

การเปรียบเทียบประสิทธิภาพอัลกอริทึม Apriori และ FP-Growth

ด้วยชุดข้อมูลร้านขายของชำ

A Comparison of Apriori and FP-Growth Algorithms with Groceries dataset

กริชบดินทร์ ผิวหอม

Kritbodin Phiwhorm

สาขาวิชาวิทยาการคอมพิวเตอร์ คณะศิลปศาสตร์และวิทยาศาสตร์ มหาวิทยาลัยราชภัฏศรีสะเกษ

Department of Computer Science, Faculty of Liberal Arts and Sciences, Sisaket Rajabhat University

E-Mail: kritbodin.p@sskru.ac.th

(Received: March 11, 2024; Revised: May 22, 2024; Accepted: June 05, 2024)

บทคัดย่อ

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อ 1) ศึกษาและเปรียบเทียบประสิทธิภาพของอัลกอริทึม Apriori และ FP-Growth ในการค้นหาไอเทมเซตที่เกิดขึ้นบ่อย (Frequent itemsets) ซึ่งเป็นขั้นตอนสำคัญในกระบวนการทำเหมืองข้อมูลกฎความสัมพันธ์ (Association rules mining) และ 2) เพื่อเปรียบเทียบความเร็วและการใช้หน่วยความจำในการค้นหาไอเทมเซตที่เกิดขึ้นบ่อย

ผลการวิจัยพบว่า 1) แต่ละอัลกอริทึมที่ใช้ในการค้นหาไอเทมเซตที่เกิดขึ้นบ่อยมีข้อดีและข้อเสียที่ต่างกัน สำหรับชุดข้อมูลและค่าสนับสนุนที่แตกต่างกัน และ 2) การเปรียบเทียบความเร็วและการใช้หน่วยความจำชี้ให้เห็นว่าอัลกอริทึม FP-Growth ทำงานได้เร็วกว่าอัลกอริทึม Apriori และใช้หน่วยความจำน้อยกว่า เนื่องจากอัลกอริทึม Apriori ต้องสแกนฐานข้อมูลหลายรอบ และ มีการสร้างชุดข้อมูลตัวแทน (Candidate itemsets) ทำให้ใช้เวลานาน ส่วนอัลกอริทึม FP-Growth เข้าสแกนฐานข้อมูลแค่ 2 รอบและไม่มีการสร้างชุดตัวแทน ทำให้ทำงานได้ดีกว่า

คำสำคัญ: อัลกอริทึมเอพริออริ, อัลกอริทึมเอฟพี-โกรธ, เหมืองข้อมูล, ไอเทมเซตที่เกิดขึ้นบ่อย

ABSTRACT

This research aims to 1) study and compare the performance of the Apriori and FP-Growth algorithms in finding frequent itemsets, which is a crucial step in the process of association rules mining, and 2) compare the speed and memory usage of the algorithms in finding frequent itemsets.

The research findings indicate that 1) each algorithm used for discovering frequent itemsets has different advantages and disadvantages depending on there dataset and support threshold used. And 2) experimental comparison of algorithm speeds and memory usage in finding frequent itemsets reveals that the FP-Growth algorithm performs faster than the Apriori algorithm and utilizes less memory. This is because the Apriori algorithm requires multiple scans of the database and generates candidate itemsets, leading to longer processing times. On the other hand, the FP-Growth

algorithm scans the database only twice and does not generate candidate itemsets, resulting in faster execution.

Keywords: Apriori algorithm, FP-Growth algorithm, Data Mining, Frequent itemsets.

บทนำ

การทำเหมืองข้อมูลด้วยการสร้างกฎความสัมพันธ์ช่วยค้นหาความสัมพันธ์ระหว่างข้อมูลในฐานข้อมูล โดยกฎความสัมพันธ์เชื่อมโยงข้อมูลที่ถูกรวบรวมอยู่ ยกตัวอย่างเช่น การวิเคราะห์ตะกร้าสินค้า (Market Basket Analysis) เป็นเทคนิคที่ร้านค้าปลีกใช้ค้นหาความสัมพันธ์ระหว่างสินค้า โดยวิเคราะห์สินค้าที่ลูกค้าซื้อพร้อมกันบ่อย ซึ่งช่วยให้ร้านค้าเข้าใจพฤติกรรมซื้อของลูกค้าและค้นหาความสัมพันธ์ระหว่างสินค้า เช่น ลูกค้าที่ซื้อนมมักจะซื้อขนมปังด้วย ข้อมูลนี้ช่วยให้ร้านค้าจัดวางสินค้าใกล้เคียงหรือจัดโปรโมชั่นดึงดูดลูกค้าได้ [1]

การหาความสัมพันธ์ (Association rules) ประกอบด้วย 2 ขั้นตอนหลัก [2] คือ 1) การค้นหาไอเทมเซตที่เกิดบ่อย และ 2) การสร้างกฎความสัมพันธ์ โดยต้องค้นหาไอเทมเซตที่เกิดขึ้นบ่อยอย่างมีประสิทธิภาพ อัลกอริทึมที่นิยมใช้คือ อัลกอริทึม Apriori [3] และอัลกอริทึม FP-Growth [4]

จากงานวิจัยของ Gulzar et al. [5] พบว่าอัลกอริทึม Apriori ใช้เวลาและหน่วยความจำมากเมื่อใช้กับข้อมูลขนาดใหญ่และซับซ้อน ขณะที่งานวิจัยของ Aldino et al. [6] และ Wicaksono et al. [7] พบว่าอัลกอริทึม FP-Growth มีประสิทธิภาพดีกว่า ใช้เวลาน้อยกว่า ต้องการหน่วยความจำน้อยกว่า และจัดการข้อมูลซับซ้อนได้ดีกว่า

ดังนั้น งานวิจัยนี้มุ่งเน้นการเปรียบเทียบประสิทธิภาพของอัลกอริทึม Apriori และ FP-Growth ในการค้นหาไอเทมเซตที่เกิดขึ้นบ่อยในชุดข้อมูลร้านขายของชำ โดยพิจารณาประสิทธิภาพด้านความเร็วและการใช้หน่วยความจำในการประมวลผลข้อมูลรายวัน (Transaction) พร้อมกำหนดค่าสนับสนุน (Support) ที่แตกต่างกัน

1. วัตถุประสงค์การวิจัย

1. เพื่อเปรียบเทียบประสิทธิภาพระหว่างอัลกอริทึม Apriori และ FP-Growth ทำให้เห็นถึงประสิทธิภาพของแต่ละอัลกอริทึมในการค้นหาไอเทมเซตที่เกิดขึ้นบ่อย
2. เพื่อเปรียบเทียบความเร็วและการใช้หน่วยความจำของอัลกอริทึมในการค้นหาไอเทมเซตที่เกิดขึ้นบ่อย โดยกำหนดค่า Support และจำนวนข้อมูล Transaction ที่แตกต่างกัน

2. เอกสารและงานวิจัยที่เกี่ยวข้อง

เทคนิคการสร้างกฎความสัมพันธ์มุ่งค้นหากฎและรูปแบบไอเทมเซตที่เกิดบ่อยในฐานข้อมูล มีหลายอัลกอริทึมที่นิยมใช้ เช่น อัลกอริทึม Apriori [3] และอัลกอริทึม FP-Growth [4]

อัลกอริทึม Apriori ใช้วิธีวนตรวจสอบฐานข้อมูลหลายรอบ โดยสร้างชุดตัวแทนที่ระดับเพื่อค้นหาไอเทมเซตที่เกิดบ่อยและเปรียบเทียบกับค่าสนับสนุนขั้นต่ำ ชุดไอเทมเซตที่มีค่า Support เท่ากับหรือมากกว่าจะถือว่า

เกิดบ่อย ส่วนที่ต่ำกว่าจะถูกตัดทิ้ง จากนั้นจะสร้างชุดตัวแทนใหม่จากไอเทมเซตที่เกิดบ่อย ข้อดีคือเข้าใจง่าย แต่ทำงานช้ากับฐานข้อมูลขนาดใหญ่เพราะต้องตรวจสอบหลายรอบ [3]

เทคนิคอัลกอริทึม FP-Growth ใช้หลักการสร้างต้นไม้ (FP-tree) และการแบ่งแยกและพิชิต (Divide-and-conquer) เพื่อประมวลผลข้อมูลอย่างรวดเร็ว [8] ในขั้นแรกจะสร้างรากต้นไม้เป็นค่าว่าง (null) แล้วจัดเรียงข้อมูลโดยแทนค่าตัวแปรด้วยโหนด ซึ่งเทคนิคนี้ไม่สร้างชุดข้อมูล Candidate itemsets ทำให้ได้กฎความสัมพันธ์เร็ว แต่ข้อมูลต้องอยู่ในรูปแบบไบนารี [9]

จากงานวิจัยของ Gulzar et al. [5] พบว่าอัลกอริทึม Apriori สามารถค้นพบรูปแบบการเกิดโรคตาในวัยรุ่นได้ดี แต่ใช้เวลาและหน่วยความจำมากเมื่อใช้กับข้อมูลขนาดใหญ่และซับซ้อน ในขณะที่งานวิจัยของ Aldino et al. [6] และ Wicaksono et al. [7] พบว่าอัลกอริทึม FP-Growth มีประสิทธิภาพดีกว่า Apriori ในการวิเคราะห์ข้อมูลจำนวนมาก เนื่องจากใช้เวลาน้อยกว่าและต้องการหน่วยความจำน้อยกว่า นอกจากนี้ FP-Growth ยังสามารถจัดการข้อมูลที่ซับซ้อนได้ดีกว่า

วิธีดำเนินการวิจัย

1. เครื่องมือการวิจัย

- 1.1 ใช้เครื่องคอมพิวเตอร์ Macbook Pro CPU 1.4GHz Quad-Core intel Core i5 RAM 8GB.
- 1.2 ใช้ภาษาโปรแกรม Python ผ่าน Colab Jupyter Notebook บน Cloud ของ Google
- 1.3 ใช้ชุดโปรแกรม Microsoft Office 365 (Microsoft Excel) เก็บรวบรวมและวิเคราะห์ข้อมูล

2. กลุ่มเป้าหมาย การทดลองนี้ใช้ชุดข้อมูลร้านขายของชำ (Groceries dataset) จาก Kaggle.com [10] ประกอบด้วยจำนวนรายการสั่งซื้อสินค้า (Transaction data) 38,765 รายการ โดยแบ่งกลุ่มเป้าหมายข้อมูลออกเป็น 5 กลุ่ม จากข้อมูล 25,000 - 38,000 รายการ ดังตารางที่ 1 เพื่อทดสอบประสิทธิภาพของอัลกอริทึม Apriori และ FP-Growth เปรียบเทียบความเร็ว (Execution time) และการใช้หน่วยความจำ (Memory usage) ในสถานะที่จำนวนรายการต่างกัน ทำให้ผลการทดสอบมีความน่าเชื่อถือและสามารถนำไปใช้ปรับปรุงหรือเลือกใช้อัลกอริทึมในการทำงานได้ดี

ตารางที่ 1 รายละเอียดการแบ่งชุดข้อมูลร้านขายของชำ

จำนวน Transaction (รายการ)	ค่า Support
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

3. ขั้นตอนการดำเนินการวิจัย ในงานวิจัยนี้ได้นำเอาหลักของการทำเหมืองข้อมูลของ Han et al. [11] ซึ่งประกอบด้วย 4 ขั้นตอน ดังนี้

3.1 ขั้นตอนการรวบรวมและวิเคราะห์ข้อมูล งานวิจัยนี้ได้นำข้อมูลร้านขายของชำ มาวิเคราะห์ กำจัดความซ้ำซ้อนและจัดกลุ่ม พร้อมบันทึกไฟล์เป็น .csv ก่อนนำเข้าทดลอง

3.2 ขั้นตอนการสร้างแบบจำลอง ผู้วิจัยได้ทำการแบ่งชุดข้อมูลทดลองออกเป็นกลุ่มย่อย แต่ละกลุ่มมีจำนวนข้อมูลตามตารางที่ 1

3.3 ขั้นตอนการค้นหาไอเทมเซตที่เกิดขึ้นบ่อย ด้วยเทคนิคอัลกอริทึม Apriori และ FP-Growth

3.4 ขั้นตอนการวัดประสิทธิภาพ แบ่งการทำงานได้ดังนี้

1) กำหนดค่า Support ที่ 0.2, 0.4, 0.6, 0.8, และ 1.0 เพื่อทดสอบอัลกอริทึมในสถานะต่างๆ ให้เห็นประสิทธิภาพและข้อจำกัดในแต่ละสถานการณ์ โดยค่า Support ต่ำจะเก็บไอเทมเซตมากขึ้น ใช้หน่วยความจำมากขึ้น ส่วนค่า Support สูงจะลดจำนวนไอเทมเซต ทำให้เห็นผลกระทบต่อความเร็วและหน่วยความจำของอัลกอริทึม

2) การหาค่าเฉลี่ยความเร็วและการใช้หน่วยความจำของกลุ่มข้อมูล ด้วยการทดสอบกลุ่มละ 10 รอบ เพื่อให้ผลลัพธ์มีความแม่นยำและเสถียร ลดความเสี่ยงจากค่าผิดพลาดที่เกิดจากการทดลองไม่เพียงพอ แสดงตัวอย่างการคำนวณได้ดังนี้ (1) หากค่าเวลาที่ใช้ในการทดลอง 10 รอบของค่า Support เท่ากับ 0.2 ได้ค่า 5, 6, 5, 7, 6, 5, 6, 5, 7, 6 วินาที ดังนั้นค่าเฉลี่ยของความเร็ว เท่ากับ $(5+6+5+7+6+5+6+5+7+6)/10$ หรือ 5.8 วินาที (2) หากค่าการใช้หน่วยความจำในการทดลอง 10 รอบของค่า Support เท่ากับ 0.2 ได้ค่า 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

4. สถิติที่ใช้ในการวิจัย แสดงข้อมูลการเปรียบเทียบ (1) ความเร็วหน่วยเป็นวินาที (Second) และ (2) การใช้หน่วยความจำหน่วยเป็นกิกะไบต์ (Gigabyte)

ผลการวิจัย

1. ผลการพัฒนาระบบ งานวิจัยนี้ได้ทำการทดลองด้วยเทคนิคกฎความสัมพันธ์ (Association rules) โดยใช้ อัลกอริทึม Apriori และ FP-Growth สามารถแสดงตัวอย่างไอเทมเซตที่เกิดขึ้นบ่อย ได้ดังนี้

ตารางที่ 2 แสดงตัวอย่างไอเทมเซตที่เกิดขึ้นบ่อยด้วยอัลกอริทึม Apriori

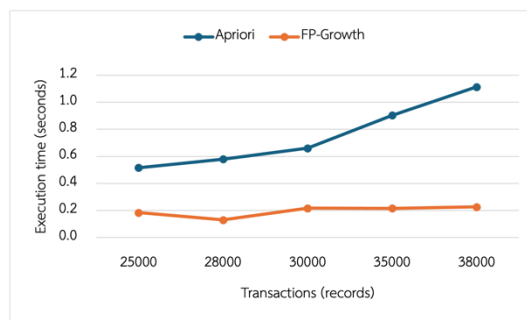
นับความถี่ของไอเทมเซตทั้งหมด	ไอเทมเซตที่เกิดขึ้นบ่อย (กรณี Support = 3)
{นม} (Milk) ความถี่ = 6	{นม} (Milk) ความถี่ = 6
{ขนมปัง} (Bread) ความถี่ = 5	{ขนมปัง} (Bread) ความถี่ = 5
{เนย} (Butter) ความถี่ = 2 (ตัดออกกรณี support=3)	{นม, ขนมปัง} (Milk, Bread) ความถี่ = 4
{นม, ขนมปัง} (Milk, Bread) ความถี่ = 4	

ตารางที่ 3 แสดงตัวอย่างไอเทมเซตที่เกิดบ่อยด้วยอัลกอริทึม FP-Growth

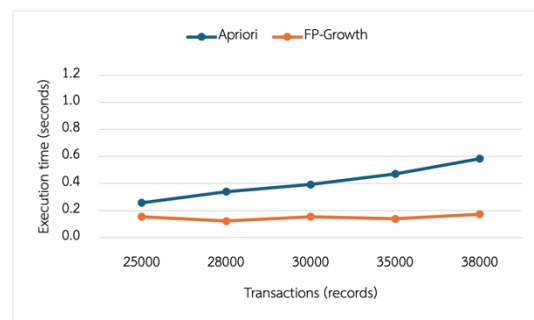
สร้าง FP-Tree หลังตัดไอเทมเซตที่น้อยกว่า Support = 3	ไอเทมเซตที่เกิดบ่อย (กรณี Support = 3)
โหนดราก (Root) ไม่มีไอเทม (ความถี่ = 0)	{นม} (Milk) ความถี่ = 6
{นม} (Milk) ความถี่ = 6	{ขนมปัง} (Bread) ความถี่ = 5
{ขนมปัง} (Bread) ความถี่ = 4 (ได้โหนด milk)	{นม, ขนมปัง} (Milk, Bread) ความถี่ = 4
{ขนมปัง} (Bread) ความถี่ = 5	

จากตารางที่ 2 แสดงการหาไอเทมเซตที่เกิดบ่อยด้วยอัลกอริทึม Apriori เปรียบเทียบกับค่าสนับสนุนขั้นต่ำ ชุดไอเทมเซตที่มีค่า Support เท่ากับหรือมากกว่าจะถูกนับเป็นไอเทมเซตที่เกิดบ่อย ส่วนที่ต่ำกว่าจะถูกตัดออก จากตารางที่ 3 แสดงการหาไอเทมเซตที่เกิดบ่อยด้วยอัลกอริทึม FP-Growth ด้วยโครงสร้างต้นไม้ FP-Tree เริ่มจากไอเทมที่มีความถี่สูงสุดและตรวจสอบความถี่ของการรวมไอเทมต่าง ๆ

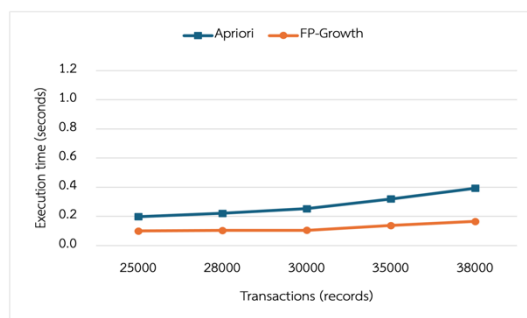
2. ผลการทดลอง การเปรียบเทียบประสิทธิภาพอัลกอริทึม Apriori และ FP-Growth ด้วยชุดข้อมูลร้านขายของชำ แสดงข้อมูลด้วยกราฟความเร็วและหน่วยความจำที่ใช้ในการค้นหาไอเทมเซตที่เกิดบ่อย ได้ดังนี้



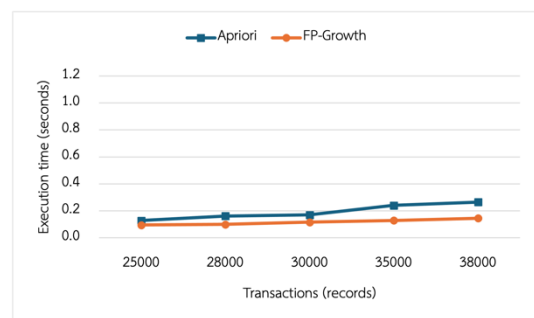
a. Support = 0.2



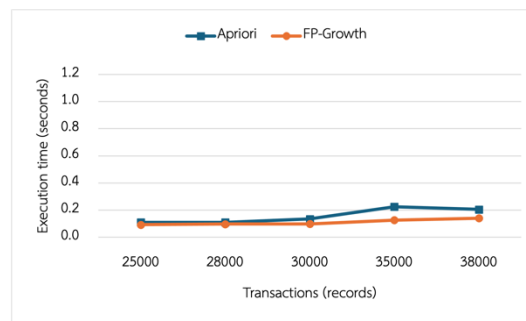
b. Support = 0.4



c. Support = 0.6



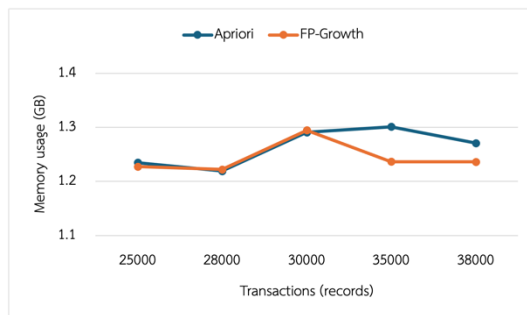
d. Support = 0.8



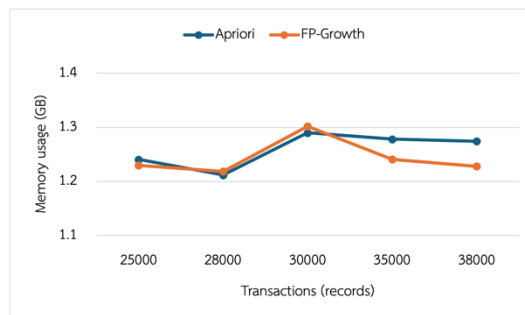
e. Support = 1.0

ภาพที่ 1 แสดงการเปรียบเทียบเวลาความเร็วที่ค้นหาไอเทมเซตที่เกิดขึ้นบ่อยระหว่างอัลกอริทึม Apriori และอัลกอริทึม FP-Growth สำหรับการเพิ่มขึ้นของ Transaction

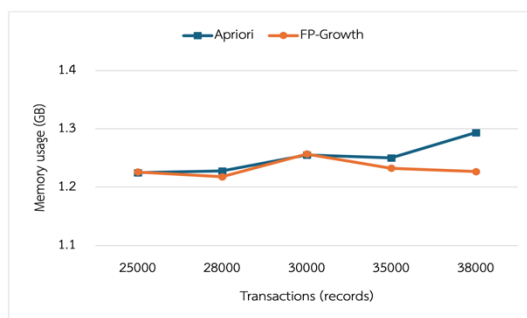
จากภาพที่ 1 กำหนดค่า Support ระหว่าง 0.2 – 1.0 (ภาพที่ 1.a – 1.e) พบว่ากรณีข้อมูล Transaction เพิ่มขึ้น จะทำให้เวลาความเร็วการค้นหามีเพิ่มขึ้น และอัลกอริทึม FP-Growth จะทำงานได้เร็วกว่า Apriori โดยเวลาการทำงานของทั้งสองอัลกอริทึมจะขยับเข้ามาใกล้กันเมื่อกำหนดค่า Support ที่มากขึ้น ซึ่งสอดคล้องกับงานวิจัยของ Han et al. [11] และ Patil et al. [12]



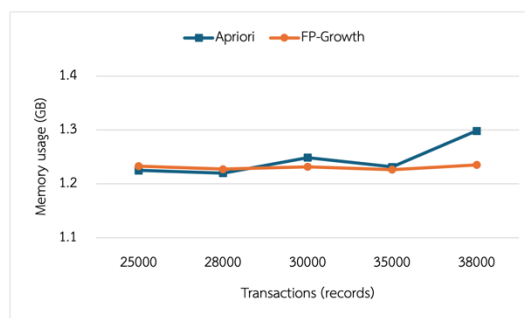
a. Support = 0.2



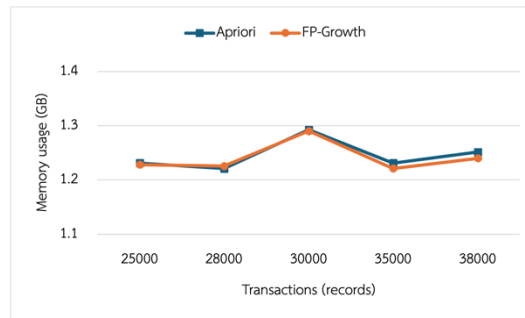
b. Support = 0.4



c. Support = 0.6



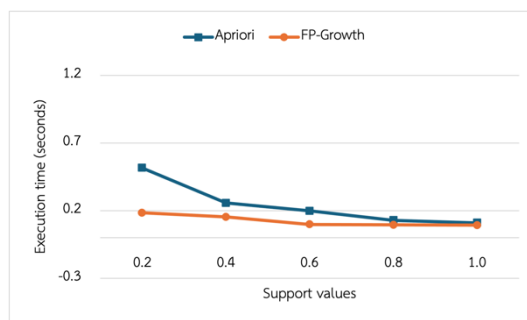
d. Support = 0.8



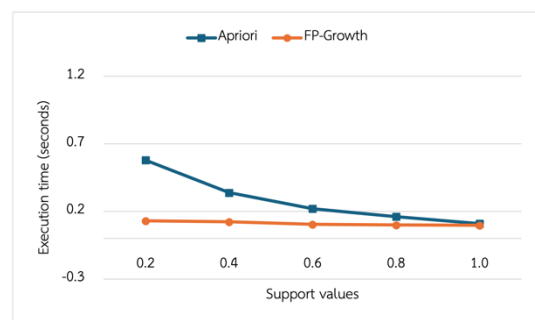
e. Support = 1.0

ภาพที่ 2 แสดงการเปรียบเทียบหน่วยความจำที่ใช้ในการค้นหาไอเทมเซตที่เกิดขึ้นบ่อยระหว่างอัลกอริทึม Apriori และอัลกอริทึม FP-Growth สำหรับการเพิ่มขึ้นของ Transaction

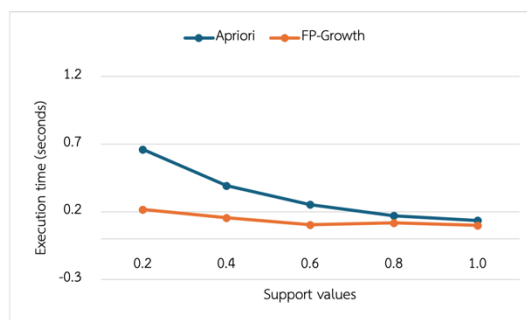
จากภาพที่ 2 กำหนดค่า Support ระหว่าง 0.2 – 1.0 (ภาพที่ 2.a – 2.e) พบว่ากรณีข้อมูล Transaction เพิ่มขึ้น จะทำให้หน่วยความจำที่ใช้ในการค้นหาของอัลกอริทึม Apriori เพิ่มขึ้นมากกว่าอัลกอริทึม FP-Growth ซึ่งสอดคล้องกับงานวิจัยของ Patil et al. [12] ที่พบว่า FP-Growth ใช้หน่วยความจำน้อยกว่า Apriori อย่างมีนัยสำคัญเมื่อจำนวน Transaction เพิ่มขึ้น เพราะใช้โครงสร้างข้อมูลแบบ FP-Tree ช่วยลดการเก็บข้อมูลซ้ำซ้อน และเพิ่มประสิทธิภาพในการค้นหาไอเทมเซตที่เกิดขึ้นบ่อย ในขณะที่ Apriori ต้องสร้างและตรวจสอบ Candidate Itemsets หลายครั้ง ทำให้ใช้หน่วยความจำมากกว่า



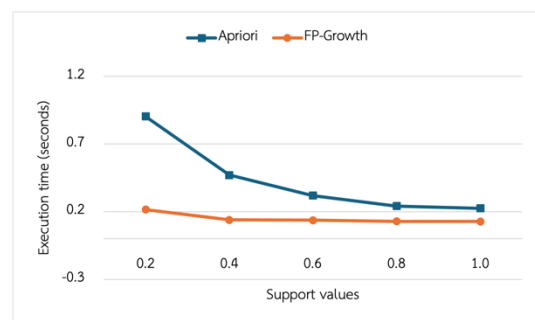
a. Transaction = 25,000



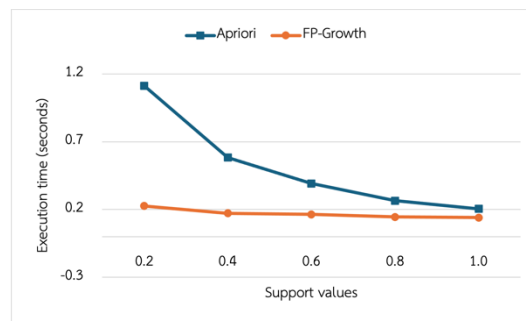
b. Transaction = 28,000



c. Transaction = 30,000



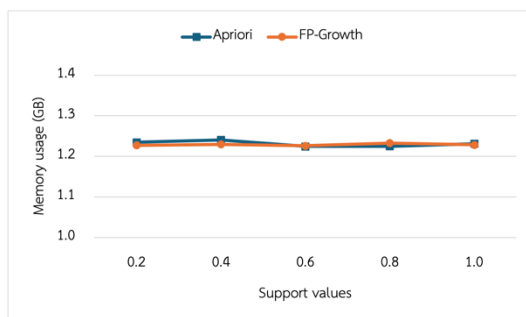
d. Transaction = 35,000



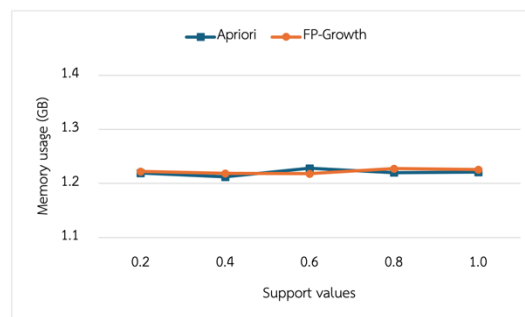
e. Transaction = 38,000

ภาพที่ 3 แสดงการเปรียบเทียบความเร็วที่ใช้ค้นหาไอเทมเซตที่เกิดขึ้นบ่อยระหว่างอัลกอริทึม Apriori และอัลกอริทึม FP-Growth สำหรับการเพิ่มขึ้นของค่า Support

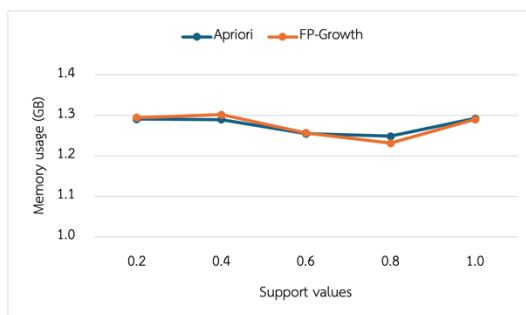
จากภาพที่ 3 กำหนดจำนวนข้อมูล Transaction ระหว่าง 25,000 – 38,000 (ภาพที่ 3.a – 3.e) พบว่า กรณีค่า Support เพิ่มขึ้น จะทำให้ความเร็วการค้นหาลดลงตามไปด้วย และอัลกอริทึม FP-Growth จะทำงานได้เร็วกว่าอัลกอริทึม Apriori โดยเวลาการทำงานของทั้งสองอัลกอริทึมจะขยับเข้ามาใกล้กันเมื่อกำหนดค่า Support เพิ่มขึ้น ซึ่งสอดคล้องกับงานวิจัยของ Aldino et al. [6] และ Wicaksono et al. [7] ที่พบว่าอัลกอริทึม FP-Growth ประมวลผลได้เร็วกว่า Apriori อย่างมีนัยสำคัญ



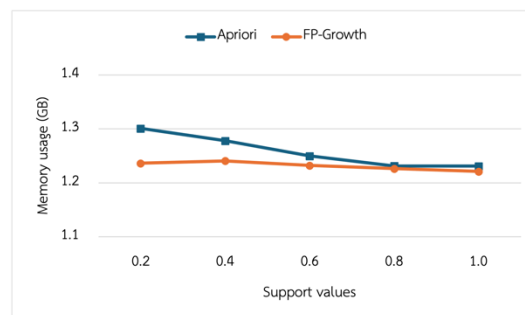
a. Transaction = 25,000



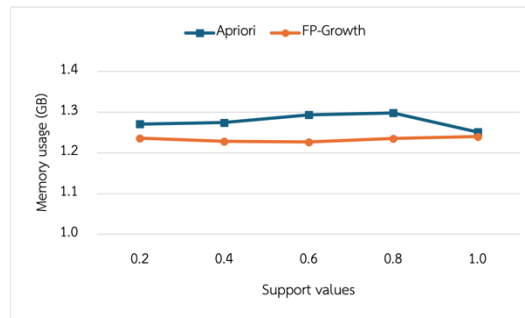
b. Transaction = 28,000



c. Transaction = 30,000



d. Transaction = 35,000



e. Transaction = 38,000

ภาพที่ 4 แสดงการเปรียบเทียบหน่วยความจำที่ใช้ค้นหาไอเทมเซตที่เกิดขึ้นบ่อยระหว่างอัลกอริทึม Apriori และอัลกอริทึม FP-Growth สำหรับการเพิ่มขึ้นของค่า Support

จากภาพที่ 4 กำหนดจำนวนข้อมูล Transaction ระหว่าง 25,000 – 38,000 (ภาพที่ 4.a – 4.e) พบว่า กรณีค่า Support เพิ่มขึ้น จะทำให้แนวโน้มของความเร็วการค้นหาลดลงตามไปด้วย และอัลกอริทึม FP-Growth จะใช้หน่วยความจำในการค้นหาได้น้อยกว่าอัลกอริทึม Apriori ซึ่งสอดคล้องกับงานวิจัยของ Wicaksono et al. [7]

อภิปรายผลการวิจัย

งานวิจัยเปรียบเทียบประสิทธิภาพของอัลกอริทึม Apriori และ FP-Growth ในการค้นหาไอเทมเซตที่เกิดขึ้นบ่อย พบว่า FP-Growth ทำงานเร็วกว่าในชุดข้อมูลร้านขายของชำ ผลลัพธ์สอดคล้องกับงานวิจัยของ Nasreen et al. [13] ดังตารางที่ 4

ตารางที่ 4 แสดงการวิเคราะห์การทำงานระหว่างอัลกอริทึม Apriori และ FP-Growth

	อัลกอริทึม Apriori	อัลกอริทึม FP-Growth
หลักการทำงาน	ค้นหาด้วยวิธี Breadth first search	ค้นหาด้วยวิธี Depth first search
การสแกนฐานข้อมูล	สแกนฐานข้อมูลหลายรอบ	สแกนฐานข้อมูล 2 รอบ
เวลาที่ใช้ทำงาน	ใช้เวลานาน โดยเฉพาะกับฐานข้อมูลขนาดใหญ่	ใช้เวลาน้อยลง เมื่อเทียบกับอัลกอริทึม Apriori
ข้อเสีย	ใช้ทรัพยากรหน่วยความจำสูงเพื่อเก็บชุดไอเทมเซตจำนวนมาก	โครงสร้าง FP-Tree ใช้เวลาในการสร้างนาน
ข้อดี	ออกแบบใหม่มีประสิทธิภาพในการทำงานที่รวดเร็ว และง่ายต่อการใช้งาน	ฐานข้อมูลถูกสแกนเพียง 2 รอบ และไม่มีการสร้างชุดข้อมูล Candidate itemsets

จากตารางที่ 4 พบว่าอัลกอริทึม Apriori มีข้อจำกัด ต้องสแกนฐานข้อมูลหลายรอบ ทำให้ใช้เวลานานและหน่วยความจำสูง โดยเฉพาะกับฐานข้อมูลขนาดใหญ่ ในขณะที่อัลกอริทึม FP-Growth ไม่สร้าง

Candidate itemsets ช่วยลดการใช้หน่วยความจำและเพิ่มประสิทธิภาพการทำงาน [7,12] โดยสแกนฐานข้อมูลเพียง 2 รอบ จึงทำให้การประมวลผลเร็วกว่า ดังนั้น FP-Growth เป็นทางเลือกที่ดีกว่า Apriori สำหรับฐานข้อมูลขนาดใหญ่ [8]

ข้อเสนอแนะ

สำหรับงานวิจัยในครั้งต่อไป จะทำการศึกษาและเปรียบเทียบประสิทธิภาพของกฎที่ได้โดยใช้ค่าความเชื่อมั่น (Confidence) และค่าลิฟท์ (Lift) เพื่อวิเคราะห์ผลกระทบต่อประสิทธิภาพของไอเทมเซต นอกจากนี้ ยังจะศึกษาด้านความเร็วในการทำงาน โดยเปรียบเทียบความเร็วของอินเทอร์เนตที่ใช้ในการทดลองระหว่างโปรแกรมที่ติดตั้งบนเครื่องคอมพิวเตอร์กับการทำงานบนระบบคลาวด์ ซึ่งอาจมีผลต่อความเร็วของการทำงาน

เอกสารอ้างอิง

- [1] Tan, P.-N., Steinbach, M., & Kumar, V. (2005). Market Basket Analysis using Association Rule Mining. *Introduction to Data Mining*, 327-352. Boston: Pearson Addison Wesley.
- [2] Zhao, Q., & Bhowmick, S. S. (2003). Association rule mining: A survey. *Nanyang Technological University, Singapore*, 135.
- [3] Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *Proceedings of the 20th VLDB Conference Santiago, Chile*, 487-499.
- [4] Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. *ACM Sigmod Record*, 29(2), 1-12.
- [5] Gulzar, K., Ayooob Memon, M., Mohsin, S. M., Aslam, S., Akber, S. M., & Nadeem, M. A. (2023). An Efficient Healthcare Data Mining Approach Using Apriori Algorithm: A Case Study of Eye Disorders in Young Adults. *Information*, 14(4), 203; <https://doi.org/10.3390/info14040203>
- [6] Aldino, A. A., Pratiwi, E. D., Sintaro, S., & Putra, A. D. (2021). Comparison of market basket analysis to determine consumer purchasing patterns using fp-growth and apriori algorithm. *2021 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE)*, 29-34.
- [7] Wicaksono, D., JAMBAK, M. I., & SAPUTRA, D. M. (2020). The comparison of apriori algorithm with preprocessing and FP-growth algorithm for finding frequent data pattern in association rule. *Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019)*, 315-319. <https://doi.org/10.2991/aisr.k.200424.047>
- [8] Srinadh, V. (2022). Evaluation of Apriori, FP growth and Eclat Association rule mining algorithms. *International Journal of Health Sciences*, (II), 7475-7485.
- [9] Dwiputra, D., Widodo, A. M., Akbar, H., & Firmansyah, G. (2023). Evaluating the Performance of Association Rules in Apriori and FP-Growth Algorithms: Market Basket Analysis to Discover Rules of Item Combinations. *Journal of World Science*, 2(8), 1229-1248.
- [10] DEDHIA, H. (2022). *Groceries dataset*. <https://www.kaggle.com/datasets/heeraldedhia/groceries-dataset/data>.

- [11] Han, J., Kamber, M., & Mining, D. (2006). Concepts and techniques. *Morgan Kaufmann*, 340, 94104-3205.
- [12] Patil, M., & Patil, T. (2022). Apriori Algorithm against Fp Growth Algorithm: A Comparative Study of Data Mining Algorithms. *Available at SSRN 4113695*.
- [13] Nasreen, S., Azam, M. A., Shehzad, K., Naeem, U., & Ghazanfar, M. A. (2014). Frequent Pattern Mining Algorithms for Finding Associated Frequent Patterns for Data Streams: A Survey. *Procedia Computer Science*, 37, 109–116. <https://doi.org/10.1016/j.procs.2014.08.019>.