การเปรียบเทียบประสิทธิภาพอัลกอริทึมการทำเหมืองกฎความสัมพันธ์ระหว่าง Apriori, FP-Growth, FP-Max และ H-Mine สำหรับการวิเคราะห์ตะกร้าสินค้า

Performance comparison of association rule mining algorithms among Apriori, FP-Growth, FP-Max, and H-Mine for market basket analysis

กริชบดินทร์ ผิวหอม[1*]

Kritbodin Phiwhorm[1*]

Received: 11 September 2024 ; **Revised**: 20 December 2024 ; **Accepted**: 19 February 2025

บทคัดย่อ

การทำเหมืองกฎความสัมพันธ์เป็นเทคนิคสำคัญในการวิเคราะห์ตะกร้าสินค้าสำหรับธุรกิจค้าปลีก แต่มักประสบปัญหาด้านความเร็วในการประมวลผลและการใช้หน่วยความจำ โดยเฉพาะกับชุดข้อมูลขนาดใหญ่ งานวิจัยนี้นำเสนอการเปรียบเทียบประสิทธิภาพของ 4 อัลกอริทึม ได้แก่ Apriori, FP-Growth, FP-Max และ H-Mine โดยใช้ชุดข้อมูลร้านขายของชำสำหรับการวิเคราะห์ตะกร้าสินค้า ภายใต้ค่าสนับสนุนที่แตกต่างกัน ผลการวิจัยพบว่าอัลกอริทึม H-Mine มีประสิทธิภาพดีที่สุดทั้งด้านความเร็วในการประมวลผลและการใช้หน่วยความจำ เนื่องจากใช้โครงสร้างข้อมูลแบบ Hyperlink ที่มีประสิทธิภาพ ตามด้วยอัลกอริทึม FP-Growth และ FP-Max ที่ใช้โครงสร้าง FP-Tree ช่วยลดการสแกนฐานข้อมูล ในขณะที่อัลกอริทึม Apriori แสดงประสิทธิภาพต่ำที่สุด

คำสำคัญ: กฎความสัมพันธ์, อัลกอริทึมอะพริโอริ, อัลกอริทึมเอฟพี-โกรท, อัลกอริทึมเอฟพี-แมก, อัลกอริทึมเอช-ไมน์

Abstract

Association rule mining is a crucial technique for market basket analysis in retail businesses, but it often faces challenges in processing speed and memory usage, particularly with large-scale datasets. This research presents a performance comparison of four algorithms: Apriori, FP-Growth, FP-Max, and H-Mine, using a grocery store dataset for market basket analysis under varying support thresholds. The results showed that the H-Mine algorithm demonstrated superior performance in both execution time and memory usage, attributed to its efficient Hyperlink data structure, followed by FP-Growth and FP-Max algorithms, which employ FP-Tree structure to minimize database scanning. Meanwhile, the Apriori algorithm exhibited the lowest performance.

Keywords: Association rule mining, Apriori, FP-Growth, FP-Max, H-Mine

[1] อาจารย์, สาขาวิทยาการคอมพิวเตอร์ คณะศิลปศาสตร์และวิทยาศาสตร์ มหาวิทยาลัยราชภัฏศรีสะเกษ 33000

[1] Lecturer, Department of Computer Science, Faculty of Liberal Arts and Sciences, Sisaket Rajabhat University 33000

* Corresponding author, e-mail: kritbodin.p@sskru.ac.th

บทนำ

การทำเหมืองข้อมูลกฎความสัมพันธ์ (Association Rule Mining) เป็นเทคนิคเพื่อช่วยวิเคราะห์พฤติกรรมผู้บริโภค โดยเฉพาะการวิเคราะห์การซื้อสินค้าของลูกค้า (Market Basket Analysis) ที่ช่วยให้สามารถเข้าใจพฤติกรรมการซื้อของลูกค้าได้ดีขึ้น (Adeniji *et al*., 2015; Mustakim *et al*., 2018)

เทคนิคการทำเหมืองข้อมูลกฎความสัมพันธ์ได้ถูกนำมาใช้ค้นหาความสัมพันธ์ที่ซ่อนอยู่ในข้อมูล (Mustakim *et al*., 2018; Slimani & Lazzez, 2014) และมีหลายอัลกอริทึมที่ได้รับความนิยมในการค้นหาความสัมพันธ์ เช่น อัลกอริทึม Apriori, FP-Growth, FP-Max และ H-Mine เป็นต้น ซึ่งแต่ละวิธีมีแนวคิด วิธีการ จุดแข็ง และข้อจำกัดที่แตกต่างกันไป (Nigam *et al*., 2017) อย่างไรก็ตาม การเพิ่มขึ้นของปริมาณข้อมูลอย่างรวดเร็วได้นำมาซึ่งความท้าทายด้านประสิทธิภาพในการประมวลผลและการใช้ทรัพยากรหน่วยความจำเป็นอย่างมาก (Garg & Kumar, 2013)

งานวิจัยนี้มีวัตถุประสงค์เพื่อเปรียบเทียบประสิทธิภาพของ 4 อัลกอริทึมในการทดลองกับชุดข้อมูลร้านขายของชำ (Groceries dataset) ทั้งด้านความเร็วในการประมวลผลและด้านการใช้หน่วยความจำ เพื่อเป็นข้อมูลที่ช่วยให้การเลือกใช้อัลกอริทึมได้เหมาะสมกับข้อมูลที่หลากหลาย และการทำงานมีประสิทธิภาพยิ่งขึ้น

วรรณกรรมที่เกี่ยวข้อง

งานวิจัยนี้ใช้เทคนิคการค้นหาความสัมพันธ์ของข้อมูลที่ได้รับความนิยม 4 อัลกอริทึม ดังนี้

1. อัลกอริทึม Apriori

อัลกอริทึม Apriori เป็นอัลกอริทึมพื้นฐานที่นำเสนอโดย Agrawal and Srikant (1994) ซึ่งอาศัยแนวคิดการค้นหาชุดรายการที่เกิดขึ้นบ่อยแบบเป็นลำดับขั้น (level-wise approach) โดยเริ่มจากชุดรายการขนาดเล็ก และค่อย ๆ ขยายไปสู่ชุดรายการที่มีขนาดใหญ่ขึ้น

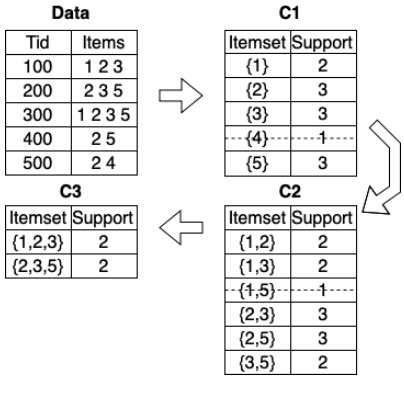

Figure 1 The Apriori algorithm working process

จาก Figure 1 แสดงการทำงานของอัลกอริทึม Apriori เพื่อหาไอเทมเซตที่เกิดขึ้นบ่อย (Frequent itemsets) หรือ ไอเทมที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ (Minimum support) โดยเริ่มจากการนำข้อมูล (Data) มาสร้างไอเทมเซตขนาด 1 (C1) และคำนวณค่าสนับสนุน (Support value) ของแต่ละไอเทม หากค่าสนับสนุนต่ำกว่าเกณฑ์ที่กำหนดจะถูกตัดออก (จากตัวอย่างกำหนดค่าสนับสนุนมากกว่าหรือเท่ากับ 2) เช่น ไอเทมเซต {4} มีค่าสนับสนุนเป็น 1 จึงถูกตัดออก จากนั้นจะนำไอเทมที่เหลือมาสร้างเป็นไอเทมเซตขนาด 2 (C2) และคำนวณค่าสนับสนุนอีกครั้ง โดยไอเทมเซต {1,5} มีค่าสนับสนุนต่ำกว่า 2 จึงถูกตัดออกเช่นกัน

กระบวนการนี้จะดำเนินต่อไปโดยนำไอเทมเซตที่เหลือมาสร้างเป็นไอเทมเซตขนาด 3 (C3) ซึ่งในขั้นตอนสุดท้ายจะเหลือเพียงไอเทมเซต {1,2,3} และ {2,3,5} แสดงให้เห็นว่าอัลกอริทึม Apriori ต้องสแกนฐานข้อมูลหลายรอบและสร้างชุดตัวแทนจำนวนมากในแต่ละรอบ แม้ว่าจะมีการตัดไอเทมเซตที่มีค่าสนับสนุนต่ำออกเพื่อลดขนาดของชุดตัวแทนในรอบถัดไป แต่ก็ยังคงต้องใช้ทรัพยากรในการประมวลผลค่อนข้างมาก

2. อัลกอริทึม FP-Growth

เพื่อแก้ไขข้อจำกัดของอัลกอริทึม Apriori จึงได้มีการพัฒนาอัลกอริทึม FP-Growth (Frequent Pattern Growth) โดย Han *et al*. (2000) ซึ่งใช้อาศัยโครงสร้างข้อมูลแบบต้นไม้ที่เรียกว่า FP-Tree (Frequent Pattern Tree) ในการจัดเก็บข้อมูลและค้นหารูปแบบที่เกิดขึ้นบ่อยอย่างมีประสิทธิภาพ

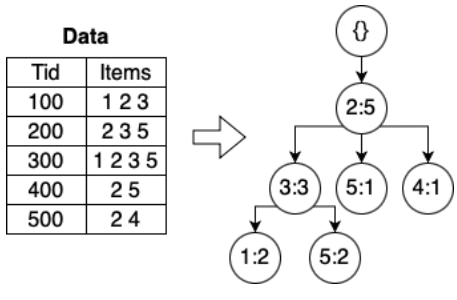

Figure 2 The FP-Growth algorithm working process

จาก Figure 2 แสดงการทำงานของอัลกอริทึม FP-Growth โดยเริ่มจากการสร้างโหนดราก (Root) เป็น {} จากนั้นจะแทรกรายการข้อมูลลงในต้นไม้ทีละรายการ โดยรายการที่มีค่าความถี่สูงจะอยู่ใกล้รากมากกว่า เช่น โหนด (2:5) แสดงว่าไอเทม 2 มีความถี่ 5 ครั้ง และมีการเชื่อมโยงไปยังไอเทมอื่นๆ ที่เกิดร่วมกัน ทำให้ลดพื้นที่การจัดเก็บข้อมูล

และประสิทธิภาพในการค้นหาไอเทมเซตที่เกิดขึ้นบ่อย เนื่องจากสามารถใช้โครงสร้างต้นไม้แทนการสแกนฐานข้อมูลซ้ำๆ เหมือนอัลกอริทึม Apriori

3. อัลกอริทึม FP-Max

อัลกอริทึม FP-Max เป็นการพัฒนาต่อยอดจากอัลกอริทึม FP-Growth โดย Grahne and Zhu (2003) ซึ่งมุ่งเน้นการค้นหาเฉพาะชุดรายการที่เกิดขึ้นบ่อยสูงสุด (maximal frequent itemsets) ส่งผลให้สามารถลดพื้นที่การค้นหา และเพิ่มประสิทธิภาพในการประมวลผล โดยเฉพาะในกรณีของฐานข้อมูลที่มีชุดรายการที่เกิดขึ้นบ่อยจำนวนมาก

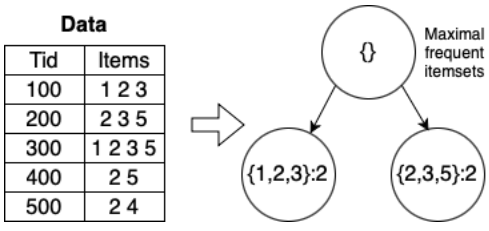

Figure 3 The FP-Max algorithm working process

จาก Figure 3 เป็นการทำงานของอัลกอริทึม FP-Max แสดงให้เห็นว่าโหนดราก {} แตกออกเป็นสองกิ่งที่เก็บไอเทมเซตที่เกิดขึ้นบ่อยสูงสุด คือ {1,2,3} และ {2,3,5} โดยแต่ละชุดมีความถี่เท่ากับ 2 ทำให้ลดการจัดเก็บไอเทมเซตย่อยที่ไม่จำเป็น ส่งผลให้ใช้พื้นที่จัดเก็บน้อยลงและค้นหาได้เร็วขึ้น (Borah & Nath, 2021)

4. อัลกอริทึม H-Mine

อัลกอริทึม H-Mine พัฒนาโดย Jian และคณะ ในปี 2001 (Pei *et al*., 2001) เพื่อแก้ไขปัญหาการใช้หน่วยความจำของอัลกอริทึม FP-Growth โดยใช้โครงสร้าง H-struct ไฮเปอร์ลิงค์ (Hyperlink) ไปยังข้อมูลที่มีไอเทม 1 อยู่ วิธีนี้ช่วยลดการใช้หน่วยความจำเมื่อเทียบกับการสร้างต้นไม้แบบ FP-Tree (Pei *et al*., 2001)

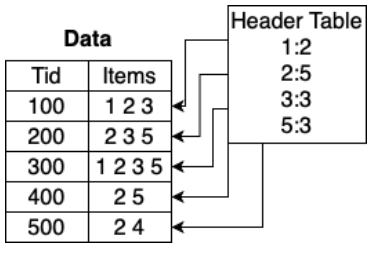

Figure 4 The H-Mine algorithm working process

จาก Figure 4 เป็นการทำงานของอัลกอริทึม H-Mine โดยใช้โครงสร้างข้อมูลแบบตารางส่วนหัว (Header Table) ในการเก็บข้อมูลไอเทมและความถี่ที่เกิดขึ้น เช่น 1:2 หมายถึงไอเทม 1 มีความถี่ 2 ครั้ง และมีลูกศรเชื่อมโยงแบบ

วิธีดำเนินการวิจัย

1. เครื่องมือวิจัย

- ใช้เครื่องคอมพิวเตอร์ Macbook Pro CPU 1.4GHz Quad-Core intel Core i5 RAM 8GB.

- ใช้ภาษาโปรแกรม Python ผ่าน Colab Jupyter Notebook บน Cloud ของ Google ใช้ชุดโปรแกรม Microsoft Office 365 (Microsoft Excel) เก็บรวบรวมและวิเคราะห์ข้อมูล

2. ข้อมูลการทดลอง

การทดลองนี้ใช้ชุดข้อมูลร้านขายของชำ (Groceries dataset) จาก Kaggle.com (Dedhia, 2022) ประกอบด้วยจำนวนข้อมูลรายการ (Transaction data) จำนวน 25,000 - 38,000 รายการ ดัง Table 1 และตัวอย่างข้อมูลดัง Table 2

Table 1 Details of Groceries Dataset Division

Number of Transactions (records)	Support values
25,000	0.2-1.0
28,000	0.2-1.0
30,000	0.2-1.0
35,000	0.2-1.0
38,000	0.2-1.0

จาก Table 1 แสดงรายละเอียดการแบ่งชุดข้อมูลร้านขายของชำออกเป็น 5 กลุ่ม ตามจำนวน Transaction โดยแต่ละกลุ่มจะกำหนดค่า support ขั้นต่ำที่ 0.2 ถูกเลือกเพื่อกรองรูปแบบความสัมพันธ์ที่มีความถี่ต่ำและอาจไม่มีนัยสำคัญออก ซึ่งช่วยลดเวลาในการประมวลผลและจำนวนกฎความสัมพันธ์ที่ไม่จำเป็น ในขณะที่ค่า support สูงสุดที่ 1.0 ถูกใช้เพื่อค้นหารูปแบบความสัมพันธ์ที่เกิดขึ้นบ่อยที่สุด ซึ่งสะท้อนถึงพฤติกรรมการซื้อที่เป็นที่นิยมของลูกค้า นอกจากนี้ การกระจายค่า support เป็น 0.2, 0.4, 0.6, 0.8 และ 1.0 ยังช่วยให้สามารถเปรียบเทียบประสิทธิภาพของแต่ละอัลกอริทึมภายใต้เงื่อนไขที่แตกต่างกันได้อย่างครอบคลุมและเป็นระบบ

Table 2 Sample of Groceries Dataset

Transaction	Items
T1	tropical fruit
T2	whole milk
T3	pip fruit
T4	citrus fruit, whole milk
T5	frankfurter, citrus fruit
T6	whole milk, dessert
T7	tropical fruit, sausage, pork
T8	pastry, citrus fruit, soda
T9	pot plants, dessert, root vegetables, sausage
T10	whole milk, chicken, butter, turkey

จาก Table 2 แสดงตัวอย่างข้อมูลที่ใช้ในการทดลอง โดยแต่ละรายการ (Transaction) ประกอบด้วย Transaction ID และรายการสินค้า (Items) ที่ลูกค้าซื้อในครั้งนั้น จะเห็นได้ว่าแต่ละ Transaction มีจำนวนสินค้าที่แตกต่างกัน เช่น T1 มีการซื้อเพียง tropical fruit อย่างเดียว ในขณะที่ T9 มีการซื้อถึง 4 รายการ ได้แก่ pot plants, dessert, root vegetables และ sausage นอกจากนี้ ยังพบรูปแบบที่น่าสนใจ เช่น citrus fruit มักถูกซื้อร่วมกับสินค้าอื่น (T4, T5, T8) และ whole milk ปรากฏในหลาย Transactions (T2, T4, T6, T10) ซึ่งความถี่ในการปรากฏของสินค้าเหล่านี้จะถูกนำไปเปรียบเทียบกับค่า Support values ที่กำหนดใน Table 1 (0.2-1.0) เพื่อค้นหารูปแบบไอเทมเซตที่เกิดขึ้นบ่อยด้วยอัลกอริทึมที่นำมาเปรียบเทียบต่อไป

จากลักษณะข้อมูลข้างต้น เมื่อพิจารณาในบริบทของการวิเคราะห์ข้อมูลขนาดใหญ่ พบว่าชุดข้อมูลที่ใช้ในการทดลองนี้มีคุณสมบัติของ Big Data (Gandomi & Haider, 2015; Han *et al*., 2006) หลายประการที่สำคัญ ได้แก่ Volume (ปริมาณ) ด้วยจำนวน Transactions ตั้งแต่ 25,000 ถึง 38,000 รายการ ซึ่งเพียงพอที่จะแสดงให้เห็นความแตกต่างของประสิทธิภาพในการประมวลผลของแต่ละอัลกอริทึม Variety (ความหลากหลาย) จากรายการสินค้าที่หลากหลายประเภท ซึ่งก่อให้เกิดรูปแบบความสัมพันธ์ที่ซับซ้อน Velocity (ความเร็ว) ที่สะท้อนผ่านการทดสอบที่ค่าสนับสนุนต่ำ (0.2) เพื่อจำลองสถานการณ์ที่ต้องการการประมวลผลอย่างรวดเร็วในการค้นหาความสัมพันธ์ทั้งหมด และ Veracity (ความถูกต้อง) เป็นการนำข้อมูล Transactions จากร้านขายของชำมาทดสอบความสามารถด้วยอัลกอริทึมในการจัดการกับข้อมูลที่มีความไม่แน่นอนและความซับซ้อนในสภาพแวดล้อมจริง

3. ขั้นตอนการวิจัย

งานวิจัยนี้ได้ใช้หลักขั้นตอนการวิจัยการทำเหมืองข้อมูลของ Han *et al*. (2006) ซึ่งประกอบด้วย 4 ขั้นตอนหลัก ดังนี้

3.1 ขั้นตอนการรวบรวมและวิเคราะห์ข้อมูล โดยงานวิจัยนี้ได้นำข้อมูลร้านขายของชำมาวิเคราะห์ กำจัดความซ้ำซ้อนและจัดกลุ่ม พร้อมบันทึกไฟล์เป็น .csv ก่อนทำการทดลอง

3.2 ขั้นตอนการสร้างแบบจำลอง ผู้วิจัยได้ทำการแบ่งชุดข้อมูลการทดลองออกเป็นหลายกลุ่มย่อยตามตารางที่ 1

3.3 ขั้นตอนการค้นหาไอเทมเซตที่เกิดขึ้นบ่อยด้วยอัลกอริทึม Apriori, FP-Growth, FP-Max และ H-Mine โดยงานวิจัยนี้ได้นำไลบรารี mlxtend.frequent_patterns เวอร์ชัน 0.23.4 (Raschka, 2018) มาใช้ในการทดลอง เนื่องจากเป็นไลบรารีที่รวบรวมอัลกอริทึมสำหรับการทำเหมืองกฎความสัมพันธ์ที่มีประสิทธิภาพและได้รับความนิยมในงานวิเคราะห์ตะกร้าสินค้า โดยในการทดลองได้กำหนดค่า min_support ตั้งแต่ 0.2 ถึง 1.0 ซึ่งครอบคลุมทั้งค่าที่ต่ำ (0.2) เพื่อค้นหารูปแบบที่เกิดขึ้นน้อยแต่อาจมีความสำคัญ ไปจนถึงค่าสูง (1.0) เพื่อค้นหารูปแบบที่พบบ่อยที่สุด ทั้งนี้เพื่อศึกษาผลกระทบต่อประสิทธิภาพการทำงานของแต่ละอัลกอริทึม ส่วนค่าพารามิเตอร์อื่นๆ ใช้ค่าเริ่มต้นของไลบรารี

3.4 ขั้นตอนการวัดผลและการประเมินผลลัพธ์ เพื่อทำการเปรียบเทียบความเร็ว (Execution time) หน่วยเป็นวินาที (Second) และการใช้หน่วยความจำ (Memory usage) หน่วยเป็นกิกะไบต์ (Gigabyte) โดยกำหนดค่าสนับสนุนที่ 0.2, 0.4, 0.6, 0.8 และ 1.0 แล้วนำแต่ละกลุ่มข้อมูลมาทดสอบจำนวน 10 รอบ เพื่อลดความเสี่ยงจากค่าผิดพลาดที่เกิดจากการทดลองไม่เพียงพอ (Borah & Nath, 2021) แสดงตัวอย่างการคำนวณ ดังนี้

ตัวอย่างการคำนวณหาความเร็ว เมื่อกำหนดค่าสนับสนุนเท่ากับ 0.2 ความเร็วแต่ละรอบเท่ากับ 5, 6, 5, 7, 6, 5, 6, 5, 7, 6 วินาที ดังนั้นค่าเฉลี่ยของความเร็วมีค่าเท่ากับ(5+6+5+7+6+5+6+5+7+6)/10 หรือ 5.8 วินาที

ตัวอย่างการคำนวณหาการใช้หน่วยความจำ เมื่อกำหนดค่าสนับสนุนเท่ากับ 0.2 ค่าการใช้หน่วยความจำแต่ละรอบเท่ากับ 1, 2, 1, 3, 2, 1, 2, 1, 3, 2 GB ดังนั้นค่าเฉลี่ยของการใช้หน่วยความจำ เท่ากับ(1+2+1+3+2+1+2+1+3+2)/10 หรือ 1.8 GB

ผลการทดลองและอภิปรายผล

การเปรียบเทียบประสิทธิภาพของอัลกอริทึมทั้ง 4 ประเภท ได้แก่ Apriori, FP-Growth, FP-Max และ H-Mine ด้วยชุดข้อมูลร้านขายของชำ ด้วยความเร็วและการใช้หน่วยความจำในการค้นหาไอเทมเซตที่เกิดขึ้นบ่อย ด้วยค่าสนับสนุนที่แตกต่างกัน แสดงผลได้ดังนี้

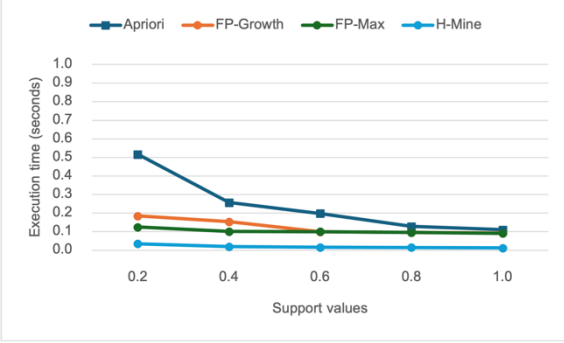

Figure 5 Comparison of execution time and support values for 25,000 transaction records

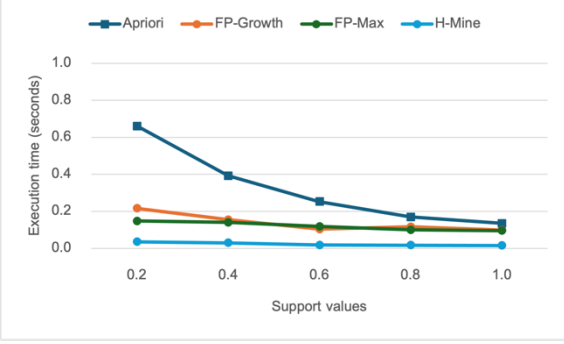

Figure 6 Comparison of execution time and support values for 28,000 transaction records

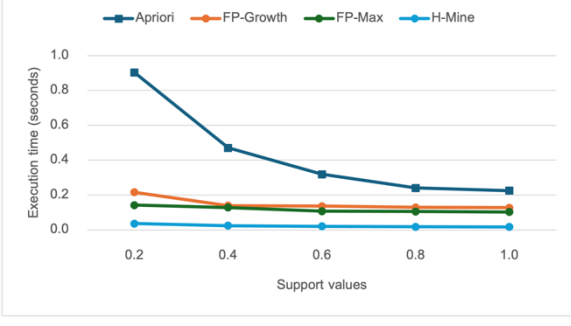

Figure 7 Comparison of execution time and support values for 30,000 transaction records

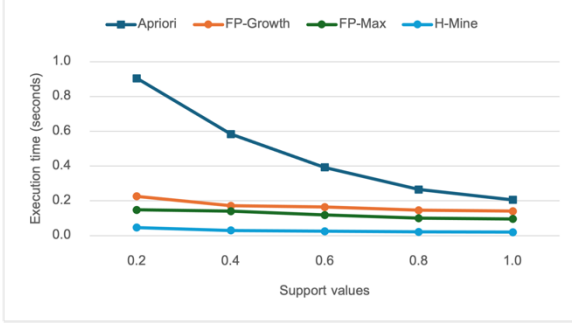

Figure 8 Comparison of execution time and support values for 35,000 transaction records

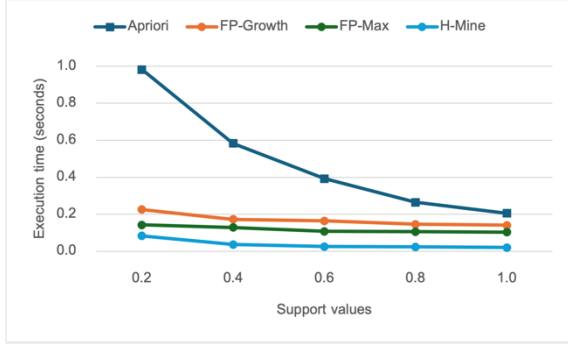

Figure 9 Comparison of execution time and support values for 38,000 transaction records

จาก Figure 5-9 แสดงการเปรียบเทียบประสิทธิภาพด้านเวลาการทำงานของอัลกอริทึม Apriori, FP-Growth, FP-Max และ H-mine โดยทดสอบกับชุดข้อมูล Transaction ขนาด 25,000 ถึง 38,000 รายการ และกำหนดค่าสนับสนุนตั้งแต่ 0.2 ถึง 1.0 ผลการทดลองพบว่า H-mine มีประสิทธิภาพดีที่สุดโดยใช้เวลาประมวลผลน้อยที่สุดและค่อนข้างคงที่แม้ขนาดข้อมูลจะเพิ่มขึ้น ในขณะที่อัลกอริทึม Apriori ใช้เวลามากที่สุด โดยเฉพาะที่ค่าสนับสนุนต่ำ และเวลาจะเพิ่มขึ้นอย่างชัดเจนเมื่อขนาดข้อมูลเพิ่มขึ้น ส่วนอัลกอริทึม FP-Growth และ FP-Max มีประสิทธิภาพใกล้เคียงกันและดีกว่า อัลกอริทึม Apriori แต่ยังใช้เวลามากกว่าอัลกอริทึม H-Mine อย่างไรก็ตาม เมื่อค่าสนับสนุนสูงขึ้น ประสิทธิภาพของทุกอัลกอริทึมจะใกล้เคียงกันมากขึ้น เนื่องจากมีจำนวนไอเทมเซตที่ต้องตรวจสอบน้อยลง (Borah & Nath, 2021; Garg & Kumar, 2013)

เมื่อพิจารณาที่ค่าสนับสนุนต่ำสุด (0.2) โดยเฉลี่ยจากทุกชุดข้อมูล (25,000-38,000 transactions) พบว่าอัลกอริทึม H-Mine มีความเร็วสูงกว่าอัลกอริทึมอื่นอย่างมีนัยสำคัญ โดยเร็วกว่าอัลกอริทึม Apriori ประมาณ 94% (คำนวณจากค่าเฉลี่ยของ H-Mine 0.047 วินาที เทียบกับ Apriori 0.794 วินาที), เร็วกว่า FP-Growth ประมาณ 78% (H-Mine 0.047 วินาที เทียบกับ 0.214 วินาที) และเร็วกว่า FP-Max ประมาณ

72% (H-Mine 0.047 วินาที เทียบกับ 0.167 วินาที) โดยคำนวณจากสูตร ((เวลาเฉลี่ยของอัลกอริทึมที่ช้ากว่า - เวลาเฉลี่ยของ H-Mine) / เวลาเฉลี่ยของอัลกอริทึมที่ช้ากว่า) × 100 ทั้งนี้เนื่องจากโครงสร้าง Hyperlink ที่มีประสิทธิภาพในการค้นหาและการจัดการหน่วยความจำ

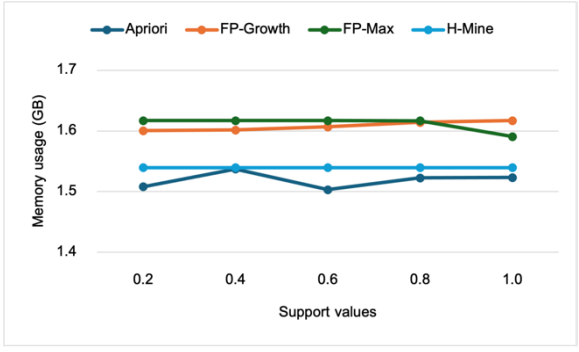

Figure 10 Comparison of memory usage and support values for 25,000 transaction records

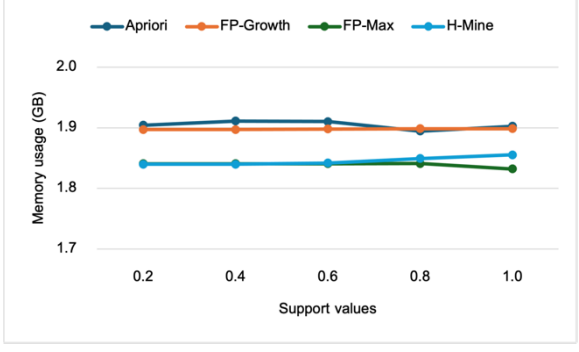

Figure 11 Comparison of memory usage and support values for 28,000 transaction records

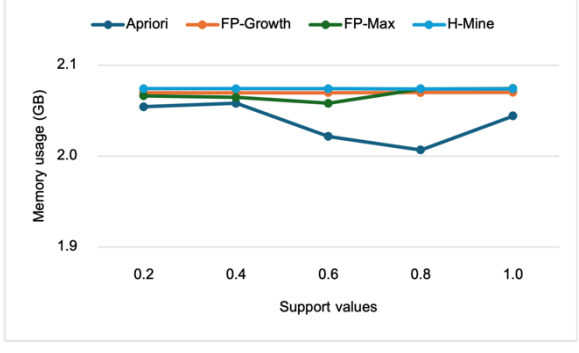

Figure 12 Comparison of memory usage and support values for 30,000 transaction records

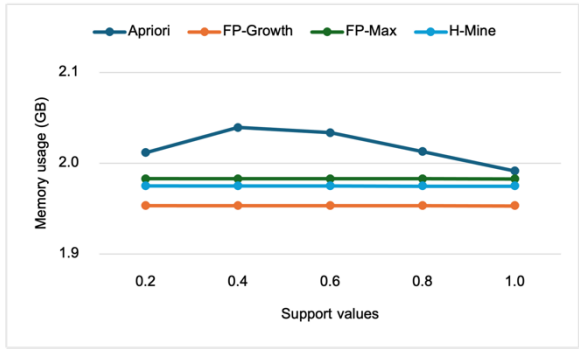

Figure 13 Comparison of memory usage and support values for 35,000 transaction records

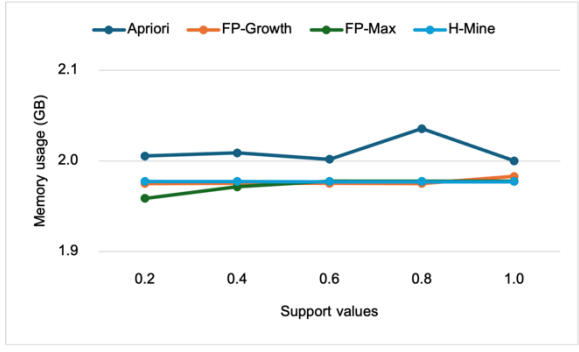

Figure 14 Comparison of memory usage and support values for 38,000 transaction records

จาก Figure 10-14 แสดงการเปรียบเทียบการใช้หน่วยความจำของอัลกอริทึม Apriori, FP-Growth, FP-Max และ H-mine บนชุดข้อมูล Transaction ขนาด 25,000 ถึง 38,000 รายการ โดยทดสอบที่ค่าสนับสนุน ตั้งแต่ 0.2 ถึง 1.0 พบว่าปริมาณการใช้หน่วยความจำมีความแตกต่างกันในแต่ละอัลกอริทึม โดยอัลกอริทึม H-mine และ FP-Max มีแนวโน้มใช้หน่วยความจำคงที่และน้อยกว่าอัลกอริทึมอื่น ในขณะที่อัลกอริทึม Apriori มีการใช้หน่วยความจำผันผวนและมีแนวโน้มเพิ่มขึ้นเมื่อขนาดข้อมูลเพิ่มขึ้น เนื่องจากต้องสร้างและเก็บ Candidate itemsets จำนวนมาก ส่วนอัลกอริทึม FP-Growth ใช้หน่วยความจำค่อนข้างคงที่แต่มากกว่าอัลกอริทึม H-Mine และ FP-Max เล็กน้อย เนื่องจากต้องเก็บโครงสร้าง FP-Tree ทั้งหมดไว้ในหน่วยความจำ (Borah & Nath, 2021; Wicaksono *et al*., 2020)

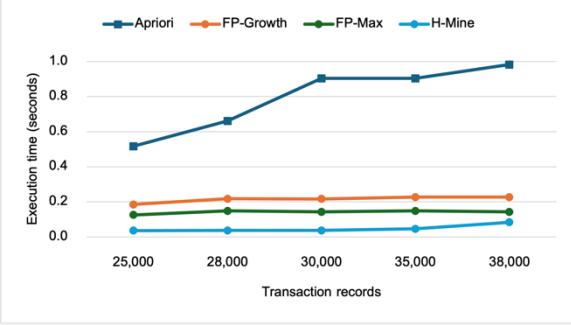

Figure 15 Comparison of execution time among association rule mining algorithms with different transaction records at support value = 0.2

จาก Figure 15 แสดงการเปรียบเทียบด้านเวลาประมวลผลกับจำนวนข้อมูล Transaction ของอัลกอริทึมทั้ง 4 ในการประมวลผลข้อมูลขนาดใหญ่ที่ค่า support เท่ากับ 0.2 พบว่าอัลกอริทึม H-Mine มีประสิทธิภาพดีที่สุด แม้ว่าจะมีแนวโน้มเพิ่มขึ้นเล็กน้อยเมื่อข้อมูลมีขนาดใหญ่ขึ้น ในทางตรงกันข้ามอัลกอริทึม Apriori แสดงให้เห็นข้อจำกัดในการรับมือกับข้อมูลขนาดใหญ่อย่างชัดเจน โดยเวลาประมวลผลเพิ่มขึ้นอย่างมีนัยสำคัญ สำหรับอัลกอริทึม FP-Growth และ FP-Max มีประสิทธิภาพปานกลางและรักษาเสถียรภาพได้ดี

จาก Figure 16 แสดงการเปรียบเทียบด้านการใช้หน่วยความจำกับจำนวนข้อมูล Transaction เผยให้เห็นพฤติกรรมที่น่าสนใจ โดยทุกอัลกอริทึมมีจุดเปลี่ยนสำคัญที่ 30,000 transactions ซึ่งเป็นจุดที่ใช้หน่วยความจำสูงสุด (ประมาณ 2.05-2.07 GB) จากนั้นการใช้หน่วยความจำมีแนวโน้มลดลงเล็กน้อย สังเกตได้ว่าอัลกอริทึม H-Mine แม้จะเริ่มต้นด้วยการใช้หน่วยความจำน้อยที่สุด แต่เมื่อข้อมูลมีขนาดใหญ่ขึ้น การใช้หน่วยความจำของทุกอัลกอริทึมมีค่าใกล้เคียงกัน

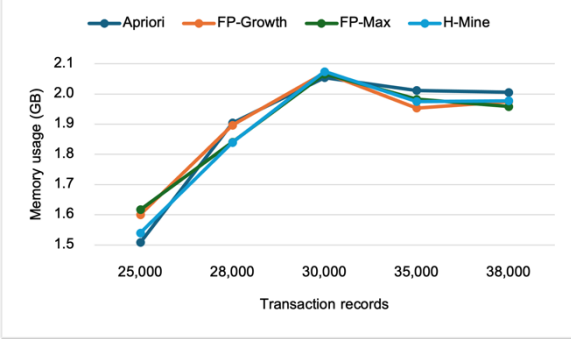

Figure 16 Comparison of memory usage among association rule mining algorithms with different transaction records at support value = 0.2

จากผลการวิเคราะห์ประสิทธิภาพของทั้ง 4 อัลกอริทึม พบว่าแต่ละอัลกอริทึมมีความเหมาะสมกับลักษณะข้อมูลที่แตกต่างกัน โดยอัลกอริทึม H-Mine แสดงประสิทธิภาพดีที่สุดสำหรับข้อมูลขนาดใหญ่ (25,000-38,000 รายการ) และข้อมูลที่มีค่าสนับสนุนต่ำ (0.2-0.4) เนื่องจากใช้โครงสร้าง Hyperlink ที่มีประสิทธิภาพในการค้นหา ทำให้เหมาะกับการประมวลผลแบบ real-time ส่วนอัลกอริทึม FP-Growth และ FP-Max เหมาะกับข้อมูลขนาดกลาง (25,000-30,000 รายการ) และข้อมูลที่มีรูปแบบการเกิดซ้ำสูง เนื่องจากใช้โครงสร้าง FP-Tree ในการบีบอัดข้อมูล โดย FP-Max จะมีประสิทธิภาพดีในกรณีที่ต้องการเฉพาะ maximal frequent itemsets ในขณะที่อัลกอริทึม Apriori เหมาะกับข้อมูลขนาดเล็ก (น้อยกว่า 25,000 รายการ) ที่มีค่าสนับสนุนสูง (0.8-1.0) และมีรูปแบบซ้ำน้อย เนื่องจากการสแกนฐานข้อมูลหลายรอบจะส่งผลกระทบน้อยกว่าในกรณีนี้

สรุปผลการทดลองและข้อเสนอแนะ

จากการวิจัยเปรียบเทียบประสิทธิภาพของอัลกอริทึม Apriori, FP-Growth, FP-Max และ H-mine ในการค้นหาไอเทมเซตที่เกิดขึ้นบ่อยบนชุดข้อมูล Transaction ขนาด 25,000 ถึง 38,000 รายการ พบว่าอัลกอริทึม H-Mine มีประสิทธิภาพดีที่สุดในการจัดการกับข้อมูลขนาดใหญ่ โดยเฉพาะเมื่อต้องรับมือกับความท้าทายของ Big Data ทั้งด้านปริมาณข้อมูล (Volume) และความเร็วในการประมวลผล (Velocity)

ตามด้วยอัลกอริทึม FP-Growth และ FP-Max ที่ใช้โครงสร้าง FP-Tree ช่วยลดการสแกนฐานข้อมูล ในขณะที่อัลกอริทึม Apriori มีประสิทธิภาพต่ำที่สุด โดยเฉพาะเมื่อค่าสนับสนุนต่ำและข้อมูลขนาดใหญ่ เนื่องจากต้องสแกนฐานข้อมูลหลายรอบและสร้าง Candidate itemsets จำนวนมาก นอกจากนี้ยังพบว่าจำนวน Transaction ที่เพิ่มขึ้นส่งผลให้เวลาและหน่วยความจำที่ใช้เพิ่มขึ้น โดยเฉพาะกับอัลกอริทึม Apriori ในขณะที่ค่าสนับสนุนที่สูงขึ้นช่วยลดเวลาและหน่วยความจำที่ใช้ เนื่องจากมีไอเทมเซตที่เกิดขึ้นบ่อยน้อยลง

ผลการวิเคราะห์ยังแสดงให้เห็นว่าการเลือกใช้อัลกอริทึมควรพิจารณาจากลักษณะข้อมูลเป็นสำคัญ โดยข้อมูลขนาดใหญ่และต้องการประมวลผลรวดเร็วควรใช้ H-Mine ข้อมูลที่มีรูปแบบซ้ำสูงควรใช้ FP-Growth หรือ FP-Max และข้อมูลขนาดเล็กที่มีค่าสนับสนุนสูงสามารถใช้ Apriori ได้อย่างมีประสิทธิภาพ การพิจารณาลักษณะข้อมูลก่อนเลือกใช้อัลกอริทึมจะช่วยให้การวิเคราะห์มีประสิทธิภาพสูงสุด

อย่างไรก็ตาม งานวิจัยนี้มีข้อจำกัดที่ควรพิจารณา ได้แก่ (1) การทดลองใช้เพียงชุดข้อมูลร้านขายของชำเพียงชุดเดียว ซึ่งอาจไม่ครอบคลุมลักษณะข้อมูลที่หลากหลายในธุรกิจค้าปลีกประเภทอื่น (2) การทดสอบจำกัดเฉพาะขนาดข้อมูลไม่เกิน 38,000 รายการ ซึ่งอาจไม่สะท้อนประสิทธิภาพการทำงานกับข้อมูลขนาดใหญ่มาก และ (3) การทดสอบอยู่ภายใต้สภาพแวดล้อมฮาร์ดแวร์และซอฟต์แวร์ที่กำหนด ซึ่งประสิทธิภาพอาจแตกต่างในสภาพแวดล้อมอื่น

สำหรับข้อเสนอแนะในการประยุกต์ใช้งาน ควรพิจารณาเลือกใช้อัลกอริทึมให้เหมาะสมตามข้อจำกัดด้านทรัพยากรและลักษณะการใช้งาน ดังนี้

กรณีระบบมีข้อจำกัดด้านทรัพยากร อัลกอริทึม H-Mine เหมาะสำหรับระบบที่มีหน่วยความจำจำกัดแต่ต้องการประสิทธิภาพสูง เช่น ระบบฝังตัว หรือแอปพลิเคชันบนอุปกรณ์พกพา เนื่องจากใช้หน่วยความจำน้อยที่สุดและมีความเร็วสูงสุด ส่วนอัลกอริทึม FP-Max เป็นทางเลือกที่ดีสำหรับระบบที่มีข้อจำกัดด้านพื้นที่จัดเก็บ เนื่องจากเก็บเฉพาะไอเทมเซตที่มีความสำคัญสูงสุด

กรณีระบบมีทรัพยากรเพียงพอ อัลกอริทึม FP-Growth เหมาะกับระบบที่มีหน่วยความจำเพียงพอและต้องการความเสถียรในการทำงาน เช่น ระบบวิเคราะห์ข้อมูลขององค์กรขนาดใหญ่ ส่วนอัลกอริทึม Apriori ควรใช้เฉพาะกับข้อมูลขนาดเล็กหรือในกรณีที่ต้องการความเข้าใจง่ายในการพัฒนาและบำรุงรักษาระบบ

สำหรับการวิจัยในอนาคตควรศึกษาเพิ่มเติมเกี่ยวกับการพัฒนาเทคนิคการจัดการหน่วยความจำแบบไดนามิกเพื่อรองรับการทำงานในสภาพแวดล้อมที่มีทรัพยากรจำกัด และการศึกษาผลกระทบของปัจจัยอื่นๆ เช่น ความซับซ้อนของข้อมูล รูปแบบการกระจายตัวของข้อมูล และประสิทธิภาพการทำงานบนฮาร์ดแวร์ที่แตกต่างกัน

เอกสารอ้างอิง

Adeniji, I. A., Saheed, Y. K., Oladele, T. O., & Braimah, J. O. (2015). Comparative analysis of association rule mining techniques for monitoring behavioural patterns of customers in a grocery store. *International Journal of Computer Science and Information Security, 13*(1), 46–51.

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

Borah, A., & Nath, B. (2021). Comparative evaluation of pattern mining techniques: An empirical study. *Complex & Intelligent Systems, 7*(2), 589–619. https://doi.org/10.1007/s40747-020-00226-4

Dedhia, H. (2022). *Groceries dataset* [Dataset]. Kaggle. https://www.kaggle.com/datasets/heeraldedhia/groceries-dataset/data

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management, 35*(2), 137–144. https://doi.org/10.1016/j.ijinfomgt.2014.10.007

Garg, K., & Kumar, D. (2013). Comparing the performance of frequent pattern mining algorithms. *International Journal of Computer Applications, 69*(25), 21–28. https://doi.org/10.5120/12129-8502

Grahne, G., & Zhu, J. (2003). High performance mining of maximal frequent itemsets. In *Proceedings of the 3rd SIAM International Conference on Data Mining* (pp. 135–143). SIAM.

Han, J., Kamber, M., & Pei, J. (2006). *Data mining: Concepts and techniques* (2nd ed.). Morgan Kaufmann.

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data* (pp. 1–12). ACM. https://doi.org/10.1145/342009.335372

Mustakim, Herianda, D. M., Ilham, A., Daeng Gs, A., Laumal, F. E., Kurniasih, N., Iskandar, A., Manulangga, G., Indra Iswara, I. B. A., & Rahim, R. (2018). Market basket analysis using Apriori and FP-Growth for analysis consumer expenditure patterns at Berkah Mart in Pekanbaru Riau. *Journal of Physics: Conference Series, 1114*, 012131. https://doi.org/10.1088/1742-6596/1114/1/012131

Nigam, B., Nigam, A., & Dalal, P. (2017). Comparative study of top 10 algorithms for association rule mining. *International Journal of Computer Sciences and Engineering, 5*(8), 190–195. https://doi.org/10.26438/ijcse/v5i8.190195

Pei, J., Han, J., Lu, H., Nishio, S., Tang, S., & Yang, D. (2001). H-mine: Hyper-structure mining of frequent patterns in large databases. In *Proceedings 2001*

IEEE International Conference on Data Mining (pp. 441–448). IEEE. https://doi.org/10.1109/ICDM.2001.989550

Raschka, S. (2018). MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack. *Journal of Open Source Software, 3*(24), 638. https://doi.org/10.21105/joss.00638

Slimani, T., & Lazzez, A. (2014). Efficient analysis of pattern and association rule mining approaches. *International Journal of Information Technology and Computer Science, 6*(3), 70–81. https://doi.org/10.5815/ijitcs.2014.03.09

Wicaksono, D., Jambak, M. I., & Saputra, D. M. (2020). The comparison of Apriori algorithm with preprocessing and FP-Growth algorithm fWor finding frequent data pattern in association rule. In *Proceedings of the Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019)* (pp. 574–579). Atlantis Press.