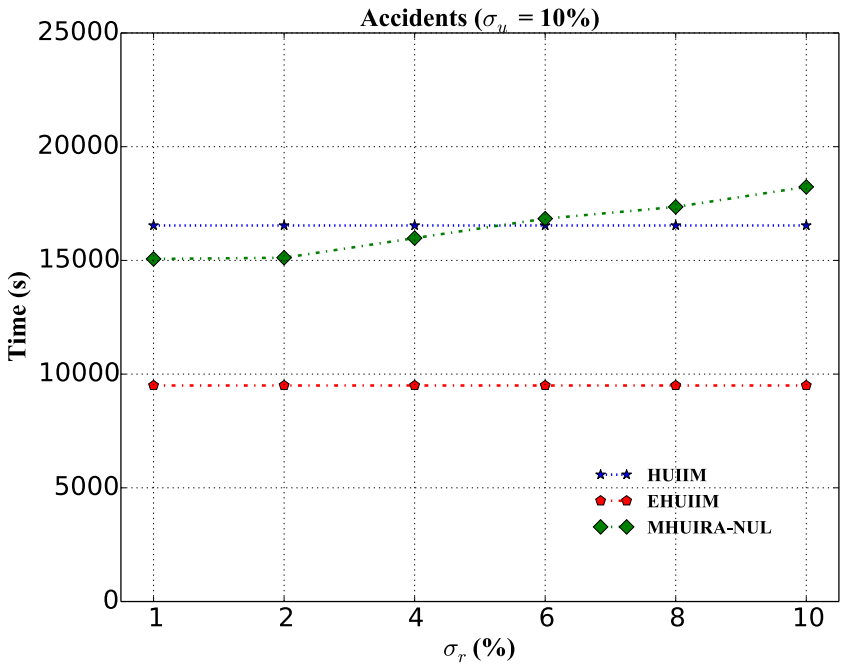
ชื่อฐานข้อมูล	จำนวนรายการที่ปรากฏ	จำนวนทรานแซกชัน	ความยาวเฉลี่ยของทรานแซกชัน	ชนิดของฐานข้อมูล
Connect	129	67,557	43	หนาแน่น
Foodmart2000	1,559	36,869	11	เบาบาง
Kosarak	41,270	990,002	7.3	เบาบาง
Mushroom	119	8,124	23	หนาแน่น
Pumsb	2,113	49,046	74	หนาแน่น
Retail	16,469	88,162	10.3	เบาบาง

การทดลองที่ผู้วิจัยได้ทำคือการทดสอบประสิทธิภาพของขั้นตอนวิธี *HUIIM* และ *EHUIIM* เปรียบเทียบกับขั้นตอนวิธี *MHUIRA-NUL* สามารถแบ่งได้เป็น 6 กรณี คือ 1) การทดสอบเวลาในการคำนวณเมื่อทำการกำหนดค่าขีดแบ่งคุณประโยชน์แบบตายตัวและกำหนดค่าขีดแบ่งความสม่ำเสมอแบบแปรปรวน 2) การทดสอบเวลาในการคำนวณเมื่อทำการกำหนดค่าขีดแบ่งสม่ำเสมอแบบตายตัวและกำหนดค่าขีดแบ่งคุณประโยชน์แบบแปรปรวน 3) หน่วยความจำที่ใช้ในการประมวลผลเมื่อทำการกำหนดค่าขีดแบ่งคุณประโยชน์แบบตายตัวและกำหนดค่าขีดแบ่งความสม่ำเสมอแบบแปรปรวน 4) หน่วยความจำที่ใช้ในการประมวลผลเมื่อทำการกำหนดค่าความสม่ำเสมอแบบตายตัวและกำหนดค่าขีดแบ่งคุณประโยชน์แบบแปรปรวน 5) การพิจารณาจำนวนผลลัพธ์ที่สามารถค้นหาได้เมื่อทำการกำหนดค่าขีดแบ่งคุณประโยชน์แบบตายตัวและกำหนดค่าขีดแบ่งความสม่ำเสมอแบบแปรปรวน 6) การพิจารณาจำนวนผลลัพธ์ที่สามารถค้นหาได้เมื่อทำการกำหนดค่าขีดแบ่งสม่ำเสมอแบบตายตัวและกำหนดค่าขีดแบ่งคุณประโยชน์แบบแปรปรวน ตามลำดับ

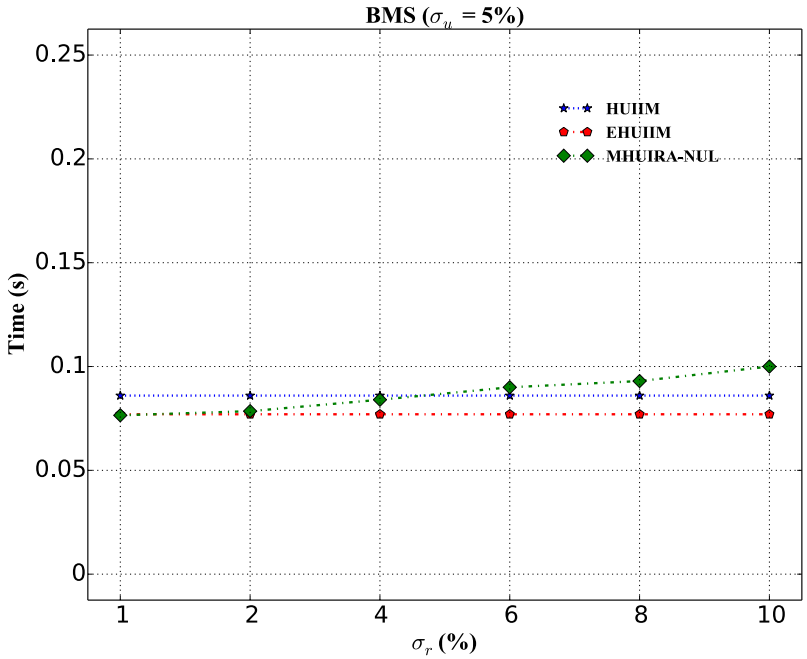
4.1 เวลาที่ใช้ในการคำนวณ

ภาพที่ 4-1 ถึง 4-10 แสดงให้เห็นถึงเวลาที่ใช้ในการประมวลผลแต่ละชุดข้อมูลด้วยขั้นตอนวิธี *HUIIM* และ *EHUIIM* เปรียบเทียบกับขั้นตอนวิธี *MHURIA-NUL* ภายใต้การกำหนดค่าขีดแบ่งความสม่ำเสมอแบบแปรปรวนและค่าขีดแบ่งคุณประโยชน์แบบตายตัวที่ต่ำสุด ซึ่งจากการกำหนดค่าให้กับพารามิเตอร์ดังกล่าวจะทำให้มีเซตรายการเป็นจำนวนมากมีค่าคุณประโยชน์มากกว่าหรือเท่ากับค่าขีดแบ่งคุณประโยชน์ และจะทำให้การค้นหาเซตรายการใช้เวลามาก ที่ซึ่งสะท้อนให้เห็นถึงประสิทธิภาพของขั้นตอนวิธี *HUIIM* และ *EHUIIM* เมื่อต้องพิจารณาเซตรายการเป็นจำนวนมาก โดยจากภาพจะสามารถสังเกตุได้ว่าขั้นตอนวิธี *MHURIA-NUL* ใช้เวลาในการประมวลผลน้อยกว่าขั้นตอนวิธี *HUIIM* และ *EHUIIM* ในหลายกรณี เนื่องจากขั้นตอนวิธี *MHURIA-NUL* สามารถใช้ประโยชน์จากค่าขีดแบ่งความสม่ำเสมอ ในการตัดเซตรายการที่ปรากฏอย่างไม่สม่ำเสมอออกจากการพิจารณา ซึ่งส่งผลให้พื้นที่การค้นหาลดลงอย่างมีนัยสำคัญ โดยเมื่อค่าขีดแบ่งความสม่ำเสมอมีค่าที่ต่ำจะทำให้ *MHURIA-NUL* ใช้เวลาในพิจารณาเซตรายการน้อยลง แต่อย่างไรก็ตาม ในชุดข้อมูล Accidents, Chainstore, Kosarak และ PUMSB ขั้นตอนวิธี *HUIIM* และ *EHUIIM* ใช้เวลาประมวลผลน้อยกว่า

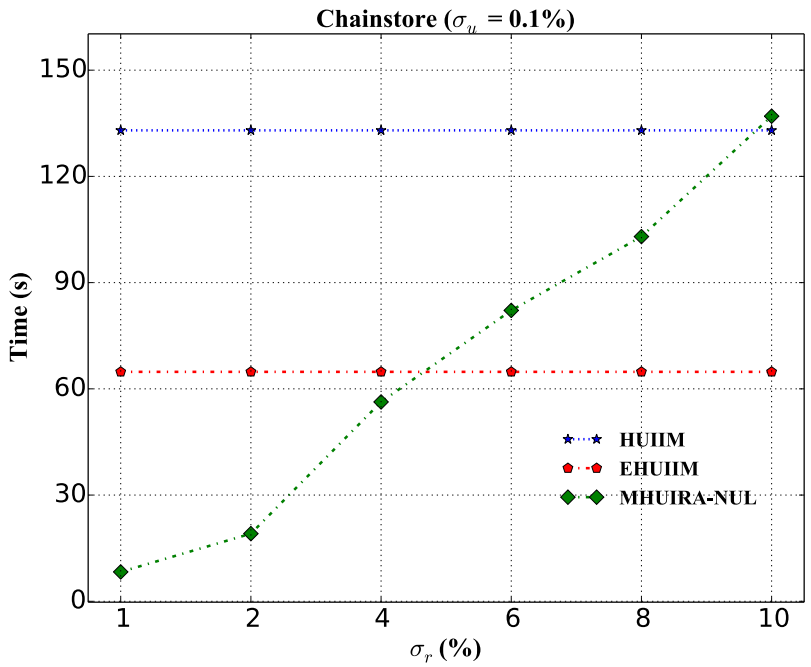
ขั้นตอนวิธี *MHURIA-NUL* เนื่องจาก *MHURIA-NUL* ไม่สามารถใช้ประโยชน์จากค่าขีดแบ่งความสม่ำเสมอ ในการลดทอนเซตรายการที่ปรากฏไม่สม่ำเสมอออกจากการพิจารณาได้มากเหมือนกับฐานข้อมูลอื่นๆ แต่ขั้นตอนวิธี *HUIIM* และ *EHUIIM* สามารถใช้ประโยชน์จากค่าประมาณคุณประโยชน์ *TWU* เพื่อระบุเซตรายการที่มีค่าประมาณคุณประโยชน์ต่ำ และทำการลบเซตรายการเหล่านั้นออกจากการพิจารณา นอกจากนี้เรายังสังเกตุได้ว่าขั้นตอนวิธี *EHUIIM* ใช้เวลาคำนวณน้อยกว่า *HUIIM* ในทุกกรณีโดยประมาณ 5-40% เนื่องจากขั้นตอนวิธี *EHUIIM* สามารถใช้ประโยชน์จากเทคนิคการลดทอนแบบใหม่ ที่ซึ่งจะช่วยให้ *EHUIIM* สามารถตัดรายการที่มีค่าคุณประโยชน์ต่ำออกจากการพิจารณาได้อย่างรวดเร็ว ซึ่งจะช่วยลดทอนจำนวนเซตรายการที่ต้องทำการพิจารณาได้



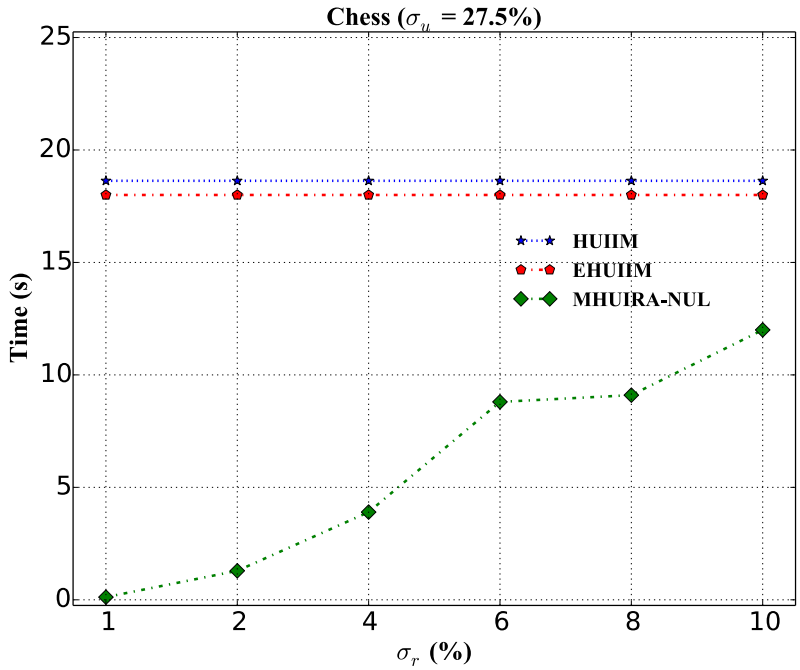
ภาพที่ 4-1 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ ของฐานข้อมูล Accidents



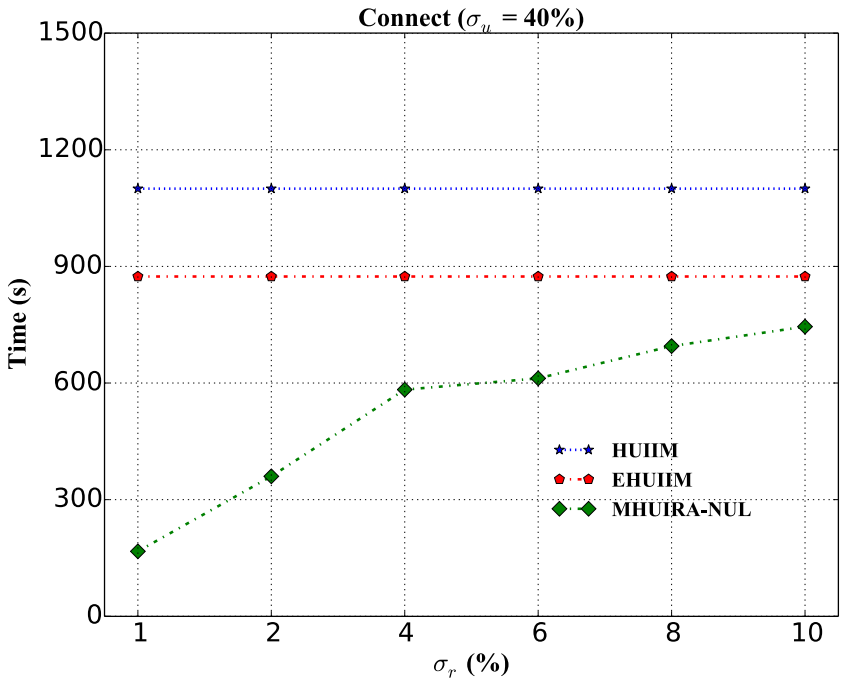
ภาพที่ 4-2 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ ของฐานข้อมูล BMS



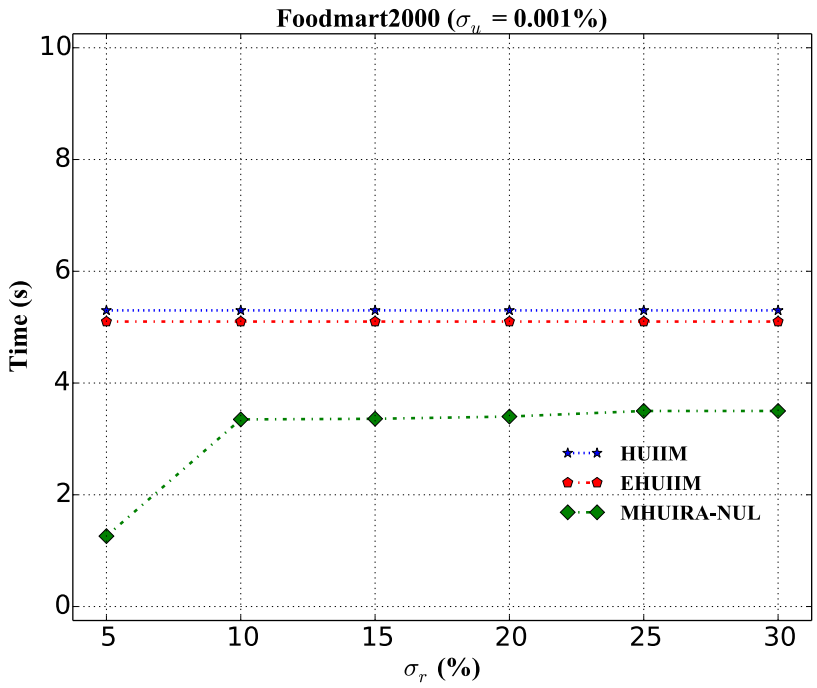
ภาพที่ 4-3 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ ของฐานข้อมูล Chainstore



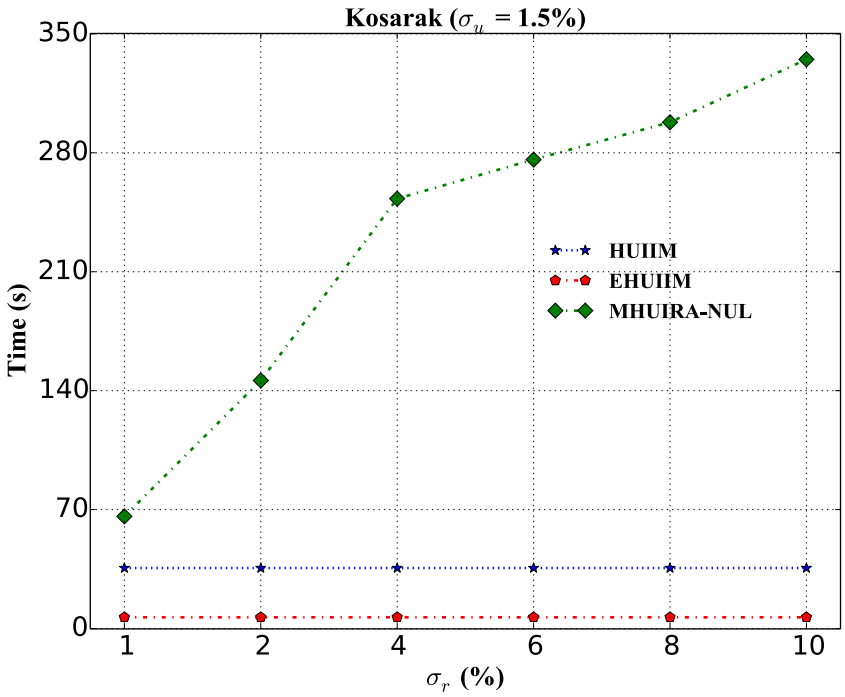
ภาพที่ 4-4 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ ของฐานข้อมูล Chess



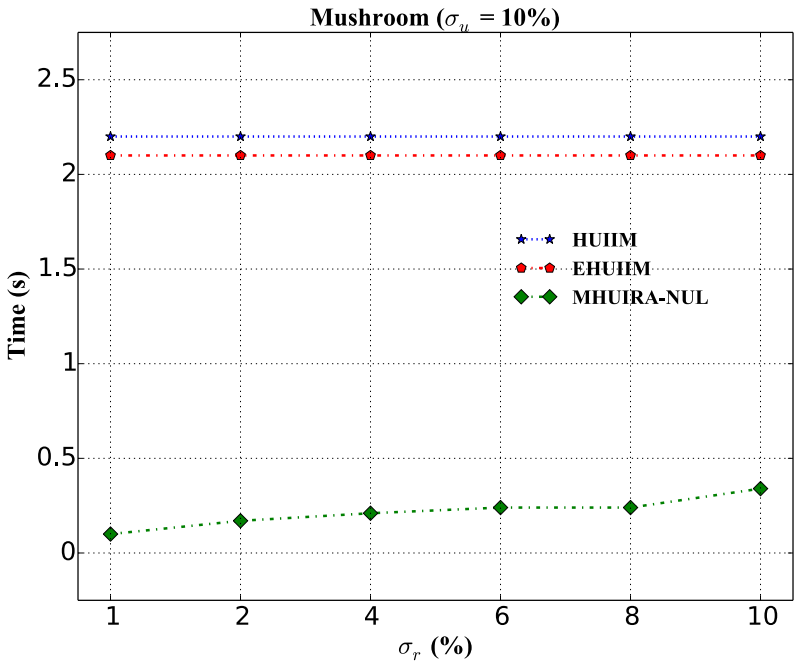
ภาพที่ 4-5 เวลาที่ใช้ในการคำนวณขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอของฐานข้อมูล Connect



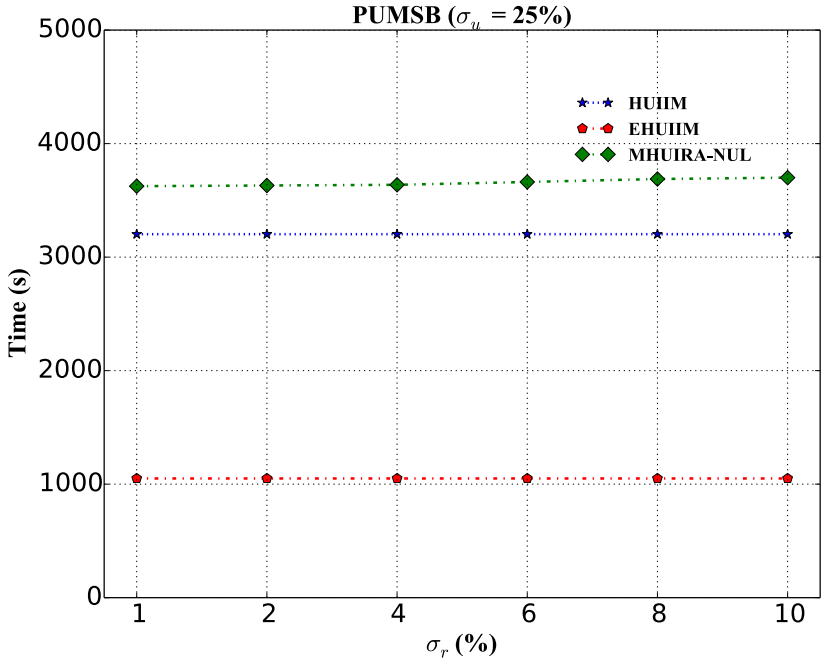
ภาพที่ 4-6 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอของฐานข้อมูล Foodmart2000



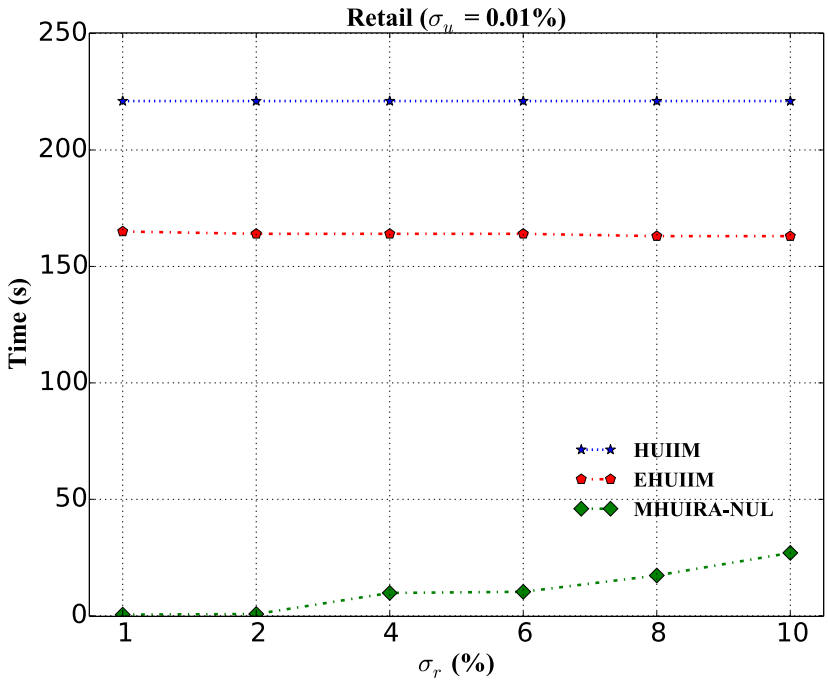
ภาพที่ 4-7 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอของฐานข้อมูล Kosarak



ภาพที่ 4-8 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอของฐานข้อมูล Mushroom

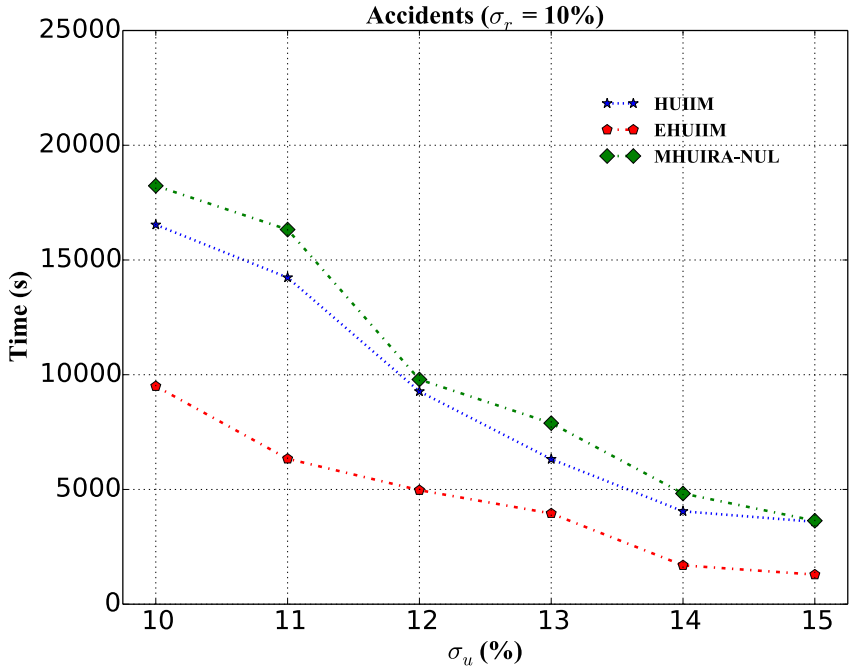


ภาพที่ 4-9 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอของฐานข้อมูล PUMSB

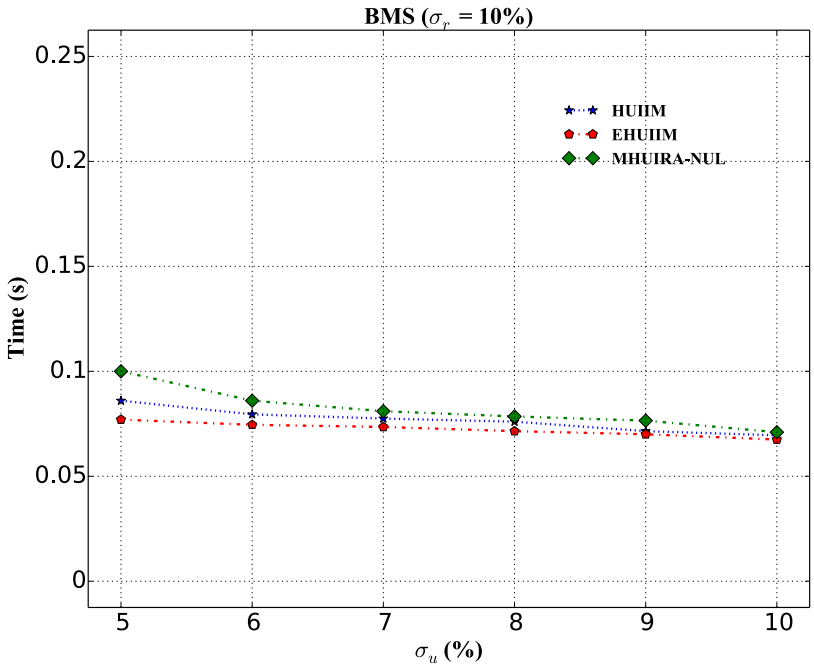


ภาพที่ 4-10 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอของฐานข้อมูล Retail

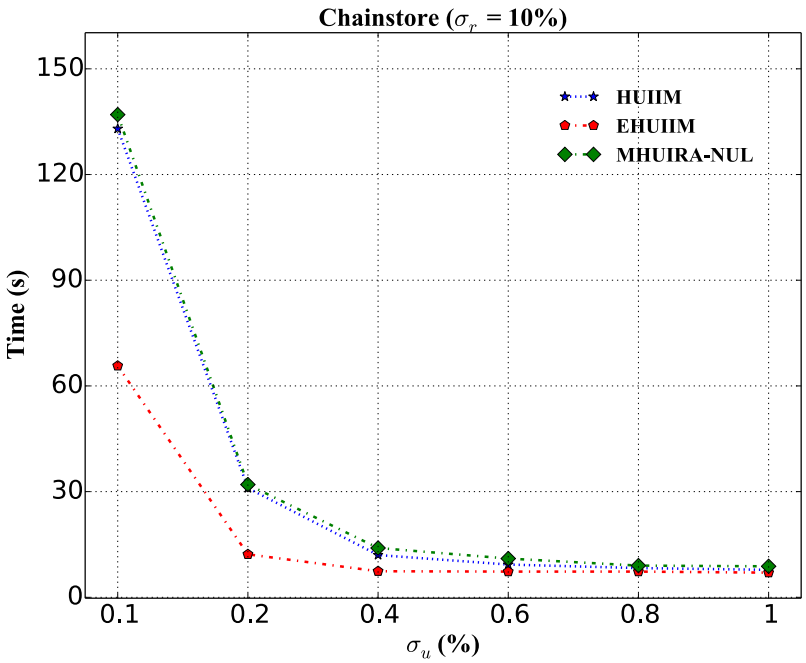
ภาพที่ 4-11 ถึง 4-20 แสดงให้เห็นถึงเวลาที่ใช้ในการประมวลผลแต่ละชุดข้อมูลด้วยขั้นตอนวิธี *HUIIM* และ *EHUIIM* เปรียบเทียบกับวิธี *MHURIA-NUL* ภายใต้การกำหนดค่าขีดแบ่งคุณประโยชน์แบบแปรปรวน และค่าขีดแบ่งความสม่ำเสมอแบบตายตัวที่ต่ำสุด จากภาพสังเกตุได้ว่าเวลาในการประมวลผลของทั้งสามวิธีจะลดลงเมื่อเพิ่มค่าขีดแบ่งคุณประโยชน์ เนื่องจากทั้งสามขั้นตอนวิธีสามารถใช้ประโยชน์จากค่าขีดแบ่งคุณประโยชน์ในการลดทอนเซตรายการที่มีค่าคุณประโยชน์ต่ำ นอกจากนี้เรายังสังเกตได้ว่าเวลาในการประมวลผลของขั้นตอนวิธี *HUIIM* และ *EHUIIM* มากกว่า *MHURIA-NUL* ในชุดข้อมูล Chess, Connect, Foodmart2000, Mushroom และ Retail เนื่องจากขั้นตอนวิธี *MHURIA-NUL* สามารถใช้ประโยชน์จากค่าขีดแบ่งความสม่ำเสมอในการลดทอนเซตรายการที่ปรากฏอย่างไม่สม่ำเสมอ แต่ในส่วนของชุดข้อมูล Accidents, Chainstore, Kosarak และ PUMSB ขั้นตอนวิธี *HUIIM* และ *EHUIIM* มีประสิทธิภาพที่ดีกว่าขั้นตอนวิธี *MHURIA-NUL* เนื่องจากขั้นตอนวิธี *MHURIA-NUL* ไม่สามารถใช้ประโยชน์จากค่าขีดแบ่งความสม่ำเสมอในการตัดเซตรายการที่ปรากฏอย่างไม่สม่ำเสมอ แต่อย่างไรก็ตามเมื่อเราสังเกตุประสิทธิภาพของขั้นตอนวิธีดังกล่าว เห็นได้ชัดว่า *EHUIIM* ใช้เวลาในการประมวลผลน้อยกว่า *HUIIM* เมื่อมีการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์



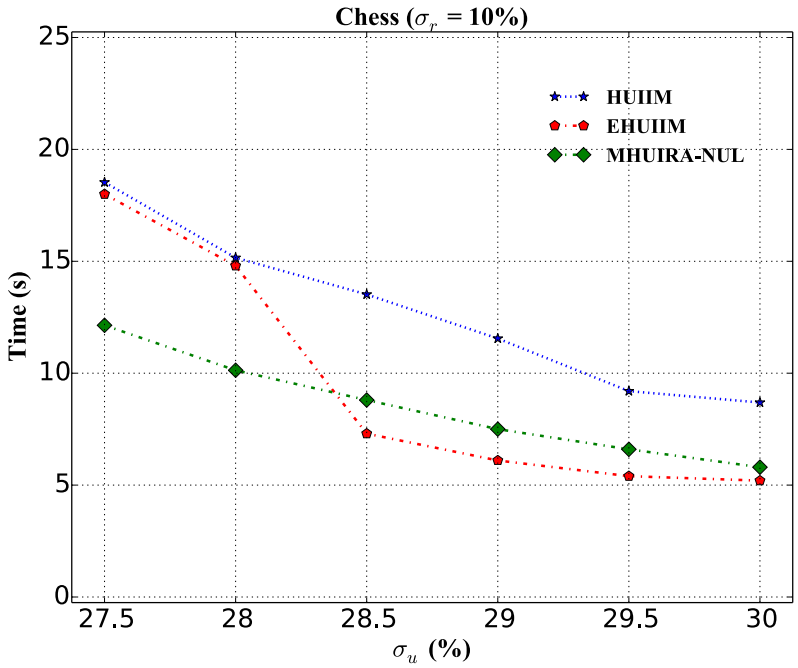
ภาพที่ 4-11 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล Accidents



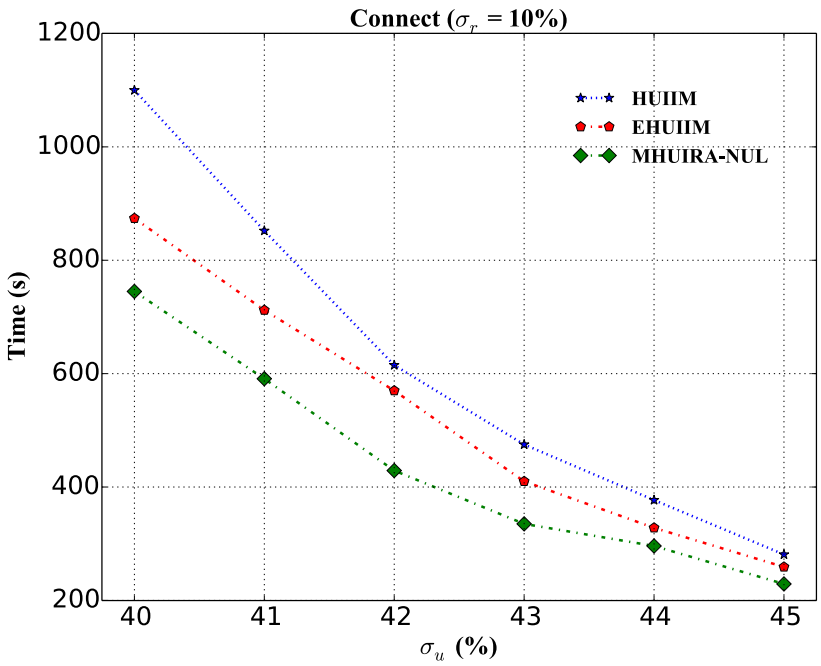
ภาพที่ 4-12 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล BMS



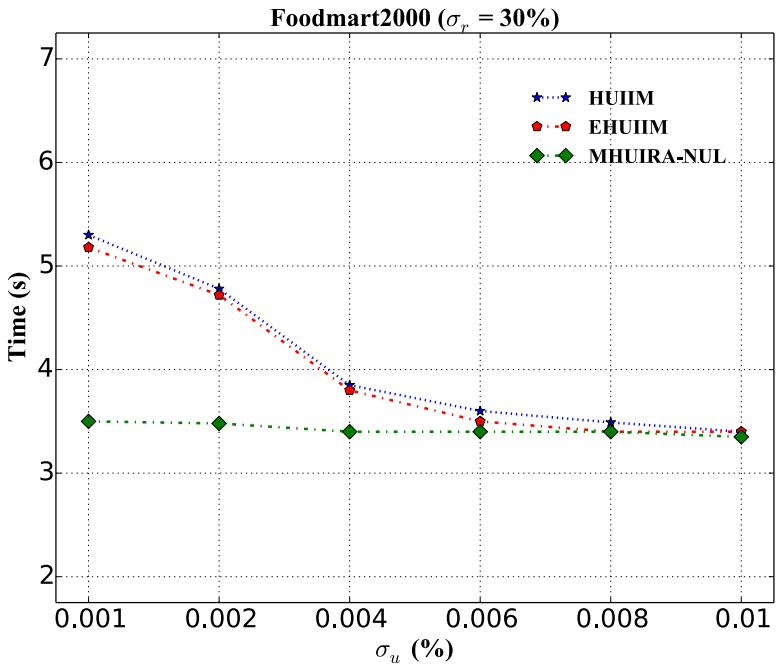
ภาพที่ 4-13 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล Chainstore



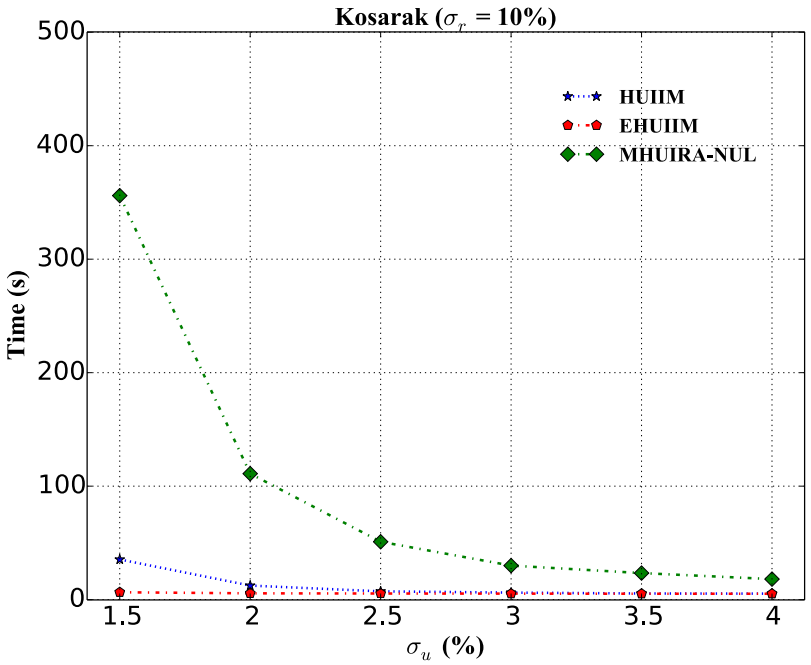
ภาพที่ 4-14 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล Chess



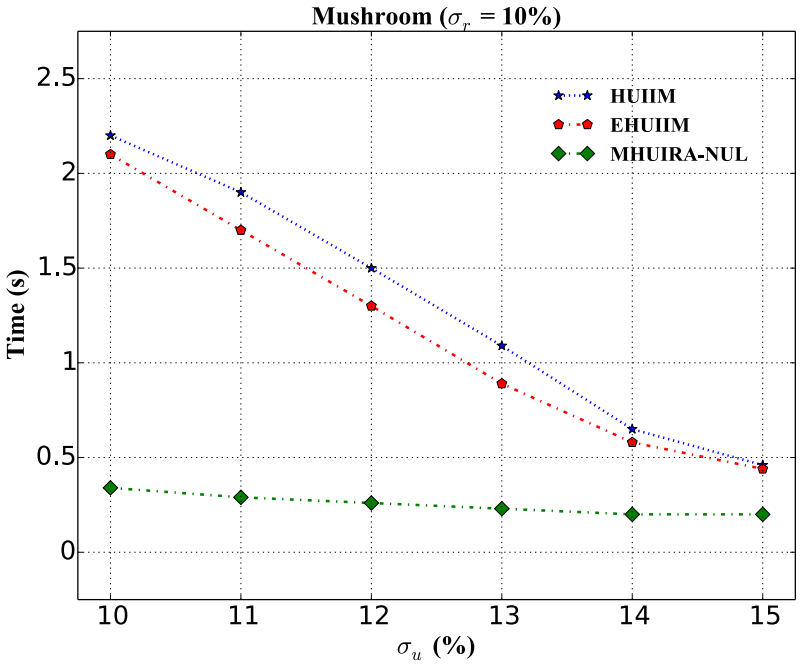
ภาพที่ 4-15 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล Connect



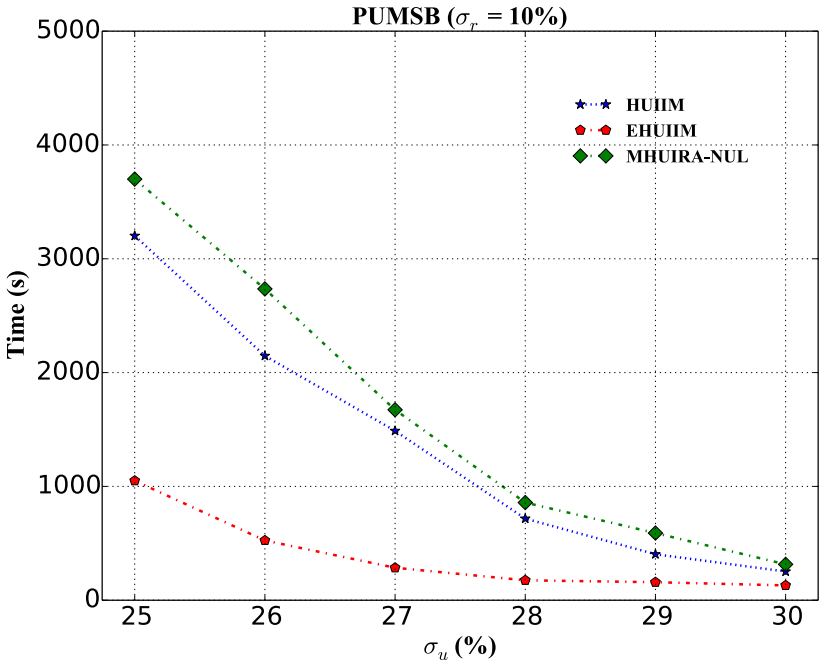
ภาพที่ 4-16 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล Foodmart2000



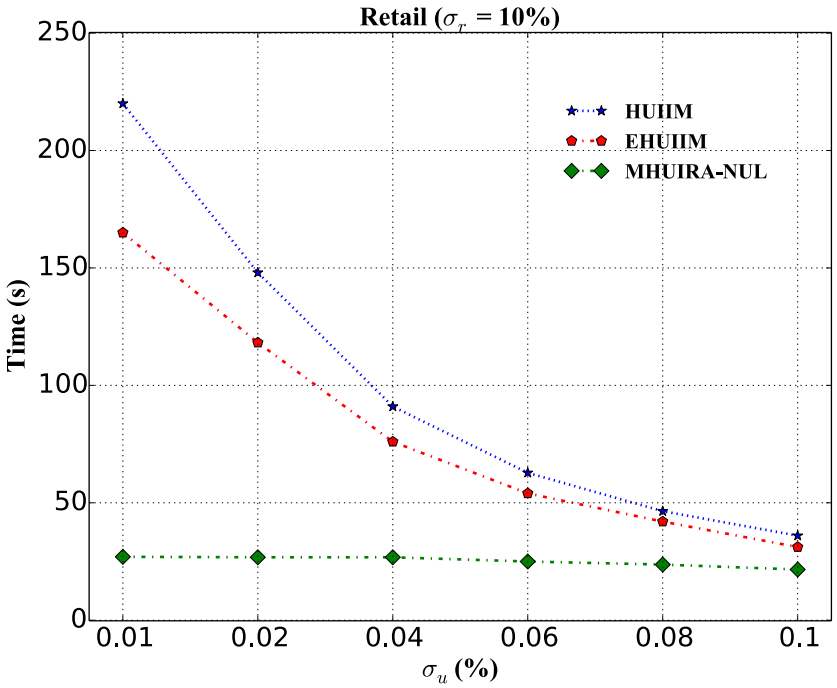
ภาพที่ 4-17 เวลาที่ใช้ในการคำนวณขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล Kosarak



ภาพที่ 4-18 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล Mushroom



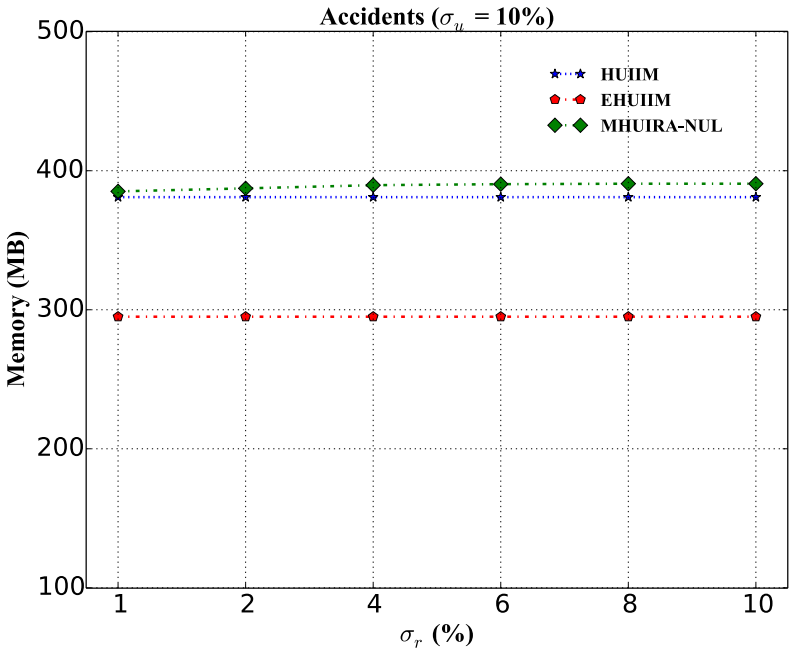
ภาพที่ 4-19 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล PUMSB



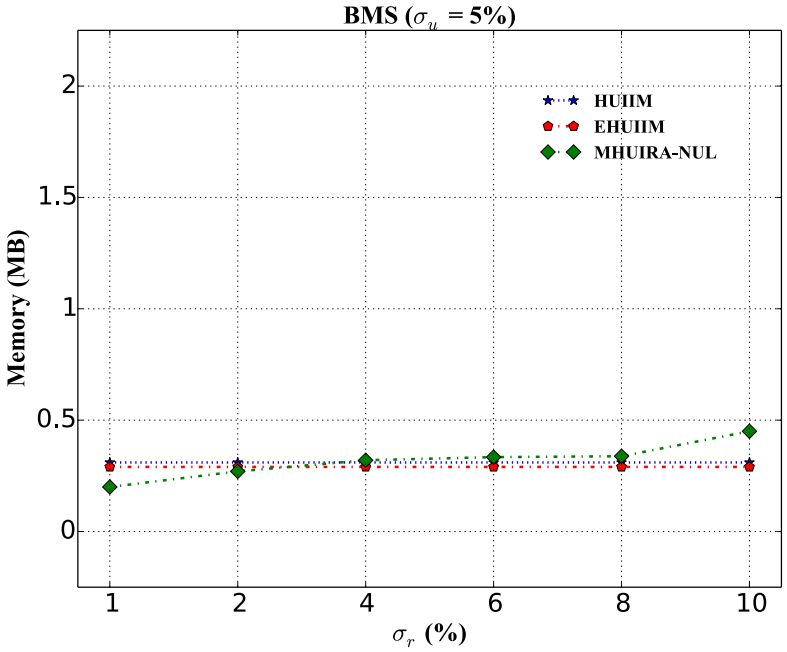
ภาพที่ 4-20 เวลาที่ใช้ในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล Retail

4.2 หน่วยความจำที่ใช้ในการคำนวณ

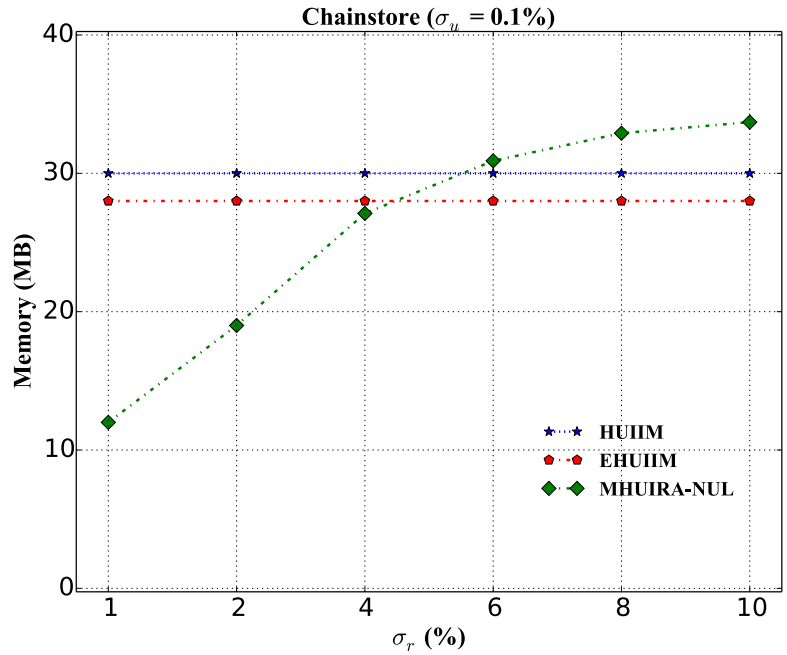
จากภาพที่ 4-21 ถึง 4-40 แสดงการใช้หน่วยความจำสูงสุดในการคำนวณของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* โดยขึ้นอยู่กับค่าขีดแบ่งความสม่ำเสมอแบบแปรปรวน และค่าขีดแบ่งคุณประโยชน์แบบแปรปรวน จากภาพสังเกตได้ว่า *MHURIA-NUL* ใช้หน่วยความจำน้อยกว่าวิธีการอื่นๆ เนื่องจาก *MHURIA-NUL* สามารถใช้ค่าขีดแบ่งความสม่ำเสมอในการลดทอนการพิจารณารายการ ซึ่งเป็นสาเหตุให้ *MHURIA-NUL* เก็บรายการหรือเซตรายการในหน่วยความจำเพียงไม่กี่รายการ นอกจากนี้เรายังสามารถสังเกตได้ว่า *EHUIIM* ใช้หน่วยความจำน้อยกว่า *HUIIM* เนื่องจากสามารถใช้ประโยชน์จากเทคนิคการลดทอนการพิจารณาแบบใหม่ ในการตัดรายการหรือเซตรายการทั้งหมดที่มีค่าคุณประโยชน์ต่ำ



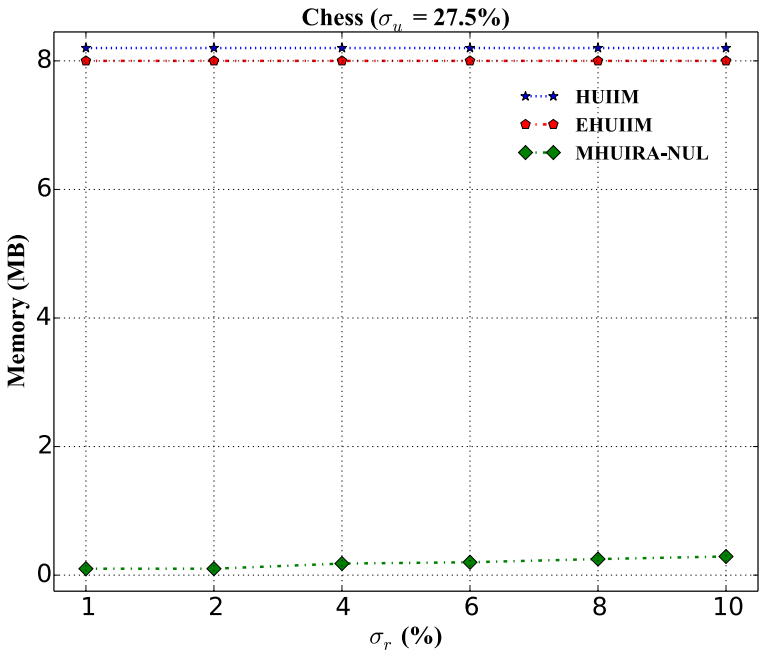
ภาพที่ 4-21 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ ของฐานข้อมูล Accidents



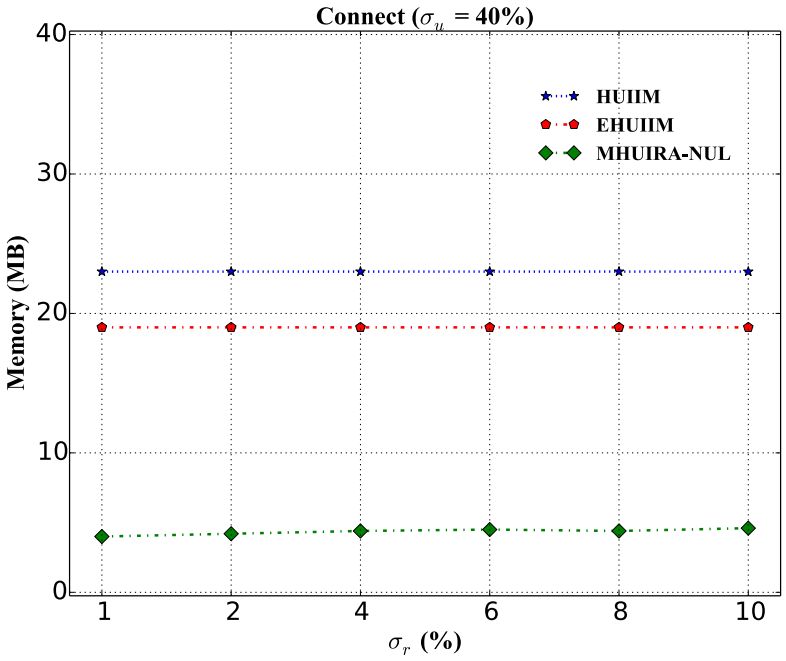
ภาพที่ 4-22 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ ของฐานข้อมูล BMS



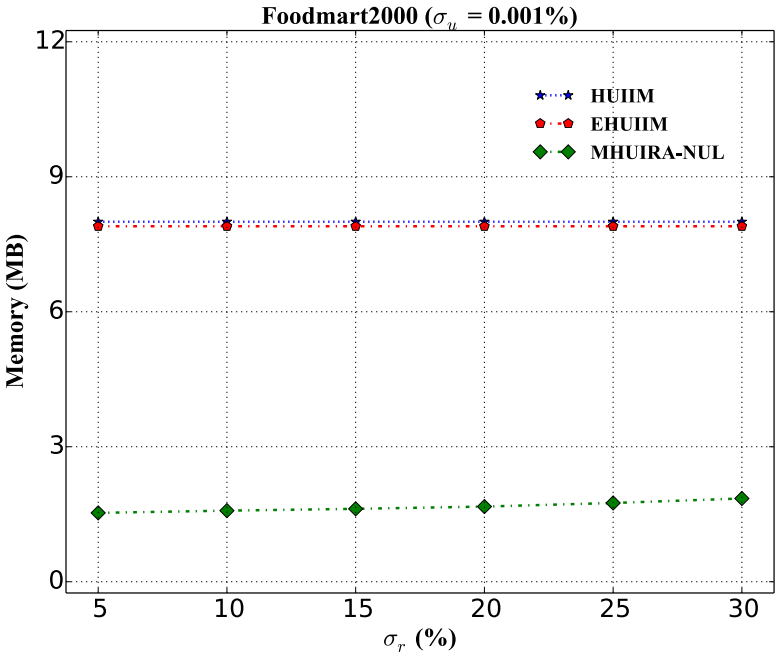
ภาพที่ 4-23 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ ของฐานข้อมูล Chainstore



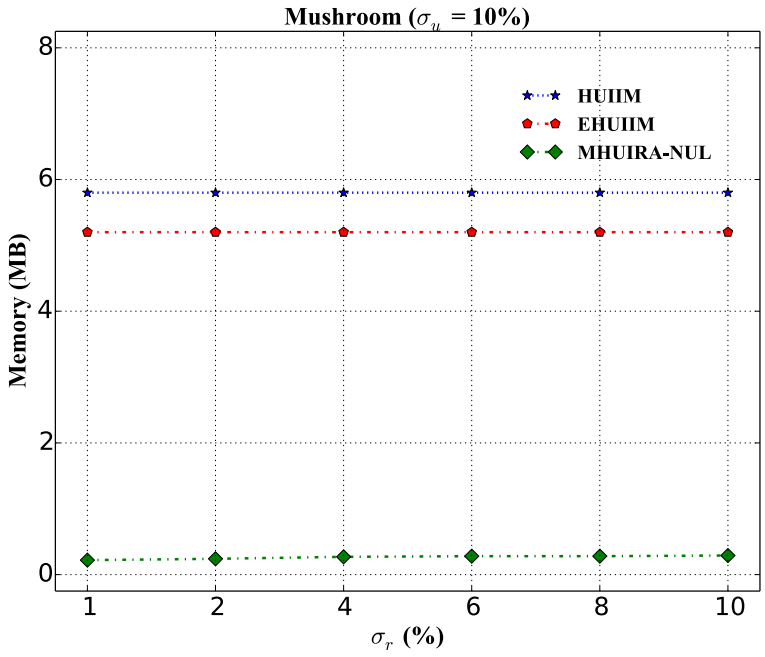
ภาพที่ 4-24 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ ของฐานข้อมูล Chess



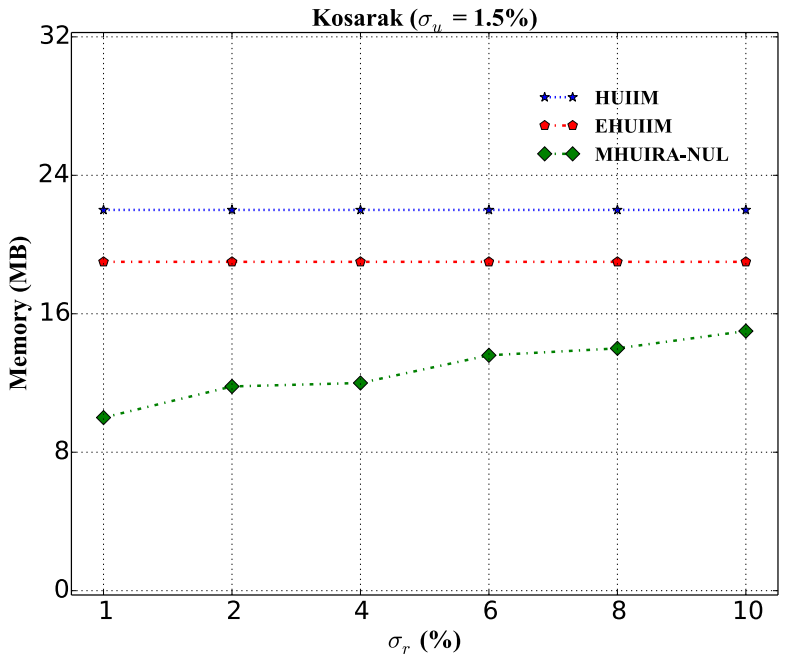
ภาพที่ 4-25 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ ของฐานข้อมูล Connect



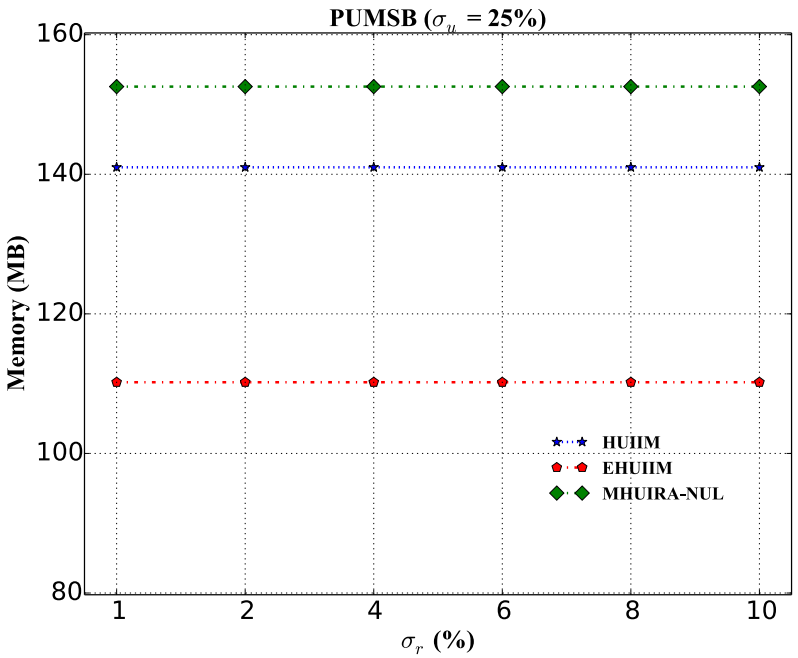
ภาพที่ 4-26 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ ของฐานข้อมูล Foodmart2000



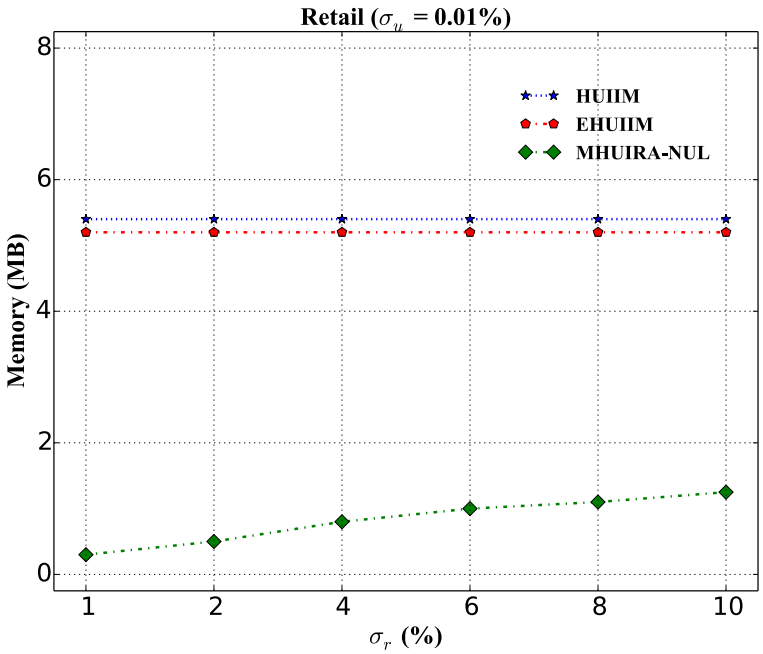
ภาพที่ 4-27 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ ของฐานข้อมูล Mushroom



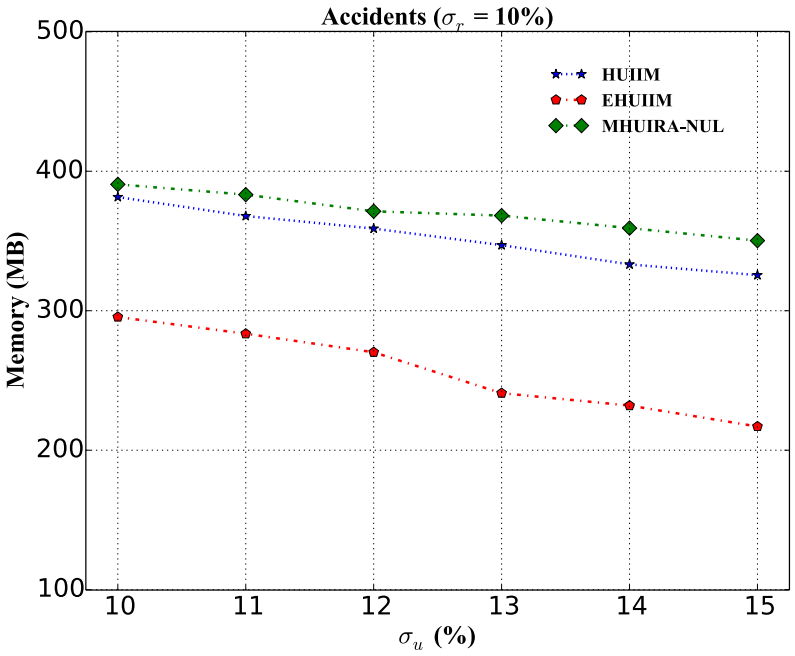
ภาพที่ 4-28 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ ของฐานข้อมูล Kosarak



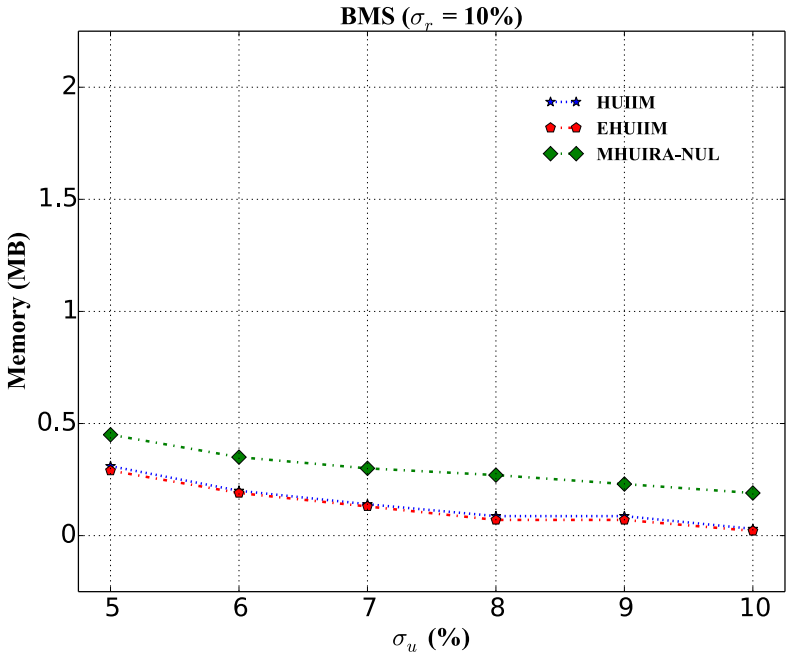
ภาพที่ 4-29 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ ของฐานข้อมูล PUMSB



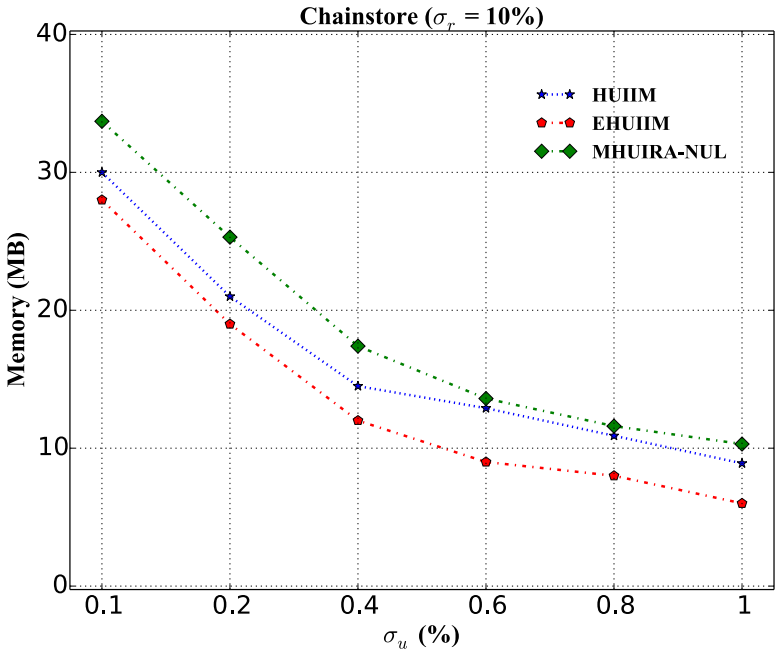
ภาพที่ 4-30 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ ของฐานข้อมูล Retail



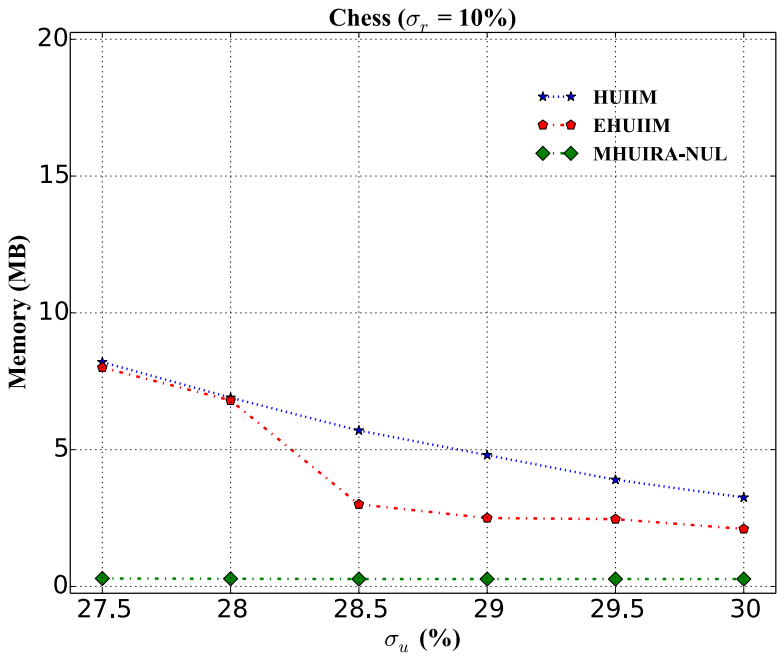
ภาพที่ 4-31 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล Accidents



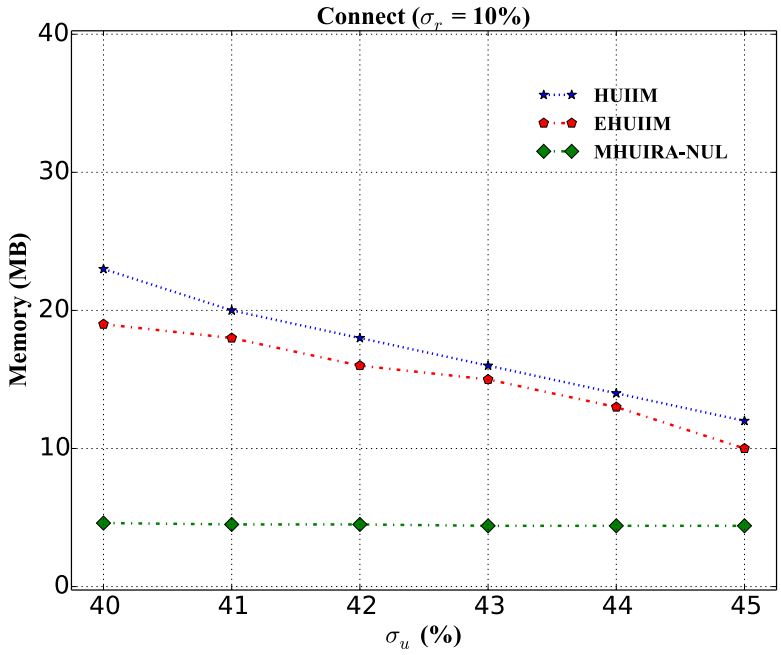
ภาพที่ 4-32 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล BMS



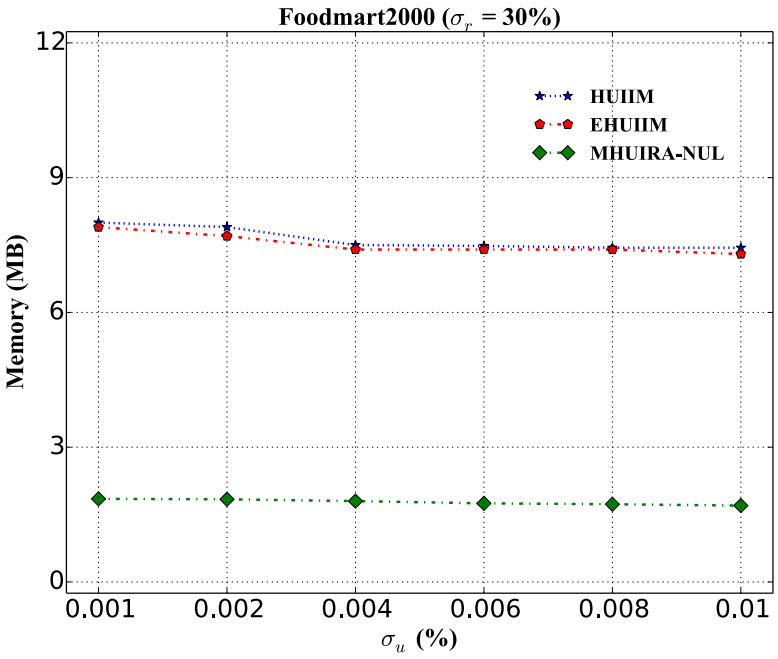
ภาพที่ 4-33 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล Chainstore



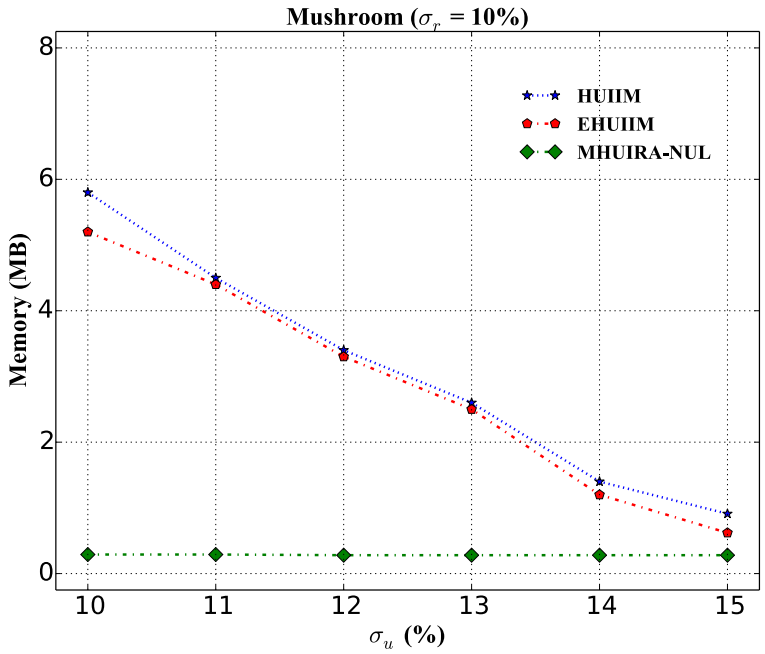
ภาพที่ 4-34 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล Chess



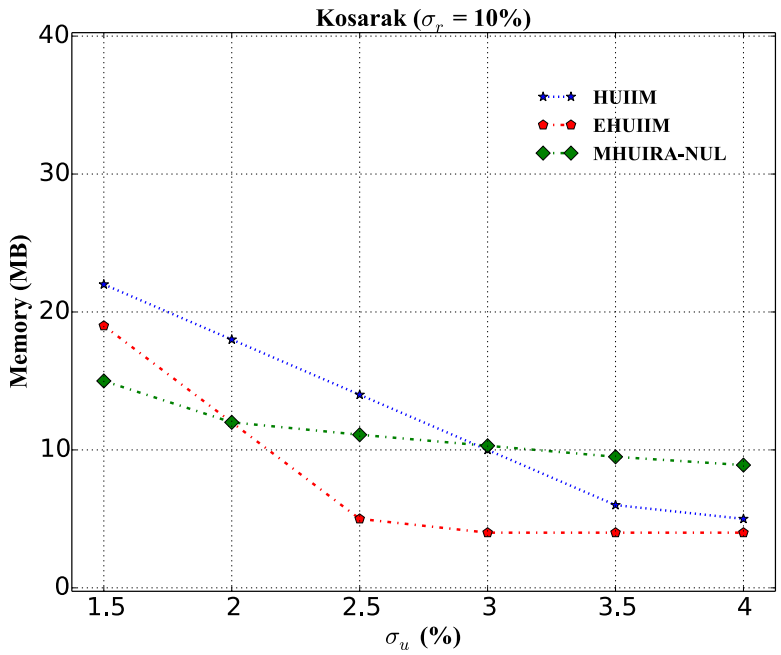
ภาพที่ 4-35 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล Connect



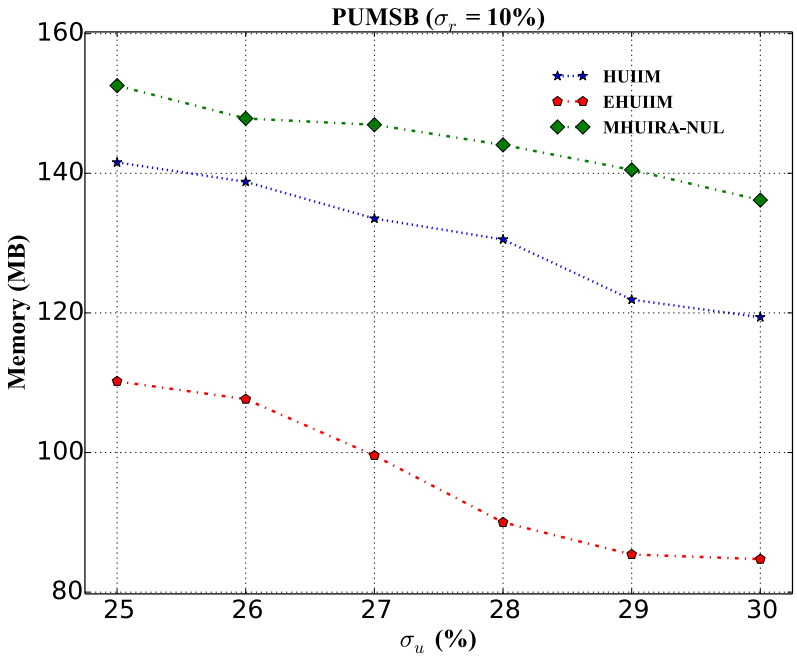
ภาพที่ 4-36 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล Foodmart2000



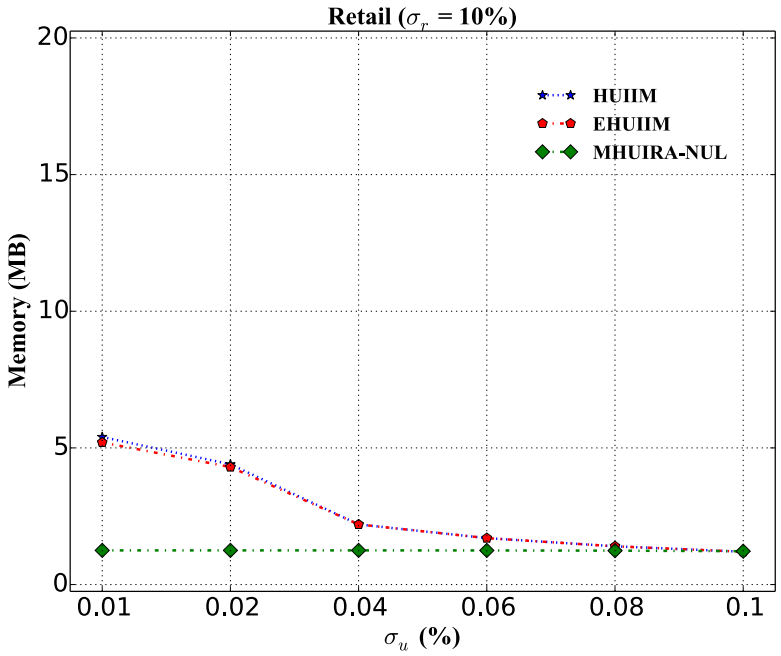
ภาพที่ 4-37 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล Mushroom



ภาพที่ 4-38 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล Kosarak



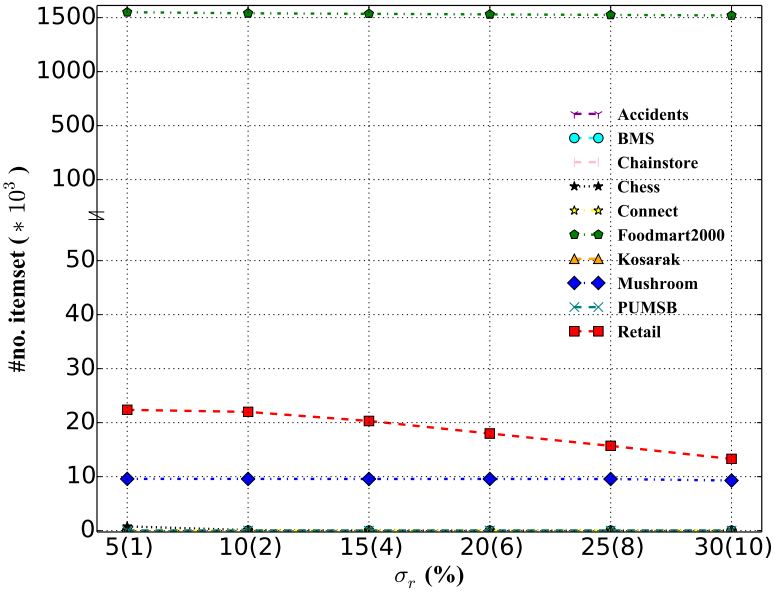
ภาพที่ 4-39 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล PUMSB



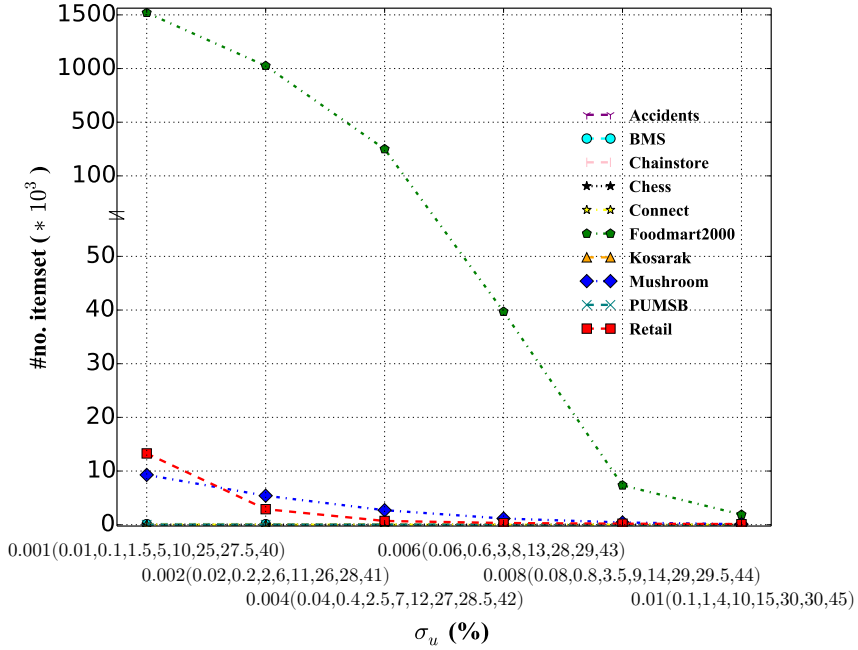
ภาพที่ 4-40 หน่วยความจำที่ใช้ในการประมวลผลเปรียบเทียบของขั้นตอนวิธี *HUIIM, EHUIIM* และ *MHURIA-NUL* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์ ของฐานข้อมูล Retail

4.3 จำนวนผลลัพธ์ที่ค้นหาได้

ในส่วนสุดท้าย แสดงให้เห็นถึงจำนวนของเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอที่ค้นหาได้จากชุดข้อมูลทั้งหมด ภายใต้การกำหนดค่าขีดแบ่งคุณประโยชน์แบบตายตัวและกำหนดค่าขีดแบ่งความสม่ำเสมอแบบแปรปรวน (แสดงในภาพที่ 4-41) และการกำหนดค่าขีดแบ่งความสม่ำเสมอแบบตายตัวและกำหนดค่าขีดแบ่งคุณประโยชน์แบบแปรปรวน (แสดงในภาพที่ 4-42) จากภาพทั้งสองจะสังเกตุได้ว่าจำนวนของเซตรายการจะลดลงอย่างมีนัยสำคัญเมื่อทำการกำหนดค่าขีดแบ่งความสม่ำเสมอหรือค่าขีดแบ่งคุณประโยชน์เพิ่มขึ้น นอกจากนี้เรายังสังเกตุได้ว่า ภาพที่ 4-41 เมื่อค่าขีดแบ่งความสม่ำเสมอมีค่าที่ต่ำจะทำให้มีเซตรายการจำนวนมากที่มีค่าความสม่ำเสมอมากกว่าค่าขีดแบ่งความสม่ำเสมอ ในทำนองเดียวกัน ภาพที่ 4-42 เมื่อค่าขีดแบ่งคุณประโยชน์มีค่าต่ำจะทำให้มีเซตรายการจำนวนมากที่มีค่าคุณประโยชน์ไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์



ภาพที่ 4-41 จำนวนรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอที่ค้นหาได้จากขั้นตอนวิธี *EHUIIM* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งความสม่ำเสมอ



ภาพที่ 4-42 จำนวนรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอที่ค้นหาได้จากขั้นตอนวิธี *EHUIIM* เมื่อทำการเปลี่ยนแปลงค่าขีดแบ่งคุณประโยชน์

4.4 การวิเคราะห์ความซับซ้อนของอัลกอริทึม

4.4.1 ความซับซ้อนเชิงเวลาของขั้นตอนวิธี *HUIIM* เป็น *O((nm)+(n)+(2^{p}2m))* เมื่อ *n* คือ จำนวนรายการทุกรายการที่เป็นสมาชิกของเซต *I, m* คือ จำนวนทรานแซกชันทั้งหมดในฐานข้อมูล *D* และ *p* คือ จำนวนรายการที่มีค่า *TUW* สูงหลังจากทำการพิจารณาลบรายการ

จากรายละเอียดในภาพที่ 3-1 ขั้นตอนวิธี *HUIIM* ทำการอ่านฐานข้อมูลทั้งหมด *m* ทรานแซกชัน โดยแต่ละทรานแซกชันอาจมีรายการทั้งหมดที่เป็นสมาชิกของเซต *I* เท่ากับ *n* ดังนั้นการอ่านฐานข้อมูลมีค่าเท่ากับ *m* × *n* จากนั้นรายการบางส่วนที่มีค่า *TWU* ต่ำจะถูกลบออก โดยการพิจารณาเพียงครั้งเดียวทำให้จำนวนของรายการที่มีค่า *TWU* สูง และเหลือพิจารณาต่อไปมีค่าเท่ากับ *p* ที่ซึ่งจำนวนครั้งในการพิจารณารายการมีค่าเท่ากับ *n*

ขั้นตอนการหาเซตรายการ *HUIIs* ทั้งหมด (ภาพที่ 3-2) แต่ละรายการจะถูกรวมเข้ากับรายการอื่นๆ เพื่อสร้างเซตรายการ 2 รายการ (แต่ละเซตรายการที่ถูกสร้างขึ้นจากแต่ละคู่ของรายการจะถูกรวมกับเซตรายการที่สร้างขึ้นจากโหนดพ่อแม่เดียวกัน) จำนวนรวมของเซตรายการที่รวมกันจะต้องพิจารณามีค่าเท่ากับ *2^p* สำหรับการรวมกันแต่ละครั้ง *NUL* ของสองรายการหรือเซตรายการจะต้องทำการอินเตอร์เซกชัน เพื่อคำนวณหาค่าคุณประโยชน์และค่าความสม่ำเสมอของเซตรายการที่ทำการรวมกัน จำนวนของการอินเตอร์เซกชันจะขึ้นอยู่กับขนาด(ทรานแซกชัน)ของ *NUL* ที่ซึ่งอาจมีทรานแซกชันทั้งหมดคือ *m* ดังนั้นการอินเตอร์เซกชันจะมีค่าเท่ากับ *m* × *m* และความซับซ้อนของเวลาของขั้นตอนการหาเซตรายการ *HUIIs* ทั้งหมดจะมีค่าเท่ากับ *2^p* × *2m* สุดท้ายจากขั้นตอนทั้งหมดใน *HUIIM* ที่กล่าวมาข้างต้นความซับซ้อนของเวลาทั้งหมดของขั้นตอนวิธี *HUIIM* คือ *O((nm)+(n)+(2^{p}2m))*

4.4.2 ความซับซ้อนเชิงเวลาของขั้นตอนวิธี *EHUIIM* เป็น *O((nm)+(n^2)+(2^{q}2m))* เมื่อ *n* คือ จำนวนรายการทุกรายการที่เป็นสมาชิกของเซต *I, m* คือ จำนวนทรานแซกชันทั้งหมดในฐานข้อมูล *D* และ *q* คือ จำนวนรายการที่มีค่า *TUW* สูงหลังจากทำการลดทอนปริภูมิสถานะ

จากรายละเอียดในภาพที่ 3-3 ขั้นตอนวิธี *EHUIIM* ทำการอ่านฐานข้อมูลทั้งหมด *m* ทรานแซกชัน โดยแต่ละทรานแซกชันอาจมีรายการทั้งหมดที่เป็นสมาชิกของเซต *I* เท่ากับ *n* ดังนั้นการอ่านฐานข้อมูลจะมีค่าเท่ากับ *m* × *n* จากนั้น *EHUIIM* จะทำการลบรายการที่มีค่า *TWU* ต่ำทั้งหมดมีค่าเท่ากับ *n^2* ซึ่งทำให้จำนวนรายการที่เหลือพิจารณาต่อไปมีค่าเท่ากับ *q* (<u>หมายเหตุ</u> *q* จะน้อยกว่าหรือเท่ากับ *p* จำนวนรายการที่มีค่า *TWU* สูงของขั้นตอนวิธี *HUIIM*) ขั้นตอนการหาเซตรายการ *HUIIs* ทั้งหมด *EHUIIM* จะทำแบบเดียวกับ *HUIIM* ซึ่งจะมีค่าเท่ากับ *2^q* × *2m* สุดท้ายจากขั้นตอนทั้งหมดใน *EHUIIM* ที่กล่าวมาข้างต้นความซับซ้อนของเวลาทั้งหมดของขั้นตอนวิธี *EHUIIM* คือ *O((nm)+(n^2)+(2^{q}2m))*

4.4.3 ความซับซ้อนของการใช้พื้นที่ของขั้นตอนวิธี *HUIIM* และ *EHUIIM* เป็น *O((nm)+ (u^{2}m)+((u-2)um))* เมื่อ *n* คือ จำนวนรายการทุกรายการที่เป็นสมาชิกของเซต *I, m* คือ จำนวนทรานแซกชันทั้งหมดในฐานข้อมูล *D* และ *u* คือ จำนวนรายการที่มีค่า *TUW* สูงหลังจากทำการลดทอนปริภูมิสถานะ

การใช้พื้นที่ของทั้งสองวิธีสามารถคำนวณได้จากจำนวนรายการหรือเซตรายการและ *NUL* ที่ถูกจัดเก็บอยู่ในหน่วยความจำ แต่ละรายการหรือเซตรายการสามารถมีสมาชิกใน *NUL* มากที่สุดคือ

m ดังนั้นการใช้พื้นที่ทั้งหมดของรายการทั้งหมดจึงเป็น $n \times m$ จากนั้นรายการที่มีค่า *TWU* ต่ำจะถูกลบออกและรายการที่เหลือพิจารณาต่อคือ *u* รายการที่เหลือทั้งหมดจะถูกรวมกับรายการอื่นๆ เพื่อสร้าง u^2 เซตรายการ 2 รายการ จากนั้นการใช้พื้นที่เพื่อเก็บ *NUL* ของเซตรายการที่บรรจุไปด้วย 2 รายการทั้งหมด ซึ่งจะมีค่าเท่ากับ $u^2 \times m$ ต่อไปพิจารณาและรวมเซตรายการ ที่มีรายการนำหน้าเดียวกันเพื่อสร้างเซตรายการ *3, 4, 5, ..., u* ที่ค่า *TWU* สูง ซึ่งชุดข้อมูล *p* เซตรายการ แต่ละชุดสามารถมีได้สูงสุด *u* เซตรายการ การใช้พื้นที่เพื่อเก็บ *3, 4, 5, ..., u* เซตรายการที่มีรายการนำหน้าเดียวกันระหว่างกระบวนการจะเท่ากับ $(u - 2) \times u \times m$ สุดท้ายต้องมีการเก็บเซตรายการทุกขนาดไว้ในระหว่างกระบวนการ การใช้พื้นที่ทั้งหมดของขั้นตอนวิธี *HUIIM* และ *EHUIIM* คือ $O((nm)+(u^2m)+((u-2)um))$

บทที่ 5

สรุปผลวิจัย

งานวิจัยนี้ได้นำเสนอแนวความคิดในการค้นหารูปแบบ ภายใต้การพิจารณาความไม่สม่ำเสมอของการปรากฏและค่าคุณประโยชน์ของรูปแบบ (ผลกำไร ต้นทุน หรือ อื่นๆ) ซึ่งการค้นหารูปแบบภายใต้การพิจารณาข้างต้น จะมีส่วนขยายในการสังเกตุพฤติกรรมการซื้อของลูกค้าที่ทำการซื้อสินค้าที่ให้ผลตอบแทนสูง จึงเป็นข้อมูลที่ช่วยในการจัดการบริหารสินค้าคงคลัง วางแผนการผลิต และทำให้ผู้บริหารสามารถวิเคราะห์ความต้องการลูกค้า สามารถวางแผนการตลาดได้ดียิ่งขึ้น จากกรอบความคิดข้างต้น รูปแบบหนึ่งๆ จะเป็นรูปแบบที่น่าสนใจก็ต่อเมื่อเป็นรูปแบบที่ปรากฏอย่างไม่สม่ำเสมอ (กล่าวคือ รูปแบบมีค่าความสม่ำเสมอมากกว่าค่าขีดแบ่งความสม่ำเสมอที่ผู้ใช้กำหนด) และเป็นรูปแบบที่มีค่าคุณประโยชน์สูง (กล่าวคือ รูปแบบมีค่าคุณประโยชน์ไม่น้อยกว่าค่าขีดแบ่งคุณประโยชน์ที่ผู้ใช้กำหนด)

ในการค้นหารูปแบบอย่างมีประสิทธิภาพ 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"* เพื่อทำการลดทอนปริภูมิสถานะของการค้นหาเซตรายการ นอกจากนี้เทคนิคใหม่ *"Efficient Pruning technique"* ถูกออกแบบและประยุกต์ใช้กับ *HUIIM* เพื่อเพิ่มประสิทธิภาพในการคำนวณเรียกว่า *"Efficient High Utility Itemset with Irregular occurrence Miner, EHUIIM"*

ผู้วิจัยได้ทำการทดลองเพื่อทดสอบประสิทธิภาพของขั้นตอนวิธีที่เสนอกับชุดข้อมูลจริง 10 ชุดข้อมูล โดยทำการทดลองในสามแง่มุมด้วยกันคือ เวลาที่ใช้ในการหาผลลัพธ์ หน่วยความจำที่ใช้ในการหาผลลัพธ์ และจำนวนผลลัพธ์ที่สามารถค้นหาได้ โดยการทดลองชี้ให้เห็นว่า ขั้นตอนวิธี *EHUIIM* สามารถค้นหารูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอได้อย่างมีประสิทธิภาพ

วิจารณ์ผลการดำเนินงาน

ข้อดีของวิทยานิพนธ์

1. การวิเคราะห์พฤติกรรมภายใต้กรอบแนวคิดใหม่ที่เกี่ยวกับการค้นหารูปแบบที่ให้ผลตอบแทนสูงและการค้นหารูปแบบที่ปรากฏแบบไม่สม่ำเสมอ (*HUIIM*)
2. ทราบถึงพฤติกรรมของผู้บริโภคจะนำไปสู่การค้นหาสาเหตุของการเกิดขึ้นของพฤติกรรมหรือรูปแบบการบริโภคซึ่งจะสามารถนำข้อมูลดังกล่าวไปประกอบการตัดสินใจเพื่อทำการพัฒนาผลิตภัณฑ์และหรือขั้นตอนการดำเนินธุรกิจได้

ปัญหาของวิทยานิพนธ์

การทดสอบการหาเซตรายการที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอจากขั้นตอน *HUIIM* และ *EHUIIM* พบว่าเมื่อข้อมูลมีขนาดใหญ่หรือค่าขีดแบ่งคุณประโยชน์ (utility threshold) ขั้นต่ำที่กำหนดไว้มีค่าน้อย จะทำให้ขั้นตอน *HUIIM* และ *EHUIIM* ใช้เวลาในการคำนวณมาก

ข้อจำกัดของวิทยานิพนธ์

1. ข้อมูลที่จะทำการพิจารณาจะต้องมีลักษณะเป็นแบบทรานแซกชันที่ประกอบไปด้วยรายการสินค้าและจำนวนครั้งการปรากฏ
2. ผู้ที่ต้องการผลลัพธ์จะต้องทำการกำหนดค่าขีดแบ่ง (threshold) เพื่อใช้เป็นเกณฑ์สำหรับวัดความน่าสนใจของเซตรายการที่จะทำการค้นหาจากข้อมูลที่ต้องการวิเคราะห์

ข้อเสนอแนะของวิทยานิพนธ์

งานวิจัยนี้นำเสนอการค้นหารูปแบบที่มีค่าคุณประโยชน์สูงและปรากฏอย่างไม่สม่ำเสมอจากฐานข้อมูล สามารถนำวิธีดังกล่าวไปสร้างเป็นโปรแกรมเพื่อนำไปประกอบการตัดสินใจในการพัฒนาหรือปรับปรุงกระบวนการในการดำเนินธุรกิจ

บรรณานุกรม

Chan, R., Yang, Q., Shen, Y.D. (2003), Mining high utility itemsets, in: *IEEE International Conference on Data Mining*, pp. 19–26.

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.

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.

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.

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.

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.

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.

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.

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.

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.

MacKinnon, R.K., Strauss, T.D., and Leung, C.K. (2014), DISC: efficient uncertain frequent pattern mining with tightened upper bounds. *In Proc. EEE ICDM Workshops*, pp. 1038 – 1045.

Krishnamoorthy, S. (2015), Pruning strategies for mining high utility itesmets, *Expert Systems with Applications*, 42, 2015, pp. 2371 – 2381.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

Zihayat, M., and An, A. (2014), Mining top-k high utility patterns over data streams, *Information Science*, 285, 2014, pp. 138 - 161.

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.

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.

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.

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.

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 (ICCIC)*, 2012, pp. 1–4.

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.

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.

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.

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, I*nternational Workshop on Databases in Networked Information Systems*, 2015, pp. 12 – 26.

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.

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

Ampawan, 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,

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

Amphawan, K., Surarerks, A. (2015-b), Pushing regularity constraint on high utility itemsets mining, *Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*

ภาคผนวก

ภาพผนวก ก

จริยธรรมในงานวิจัย

ที่ ๗๕/๒๕๖๐

เอกสารรับรองผลการพิจารณาจริยธรรมการวิจัยในมนุษย์
มหาวิทยาลัยบูรพา

คณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์ มหาวิทยาลัยบูรพา ได้พิจารณาโครงการวิจัย

รหัสโครงการวิจัย Sci 015/2560
โครงการวิจัยเรื่อง การค้นหารูปแบบที่ให้ผลตอบแทนสูงโดยปรากฏไม่สม่ำเสมอ
หัวหน้าโครงการวิจัย นายศุภชัย เล้าวิบูลย์
หน่วยงานที่สังกัด นิสิตระดับบัณฑิตศึกษา คณะวิทยาการสารสนเทศ

คณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์ มหาวิทยาลัยบูรพา ได้พิจารณาแล้วเห็นว่าโครงการวิจัยดังกล่าวเป็นไปตามหลักการของจริยธรรมการวิจัยในมนุษย์ โดยที่ผู้วิจัยเคารพสิทธิและศักดิ์ศรีในความเป็นมนุษย์ ไม่มีการล่วงละเมิดสิทธิ สวัสดิภาพ และไม่ก่อให้เกิดภยันตรายแก่ตัวอย่างการวิจัยและผู้เข้าร่วมโครงการวิจัย

จึงเห็นสมควรให้ดำเนินการวิจัยในขอบข่ายของโครงการวิจัยที่เสนอได้ (ดูตามเอกสารตรวจสอบ)

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

การรับรองผลการพิจารณาจริยธรรมการวิจัยในมนุษย์ฉบับนี้ มีผลถึงวันที่ ๑ เดือน เมษายน พ.ศ. ๒๕๖๑

ออกให้ ณ วันที่ ๒๒ เดือน มีนาคม พ.ศ. ๒๕๖๐

ลงนาม
(ผู้ช่วยศาสตราจารย์ ดร.วิทวัส แจ้งเอี่ยม)
ประธานคณะกรรมการพิจารณาจริยธรรมการวิจัยในมนุษย์
มหาวิทยาลัยบูรพา

ภาพผนวก ข

การเผยแพร่ผลงานวิจัย

Certificate of Contributions

Supachai Laoviboon and Komate Amphawan

Entitled

Mining High-Utility Itemsets with Irregular Occurrence

Has Contributed To

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

February 1 - 4, 2017

Amari Pattaya, Chon Buri, Thailand

Organized by

Faculty of Informatics, Burapha University, Thailand

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

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

Technical Co-Sponsor

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 uitlity mining; regular/irregular itemsets; irrgular 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 shelfs 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 *HUI*s, 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], d^{2}HUP [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 $I = \{i_1, i_2, \ldots, i_n\}$ 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 $X = \{i_j, \ldots, i_k\} \subseteq I$ is called a k-itemset, if X contains k items. A transactional database $D = \{t_1, t_2, \ldots, t_m\}$ 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

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

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 $T^X = \{p^X, \ldots, q^X\}$ 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.

tid	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 $X = \{i_j, \ldots, i_k\}$ 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 occuring 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 $r(X) = max(fr(X, t_p), r(X, t_p, t_q), \ldots, r(X, t_y, t_z), lr(X, t_z))$.

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 $NUL^X = \{e_1, e_2, \ldots, e_k\}$ 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 $X = \{i_p, ..., i_q, i_r\}$,

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.* $\{< 1, 4, 319, 0 >, < 3, 6, 6, 0 >, < 4, 6, 30, 0 >, < 5, 2, 26, 0 >, < 8, 2, 33, 0 >\}$. The first entry $< 1, 4, 319, 0 >$ 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 $tu(t_1) = (2\times2)+(3\times3)+(13\times20)+(2\times25) = 323$ and then collected in *tuList*. In addition, transaction weighted utilities and *NUL*s 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).

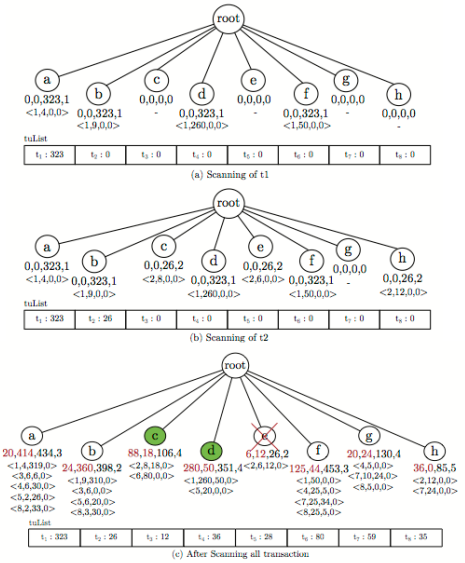


Fig. 1: *HUIIM-ScanningDatabase* : *tulist* and *HUII-tree*

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 $\in I$
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 $\langle p, u(i, t_p), 0, 0\rangle$ 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 $e = \langle p, u(i, t_p), 0, 0\rangle$ 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 $e = \langle p, u(i, t_p), 0, 0\rangle$ 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 $e = \langle p, u(i, t_p), ru(i, t_p), 0\rangle$
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 $X = \{i_j, \ldots, i_k\}$ 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 mering 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', $NUL^a = \{< 1, 4, 319, 0 >, < 3, 6, 6, 0 >, < 4, 6, 30, 0 >, < 5, 2, 26, 0 >, < 8, 2, 33, 0 >\}$ and $NUL^b = \{< 1, 9, 310, 0 >, < 3, 6, 0, 0 >, < 5, 6, 20, 0 >, < 8, 3, 30, 0 >\}$ 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 $NUL^{ab} = \{< 1, 13, 310, 4 >, < 3, 12, 0, 6 >, < 5, 8, 20, 2 >, < 8, 5, 30, 2 >\}$, 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 $NUL^c = \{< 2, 8, 12, 0 >, < 6, 80, 0, 0 >\}$ 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.

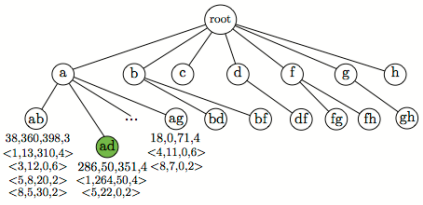


Fig. 2: HUII-tree containing 2-HUIIs

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 $NUL^{abd} = \{< 1, 273, 50, 13 >, < 5, 28, 0, 8 >, \}$. 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*, dut 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-Mining*HUIIs*

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** $tou(i) \geq \sigma_u$ **then**
3: **for** item j in HUII-tree (where $i \prec j$) **do**
4: $NUL^{ij} \leftarrow \emptyset$
5: **IntersectNUL** (NUL^{ij}, NUL^{i}, NUL^{j})
6: calculate $r(ij)$, $twu(ij)$ $u(ij)$, $ru(ij)$, $up(ij)$ from NUL^{ij}
7: calculate $tou(ij) \leftarrow u(ij) + ru(ij) - up(ij)$
8: **if** $twu(ij) \geq \sigma_u$ **then**
9: create a node of itemset ij with $ps(ij)$, $twu(ij)$, $u(ij)$, $ru(ij)$, $up(ij)$ 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, σ_u, σ_r)
15: **Procedure** MiningAllHUII(*HUII-tree*, node of X, σ_u, σ_r)
16: **for** child node of X with itemset Y **do**
17: **if** $tou(Y) \geq \sigma_u$ **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 $r(YZ)$, $twu(YZ)$ $u(YZ)$, $ru(YZ)$, $up(YZ)$ from NUL^{YZ}
24: calculate $tou(YZ) \leftarrow u(YZ) + ru(YZ) - up(YZ)$
25: **if** $twu(ij) \geq \sigma_u$ **then**
26: create a node of itemset YZ with $ps(YZ)$, $twu(YZ)$, $u(YZ)$, $ru(YZ)$, $up(YZ)$ 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 $e = \langle p, u(YZ, t_p), ru(YZ, t_p), up(YZ, t_p)\rangle$ 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 $twu(Q) \leftarrow 0$ and set $ru(Q) \leftarrow 0$
35: **for** each entry $e = \langle p, u(Q, t_p), ru(Q, t_p), up(Q, t_p)\rangle$ 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: $twu(Q) \leftarrow twu(Q) + u(Q, t_p) + ru(Q, t_p) + up(Q, t_p)$
38: $ru(Q) \leftarrow ru(Q) + ru(Q, t_p)$
39: update $tou(Q) \leftarrow u(Q) + ru(Q)$

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 theshold (Thanks to *downward closuse 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.

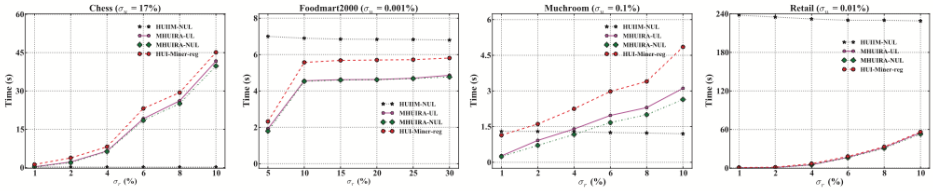


Fig. 3: Runtime with variation of regularity threshold

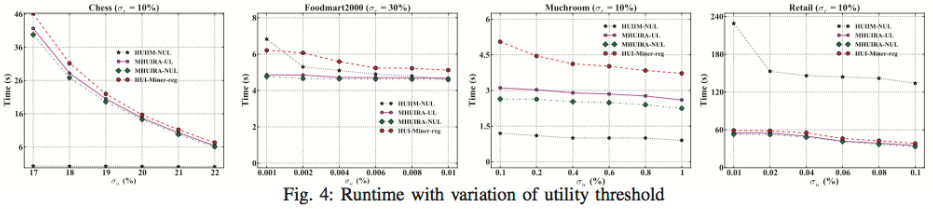


Fig. 4: Runtime with variation of utility threshold

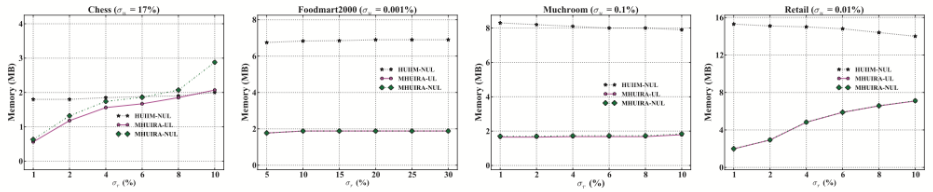


Fig. 5: Memory usage with variation of regularity threshold

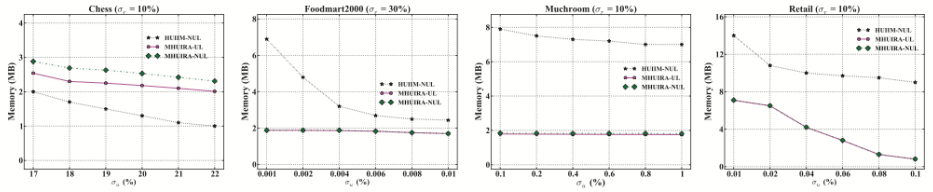


Fig. 6: Memory usage with variation of utility threshold

[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.