อัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น

PROBABILITY-BASED INCREMENTAL ASSOCIATION RULE DISCOVERY ALGORITHM

รัชดาภรณ์ อมรชีวิน

RATCHADAPORN AMORNCHEWIN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา[illegible]ปรัชญาดุษฎีบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2553

KMITL-2010-IT-D-001-001

อัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น

PROBABILITY-BASED INCREMENTAL ASSOCIATION RULE DISCOVERY ALGORITHM

รัชดาภรณ์ อมรชีวิน

RATCHADAPORN AMORNCHEWIN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาปรัชญาดุษฎีบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2553

KMITL-2010-IT-D-001-001

PROBABILITY-BASED INCREMENTAL ASSOCIATION RULE DISCOVERY ALGORITHM

RATCHADAPORN AMORNCHEWIN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2010

KMITL-2010-IT-D-001-001

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา รศ.ดร.วรพจน์ กรีสุระเดช ที่ให้ความช่วยเหลือ ให้คำชี้แนะช่วยแก้ปัญหาตลอดจนให้ความรู้และประสบการณ์ที่ดีแก่ข้าพเจ้า

ขอขอบพระคุณ ศ.ดร. ชิดชนก เหลือสินทรัพย์และท่านคณะกรรมการสอบหัวข้อและโครงร่างวิทยานิพนธ์ทุกท่านที่ได้กรุณาให้คำแนะนำตลอดจนข้อชี้แนะ จนในที่สุดทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลงได้

ขอขอบพระคุณ มหาวิทยาลัยราชภัฏเทพสตรี ที่ได้ให้ทุนสนับสนุนในการเรียน ขอขอบคุณท่านคณบดีและ พี่ๆ เพื่อนๆ และน้องๆ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยราชภัฏเทพสตรีที่ให้ความช่วยเหลือและกำลังใจที่ดี

สุดท้ายต้องขอขอบพระคุณบิดา มารดา และขอบคุณพี่ๆ และหลานๆ ที่เป็นกำลังใจที่ดีตลอดมา

สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดามารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า

รัชดาภรณ์ อมรชีวิน

หัวข้อวิทยานิพนธ์	อัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักของความน่าจะเป็น
นักศึกษา	นางสาวรัชดาภรณ์ อมรชีวิน
รหัสประจำตัว	47066008
ปริญญา	ปรัชญาดุษฎีบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
พ.ศ.	2553
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.วรพจน์ กรีสุระเดช

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้นำเสนอการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยอาศัยหลักความน่าจะเป็นโดยมีวัตถุประสงค์เพื่อค้นคว้าและพัฒนาอัลกอริทึมที่ใช้ในการค้นหากฎความสัมพันธ์เมื่อฐานข้อมูลมีการเปลี่ยนแปลง ด้วยลักษณะของฐานข้อมูลที่นำมาใช้ค้นหากฎความสัมพันธ์มักมีการเปลี่ยนแปลงในช่วงเวลาต่างๆกัน ดังนั้นเมื่อมีการเพิ่มข้อมูลใหม่จำนวนหนึ่งเข้ามาในฐานข้อมูลจะมีผลต่อกฎความสัมพันธ์ที่ได้ทำการค้นหาไว้แล้ว เนื่องจากอาจพบว่าฟรีเควนท์ไอเทมเซตเดิมที่นำมาสร้างกฎความสัมพันธ์นั้นกลายเป็นสิ่งที่ไม่น่าสนใจอีกต่อไป ในขณะเดียวกันอาจพบฟรีเควนท์ไอเทมเซตใหม่ที่น่าสนใจเกิดขึ้น เพื่อแก้ปัญหาการเพิ่มขยายการค้นหากฎความสัมพันธ์ที่เกิดขึ้นเมื่อมีข้อมูลใหม่เพิ่มเข้ามา ในงานวิทยานิพนธ์ฉบับนี้ได้นำแนวคิดของหลักความน่าจะเป็นโดยอาศัยทฤษฎีของเบอร์นูลลีมาใช้ในการทำนายและจัดเก็บไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตทำให้ลดการสแกนฐานข้อมูลเดิมและสามารถทำการค้นหาฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงได้อย่างครบถ้วนถูกต้องและมีประสิทธิภาพดีกว่าอัลกอริทึมในการเพิ่มขยายการค้นหากฎความสัมพันธ์ที่มีมาก่อนหน้าเช่น อัลกอริทึมเอฟยูพี, บอร์เดอร์ และพรีลาจก์

Thesis Title	Probability-Based Incremental Association Rule Discovery Algorithm
Student	Ms.Ratchadaporn Amornchewin
Student ID	47066008
Degree	Doctor of Philosophy
Programme	Information Technology
Year	2010
Thesis Advisor	Assoc.Prof.Dr.Worapoj Kreesuradej

ABSTRACT

This thesis proposes the Probability-Based incremental association rule discovery Algorithm to develop an algorithm for discovering association rules from dynamic database. In dynamic databases, new transactions are appended as time advances. This may introduce new association rules and some existing association rules would become invalid. Thus, the maintenance of association rules for dynamic databases is an important problem. In this thesis, probability-based incremental association rule discovery algorithm is proposed to deal with this problem. The proposed algorithm uses the principle of Bernoulli theorem to predict and keep expected frequent itemsets. This can reduce a number of times to scan an original database. The simulation results show that the proposed algorithm can discover all frequent k-itemset correctly with better performance than that of the previous incremental association rule algorithms such as FUP, Border and Pre-Large algorithms.

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

สารบัญรูป

สารบัญรูป (ต่อ)

รูปที่ หน้า

สารบัญรูป (ต่อ)

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเทคโนโลยีสารสนเทศเข้ามามีบทบาทกับหน่วยงานและองค์กรต่างๆ ทำให้มีการจัดเก็บข้อมูลจำนวนมากไว้ในฐานข้อมูล และมีแนวโน้มที่จะมีจำนวนเพิ่มขึ้นเรื่อยๆ โดยไม่ได้นำข้อมูลต่างๆ มาใช้ให้เกิดประโยชน์สูงสุด ซึ่งหากนำข้อมูลที่จัดเก็บไว้แล้วมาศึกษาสิ่งที่อาจจะพบคือความสำคัญของข้อมูลที่มีความสัมพันธ์อย่างคาดไม่ถึงซ่อนอยู่ก็คือความรู้ (Knowledge) นั่นเอง

การจัดการความรู้ได้ถูกพัฒนาอย่างกว้างขวางทั้งทางเทคโนโลยีและโปรแกรมประยุกต์ สำหรับงานวิจัยนี้ให้ความสนใจในเรื่องการค้นหาความรู้ในฐานข้อมูล (Knowledge discovery in database: KDD) ที่จัดอยู่ในกลุ่มของงานวิจัยด้านเทคโนโลยีฐานข้อมูล (Database technology) [1] การค้นหาความรู้ในฐานข้อมูลเป็นการอ้างถึงกระบวนการทั้งหมดของการค้นหาความรู้ที่เป็นประโยชน์จากข้อมูล ในหลายๆ งานวิจัยอาจจะพบคำที่ใช้ในความหมายเดียวกันคือ ดาต้าไมน์นิ่ง (data mining) แต่ดาต้าไมน์นิ่งเป็นเพียงขั้นตอนหนึ่งในกระบวนการค้นหาข้อมูลจากฐานข้อมูลที่นำมาหารูปแบบ (Pattern) เพื่อใช้ในการสกัดข้อมูล (Extract data) ที่สามารถนำมาใช้ประโยชน์ได้

การไมน์นิ่งกฎความสัมพันธ์ (Association rule mining) เป็นกระบวนการสำคัญในการทำดาต้าไมน์นิ่ง โดยจะทำการค้นหาและดึงรูปแบบของข้อมูลระหว่างรายการต่างๆ ทั้งหมดที่ได้จัดเก็บไว้ในฐานข้อมูลมาใช้ในการทำนายความสัมพันธ์ระหว่างข้อมูลด้วยการสร้างให้อยู่ในรูปของกฎความสัมพันธ์ เพื่อช่วยในการตัดสินใจและวางแผนด้านบริหารได้

อัลกอริทึมสำหรับค้นหากฎความสัมพันธ์ที่ได้รับความนิยมคืออัลกอริทึมอะพริโอริ [2] ซึ่งทำการค้นหากฎความสัมพันธ์ของฐานข้อมูลโดยเริ่มจากการหาฟรีเควนท์ k-ไอเทมเซต (Frequent k-itemset) ซึ่งหมายถึงไอเทมเซตที่ประกอบด้วย k-ไอเทม (k=1,2,…,n) ที่เกิดขึ้นร่วมกันและได้รับการพิจารณาว่ามีจำนวนมากกว่าหรือเท่ากับค่าที่ใช้ในการวัดความสัมพันธ์ที่ได้กำหนดไว้ (Minimum support) และการนำฟรีเควนท์ k-ไอเทมเซต (k >= 2) ที่ได้มาสร้างเป็นกฎความสัมพันธ์ให้อยู่ในรูปของ IF…THEN rules เช่น IF milk THEN bread หมายถึง ถ้าซื้อนมแล้วจะซื้อขนมปังด้วย เป็นต้น ดังนั้นเมื่อทรานแซคชัน (Transaction) ต่างๆ ที่ถูกจัดเก็บในฐานข้อมูลนั้นมีการปรับเปลี่ยนให้มีความทันสมัยหรือทันต่อเวลาจะมีผลต่อการเปลี่ยนแปลงค่าความสัมพันธ์ระหว่างฟรีเควนท์ k-ไอเทมทำให้กฎความสัมพันธ์ที่มีอยู่เดิมอาจจะไม่มีความถูกต้องอีกต่อไป

การเพิ่มขยายการค้นหากฎความสัมพันธ์เป็นการทำดาต้าไมน์นิ่งเพื่อค้นหากฎความสัมพันธ์ที่ได้จากเปลี่ยนแปลงทรานแซคชันในช่วงเวลาหนึ่งๆ เช่นการเพิ่ม, ลบ หรือแก้ไข

เป็นต้น ซึ่งมีผลต่อฟรีเควนท์ k-ไอเทมเซตที่มีอยู่เดิม ทำให้ข้อมูลที่ไม่เป็นฟรีเควนท์ไอเทมเซตอาจกลายมาเป็นฟรีเควนท์ไอเทมเซต หรือในทางกลับกันข้อมูลที่เคยเป็นฟรีเควนท์ k-ไอเทมเซตอาจจะเป็นหรือไม่เป็นฟรีเควนท์ k-ไอเทมเซตในฐานข้อมูลที่ได้รับการปรับปรุงแล้วอีกต่อไป ซึ่งมีผลต่อการเปลี่ยนแปลงกฎความสัมพันธ์ทำให้ต้องหาฟรีเควนท์ k-ไอเทมเซตใหม่ ลักษณะปัญหาของการหากฎความสัมพันธ์ใหม่สามารถสรุปได้ดังนี้คือ

- จำนวนการวนรอบที่ใช้ในการสแกนฐานข้อมูลเดิมและฐานข้อมูลใหม่ที่ต้องการปรับปรุงใหม่ ซึ่งใช้เวลามากในการค้นหาความสัมพันธ์ด้วยขั้นตอนการหากฎความสัมพันธ์ใหม่ทั้งหมด โดยไม่นำความรู้ที่ได้จากการทำดาต้าไมน์นิ่งก่อนที่จะทำการปรับปรุงมาใช้
- ปัญหาในการหาฟรีเควนท์ k-ไอเทมเซตที่เกิดในฐานข้อมูลที่ปรับปรุง (L_{DB+db}) ซึ่งจะต้องมีค่าสนับสนุนที่เกิดขึ้นมากกว่าค่าค่าสนับสนุนน้อยที่สุดที่กำหนด เพื่อนำมาสร้างกฎความสัมพันธ์ใหม่ที่เกิดขึ้น
- จำนวนไอเทมเซตที่ถูกสแกนในฐานข้อมูลเดิมและฐานข้อมูลใหม่ เมื่อข้อมูลในฐานข้อมูลเปลี่ยนไป อัลกอริทึมอะพริโอริจะทำการค้นหากฎความสัมพันธ์ที่ปรากฏในฐานข้อมูลปรับปรุงทั้งหมด โดยไม่นำความรู้ที่ได้จากการค้นหากฎความสัมพันธ์ที่ได้จากฐานข้อมูลเดิมมาใช้ทำให้ต้องใช้เวลานานในการค้นหากฎความสัมพันธ์ใหม่ทั้งหมด งานวิจัยต่างๆได้นำเสนอเพื่อแก้ปัญหาการทำไมน์นิ่งกฎความสัมพันธ์ใหม่ทั้งหมดของอัลกอริทึมอะพริโอริ อัลกอริทึมเหล่านี้ ได้แก่อัลกอริทึมเอฟยูพี อัลกอริทึมบอร์เดอร์ อัลกอริทึมพรีลาจก์และอัลกอริทึมโพรมิสซิง ซึ่งได้นำเสนอเทคนิคเพื่อเพิ่มประสิทธิภาพในการเพิ่มขยายการค้นหากฎความสัมพันธ์เมื่อมีการเพิ่มข้อมูลใหม่เข้ามาภายใต้พื้นฐานการทำงานของอัลกอริทึมอะพริโอริ (Apriori-Based Algorithm)

โดยแนวคิดต่างๆ ที่นำเสนอนี้ได้นำความรู้ที่ได้จากการไมน์นิ่งในฐานข้อมูลเดิมมาใช้โดยมีการเก็บไอเทมเซตต่างๆ จำนวนมากที่พบในฐานข้อมูลเดิมไว้เพื่อลดการค้นหาไอเทมเซตและลดการสแกนฐานข้อมูลเดิม ในขณะที่เมื่อมีการเพิ่มข้อมูลใหม่เข้ามาในฐานข้อมูลเดิมจะพบว่านอกจากจะต้องใช้เวลานานในการวนรอบเพื่อปรับปรุงค่าสนับสนุนให้กับไอเทมเซตที่เก็บไว้แล้วยังคงต้องใช้เวลาในการกลับไปสแกนฐานข้อมูลเดิมเพื่อหาค่าสนับสนุนให้กับไอเทมเซตใหม่ทุกตัวที่เกิดขึ้นด้วย

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

งานวิจัยนี้มีวัตถุประสงค์เพื่อค้นคว้าและพัฒนาอัลกอริทึมที่ใช้ในการเพิ่มขยายการค้นหากฎความสัมพันธ์เมื่อมีการเพิ่มทรานแซคชันใหม่เข้าไปในฐานข้อมูลเดิม ซึ่งมีผลต่อการเปลี่ยนแปลงกฎความสัมพันธ์ที่ได้ทำการค้นหาไว้แล้วจากการทำดาต้าไมน์นิ่งในฐานข้อมูลเดิม (Original database) และนำแนวความคิดที่จะหลีกเลี่ยงหรือใช้จำนวนครั้งน้อยที่สุดในการสแกนฐานข้อมูลเดิมที่มีจำนวนของข้อมูลที่จัดเก็บไว้จำนวนมาก รวมถึงการค้นหาฟรีเควนท์ไอเทมเซตที่ได้จากการ

เพิ่มฐานข้อมูลใหม่เข้ามา (Increment database) ที่มีผลต่อการเปลี่ยนแปลงกฎความสัมพันธ์ใหม่ที่เกิดขึ้นภายหลังจากการปรับปรุงฐานข้อมูล (Updated database)

เพื่อให้การค้นหากฎความสัมพันธ์ระหว่างข้อมูลครอบคลุมข้อมูลทั้งหมด ด้วยอัลกอริทึมที่มีประสิทธิภาพในการปรับปรุงค่าของแต่ละฟรีเควนท์ k-ไอเทมเซต และความสามารถในการค้นหาฟรีเควนท์ k-ไอเทมใหม่ที่เกิดขึ้นจากการปรับปรุงได้อย่างถูกต้องและรวดเร็วโดยใช้ทรัพยากรที่มีอยู่ให้เกิดประโยชน์มากที่สุด

1.3 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

งานวิจัยนี้ได้นำทฤษฎีและแนวความคิดต่างๆมาใช้ คือ การประยุกต์ใช้ในการหากฎความสัมพันธ์ โดยมีทฤษฎีและแนวคิดที่สามารถนำมาประยุกต์ใช้ได้ดังนี้คือ

1. การค้นหาความรู้ในฐานข้อมูล (Knowledge discovery in database : KDD)
2. กฎความสัมพันธ์ (Association rule)
3. การเพิ่มขยายการค้นหากฎความสัมพันธ์ (Incremental association rule)
4. ทฤษฎีเบอร์นูลลี (Bernoulli Theorem)

1.4 ขอบเขตของการวิจัย

งานวิจัยที่นำเสนอนี้เป็นโมเดลที่ประยุกต์ใช้กับการเพิ่มขยายการค้นหากฎความสัมพันธ์เมื่อฐานข้อมูลถูกปรับปรุงให้สอดคล้องกับข้อมูลที่ต้องการหากฎความสัมพันธ์ได้

1.5 ขั้นตอนของการศึกษา

ในขั้นตอนของการศึกษานี้ได้แสดงลำดับการทำงานตั้งแต่เริ่มต้นจนถึงสิ้นสุดการทำงานดังรายละเอียดต่อไปนี้

1.5.1 ศึกษาทฤษฎีและงานวิจัยจากเอกสารบทความต่าง ๆ ที่เกี่ยวข้องกับการทำงานวิจัย

1.5.2 กำหนดหัวข้อ วัตถุประสงค์ และขอบเขตการทำงานวิจัย

1.5.3 วิเคราะห์อัลกอริทึมและออกแบบโมเดลใหม่

1.5.4 พัฒนาโปรแกรมโดยใช้ซอฟท์แวร์แมทแล็บ (MATLAB) พร้อมทั้งแก้ไขข้อผิดพลาดและทดสอบการทำงานของโมเดลกับข้อมูลที่กำหนดขึ้นเอง

1.5.5 เตรียมข้อมูลที่ใช้งานจริง เพื่อนำมาทดสอบการทำงานโมเดล โดยในที่นี้จะใช้ชุดข้อมูลสังเคราะห์ (Synthetic dataset) ที่ได้มีการคิดค้นโดย Agrawal et.al.[2] ด้วยหลักการสร้างชุดข้อมูลสังเคราะห์จากหลักการทางสถิติที่เลียนแบบมาจากลักษณะการซื้อขายสินค้าจริงและได้มีการนำมาใช้ในการทดสอบอย่างแพร่หลายในงานวิจัยต่างๆ

1.5.6 ทดลองกับชุดข้อมูลสังเคราะห์ที่สร้างขึ้น เพื่อวัดประสิทธิภาพการทำงานของโมเดล

1.5.7 รวบรวมผลการทดลองจากการทำงานของโมเดล

1.5.8 วิเคราะห์และสรุปผลการทดลอง

1.5.9 เรียบเรียงเอกสารประกอบวิทยานิพนธ์

การค้นหากฎความสัมพันธ์เป็นงานวิจัยหนึ่งที่ได้รับความนิยมเพื่อค้นหาความรู้หรือสารสนเทศ (Information) ที่ซ่อนอยู่ในฐานข้อมูล และนำความรู้ที่ได้มาแสดงอยู่ในรูปของกฎความสัมพันธ์ เพื่อใช้ในงานด้านต่างๆ เช่น การบริหาร, การตัดสินใจ และการวางแผน เป็นต้น ซึ่งการหากฎความสัมพันธ์มีการทำวิจัยในหลายรูปแบบแตกต่างกันตามลักษณะของข้อมูล

ในบทที่ 2 จะกล่าวถึง ทฤษฎีและงานวิจัยต่างๆ ที่เกี่ยวข้องกับการทำงานวิจัยที่นำเสนอนี้

บทที่ 2
ทฤษฎีพื้นฐานที่ใช้ในการวิจัย และงานวิจัยที่เกี่ยวข้อง

การหากฎความสัมพันธ์เป็นการทำไมน์นิ่งอย่างหนึ่งที่ได้รับความสนใจในการทำงานวิจัย เพื่อค้นหาสารสนเทศที่ซ่อนอยู่ในฐานข้อมูลโดยเฉพาะในเรื่องที่เกี่ยวกับการวิเคราะห์สินค้าที่ถูกซื้อร่วมกัน (Market basket analysis) ทั้งนี้เพื่อให้สามารถนำข้อมูลหรือความรู้ที่ได้จากการค้นหากฎความสัมพันธ์มาใช้ในด้านต่างๆ เช่น การวางแผนกลยุทธ์ทางการตลาด, ช่วยในการตัดสินใจหรือวางแผนเกี่ยวกับผลิตภัณฑ์ เป็นต้น

ในบทนี้จะกล่าวถึงทฤษฎีพื้นฐานต่างๆ ของการหากฎความสัมพันธ์ (Association rules), การเพิ่มขยายกฎความสัมพันธ์ (Incremental Association rule) และงานวิจัยด้านต่างๆ ที่เกี่ยวข้องกับการหากฎความสัมพันธ์ของข้อมูลที่ปรากฏในฐานข้อมูลซึ่งมีรายละเอียดดังนี้

2.1 การไมน์นิ่งกฎความสัมพันธ์

กระบวนการไมน์นิ่งกฎความสัมพันธ์ (Association rules mining) ได้ถูกนำเสนอในปี ค.ศ. 1993[2] เพื่อใช้ในการค้นหาความสัมพันธ์ระหว่างทรานแซคชันที่ปรากฏในฐานข้อมูลขนาดใหญ่ ความสัมพันธ์ที่ได้สามารถใช้บอกลักษณะของข้อมูลที่เกิดขึ้นร่วมกันหรือใช้ในการทำนายลักษณะของข้อมูลที่อาจเกิดขึ้นต่อไป โดยจะแสดงอยู่ในรูปของกฎความสัมพันธ์

กฎความสัมพันธ์ เป็นเทคนิคที่ใช้หากฎเกณฑ์ความสัมพันธ์ที่ปรากฏขึ้นระหว่างข้อมูลทั้งหมดในฐานข้อมูล โดยกำหนดให้ I เป็นเซตของไอเทม I โดย $I = I_1, I_2, \ldots, I_m$, เซตของทรานแซคชันในฐานข้อมูล แทนด้วย D ซึ่งแต่ละทรานแซคชัน T เป็นเซตของไอเทม โดย $T \subseteq I$ ทรานแซคชัน T แต่ละทรานแซคชันจะมีตัวระบุ (Transaction Identifier : TID), ถ้า A เป็นเซตของไอเทมในทรานแซคชันหนึ่ง ซึ่ง $A \subseteq T$ ไอเทมต่างๆ ที่ปรากฏในฐานข้อมูลจะถูกนำมาหาความสัมพันธ์ โดยไอเทมที่จะนำมาสร้างเป็นกฎความสัมพันธ์ได้จะต้องมีจำนวนของข้อมูลที่เกิดขึ้นมากกว่าหรือเท่ากับตัววัด 2 ตัวคือ ค่าสนับสนุนน้อยที่สุด (minimum support) และค่าความเชื่อมั่นน้อยที่สุด (minimum confidence)

กฎความสัมพันธ์ที่ได้จะแสดงอยู่ในรูปของกฎ “ถ้า ... แล้ว...” (IF...THEN ...) ซึ่งในกฎหนึ่งๆ ประกอบด้วย 2 ส่วนคือ ส่วนด้านซ้ายของกฎ (Left-hand side)หรือส่วนที่อยู่หลัง IF หรือก่อน THEN (Antecedent) ที่ประกอบด้วยเงื่อนไขที่เป็นจริงหนึ่งหรือมากกว่าหนึ่งเงื่อนไข เรียกส่วนนี้ว่า ส่วนตัวกฎ (Rule Body) และส่วนด้านขวาของกฎ (Right-hand side) หรือส่วนที่อยู่หลัง THEN (Consequent) เป็นส่วนที่แสดงผลเมื่อเงื่อนไขที่ระบุในส่วนของ IF เป็นจริง เรียกส่วนนี้ว่า ส่วนหัวของกฎ (Rule head) ตัวอย่างของกฎความสัมพันธ์เช่น IF A THEN B หรือ เขียนแทนด้วย

สัญลักษณ์ $A \Rightarrow B$ โดยค่า $A \subset I$, $B \subset I$ และ $A \cap B = \phi$ [4] กฎ $A \Rightarrow B$ จะถูกจัดให้มีในเซตของทรานแซคชันในฐานข้อมูล D ด้วยค่าสนับสนุน s ซึ่งเป็นเปอร์เซ็นต์ของทรานแซคชันข้อมูลในฐานข้อมูล D ที่มีทั้ง A และ B ($A \cup B$) โดยจะเป็นค่าความน่าจะเป็นที่จะเกิด A และ B พร้อมกัน ($P(A \cup B)$) กฎ $A \Rightarrow B$ จะมีค่าความเชื่อมั่น c ในเซตของข้อมูลทรานแซคชัน ใน D ถ้า c เป็นเปอร์เซ็นต์ของข้อมูลทรานแซคชันในฐานข้อมูล D ที่มี A แล้วจะมี B ด้วย ซึ่งเป็นเรื่องของความน่าจะเป็นแบบมีเงื่อนไข ($P(B \mid A)$) ดังสามารถสรุปได้ดังนี้

$$Support\ (A \Rightarrow B) = P(A \cup B)$$

$$Confidence\ (A \Rightarrow B) = P(B \mid A)$$

นิยามและความหมายของคำต่าง ๆ ที่ใช้ในการค้นหากฎความสัมพันธ์ได้แก่

1. ไอเทม (Item) คือข้อมูลแต่ละตัวที่ใช้ในการหากฎความสัมพันธ์ เช่น bread, cheese , shampoo, milk เป็นต้น

2. ไอเทมเซต (Itemset) คือ ความสัมพันธ์ของข้อมูลที่หาได้ ไอเทมเซตประกอบด้วยไอเทมที่มีความยาวแตกต่างกัน โดยทั่วไปจะใช้ k แทนขนาดความยาวไอเทมเซต (k = 1, 2 ,3,...,n) เรียกว่า k-ไอเทมเซต ตัวอย่างไอเทมเซตเช่น 2-ไอเทมเซตหมายถึง ไอเทมเซตที่ประกอบด้วยสมาชิกของไอเทม 2 ตัว เช่น {bread, cheese}, {milk, bread}, {milk , shampoo} เป็นต้น และ 3 ไอเทมเซตหมายถึง ไอเทมเซตที่ประกอบด้วยสมาชิกของไอเทม 3 ตัว เช่น {milk, bread , shampoo}, {bread, cheese, milk} เป็นต้น

3. แคนดิเดทไอเทมเซตหรือไอเทมเซตตัวเลือก (Candidate Itemset: C) คือ ชุดของไอเทมเซตที่ได้จากการเชื่อมความสัมพันธ์ของไอเทมเซตก่อนหน้านี้ ซึ่งเป็นแคนดิเดทไอเทมเซตเป็นไอเทมเซตที่จะกลายมาเป็นฟรีเควนท์ไอเทมเซตได้เมื่อผ่านการสแกนเพื่อหาค่าสนับสนุนในฐานข้อมูล

4. ค่าสนับสนุน (Support value) เป็นค่าแสดงความสัมพันธ์ระหว่างจำนวนของไอเทมที่ปรากฏรายการข้อมูลต่างๆ ทั้งหมด ในฐานข้อมูลสามารถแสดงเป็นสมการได้ดังสมการที่ 2.1

$$Support = \frac{n}{N} \tag{2.1}$$

โดยที่ n คือจำนวนไอเทมที่ปรากฏในรายการข้อมูลต่างๆ ของฐานข้อมูล

N คือจำนวนรายการข้อมูลทั้งหมดในฐานข้อมูล

5. ค่าสนับสนุนน้อยที่สุด (Minimum support : min_sup) คือค่าสนับสนุนที่น้อยที่สุดที่ทำให้ความสัมพันธ์ที่ได้นั้นยังมีความน่าสนใจ ซึ่งค่าสนับสนุนน้อยที่สุดจะถูกกำหนดโดยผู้ใช้

$$Support\ (A \Rightarrow B) = P\ (A \cup B) \qquad (2.2)$$

โดยที่ P(A $\cup$ B) คือ เป็นค่าความน่าจะเป็นที่จะปรากฏไอเทม A และ B พร้อมกันในรายการข้อมูลต่างๆ ของฐานข้อมูล

6. ค่าความเชื่อมั่นน้อยที่สุด (Minimum confidence: min_conf) คือค่าความเชื่อมั่นที่น้อยที่สุดที่ทำให้กฎความสัมพันธ์ที่ได้นั้นยังมีความน่าสนใจ ซึ่งค่าความเชื่อมั่นน้อยที่สุดจะถูกกำหนดโดยผู้ใช้เช่นเดียวกับค่าสนับสนุนน้อยที่สุด ค่าความเชื่อมั่นนี้จะบอกถึงความเข้มแข็งของกฎความสัมพันธ์ที่เกิดขึ้น โดยพิจารณาความน่าจะเป็นของความสัมพันธ์แบบมีเงื่อนไข ที่สามารถคำนวณได้แสดงด้วยสมการที่ 2.3

$$\text{Confidence } (A \Rightarrow B) = P\ (B \mid A) \qquad (2.3)$$

$$P\ (B|A) = \frac{P(A \cap B)}{P(A)}$$

โดยที่ P (B|A) หมายถึง ความน่าจะเป็นที่ B จะเกิดขึ้นเมื่อ A เกิดขึ้นแล้ว

P (A) คือ ความน่าจะเป็นของไอเทม A

7. ฟรีเควนท์ไอเทมเซต (Frequent Itemset : F) หรือ ลาร์จไอเทมเซต (Large Itemset : L) คือชุดของไอเทมเซตโดยไอเทมเซตที่เกิดขึ้นร่วมกันและมีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุด ในฐานข้อมูล โดยจะเรียกฟรีเควนท์ไอเทมเซตตามจำนวนของสมาชิกในเซตที่เกิดขึ้นร่วมกัน เช่น ฟรีเควนท์ 2 ไอเทมเซต (L_2) จะประกอบด้วย ไอเทม 2 ตัวที่เกิดขึ้นร่วมกัน เช่น {milk, bread} เป็นต้น ฟรีเควนท์ 3 ไอเทมเซต (L_3) จะประกอบด้วย ไอเทม 3 ตัวที่เกิดขึ้นร่วมกัน เช่น {milk, bread, meat} เป็นต้นดังนั้นฟรีเควนท์ k-ไอเทมเซต (L_k) จะประกอบด้วย ไอเทม k ตัวที่เกิดขึ้นร่วมกัน โดย k มีจำนวนสมาชิกในเซตเท่ากับ 1, 2,..., n ตัว

8. อินฟรีเควนท์ไอเทมเซต (Infrequent Itemset) หรือสมอลล์ไอเทมเซต (Small Itemset) คือชุดของไอเทมเซต โดยไอเทมเซตที่เกิดขึ้นร่วมกันและมีค่าสนับสนุนน้อยกว่าค่าสนับสนุนน้อยที่สุดในฐานข้อมูล

กฎความสัมพันธ์ที่ได้จากการไมน์นิ่งจะต้องเป็นไปตามค่าที่กำหนดไว้ทั้งค่าสนับสนุนน้อยที่สุดและค่าความเชื่อมั่นน้อยที่สุดจะถูกเรียกว่า กฎที่เข้มแข็ง (Strong) โดยจะเขียนค่าสนับสนุนและค่าความเชื่อมั่นในรูปของเปอร์เซ็นต์มีค่าระหว่าง 0% ถึง 100%

กฎความสัมพันธ์จะหาค่าความสัมพันธ์ระหว่างแอททริบิวต์ที่แตกต่างกัน โดยจะมีการอ้างถึงเซตของไอเทมที่เรียกว่าไอเทมเซตจากข้อมูลทรานแซคชันจะสามารถทราบถึงความสัมพันธ์ที่เกิดขึ้นระหว่างรายการข้อมูลได้ด้วยการนับจำนวนของไอเทมเซตที่ปรากฏในทรานแซคชัน ถ้าไอเทมเซตใดๆ ปรากฏในทรานแซคชันมากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุดที่ผู้ใช้กำหนดจะเรียกว่าฟรีเควนท์ไอเทมเซต และเซตของฟรีเควนท์ k-ไอเทมเซต

2.1.1 ปัญหาของการค้นหากฎความสัมพันธ์ ประกอบด้วย 2 ปัญหาหลักๆ คือ

2.1.1.1 การหาฟรีเควนท์ไอเทมเซตทั้งหมดที่ปรากฏในฐานข้อมูล โดยค่าสนับสนุนของฟรีเควนท์ไอเทมเซตเหล่านี้จะต้องมีค่ามากกว่าหรือเท่ากับค่าสนับสุนนน้อยที่สุดที่ผู้ใช้กำหนด

2.1.1.2 การสร้างกฎความสัมพันธ์ โดยจะนำฟรีเควนท์ไอเทมเซตตั้งแต่ 2 ไอเทมเซตที่ได้จากข้อ 2.1.1.1 มาสร้างเป็นกฎความสัมพันธ์ ในส่วนนี้กฎความสัมพันธ์ที่ได้จะต้องมีค่าความเชื่อมั่นมากกว่าค่าความเชื่อมั่นน้อยที่สุดที่ผู้ใช้กำหนด สำหรับแต่ละฟรีเควนท์ไอเทมเซต I ถ้า $a \subset I \wedge a \neq \varphi$ แล้วจะสร้างกฎความสัมพันธ์ได้ดังนี้

$$a \Rightarrow I - a, \text{if} \quad \frac{p(a \cap I)}{p(a)} \geqslant \text{minimum confidence}$$

2.1.2 งานวิจัยเพื่อค้นหากฎความสัมพันธ์

งานวิจัยที่พัฒนาอัลกอริทึมเพื่อค้นหากฎความสัมพันธ์ ได้ถูกนำเสนอในหลายๆ รูปแบบด้วยกัน คือ

2.1.2.1 อัลกอริทึมอะพริโอริ (Apriori algorithm) โดยอัลกอริทึมของอะพริโอริเป็นการนำเสนอวิธีในการลดจำนวนไอเทมเซตที่จะถูกสแกนในรอบถัดไป ด้วยแนวคิดที่ว่า "ถ้าสับเซต (k-1) ใด ๆ ของแคนดิเดทไอเทมเซต ไม่ได้เป็นสมาชิก L_{k-1} ดังนั้น แคนดิเดทไอเทมเซตนั้นๆ ไม่สามารถเป็นฟรีเควนท์ไอเทมเซตในระดับต่อไปได้ ดังนั้นจะลบแคนดิเดทไอเทมเซตนั้นๆ ออกไป"

อัลกอริทึมอะพริโอริจะทำการค้นหาข้อมูลที่เป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลด้วยการสแกนข้อมูลแต่ละทรานแซคชันในฐานข้อมูลผ่านการวนรอบซ้ำ (Iterations) และในการวนรอบซ้ำแต่ละครั้งจะค้นหาจำนวนสมาชิกของฟรีเควนท์ไอเทมเซตเพิ่มขึ้นทีละ 1 ระดับ คือมีการค้นหาจำนวนไอเทมเซตเพิ่มทีละ 1 ตัว เช่น จากสมาชิก 2 ตัว รอบหรือระดับถัดไปจะค้นหาที่จำนวนสมาชิก 3 ตัวเป็นต้น โดยอาศัยความรู้ที่ได้จากขั้นตอนก่อนหน้าเช่น แคนดิเดท k - ไอเทมเซต (C_k) จะได้จากการสร้างฟรีเควนท์ (k-1) ไอเทมเซต (L_{k-1}) การค้นหาฟรีเควนท์ไอเทมเซตที่มีการเพิ่มสมาชิกไปในแต่ละระดับนี้เรียกว่า การค้นหาแบบทีละระดับ (Levelwise search) ซึ่งประกอบด้วยขั้นตอนหลักๆ 2 ขั้นตอนคือ

1. ขั้นตอนการเชื่อม (Join step)

เพื่อหาฟรีเควนท์ไอเทมเซต (L_k) การสร้างและนับแคนดิเดทไอเทมเซต (C_k) โดยอัลกอริทึมของอะพริโอริจะต้องมีการเรียงลำดับตัวอักษรของข้อมูลในทรานแซคชันจากน้อยไปมาก (lexicographic order) เพื่อใช้ในการสร้างเซตของแคนดิเดทไอเทมเซต ที่มีค่าสนับสนุนมากกว่าศูนย์

และแคนดิเดทไอเทมเซต C_k จะสามารถหาได้จากฟรีเควนท์ L_{k-1} ในกรณีที่ไม่ใช่ 1-ไอเทมเซต จะสามารถหา C_k ได้จากการนำ L_{k-1} มาเชื่อมกันโดยไอเทมเซตที่สามารถนำมาเชื่อมกันได้นั้นจะต้องมีสมาชิกทุกตัวก่อนตัวสุดท้ายเหมือนกัน เช่น

- กรณีหาแคนดิเดท 3-ไอเทมเซต จากฟรีเควนท์ไอเทมเซต AB และ AC สามารถเชื่อมกันได้เป็น ABC
- กรณีหาแคนดิเดท 4-ไอเทมเซต จากฟรีเควนท์ไอเทมเซต ABC และ ABD สามารถเชื่อมกันได้เป็น ABCD เป็นต้น

2. ขั้นตอนการตัด (Prune step)

เป็นขั้นตอนที่นำ C_k มาแตกเป็นสับเซตย่อยที่ประกอบด้วยสมาชิก k-1 ตัวแล้วพิจารณาสับเซตที่แตกย่อยมานั้นว่าทุกตัวต้องเป็นฟรีเควนท์ L_{k-1} ไอเทมเซต ถ้าพบว่ามีตัวใดตัวหนึ่งของสับเซตย่อยไม่เป็นฟรีเควนท์ L_{k-1} ไอเทมเซตจะทำการลบแคนดิเดทไอเทมเซตนั้นๆ ออกไป เนื่องจากไอเทมเซตนั้นๆ จะไม่สามารถกลายมาเป็นฟรีเควนท์ไอเทมเซตได้ เป็นการลดจำนวนไอเทมเซตที่ต้องไปสแกนในฐานข้อมูล และนี่เป็นคุณสมบัติหนึ่งของอัลกอริทึมอะพริโอริที่ได้กำหนดไว้ดังนี้คือ

"ถ้าสับเซต (k-1) ใดๆ ของแคนดิเดทไอเทมเซต ไม่ได้เป็นสมาชิกฟรีเควนท์ L_{k-1} ดังนั้น แคนดิเดทไอเทมเซตนั้นๆ ไม่สามารถเป็นฟรีเควนท์ไอเทมเซตในระดับต่อไปได้ ดังนั้นจะลบแคนดิเดทไอเทมเซตนั้นๆ ออกไป" ตัวอย่างเช่น

L_1	C_2	L_2	C_3
A	AB	AB	~~ABD~~
B	AD	AD	ABE
D	AE	AE	ADE
E	BD	BE	
	BE	DE	
	DE		

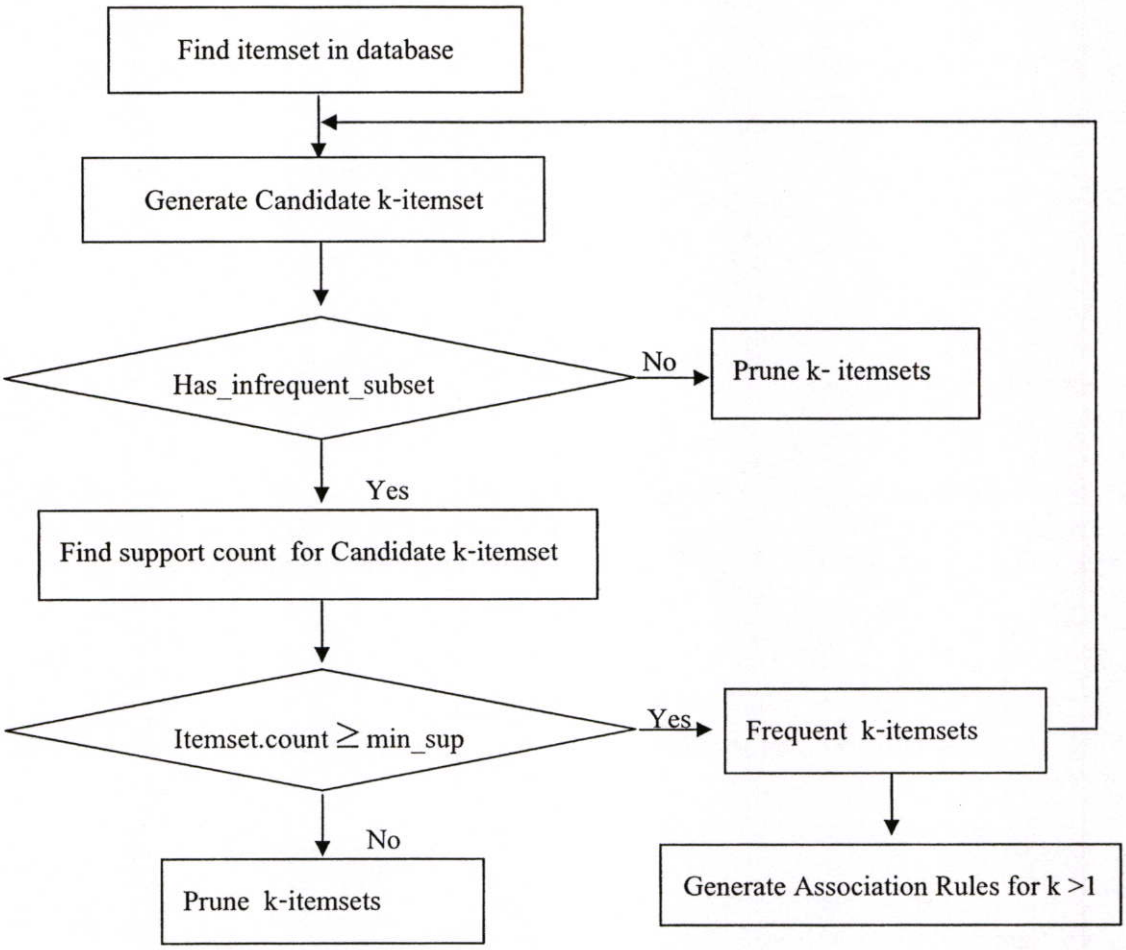
AB, AD, BD =>**Prune**

AB, AE, BE

AD, AE, DE

รูปที่ 2.1 แสดงขั้นตอนการตัดแคนดิเดทไอเทมเซต

การค้นหากฎความสัมพันธ์มีกระบวนการทำงาน 2 ขั้นตอนคือ หาฟรีเควนท์ไอเทมเซตทั้งหมด และสร้างกฎความสัมพันธ์จากฟรีเควนท์ไอเทมเซต แสดงดังรูปที่ 2.2



รูปที่ 2.2 ขั้นตอนการค้นหากฎความสัมพันธ์ของอัลกอริทึมอะพริโอริ

2.1.2.2 อัลกอริทึมพาร์ทิชัน (partition algorithm) เป็นเทคนิคหนึ่งในการค้นหากฎความสัมพันธ์โดยมีหลักการในการหาฟรีเควนท์ไอเทมเซตเช่นเดียวกับอัลกอริทึมอะพิโอริ แต่จะแบ่งฐานข้อมูลเป็นส่วนย่อยๆ เรียกว่า พาร์ทิชัน จากนั้นในช่วงเวลาหนึ่งๆ จะทำการสร้างแคนดิเดท k-ไอเทมเซตและค้นหาฟรีเควนท์ k-ไอเทมเซตจากแต่ละพาร์ทิชันในหน่วยความจำหลัก (main memory) สำหรับไอเทมเซตที่จะกลายมาเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลได้นั้น จะต้องมีคุณสมบัติดังนี้

"ไอเทม X จะกลายเป็นฟรีเควนท์ไอเทมเซตของฐานข้อมูลได้ก็ต่อเมื่อไอเทม X จะต้องเป็นฟรีเควนท์ไอเทมเซตอย่างน้อยในพาร์ทิชันใดพาร์ทิชันหนึ่ง"

กระบวนการของอัลกอริทึมพาร์ทิชันนั้นจะทำการแบ่งฐานข้อมูล เป็น n พาร์ทิชัน (partitions) จากนั้นในช่วงเวลาหนึ่งจะทำกระบวนการไมน์นิ่งแต่ละพาร์ทิชัน โดยมีขั้นตอนดังนี้

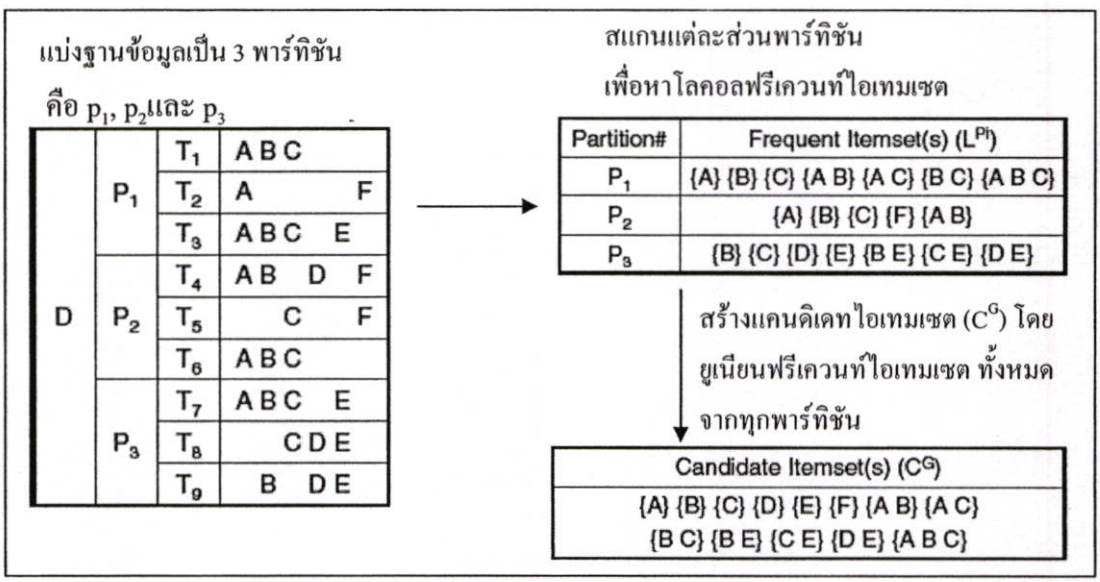
1. การสแกนพาร์ทิชันตั้งแต่พาร์ทิชันที่ 1 ถึงพาร์ทิชันที่ n (p_i , i =1..n) เพื่อหาฟรีเควนท์ไอเทมเซตทั้งหมดในแต่ละพาร์ทิชัน เรียกฟรีเควนท์ไอเทมเซตในแต่ละพาร์ทิชันนี้ว่า

โลคอลฟรีเควนท์ไอเทมเซต (local frequent itemsets) และเขียนแทนด้วย L^{pi} โดย pi แทนหมายเลขของพาร์ทิชัน

2. นำ L^{pi} สำหรับของทุกพาร์ทิชันมายูเนียน (union) เป็นแคนดิเดทไอเทมเซต (CG) ของฐานข้อมูลทั้งหมด (global candidate itemset) โดย CG นี้จะเป็นซูเปอร์เซตของฟรีเควนท์ไอเทมเซตทั้งหมดในฐานข้อมูล

3. สแกนพาร์ทิชันทั้งหมดเป็นรอบที่ 2 เพื่อหาค่าสนับสนุนให้กับแต่ละไอเทมเซตใน CG ซึ่งจะได้ CG ที่เป็นฟรีเควนท์ไอเทมเซตจริงที่ปรากฏในฐานข้อมูล

รูปที่ 2.3 แสดงตัวอย่างของอัลกอริทึมพาร์ทิชัน โดยมีการแบ่งฐานข้อมูลเป็น 3 พาร์ทิชัน ประกอบด้วย p_1, p_2 และ p_3 แต่ละพาร์ทิชันประกอบด้วย 3 ทรานแซคชัน เซตของ โลคอลฟรีเควนท์ไอเทมเซต (Local frequent itemset) สำหรับแต่ละพาร์ทิชัน แทนด้วย L_{pi} โดย pi แทนหมายเลขของพาร์ทิชัน เช่น L_{p2} ประกอบด้วย เซตดังนี้ {{A}, {B}, {C}, {F}, {A,B}} เป็นต้น เซตของโกบอลแคนดิเดทไอเทมเซต (Global candidate itemset) จะถูกนำมาพิจารณาเพื่อหาฟรีเควนท์ไอเทมเซตที่แท้จริงด้วยการสแกนฐานข้อมูลทั้งหมด



รูปที่ 2.3 แสดงตัวอย่างการสร้างโกบอลแคนดิเดทไอเทมซตของอัลกอริทึมพาร์ทิชัน

2.1.2.3 อัลกอริทึมแพทเทรินโกร์ท (Pattern Growth algorithm) แนวคิดหลักคือการนำโครงสร้างรูปต้นไม้ (tree) มาใช้เก็บสารสนเทศของฟรีเควนท์ไอเทมเซต โดยงานวิจัย[5] ได้นำเสนออัลกอริทึมทรีโพรเจคชัน (Treeprojection) ซึ่งมีการสร้างรูปต้นไม้แบบเรียงลำดับตามตัวอักษรจากน้อยไปมากและแต่ละโหนดของรูปต้นไม้ใช้แสดงถึงฟรีเควนท์ไอเทมเซตที่ได้จากการไมน์นิ่งฐานข้อมูลทั้งหมด โดยทรานแซคชันโพรเจคชัน(transaction projection) จะสามารถช่วย

ในการจัดการแคนดิเดทไอเทมเซตได้ อัลกอริทึมทรีโพรเจคชันได้ถูกนำไปปรับปรุงให้มีประสิทธิภาพอย่างกว้างขวางเพื่อใช้ในการไมน์นิ่งกฎความสัมพันธ์

นอกจากนี้งานวิจัยที่มีบทบาทอย่างมากคืออัลกอริทึมเอฟพี-โกรว์ธ (Frequent pattern growth) [6] ซึ่งแนวคิดหลักของงานวิจัยคือการหลีกเลี่ยงการสร้างแคนดิเดทไอเทมเซตโดยใช้เทคนิคการแบ่งเพื่อเอาชนะ (divide and conquer) ซึ่งมีลักษณะการแตกงานให้เป็นส่วนเล็กๆ สำหรับไมน์นิ่งแพทเทรินในฐานข้อมูลแนวคิดนี้จะช่วยลดพื้นที่ในการค้นหาฟรีเควนท์แพทเทริน ขั้นตอนในการค้นหาฟรีเควนท์ไอเทมเซต ดังนี้

1. สแกนฐานข้อมูลทั้งหมดหนึ่งรอบ เพื่อหาฟรีเควนท์ 1-ไอเทม
2. เรียงฟรีเควนท์ไอเทมเซตจากมากไปน้อยตามค่าสนับสนุนของฟรีเควนท์ไอเทมเซต
3. สแกนฐานข้อมูลอีกครั้ง เพื่อสร้างเอฟพี-ทรี

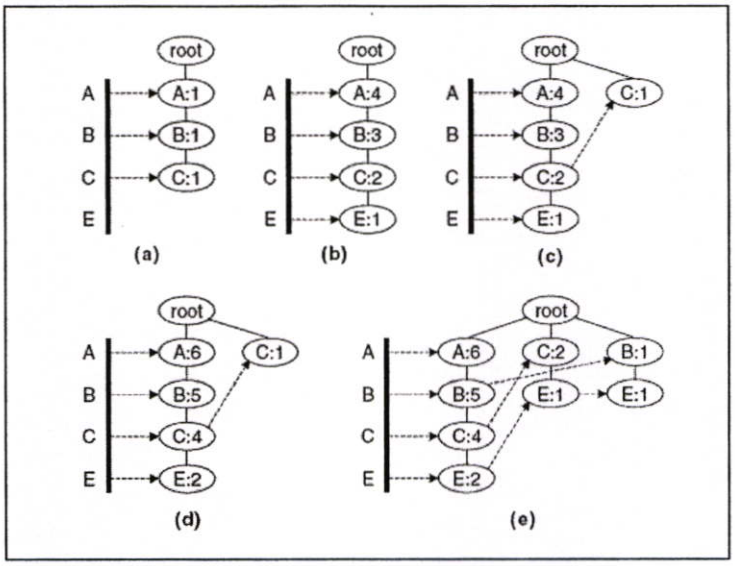
D	T_1	A B C
	T_2	A F
	T_3	A B C E
	T_4	A B D F
	T_5	C F
	T_6	A B C
	T_7	A B C E
	T_8	C D E
	T_9	B D E

รูปที่ 2. 4 แสดงตัวอย่างของฐานข้อมูลเดิม

จากรูปที่ 2. 4 แสดงตัวอย่างฐานข้อมูลที่ประกอบด้วย 9 ทรานแซคชัน เอฟพีทรีจะเริ่มจากการสแกนฐานข้อมูลทั้งหมดรอบที่ 1 เพื่อค้นหาฟรีเควนท์ 1- ไอเทมเซต เมื่อได้ฟรีเควนท์ 1-ไอเทมเซตแล้วจะนำฟรีเควนท์ 1- ไอเทมเซตที่ได้มาจัดเรียงลำดับตามค่าสนับสนุนมากไปน้อย คือ {A:6}, {B:6}, {C:4} และ {E:4}

จากนั้นจะสแกนฐานข้อมูลอีก 1 รอบเพื่อสร้างเอฟพีทรี โดยเริ่มจากทรานแซคชัน T_1 ที่ประกอบด้วย {A, B, C} จะถูกนำไปสร้างเป็น กิ่งเดี่ยวของเอฟพี-ทรีแสดงในรูปที่ 2.5 (a) ในรูปที่ 2.5 (b) เมื่อ 3 ทรานแซคชันถัดไปคือ T_2-T_4 ถูกเพิ่มเข้าไปจะทำการขยายกิ่งและเพิ่มค่าสนับสนุนให้กับโหนดของไอเทมที่ปรากฏอยู่แล้ว สำหรับกิ่งของไอเทมที่ไม่เป็นฟรีเควนท์จะถูกยกเลิกไป เช่น ไอเทม D และ F เป็นต้น อย่างไรก็ตามทรานแซคชัน T_5 จะประกอบด้วยไอเทม C เพียงอย่างเดียวที่เป็นฟรีเควนท์ไอเทม ดังในภาพที่ 2.5 (c) ฟรีเควนท์ไอเทมเซต ทั้งหมดที่เกิดขึ้นจะมีการเชื่อมโยงโหนด สำหรับแต่ละไอเทมรูปที่ 2.5 (d) ทรานแซคชัน T_6 และ T_7 จะถูกเพิ่มเข้าไปเช่นเดียวกับ $T_2 - T_4$ สุดท้ายเมื่อมีการเพิ่มทรานแซคชัน T_8 และ T_9 เข้ามาจะได้เอฟพี-ทรีที่สมบูรณ์ดังแสดงในรูปที่ 2.5 (e)

ซึ่งจะสังเกตได้ว่าไอเทมที่มีแนวโน้มการเกิดขึ้นร่วมกันจะปรากฏในเอฟพี-ทรี และผลลัพธ์ที่ได้จากเอฟพี-ทรีจะมีอัตราการบีบอัดข้อมูลสูงมาก (data compression ratio)



รูปที่ 2.5 แสดงการสร้างเอฟพี-ทรีจากฐานข้อมูล

จากอัลกอริทึมสำหรับการค้นหากฎความสัมพันธ์ที่ได้กล่าวมาข้างต้น อัลกอริทึมที่ได้รับความนิยมคือ อัลกอริทึมอะพิโอริ [2] ซึ่งเป็นการค้นหาฟรีเควนท์ไอเทมเซตจากฐานข้อมูลทั้งหมดโดยนำฟรีเควนท์ไอเทมเซตที่ค้นหาได้ในระดับก่อนหน้า (k-1) มาใช้ในการสร้างแคนดิเดทไอเทมและภายหลังจากนำแคนดิเดทไอเทมเซตที่ได้ไปสแกนในฐานข้อมูลจะได้ฟรีเควนท์ไอเทมเซตในระดับถัดไป

2. 2 ปัญหาของการเพิ่มขยายการค้นหากฎความสัมพันธ์

ข้อมูลในยุคปัจจุบันนั้นมักมีการจัดเก็บในฐานข้อมูล และมีแนวโน้มที่ข้อมูลที่ถูกจัดเก็บนั้นจะสามารถถูกปรับปรุงด้วยการเพิ่มจำนวนรายการข้อมูลเข้าไปในฐานข้อมูล ซึ่งมีผลทำให้ฐานข้อมูลมีการเปลี่ยนแปลงตลอดเวลา

การค้นหากฎความสัมพันธ์จากฐานข้อมูลเมื่อฐานข้อมูลมีการเปลี่ยนแปลงจากการเพิ่มรายการข้อมูลในฐานข้อมูลล้วนแต่มีผลต่อการเปลี่ยนแปลงกฎความสัมพันธ์ที่ได้เคยไมน์นิ่งไว้ เนื่องจากกฎความสัมพันธ์ที่ได้สามารถนำไปใช้ในการทำนายความสัมพันธ์ของข้อมูลที่เกิดขึ้นร่วมกัน ซึ่งสามารถนำข้อมูลที่ได้จากการไมน์นิ่งไปใช้ในการวางแผนกลยุทธ์และช่วยในการตัดสินใจได้ตามแนวโน้มใหม่ที่เกิดขึ้นได้ กล่าวคือกฎความสัมพันธ์ที่ได้จากการไมน์นิ่งฐานข้อมูลใหม่ทั้งหมดที่ผ่านการปรับปรุงแล้วอาจทำให้กฎความสัมพันธ์ที่ได้จากการไมน์นิ่งฐานข้อมูลเดิมไม่น่าสนใจอีกต่อไป ในขณะเดียวกันอาจพบว่ามีรายการใหม่เกิดขึ้นและกลายเป็นกฎความสนใจใหม่ที่ได้รับ

ความสนใจเพิ่มขึ้นมา เช่น ในกรณีที่มีสินค้าหรือผลิตภัณฑ์ใหม่ๆ เกิดขึ้น และได้รับความนิยมจากลูกค้าทำให้หน่วยงานต้องปรับแผนกลยุทธ์ให้สอดคล้องกับความต้องการที่เปลี่ยนไป เป็นต้น

ด้วยลักษณะของการค้นหากฎความสัมพันธ์นั้นจะทำการค้นหาในฐานข้อมูลที่อยู่ในรูปแบบออฟไลน์ (Off-line) ดังนั้นในงานวิจัยนี้ได้ทำการศึกษาเฉพาะรูปแบบของการเพิ่มข้อมูลใหม่เข้ามาในฐานข้อมูลเท่านั้น เนื่องจากถ้าข้อมูลที่เปลี่ยนแปลงมีทั้งการเพิ่มหรือลบเฉพาะส่วนของข้อมูลใหม่แล้วข้อมูลในส่วนดังกล่าวจะถูกจัดให้อยู่ในรูปของข้อมูลใหม่ที่เพิ่มเข้ามาเท่านั้น

เมื่อฐานข้อมูลมีการเปลี่ยนแปลงเกิดขึ้นทำให้ต้องทำการค้นหากฎความสัมพันธ์ใหม่จากฐานข้อมูลที่มีการเปลี่ยนแปลงนี้ เนื่องจากข้อมูลในฐานข้อมูลมีจำนวนมากการค้นหากฎความสัมพันธ์การสแกนฐานข้อมูลแต่ละรอบเพื่อค้นหาฟรีเควนท์ k-ไอเทมเซตนั้นทำให้ต้องเสียเวลาในการสร้างแคนดิเดทไอเทมเซตเพื่อค้นหาฟรีเควนท์ไอเทมเซตโดยจะต้องสแกนฐานข้อมูลจำนวนหลายรอบโดยไม่ได้นำฟรีเควนท์ไอเทมเซตซึ่งเป็นความรู้ที่ได้จากการไมน์นิ่งฐานข้อมูลเดิมมาใช้ให้เกิดประโยชน์

ถ้ากำหนดให้ L แทนฟรีเควนท์ไอเทมเซตทั้งหมดจากการไมน์นิ่งฐานข้อมูลเดิมที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุด ซึ่งอัลกอริทึมที่ใช้ในการค้นหากฎความสัมพันธ์ทั่วไปจะมีการจัดเก็บค่าสนับสนุนไว้ของฟรีเควนท์ไอเทมเซตไว้เท่านั้น ดังนั้นหลังจากมีการปรับปรุงฐานข้อมูลเดิมด้วยข้อมูลใหม่ที่เพิ่มเข้ามาแล้วด้วยเทคนิคการแบ่งเพื่อเอาชนะ (divide and conquer) ค่าสนับสนุนน้อยที่สุดเดียวกัน สามารถจัดกลุ่มไอเทมเซตที่ได้จากการปรับปรุงได้เป็น 4 กรณีดังนี้ [Tsai et al. 1999] [7]

กรณีที่ 1 ไอเทม X เป็นฟรีเควนท์ไอเทมเซตทั้งในฐานข้อมูลเดิมและฐานข้อมูลปรับปรุง

กรณีที่ 2 ไอเทม X เป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิมแต่ไม่เป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุง

กรณีที่ 3 ไอเทม X ไม่เป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิมแต่เป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุง

กรณีที่ 4 ไอเทม X ไม่เป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิมและฐานข้อมูลปรับปรุง

ตารางที่ 2.1 แสดงกรณีต่างๆ ของไอเทมเซตที่ปรากฏภายหลังจากฐานข้อมูลเดิมถูกปรับปรุงเมื่อมีการเพิ่มข้อมูลใหม่เข้ามา ซึ่งจะพบว่าไอเทมเซตที่จะกลายมาเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงได้จะต้องเป็นฟรีเควนท์ไอเทมเซตอย่างน้อยในฐานข้อมูลเดิมหรือต้องเป็นฟรีเควนท์ไอเทมเซตในส่วนของข้อมูลที่เพิ่มเข้ามา ดังนั้นกรณีที่ 4 ซึ่งพบว่าไอเทมเซตไม่เป็นฟรีเควนท์ไอเทมเซตทั้งในฐานข้อมูลเดิมและฐานข้อมูลใหม่ที่เพิ่มเข้ามา จึงไม่สามารถกลายมาเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงได้ ส่วนกรณีที่ 1 และ 2 เนื่องจากทั้ง 2 กรณีเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิมทำให้ทราบค่าสนับสนุนของไอเทมเซต และสามารถทำการค้นหาค่าสนับสนุนที่เกิดขึ้นในส่วนของข้อมูลใหม่ที่ปรับปรุงได้ แต่ในกรณีที่ 3 เนื่องจากไม่มีการ

จัดเก็บค่าสนับสนุนของไอเทมเซตที่ไม่เป็นฟรีเควนท์ไอเทมเซตไว้ ดังนั้น หากต้องการทราบค่าสนับสนุนของไอเทมเซตในกรณีที่ 3 ซึ่งอาจกลายมาเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงได้จะต้องทำการสแกนฐานข้อมูลเดิมเพื่อหาค่าสนับสนุนที่เกิดขึ้นจริง

ตารางที่ 2.1 แสดงกรณีของไอเทมเซตที่ปรากฏเมื่อฐานข้อมูลมีการเปลี่ยนแปลง

	ฟรีเควนท์ไอเทมเซต ในฐานข้อมูลปรับปรุง	ไม่เป็นฟรีเควนท์ไอเทมเซต ในฐานข้อมูลปรับปรุง
ฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิม	กรณีที่ 1	กรณีที่ 2
ไม่เป็นฟรีเควนท์ไอเทมเซตใน ฐานข้อมูลเดิม	กรณีที่ 3	กรณีที่ 4

ดังนั้นในหลายๆ งานวิจัยได้นำเสนออัลกอริทึมเพื่อปรับปรุงประสิทธิภาพของการเพิ่มขยายการค้นหากฎความสัมพันธ์เมื่อฐานข้อมูลเดิมมีการเปลี่ยนแปลงดังจะกล่าวในหัวข้อต่อไป

2.3 งานวิจัยสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์

ฐานข้อมูลเป็นแหล่งที่ใช้ในการจัดเก็บข้อมูลที่ประกอบด้วยข้อมูลต่างๆ มากมาย โดยทั่วไปการค้นหาความสัมพันธ์ระหว่างข้อมูลในฐานข้อมูลอาจพบความรู้ที่ซ่อนอยู่ภายในฐานข้อมูล ซึ่งสามารถนำความรู้ที่ได้มาสร้างให้อยู่ในรูปของกฎความสัมพันธ์

ข้อมูลที่ถูกจัดเก็บในฐานข้อมูลสามารถเกิดการเปลี่ยนแปลงได้ตลอดเวลาหรืออาจกล่าวได้ว่าข้อมูลในฐานข้อมูลนั้นไม่คงที่ เรียกฐานข้อมูลในลักษณะนี้ว่า ฐานข้อมูลไดนามิก (Dynamic database) การเปลี่ยนแปลงฐานข้อมูลอาจเกิดจากการเพิ่มรายการข้อมูลเข้าไปในฐานข้อมูล (Insert), ลบรายการข้อมูลที่มีอยู่ในฐานข้อมูล (Delete) หรืออาจมีทั้งการเพิ่มและลบรายการข้อมูลในฐานข้อมูล (Modify) โดยในที่นี้จะเรียกส่วนของฐานข้อมูลก่อนทำการเปลี่ยนแปลงว่าฐานข้อมูลเดิม (Original database : DB) และเรียกส่วนของข้อมูลใหม่ว่าฐานข้อมูลใหม่ที่เพิ่มเข้ามา (Increment database : db) และฐานข้อมูลที่ผ่านการเปลี่ยนแปลงฐานข้อมูลเดิมด้วยฐานข้อมูลใหม่แล้วจะเรียกว่า ฐานข้อมูลปรับปรุง (Updated database : UP)

เมื่อฐานข้อมูลมีการเปลี่ยนแปลงจะมีผลต่อกฎความสัมพันธ์ที่ได้ไมน์นิ่งไว้ในฐานข้อมูลเดิมเนื่องจากฟรีเควนท์ไอเทมเซตที่ได้ทำการค้นหาจากฐานข้อมูลเดิม อาจไม่สามารถเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลที่ปรับปรุง ในขณะที่สมอลล์ไอเทมเซตในฐานข้อมูลเดิมอาจกลายมาเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุง ทำให้กฎความสัมพันธ์ที่มีอยู่เดิมไม่เหมาะสมที่จะนำมาใช้ในการวางแผนกลยุทธ์อีกต่อไป เนื่องจากกฎความสัมพันธ์ที่พบในฐานข้อมูลปรับปรุงจะแสดงถึงความต้องการของลูกค้าเปลี่ยนไปจากกฎความสัมพันธ์ใหม่ที่เกิดขึ้นได้

งานวิจัยสำหรับค้นหากฎความสัมพันธ์ใหม่ในฐานข้อมูลปรับปรุงในระยะแรก เช่น อัลกอริทึมอะพริโอรินั้นจะเป็นการค้นหากฎความสัมพันธ์ระหว่างข้อมูลในฐานข้อมูลเดิมรวมกับฐานข้อมูลใหม่ที่เพิ่มเข้ามาเพื่อให้ได้กฎความสัมพันธ์ทั้งหมดที่เกิดขึ้นในฐานข้อมูลปรับปรุง โดยไม่ได้นำส่วนของฟรีเควนท์ k-ไอเทมเซตที่ได้จากการไมน์นิ่งฐานข้อมูลเดิมมาใช้ให้เกิดประโยชน์ ดังนั้นในหลายงานวิจัยต่อมาได้มีการพัฒนาอัลกอริทึมเพื่อให้สามารถทำการค้นหากฎความสัมพันธ์ที่ได้อย่างมีประสิทธิภาพ ในหลายๆ งานวิจัยได้นำเสนอรูปแบบอัลกอริทึมเพื่อใช้ในการเพิ่มขยายการค้นหากฎความสัมพันธ์ในฐานข้อมูลปรับปรุง โดยทั่วไปแล้วฐานข้อมูลเดิมมักจะมีขนาดใหญ่กว่าส่วนของข้อมูลที่เพิ่มเข้ามา ดังนั้นงานวิจัยส่วนใหญ่จะนำเสนอวิธีเพื่อลดการสแกนฐานข้อมูลเดิมและสามารถค้นหากฎความสัมพันธ์ใหม่ที่เกิดขึ้นอย่างมีประสิทธิภาพดังงานวิจัย [6]-[37]

แนวคิดของงานวิจัยที่นำเสนอในด้านการเพิ่มขยายการค้นหากฎความสัมพันธ์สามารถแบ่งได้เป็น 2 รูปแบบ คือ

2.3.1 การเพิ่มขยายค้นหากฎความสัมพันธ์ที่ให้ความสำคัญกับข้อมูลใหม่ที่เพิ่มเข้ามา

เนื่องจากมองว่าข้อมูลใหม่ที่เกิดขึ้นจะบ่งบอกแนวโน้มของความต้องการของลูกค้าในช่วงเวลาหนึ่งๆ เช่น สินค้าหรือผลิตภัณฑ์ใหม่ๆ เกิดขึ้นตามกระแสความนิยมของภาพยนตร์ที่ออกฉายในเวลาหนึ่ง เมื่อมีภาพยนตร์ใหม่ออกมาความต้องการสินค้าที่เกี่ยวข้องกับภาพยนตร์เรื่องใหม่อาจไม่สามารถกลายมาเป็นฟรีเควนท์ไอเทมเซตได้เนื่องจากจำนวนข้อมูลใหม่ที่เพิ่มเข้ามาจะมีขนาดน้อยกว่าฐานข้อมูลเดิม แต่เมื่อพิจารณาจำนวนข้อมูลใหม่ที่เกิดขึ้นนั้นอาจบ่งบอกถึงแนวโน้มความต้องการของสินค้าหรือผลิตภัณฑ์ที่ลูกค้าให้ความสนใจในช่วงเวลานั้นหรืออาจมีมากขึ้นในอนาคต ดังนั้นข้อมูลในลักษณะนี้จะให้ความสำคัญกับฟรีเควนท์ไอเทมเซตที่เกิดขึ้นในส่วนของข้อมูลใหม่ที่เพิ่มเข้ามา ซึ่งฟรีเควนท์ไอเทมเซตนี้อาจไม่เป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิมและเพื่อหลีกเลี่ยงการสแกนข้อมูลในฐานข้อมูลเดิม แนวคิดนี้จะนำเสนอวิธีการต่างๆ เพื่อใช้ในการประมาณค่าที่คาดว่าจะเกิดขึ้นในฐานข้อมูลเดิมให้กับฟรีเควนท์ไอเทมเซตใหม่ที่เกิดขึ้น

งานวิจัยที่นำเสนอเทคนิคสำหรับการเพิ่มขยายกฎความสัมพันธ์สำหรับข้อมูลที่ให้ความ สำคัญกับข้อมูลใหม่ ได้แก่

2.3.1.1 โมเดลการให้น้ำหนัก (Weight model)

เป็นงานวิจัยที่นำเสนอโดย Zhang et al.[8] แนวคิดของงานวิจัยนี้จะให้ความสำคัญกับข้อมูลใหม่ที่เพิ่มเข้ามาในฐานข้อมูลเดิมด้วยเทคนิคของการคำนวณน้ำหนัก (weight) ให้กับข้อมูลสูงกว่าข้อมูลที่ปรากฏในฐานข้อมูลเดิม นอกจากนี้จะสามารถหลีกเลี่ยงการสแกนฐานข้อมูลใหม่ทั้งหมดได้แล้วยังทำให้พบกฎความสัมพันธ์ของสินค้าใหม่ที่น่าสนใจ ในส่วนของข้อมูลที่เพิ่มเข้ามาเนื่องจากข้อมูลใหม่ที่เพิ่มเข้ามาอาจแสดงถึงแนวโน้มการเปลี่ยนแปลงรูปแบบการซื้อสินค้าหรือความสนใจสินค้าของลูกค้า แต่สินค้าใหม่ที่ลูกค้าให้ความสนใจเหล่านี้อาจมีจำนวนไม่มากพอที่กลาย

มาเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุง ดังนั้นอัลกอริทึมสำหรับค้นหากฎความสัมพันธ์ของฐานข้อมูลส่วนใหญ่ (current mining algorithm) อาจไม่สามารถดึงกฎความสัมพันธ์ที่เกิดขึ้นในส่วนของฐานข้อมูลใหม่ออกมาได้ เนื่องจาก อัลกอริทึมจะให้ความสำคัญกับทุกทรานแซคชันเท่ากัน

การทำงานของอัลกอริทึมนี้นำเสนอวิธีการที่เรียกว่า เซตคู่แข่ง (Competitive set) โดยจะเริ่มจากการค้นหากฎความสัมพันธ์ในฐานข้อมูลเดิม ซึ่งได้จากการหาฟรีเควนท์ไอเทมเซต นอกจากนี้ยังมีการหาไอเทมเซตที่มีค่า น้ำหนักที่คำนวณได้น้อยกว่าค่าสนับสนุนน้อยที่สุดแต่อาจจะสามารถกลายมาเป็นฟรีเควนท์ไอเทมเซตได้เมื่อมีข้อมูลใหม่เพิ่มเข้ามาโดยไอเทมเซตนี้จะถูกจัดเก็บ ในส่วนที่เรียกว่าเซตคู่แข่ง ไอเทมเซตที่จะเก็บในเซตคู่แข่งได้จะต้องมีค่าน้ำหนักที่คำนวณได้มากกว่าหรือเท่ากับค่าที่กำหนดเพิ่มขึ้นมาอีก 1 ค่าคือ minimum crucial

เมื่อมีข้อมูลใหม่เพิ่มเข้ามาในฐานข้อมูลเดิมจะนำโมเดลการให้น้ำหนักมาใช้เพื่อทำการปรับปรุงค่าสนับสนุนและค่าความเชื่อมั่นให้กับกฎความสัมพันธ์และเซตคู่แข่งในฐานข้อมูลเดิม ดังนั้นกฎความสัมพันธ์ที่ได้จากการไมน์นิ่งเมื่อทำการปรับปรุงฐานข้อมูลของโมเดลนี้จะแตกต่างจากอัลกอริทึมอะพริโอริ และเอฟยูพีคือ นอกจากจะได้กฎความสัมพันธ์ที่แท้จริงแล้วยังมีการพิจารณากฎความสัมพันธ์ที่อยู่ในเซตของเซตคู่แข่งอีกด้วย

ข้อดี

1. ช่วยลดการสแกนฐานข้อมูลเดิม

ข้อเสีย

1. มีการจัดเก็บไอเทมเซตทั้งส่วนที่เป็นฟรีเควนท์ไอเทมเซต และไอเทมที่ไม่ใช่ฟรีเควนท์ที่อาจกลายมาเป็นฟรีเควนท์ไอเทมเซต

2. การให้ความสำคัญกับข้อมูลใหม่ที่เกิดขึ้นมากกว่าข้อมูลในฐานข้อมูลเดิมทำให้พบจำนวนกฎความสัมพันธ์ที่ได้จากการค้นหามากกว่าอัลกอริทึมอื่น เช่น อะพริโอริ, เอฟยูเอฟ เป็นต้น

3. เนื่องจากการปรับปรุงกฎจะมีจำนวนมากกว่าฟรีเควนท์ไอเทมเซต ทำให้เสียเวลาในการปรับปรุงมาก

4. อัลกอริทึมที่มีการปรับปรุงกฎความสัมพันธ์ที่มีทั้งในส่วนที่พบในฐานข้อมูลเดิมและฐานข้อมูลใหม่ การปรับปรุงกฎความสัมพันธ์ซึ่งมีจำนวนมากทำให้ต้องใช้เวลานานมากกว่าการปรับปรุงฟรีเควนท์ไอเทมเซตมีดังนี้

2.3.1.2 การบำรุงรักษากฎ (Rule maintenance)

Dudek and Zgrzywa [9] ได้นำเสนออัลกอริทึมการบำรุงรักษากฎสำหรับการเพิ่มขยายกฎความสัมพันธ์ด้วยการปรับปรุงเฉพาะกฎความสัมพันธ์ใหม่ที่เกิดขึ้นจากการเพิ่มข้อมูลเข้า

ไปในฐานข้อมูลเดิม ซึ่งกฎความสัมพันธ์ใหม่นี้ได้มาจากกระบวนการไมน์นิ่งส่วนของข้อมูลใหม่ที่เพิ่มเข้ามาด้วยอัลกอริทึมใดๆ ที่ใช้ในการค้นหากฎความสัมพันธ์ เช่น อัลกอริทึมอะพริโอริ เป็นต้น

โดยทั่วไปกฎความสัมพันธ์ใหม่ที่ปรากฏขึ้นในกระบวนการไมน์นิ่งส่วนของข้อมูลใหม่ที่เพิ่มเข้ามาจะมีโอกาสน้อยมากที่จะกลายมาเป็นกฎความสัมพันธ์ที่ถูกเพิ่มเข้าไปในกฎพื้นฐาน (rule base) จริง ดังนั้นอัลกอริทึมนี้มีแนวคิดว่าเมื่อเวลาเปลี่ยนแปลงไปรูปแบบความสัมพันธ์ใหม่ที่เกิดขึ้นจะมีความน่าสนใจมากกว่ารูปแบบความสัมพันธ์เดิม ดังนั้นได้มีการนำเสนอฟังก์ชันที่เรียกว่า time influence function (f_T) ที่ ซึ่งสำหรับใช้ในการคำนวณค่าประมาณการของค่าสนับสนุน และ ความเชื่อมั่น ให้กับกฎความสัมพันธ์ใหม่ที่ค้นพบในส่วนของข้อมูลใหม่ที่เพิ่มเข้ามา ทั้งนี้เนื่องจากงานวิจัยนี้ให้ความสำคัญกับทรานแซคชันใหม่ที่เพิ่มเข้าในฐานข้อมูลมากกว่าทรานแซคชันเดิม แต่อย่างไรก็ดีแนวทางนี้ไม่ได้นำเสนอวิธีที่ใช้ในการประมาณค่าสนับสนุนและค่าความเชื่อมั่นให้กับกฎความสัมพันธ์ใหม่ที่เกิดขึ้น

ขั้นตอนในการค้นหากฎความสัมพันธ์จะแบ่งเป็น 3 ขั้นตอนหลักๆ คือ

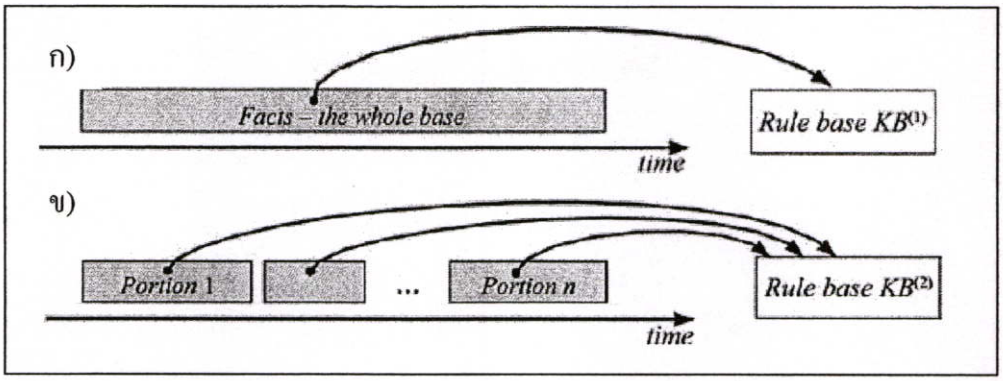
1. การค้นหาความสัมพันธ์ของทรานแซคชันปัจจุบันด้วยอัลกอริทึมสำหรับค้นหาความสัมพันธ์เช่น อะพริโอริ แนวคิดนี้มองทรานแซคชันที่เข้ามาเป็นลักษณะของเซตที่ไม่มีการเรียงลำดับ

2. การค้นหากฎที่เพิ่มขึ้นไปยังกฎพื้นฐาน โดยใช้อัลกอริทึมในการบำรุงรักษากฎ โดยในขั้นตอนนี้จะเป็นหัวใจของวิธีการที่นำเสนอ คือ มีการใช้อัลกอริทึมการบำรุงรักษากฎเพื่อทำการปรับปรุงกฎ ในการปรับปรุงกฎนั้นจะแก้ปัญหาของการบำรุงรักษาส่วนที่เพิ่ม (incremental maintenance) ของกฎพื้นฐานด้วยการประมาณค่า (approximate) ที่คำนวณได้

3. การวิเคราะห์ข้อเท็จจริงที่ใช้

ผลของกฎความสัมพันธ์ใหม่ที่ได้จากการเพิ่มข้อมูลในแต่ละช่วงเวลาเข้าไปในกฎความสัมพันธ์เดิมนั้นด้วยอัลกอริทึมการบำรุงรักษากฎนี้จะได้กฎความสัมพันธ์ที่คล้ายกับการค้นหากฎความสัมพันธ์ในฐานข้อมูลที่ได้รับการปรับปรุงทั้งหมด ดังแสดงในภาพที่ 2.6

อัลกอริทึมที่นำเสนอนี้สามารถนำไปใช้ในการเรียนรู้แบบจำลองในระบบฐานความรู้ (Knowledge based system) ซึ่งเป็นการนำข้อมูลที่ได้เก็บไว้มาทำการวิเคราะห์เพื่อหาความน่าสนใจของกฎความสัมพันธ์ซึ่งคล้ายกับเซตของกฎ เมื่อมีการค้นหาจากชุดข้อมูลฝึกฝน (training set) ทั้งหมดและใช้กฎความสัมพันธ์ที่ได้ในการปรับปรุงประสิทธิภาพตัวอย่างของระบบฐานความรู้ ได้แก่ ซอฟต์แวร์เอเยนต์ (software agent) เป็นต้น



รูปที่ 2.6 แสดงการค้นหากฎความสัมพันธ์

ข้อดี

1. มีการใช้การประมาณค่าให้กับข้อมูลใหม่เพื่อหลีกเลี่ยงการสแกนฐานข้อมูลเดิม

2. เป็นการนำเสนอแนวคิดในการปรับปรุงกฎความสัมพันธ์ที่สามารถใช้อัลกอริทึมใดในการค้นหากฎความสัมพันธ์ในส่วนของฐานข้อมูลเดิมก็ได้

ข้อเสีย

1. ใช้การประมาณค่าสนับสนุนและค่าความเชื่อมั่นให้กับไอเทมเซตในส่วนของข้อมูลที่เพิ่มเข้ามาซึ่งอาจมีข้อผิดพลาดได้เนื่องจากไม่ใช่ค่าที่แท้จริงของไอเทมเซตนั้นๆ ที่เกิดขึ้น

2. การปรับปรุงกฎจะมีจำนวนมากกว่าฟรีเควนท์ไอเทมเซต ทำให้เสียเวลาในการปรับปรุงมาก

2.3.2 การเพิ่มขยายค้นหากฎความสัมพันธ์ที่ให้ความสำคัญกับข้อมูลเก่าและข้อมูลใหม่เท่ากัน

กฎความสัมพันธ์ที่ได้จากไมน์นิ่งในแนวคิดนี้จะได้ผลลัพธ์เช่นเดียวกับการค้นหากฎความสัมพันธ์ทั้งหมดของฐานข้อมูลเดิมและส่วนของข้อมูลใหม่ที่เพิ่มเข้ามา ซึ่งเป็นลักษณะของการไมน์นิ่งเพื่อค้นหากฎความสัมพันธ์ใหม่ทั้งหมดเช่นเดียวกับการหากฎความสัมพันธ์ด้วยอัลกอริทึมอะพริโอริ

Teng and Chen [10] ได้สรุปว่างานวิจัยในแนวคิดนี้ประกอบด้วยเทคนิคหลักๆ ที่ใช้ 3 เทคนิคคือ

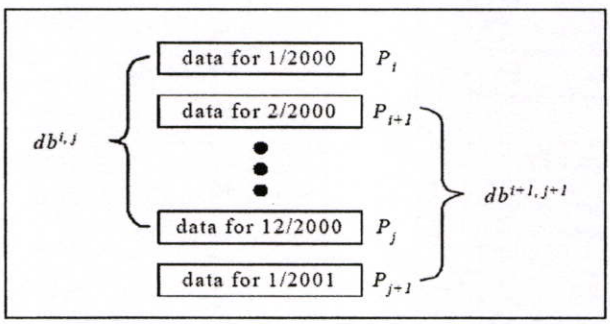
2.3.2.1 อัลกอริทึมที่มีพื้นฐานการทำงานแบบพาร์ทิชัน (Partition-Based Algorithms)

เป็นเทคนิคหนึ่งในการค้นหากฎความสัมพันธ์โดยมีหลักการคือ “ถ้าไอเทม X เป็นฟรีเควนท์ไอเทมเซต ในฐานข้อมูล ซึ่งถูกแบ่งเป็น n พาร์ทิชัน ประกอบด้วย p_1, p_2, …, p_n แล้วดังนั้น ไอเทม X จะต้องเป็นฟรีเควนท์ไอเทมเซต อย่างน้อยใน 1 พาร์ทิชัน” แนวคิดของอัลกอริทึมพาร์ทิชันสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์มีดังนี้

1. สไลด์ดิ่ง-วินโดว์ฟิลเตอร์ริง (Sliding-Window Filtering: SWF) [11]

งานวิจัยนี้จะนำเสนออัลกอริทึมที่ชื่อว่าเอสดับเบิลยูเอฟ เพื่อใช้ในการหากฎความสัมพันธ์เมื่อมีการเพิ่มทรานแซคชันใหม่หรือลบทรานแซคชันเก่าออกจากฐานข้อมูล ซึ่งลักษณะการทำงานของเอสดับเบิลยูเอฟจะมีการแบ่งฐานข้อมูลเป็นส่วนๆ ตามช่วงของเวลาที่เรียกว่าพาร์ทิชัน ดังรูปที่ 2.7 จะเป็นการแบ่งข้อมูลตามช่วงเวลาของเดือนในแต่ละปี เช่น 1/2000 หมายถึงเดือนมกราคมปี 2000, 2/2000 หมายถึงเดือนกุมภาพันธ์ ปี 2000 เป็นต้น

หลักการของเอสดับเบิลยูเอฟจะลดจำนวนรอบของการสแกนเนื่องจากกระบวนการของแต่ละพาร์ทิชันจะอยู่ในรูปแบบที่เรียกว่า เฟส (Phase) ซึ่งจะมีการสร้างแคนดิเดทไอเทมเซตและมีการสะสมสารสนเทศที่เป็นฟรีเควนท์ไอเทมเซตที่ได้จากการไมน์นิ่งในเฟสก่อนหน้าไปยังเฟสถัดไปและเทคนิคของเอสดับเบิลยูเอฟ มีประสิทธิภาพในการหากฎความสัมพันธ์ระหว่างข้อมูลด้วยการใช้ประโยชน์จาหน่วยความจำหลัก และเทคนิคของสไลด์ดิ่ง-วินโดว์พาร์ทิชัน (Sliding-window partition) เนื่องจากจะมีการแบ่งข้อมูลในลักษณะของพาร์ทิชันที่สามารถทำงานได้ภายในหน่วยความจำหลัก เพื่อลดจำนวนของ I/O และซีพียู



รูปที่ 2.7 การไมน์นิ่งส่วนของข้อมูลที่มีการแบ่งตามช่วงเวลา

อัลกอริทึมเอสดับเบิลยูเอฟ จะเริ่มจากการแบ่งทรานแซคชันในฐานข้อมูลออกเป็น n พาร์ทิชันแนวคิดหลักของอัลกอริทึมเอสดับเบิลยูเอฟคือ การคำนวณเซตของแคนดิเดท 2 ไอเทมเซตได้ใกล้เคียงกับฟรีเควนท์ 2 ไอเทมเซต โดยการทำงานของอัลกอริทึมจะเริ่มจากการหา 2 ไอเทมเซดใดๆ ในแต่ละทรานแซคชันของพาร์ทิชันที่ได้แบ่งไว้ตามลำดับ จากนั้นจะทำการนับค่าสนับสนุนและใช้ค่าแบ่งของฟิลเตอร์ริ่ง (Filtering threshold) ซึ่งได้จากผลคูณระหว่างค่าสนับสนุนน้อยที่สุดและจำนวนทรานแซคชันที่ใช้ในการไมน์นิ่งของพาร์ทิชัน ในการพิจารณาค่าสนับสนุนที่มีค่ามากกว่าหรือเท่ากับค่าแบ่งของฟิลเตอร์ริ่ง จะเก็บไว้ในส่วนของแคนดิเดท 2 ไอเทมเซตเพื่อนำไปค้นหาในพาร์ทิชันต่อๆ ไป โดยจะทำการหาค่าสนับสนุนให้กับแคนดิเดทในแต่ละพาร์ทิชัน คือถ้าพบว่าในแต่ละพาร์ทิชันมีแคนดิเดท 2 ไอเทมเซตใดๆ เหมือนกันจะทำการบวกค่าสนับสนุนให้ ในลักษณะที่เรียกว่า ค่าฟิลเตอร์สะสม (Cumulative filter) จนกระทั่งครบทุกพาร์ทิชันจะนำ จะนำแคนดิเดท 2 ไอเทมเซตที่ไม่ซ้ำกันมาเรียงตามลำดับตัวอักษรแล้วนำมาสร้าง แคนดิเดท 3 ไอเทมเซต

โดยนำ C_2*C_2 ซึ่งแตกต่างจากอัลกอริทึมของอะพริโอริที่สร้างแคนดิเดท 3-ไอเทมเซตจากการเชื่อมฟรีเควนท์ 2 ไอเทมเซต (L_2*L_2) ซึ่งจำนวนของแคนดิเดท 3 ไอเทมเซตของเอสดับเบิลยูเอฟจะมีจำนวนมากกว่าจำนวนของแคนดิเดท 3 ไอเทมเซต C_3ของอะพริโอริ

ข้อดี

1. สามารถลดจำนวนแคนดิเดทไอเทมเซตได้อย่างมีประสิทธิภาพ เนื่องจากมีการนำความรู้จากการค้นหาในเฟสก่อนหน้ามาใช้ ซึ่งมีผลต่อการลดซีพียูและหน่วยความจำหลัก

2. สร้างแคนดิเดท 2-ไอเทมเซต (C_2) ที่ใกล้เคียงกับค่า L_2 เท่าที่จะเป็นไปได้ โดยการเริ่มจาก C_2 จะมีผลต่อการลดจำนวนข้อมูลที่ต้องสแกนใน 1 ไอเทมเซต และจะสแกนฐานข้อมูลที่มีช่วงเวลาที่เปลี่ยนแปลงที่ต้องการเพียง 1 ครั้ง

3. หลีกเลี่ยง data skew ได้เนื่องจากมีการใช้สารสนเทศที่สะสมมาทำให้สามารถตัดแคนดิเดทไอเทมเซตที่ไม่เป็นฟรีเควนท์ไอเทมเซตในขั้นก่อนหน้าได้

ข้อเสีย

1. การทำงานจะเริ่มจากการหาแคนดิเดท 2 ไอเทมเซต โดยจะกำหนดให้ทุก 1-ไอเทมเซตเป็นฟรีเควนท์ 1 ไอเทมเซต ซึ่งอาจมีบางไอเทมที่ไม่สามารถเป็นฟรีเควนท์ 1 ไอเทมเซตได้ถูกนำไปสร้างเป็นแคนดิเดท 2 ไอเทมเซตด้วย

2. จำนวนของ แคนดิเดท 3 ไอเทมเซต ที่ได้จากการเชื่อมระหว่างแคนดิเดท 2-ไอเทมเซต (C_2*C_2) มีจำนวนมากกว่าการหาแคนดิเดท 3 ไอเทมเซต ที่ได้จากอะพริโอริ ซึ่งจัดเก็บไว้ในหน่วยความจำหลัก ทำให้ใช้หน่วยความจำหลักในการจัดเก็บแคนดิเดท 3 ไอเทมเซต จำนวนมาก

อัลกอริทึมเอฟไอ-เอสดับเบิลยูเอฟ (SWF with frequent itemset: FI-SWF) [12] และซีไอ-เอสดับเบิลยูเอฟ (SWF with candidate itemsest: CI-SWF) [12] ได้นำหลักการของ SWF มาพัฒนาเพิ่มเติมโดยอัลกอริทึม เอฟไอ-เอสดับเบิลยูเอฟ และ อัลกอริทึมซีไอ-เอสดับเบิลยูเอฟ ได้มีการนำฟฟรีเควนท์ไอเทมเซตและแคนดิเดทไอเทมเซตจากการไมน์นิ่งก่อนหน้ามาใช้ใหม่อีกครั้งเพื่อลดจำนวนของแคนดิเดทไอเทมเซต ดังนั้นเวลาที่ใช้ในการประมวลผลสำหรับทั้ง 2 อัลกอริทึมจะดีกว่าอัลกอริทึมเอสดับเบิลยูเอฟ

2.3.2.2 แพทเทรินโกร์ท (Pattern-Growth)

แนวคิดหลักคือการนำโครงสร้างรูปต้นไม้มาใช้เก็บสารสนเทศของฟรีเควนท์ไอเทมเซต โดยงานวิจัย [5] ที่ได้นำเสนออัลกอริทึมทรีโพรเจคชัน ซึ่งมีการทำการค้นหาฟรีเควนท์ไอเทมเซตในลักษณะของการเรียงลำดับของไอเทมตามตัวอักษรจากมากไปน้อยและใช้แต่ละโหนดของรูปต้นไม้แสดงถึงฟรีเควนท์ไอเทมเซตที่ได้จากการไมน์นิ่งฐานข้อมูลทั้งหมด โดยสามารถช่วยในการจัดการแคนดิเดทไอเทมเซตได้

งานวิจัยที่ได้นำแนวคิดของแพทเทรินโกร์ทมาใช้ในการเพิ่มขยายการค้นหากฎความสัมพันธ์มีดังนี้คือ

1. อัลกอริทึมเอฟพี-โกรท์ (Frequent Pattern growth: FP-Growth) [6]

เป็นงานวิจัยที่พยายามหลีกเลี่ยงการสร้างแคนดิเดทไอเทมเซต โดยใช้เทคนิค การเรียกซ้ำเพื่อสร้างเส้นทางของฟรีเควนท์แพทเทิร์นที่เรียกว่าเทคนิคการแบ่งเพื่อเอาชนะ (Divide and conquer) ของเอฟพี-ทรีในการค้นหาฟรีเควนท์ไอเทมเซต

เนื่องจากเอฟพี-ทรีไม่สามารถนำมาประยุกต์กับปัญหาของการไมน์นิ่งแบบเพิ่มขยาย (Incremental mining) ได้โดยตรง งานวิจัย[13] ได้นำเสนอ 2 อัลกอริทึมที่ได้นำรูปแบบ เอฟพี-ทรีของมาแก้ปัญหาของการไมน์นิ่งแบบเพิ่มขยาย คือดีบี-ทรี (DB-tree) และ พีโอทีเอฟพี-ทรี (PotFp-tree) ซึ่งดีบี-ทรีเก็บไอเทมทั้งหมดในเอฟพี-ทรีแทนที่จะเก็บฟรีเควนท์ 1ไอเทมเซต ในฐานข้อมูล โดยมีการสร้างโครงสร้างเช่นเดียวกับเอฟพี-ทรี เมื่อมีข้อมูลใหม่เพิ่มเข้ามากิ่งของดีบี-ทรีจะสามารถปรับหรือสร้างกิ่งใหม่ได้ ในขณะที่การลบทรานแซคชันเก่าจะสามารถทำได้โดยการปรับหรือลบกิ่งนั้นๆ ออกไป เมื่อฐานข้อมูลมีการเปลี่ยนแปลงจะทำให้ดีบี-ทรีมีความสะดวกและยืดหยุ่นมากขึ้น แต่เมื่อข้อมูลทั้งหมดมีขนาดใหญ่มากการปรับโครงสร้างของดีบี-ทรีอาจจะพบปัญหาในเรื่องของหน่วยความจำไม่เพียงพอ

ส่วนอีกอัลกอริทึมคือ พีโอทีเอฟพี-ทรีจะมีการจัดเก็บเพียงไอเทมที่จะพัฒนาเป็นฟรีเควนท์ ไอเทมเซต (Potentially frequent itemset) ฟรีเควนท์ เพิ่มเข้าไปในฟรีเควนท์ 1 ไอเทมเซต โดยไอเทมที่จะพัฒนาเป็นฟรีเควนท์ไอเทมเซตนี้สามารถพิจารณาได้จากค่าพารามิเตอร์ที่ยอมรับอีกค่าที่เรียกว่าโทเลอแรนซ์พารามิเตอร์ (Tolerance parameter: t) คือ ถ้าค่าสนับสนุน (s) ของไอเทมเซตใดๆ มีค่ามากกว่าหรือเท่ากับค่า t แต่น้อยกว่าค่าสนับสนุนน้อยที่สุด (min_sup) $t \le s \le \min\_\sup$ จะจัดเป็นไอเทมที่จะพัฒนาเป็นฟรีเควนท์ไอเทมเซต ทำให้การสแกนฐานข้อมูลเดิมเพื่อปรับปรุงเอฟพี-ทรีเมื่อมีการปรับปรุงฐานข้อมูลลดลง

ข้อดี

1. ไม่มีการสร้างแคนดิเดทไอเทมเซตไม่มีการกำจัดแคนดิเดทไอเทมเซตที่ไม่ใช่ฟรีเควนท์ไอเทมเซตออก

2. มีรูปแบบการใช้โครงสร้างข้อมูลที่ย่อกระชับ

3. สามารถลดการสแกนฐานข้อมูล

4. สามารถกำจัดข้อมูลที่ไม่เป็นฟรีเควนท์ไอเทมเซตออกไป ทำให้ได้ฟรีเควนท์แพทเทรินไมน์นิ่งที่สมบูรณ์

ข้อเสีย

1. ปัญหาที่พบคือ หน่วยความจำไม่เพียงพอ (Insufficient memory) เมื่อขนาดของทรีมีสมาชิกจำนวนมาก

2. อัลกอริทึมซิกแซก (Zigzag Algorithm) [27]

ซิกแซกคอัลกอริทึมเป็นอัลกอริทึมที่ใช้เทคนิคของการหาฟรีเควนท์ไอเทมเซตสูงสุด (Maximal frequent itemset: MFI) ซึ่งหมายถึงการค้นหาไอเทมใดๆ ที่ไม่เป็นซับเซตของฟรีเควนท์ไอเทมเซตอื่น โดยวัตถุประสงค์ของซิคแซคคือต้องการลดการสร้างแคนดิเดทไอเทมเซตและสามารถทำการเพิ่มขยายการค้นหากฎความสัมพันธ์ได้โดยอาศัยความรู้ที่ได้จากการไมนิ่งในฐานข้อมูลก่อนที่จะมีการทำปรับฐานข้อมูล ด้วยลักษณะการจัดเก็บในรูปของรายการระบุหมายเลขทรานแซคชัน (Transaction ID list : TIDlist) ทำให้ไม่จำเป็นต้องทำการสแกนฐานข้อมูลใหม่ทั้งหมด แต่จะนำเอาข้อมูลที่ถูกจัดเก็บไว้ในรายการระบุหมายเลขทรานแซคชันมาใช้ในการหาฟรีเควนท์ไอเทมเซตสูงสุดดังรูปที่ 2.8 แสดงตัวอย่างฐานข้อมูลที่ประกอบด้วย 3 ทรานแซคชัน การทำงานของอัลกอริทึมซิคแซคจะเริ่มจากนำไอเทมที่ปรากฏในฐานข้อมูลไปเก็บไว้เรียงตาม ลำดับตัวอักษรของไอเทมจากน้อยไปมาก ตามหลักพจนานุกรม (Lexicographic order) แล้วระบุหมายเลขทรานแซคชันที่พบไอเทมนั้นใส่เข้าไปในช่องของไอเทมแต่ละตัวจนครบทุกไอเทมที่มีในฐานข้อมูล เช่น ไอเทม A จะปรากฏในทรานแซคชันหมายเลขที่ 1 และ 3 , ไอเทม C จะปรากฏในทรานแซคชันหมายเลขที่ 1, 2 และ 3 ซึ่งไอเทมที่จัดเก็บจะเรียงลำดับจาก) การหากฎความ สัมพันธ์จะต้องหาค่าสนับสนุนของแต่ละไอเทม ซึ่งค่าสนับสนุนของไอเทมจะต้องนำมาพิจารณาว่ามีค่ามากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุดที่กำหนดไว้หรือไม่ ถ้าไอเทมใดมีค่ามากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุดจะนำไปเก็บไว้ในส่วนของเซตผสม (combine set) เพื่อนำไปหาฟรีเควนท์ไอเทมเซตสูงสุดในลักษณะของแพทเทรินโกรท์ ดังแสดงตัวอย่างในภาพที่ 2.9

TID	Transaction
1	A C T W
2	C D W
3	A C T W

	TIDlist			
A	C	D	T	W
1	1	2	1	1
3	2		3	2
	3			3
2	3	1	2	3

ค่าสนับสนุนของแต่ละไอเทม

รูปที่ 2.8 ตัวอย่างการนำฐานข้อมูลการจัดเก็บแต่ละไอเทมในรูปของรายการระบุหมายเลขทรานแซคชัน

ถ้ากำหนดให้ค่าสนับสนุนน้อยที่สุดเท่ากับ 50 % จากตัวอย่างในรูปที่ 2.8 จะพบว่าไอเทมที่จะเก็บไว้ในเซตผสมคือ {A, C, T, W} สำหรับไอเทมและต้องการหาค่าสนับสนุนระหว่าง

ไอเทมเซต AC จากตัวอย่างข้อมูลในรูปที่ 2.8 จะพบว่าไอเทม A จะปรากฏใน 2 ทรานแซคชันคือ ที่หมายเลขทรานแซคชันที่ 1 และ 3 และไอเทม C จะปรากฏในหมายเลขทรานแซคชันที่ 1, 2 และ 3 ในส่วนของการหาค่าสนับสนุนของ AC จะหาได้จากค่าสนับสนุนที่เกิดร่วมกันด้วยการหา อินเทอร์เซคชัน (A∩C) และใช้ฟังก์ชันดิฟฟ์เซต ซึ่งเป็นการหาค่าหมายเลขทรานแซคชันที่แตกต่างกัน ดังนี้คือ

$$d(AC) = t(A) - t(C)$$
$$= (1,3) - (1,2,3)$$
$$= 0$$
$$\sigma(\mathrm{AC}) = \sigma(A) - d(AC)$$
$$= 2 - 0$$
$$= 2$$

โดย d(AC) หมายถึง การดิฟฟ์เซตของไอเทม AC

t(A) หมายถึง ทรานแซคชันที่มี A ปรากฏอยู่ ในตัวอย่างนี้คือ หมายเลขทรานแซคชันที่ 1 และ 3

t(C) หมายถึงทรานแซคชันที่มี C ปรากฏอยู่ ในตัวอย่างนี้คือ หมายเลขทรานแซคชันที่ 1, 2 และ 3

t(A)-t(C) เป็นการหาทรานแซคชันที่แตกต่างของ A และ C โดยหาหมายเลขทรานแซคชันที่มี A แต่ไม่มี C ในตัวอย่างนี้ เป็น 0 เนื่องจากทรานแซคชันที่มี A จะมี C ด้วย

$\sigma(\mathrm{AC})$ หมายถึง ค่าสนับสนุนของ A และ C ที่เกิดร่วมกัน

$\sigma(A)$ หมายถึง ค่าสนับสนุนของ A

ขั้นตอนในการหาฟรีเควนท์ไอเทมเซตสูงสุด

เมื่อได้ค่าสนับสนุนแล้วจะนำไอเทมต่างๆ มาหาฟรีเควนท์ไอเทมเซตสูงสุด โดยมีขั้นตอนในการหาฟรีเควนท์ไอเทมเซตสูงสุด ดังนี้คือ

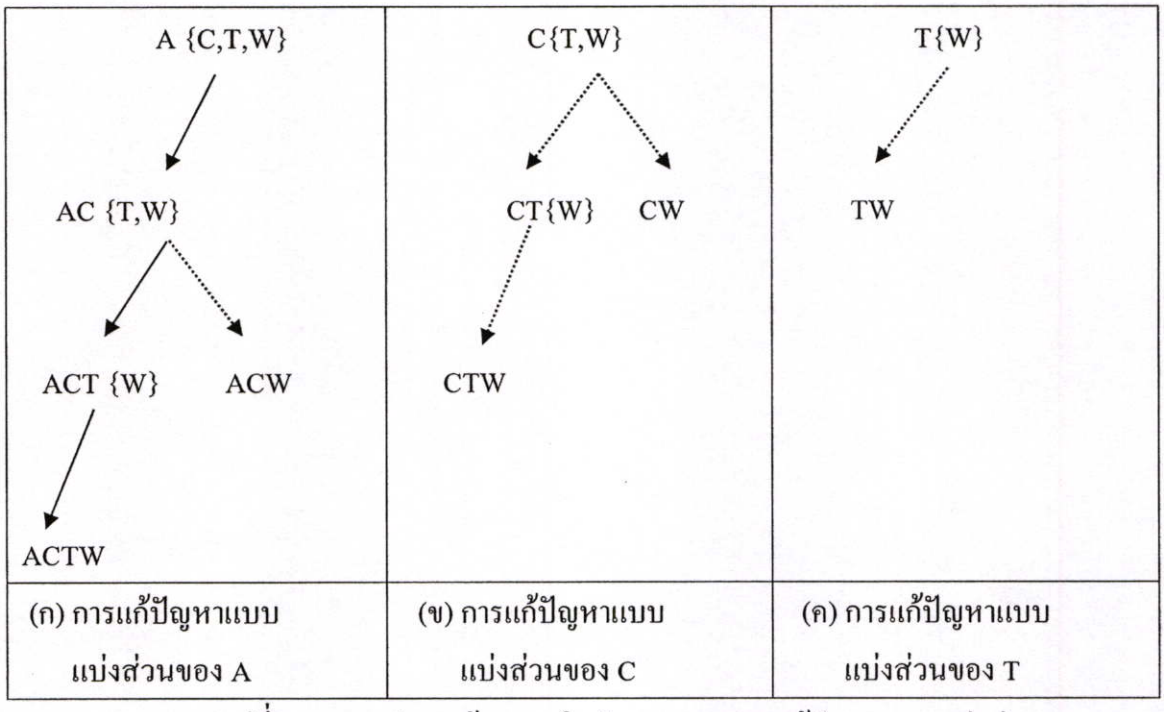
1. ไอเทมทั้งหมด ที่ปรากฏในฐานข้อมูลจะถูกจัดเก็บในเซตที่เป็นไปได้ทั้งหมด (Possible set) โดยจะมีการพิจารณาไอเทมที่เป็นเซตที่เป็นไปได้ทั้งหมด ว่าไอเทมใดมีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุดที่กำหนดไว้ จะนำไปจัดเก็บไว้ในส่วนที่เรียกว่า เซตผสม ซึ่งจะนำมาใช้ในขั้นตอนต่อไป

2. ใช้การค้นหาแบบการย้อนรอย (Backtracking search) เพื่อหาฟรีเควนท์ไอเทมเซตสูงสุด โดยจะมีลักษณะการหาฟรีเควนท์ไอเทมเซตสูงสุด เป็นแบบ การแก้ปัญหาแบบแบ่งส่วน คือจะมีการหาฟรีเควนท์สำหรับแต่ละไอเทมเซตที่เป็นสมาชิกของเซตผสมจากรูปที่ 2.8 และ 2.9 จะได้เซตที่เป็นไปได้และเซตรวมดังนี้คือ

Possible set = {A C D T W}

Combine set = { A C T W}

จากรูปที่ 2.9 (ก) จะแสดงการแก้ปัญหาแบบแบ่งส่วนให้กับไอเทม A ซึ่งเป็นสมาชิกของ C โดยจะพบว่าจะมีการหาความสัมพันธ์ของ A กับสมาชิกตัวอื่นๆ ของเซตรวมทีละตัวตามลำดับในลักษณะของการค้นหาแบบลึกก่อน (depth first search) สำหรับการค้นหาฟรีเควนท์ไอเทมเซตของไอเทม A สุดท้ายจะได้ MFI คือ ACTW รูปที่ 2.9 (ข) จะพบว่ามีลักษณะการหาความสัมพันธ์ของไอเทม C กับเซตผสม {T, W} ซึ่งจะพบว่าทั้ง CT, CW และ TW เป็นสมาชิกของ ACTW ที่เป็น MFI อยู่แล้วเช่นเดียวกับ 2.9 (ค) ที่T และ W เป็นสมาชิกของ ACTW ในส่วนนี้จะสามารถข้ามการค้นหาความสัมพันธ์ของ C, T และ W ไปได้ ทำให้ลดเวลาในการหาความสัมพันธ์ทุกไอเทมลักษณะนี้เป็นการตัดไอเทมที่เป็นสมาชิกของเซตผสมที่เป็นซับเซตของฟรีเควนท์ไอเทมเซตสูงสุดออกไป



(ก) การแก้ปัญหาแบบแบ่งส่วนของ A	(ข) การแก้ปัญหาแบบแบ่งส่วนของ C	(ค) การแก้ปัญหาแบบแบ่งส่วนของ T

รูปที่ 2.9 การทำการย้อนรอยในลักษณะของการแก้ปัญหาแบบแบ่งส่วน

ในส่วนของฟรีเควนท์ไอเทมเซตสูงสุดที่ได้จะอยู่ในรูปของ k-ไอเทมเซตที่สามารถแสดงให้เห็นความสัมพันธ์ระหว่างข้อมูลในลักษณะของกฎความสัมพันธ์ได้เช่น A=> C, A=>T, A=>W, A=> C,T , A=> CW , A=> TW , A => C, T, W เป็นต้น

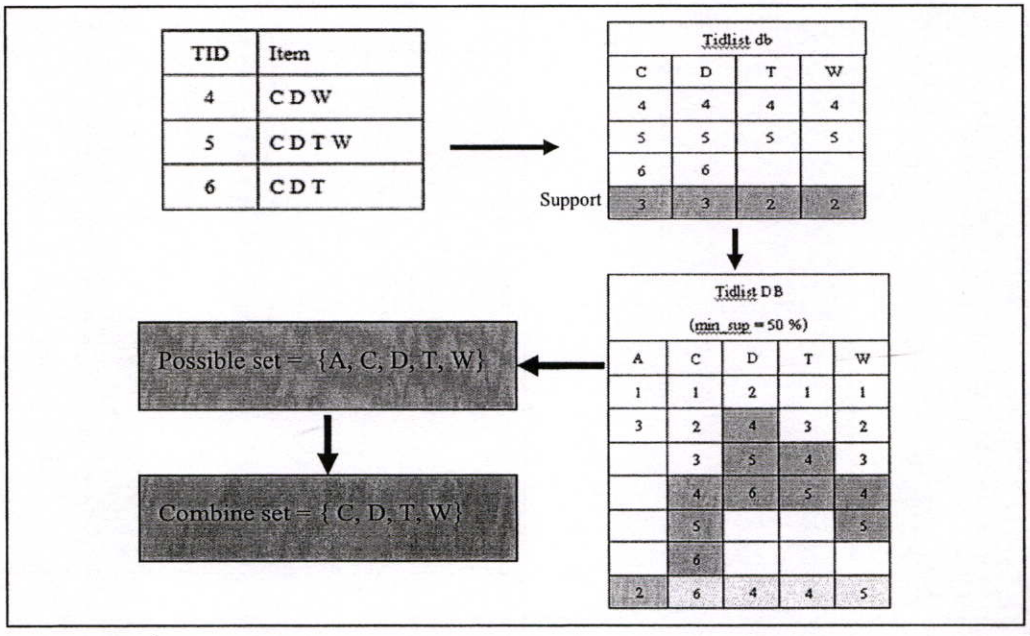
ในกรณีที่มีการเพิ่มข้อมูลใหม่เข้าไปในฐานข้อมูลเดิม โดยหลักการของซิกแซกจะสร้าง รายการระบุหมายเลขทรานแซคชันของข้อมูลที่ถูกเพิ่มเข้ามาหรือลบออกไปต่างหากก่อนที่จะนำค่าของรายการระบุหมายเลขทรานแซคชันมาทำการปรับปรุงกับรายการระบุหมายเลข

ทรานแซคชันของฐานข้อมูลเดิมดังแสดงในรูปที่ 2.10 ซึ่งเมื่อมีการเพิ่มในฐานข้อมูลจะมีผลต่อกฎความสัมพันธ์ที่ได้เคยทำการค้นหาไว้แล้วโดยซิกแซกจะมี 3 ขั้นตอนหลักๆ คือ

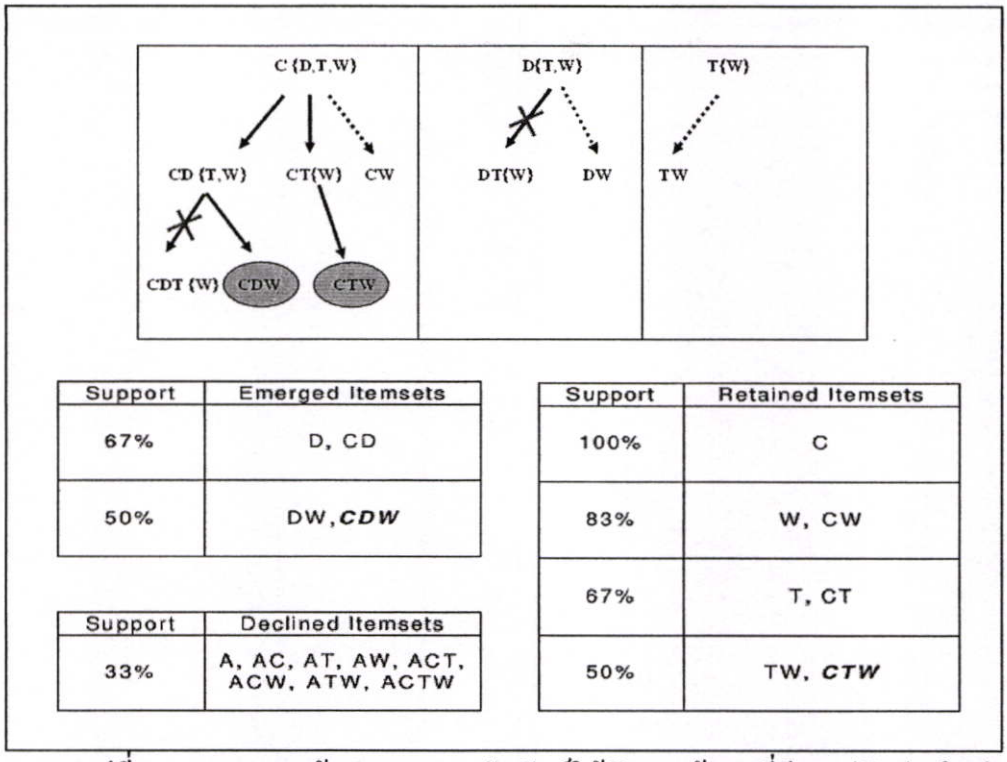
1. การบันทึกทรานแซคชันที่เพิ่มหรือลบในฐานข้อมูล
2. ปรับปรุงค่าฟรีเควนท์ไอเทมเซตสูงสุด ในฐานข้อมูลปรับปรุง (updated database)
3. ปรับค่าสนับสนุนแต่ละฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงดังนี้คือ

$$\mathcal{L}_{\Delta}(X) = (\mathcal{L}_{D}(X) \cup \mathcal{L}_{d+}(X)) - \mathcal{L}_{d-}(X)$$

ในการหากฎความสัมพันธ์ให้กับฐานข้อมูลที่มีการปรับปรุงใหม่นั้นซิกแซกจะทำการปรับปรุงกฎความสัมพันธ์ใหม่ที่เกิดขึ้นได้ในลักษณะเดียวกันดังแสดงในรูปที่ 2.11 คือจะนำสิ่งที่ได้จากการไมนิ่งในฐานข้อมูลเดิมมาใช้และเมื่อมีการเพิ่มทรานแซคชันจะทำการสแกนหาค่า สนับสนุนของแต่ละไอเทมเพื่อจัดเก็บในรายการลำดับทรานแซคชันแล้วจึงนำค่าสนับสนุนที่ได้มาทำการปรับปรุงสำหรับแต่ละไอเทม โดยในขั้นตอนการปรับปรุงเพื่อหาฟรีเควนท์ไอเทมเซตสูงสุดใหม่นั้นผู้ใช้สามารถปรับเปลี่ยนค่าสนับสนุนน้อยที่สุดที่ต้องการได้ จากนั้นจึงหาค่าสนับสนุนในแต่ละ k-ไอเทมเซต ด้วยอินเทอร์เซคชั่นของแต่ละไอเทมและใช้ฟังก์ชันดิฟฟ์เซต (Diffset) ในการหาไอเทมเซตสูงสุดเช่นเดียวกับขั้นตอนหาค่าและขั้นตอนการหาฟรีเควนท์ไอเทมเซตสูงสุด ข้างต้น โดยทั้งนี้ฟรีเควนท์ไอเทมเซตสูงสุด ใหม่ที่หาได้อาจจะแตกต่างจากเดิมซึ่งหมายถึงกฎความสัมพันธ์ที่ได้ก็จะแตกต่างไปจากเดิม



รูปที่ 2.10 การปรับปรุงรายการระบุหมายเลขทรานแซคชันของอัลกอริทึมซิกแซก



Support	Emerged Itemsets
67%	D, CD
50%	DW, ***CDW***

Support	Declined Itemsets
33%	A, AC, AT, AW, ACT, ACW, ATW, ACTW

Support	Retained Itemsets
100%	C
83%	W, CW
67%	T, CT
50%	TW, ***CTW***

รูปที่ 2.11 แสดงการค้นหากฎความสัมพันธ์ให้กับฐานข้อมูลที่มีการปรับปรุงใหม่

ข้อดี

1. สามารถลดการหาแคนดิเดทไอเทมเซต เนื่องจากถ้าไอเทมใดเป็นสมาชิกของฟรีเควนท์ไอเทมเซตสูงสุด จะสามารถข้ามการค้นหาไปได้

2. การจัดเก็บข้อมูลในลักษณะของรายการระบุหมายเลขทรานแซคชัน จะทำให้ง่ายต่อการจัดเก็บและจัดการในกรณีที่มีการเพิ่มทรานแซคชัน เนื่องจากมีการจัดเก็บเรียงตามลำดับทรานแซคชัน

3. สามารถปรับเปลี่ยนค่าสนับสนุนน้อยที่สุดได้เมื่อมีการปรับปรุงฐานข้อมูล ซึ่งจะต้องมีการหากฎความสัมพันธ์ใหม่ของไอเทมต่างๆ ที่ถูกจัดเก็บในรายการระบุหมายเลขทรานแซคชันทั้งหมด

ข้อเสีย

1. จากการจัดเก็บเป็นระบุหมายเลขทรานแซคชัน สำหรับแต่ละไอเทมจะทำให้ต้องมีการจัดเก็บข้อมูลทั้งหมดจำนวนมาก นั่นคือจำนวนข้อมูลมากที่สุดที่ต้องจัดเก็บคือทุกไอเทมและทุกทรานแซคชันที่ปรากฏในฐานข้อมูล

2. การหาความสัมพันธ์ถึงแม้จะไม่ต้องทำการสแกนฐานข้อมูลใหม่ทั้งหมด แต่จะต้องทำการหาค่าสนับสนุนที่เกิดขึ้นในแต่ละไอเทมและระหว่างไอเทมทั้งหมดใหม่ทุกครั้งที่มีการเปลี่ยนแปลง

2.3.2.3 อัลกอริทึมที่มีพื้นฐานการทำงานของอะพริโอริ (Apriori - Based Alogorithms)

อัลกอริทึมอะพริโอริเป็นอัลกอริทึมที่ใช้ความรู้หรือคุณสมบัติของฟรีเควนท์ไอเทมเซต มาช่วยให้การค้นหาในขอบเขตที่แคบลง สำหรับการค้นหาฟรีเควนท์ไอเทมเซต โดยวิธีการหลักของอะพริโอริ คือ ที่นำเซตของฟรีเควนท์ k-ไอเทมเซต มาใช้ในการค้นหาฟรีเควนท์ (k+1) ไอเทม ในลักษณะค้นหาตามระดับด้วยการนำฟรีเควนท์ k-ไอเทมมาสร้างแคนดิเดท (k+1) ไอเทมเซตด้วยขั้นตอนการเชื่อมและการตัด (ดังขั้นตอนที่ได้กล่าวมาก่อนหน้านี้) จากนั้นจึงนำแคนดิเดท (k+1) ไอเทมเซตไปสแกนเพื่อหาฟรีเควนท์ (k+1) ไอเทมเซต กระบวนการเหล่านี้จะถูกทำไปต่อเนื่องจนกว่าจะไม่สามารถสร้างแคนดิเดท (k+1) ไอเทมเซตได้

งานวิจัยที่ใช้หลักการของอะพริโอริมาพัฒนาสำหรับค้นหาการเพิ่มขยายกฎความ สัมพันธ์มีดังนี้ คือ

1. อัลกอริทึมที่มีพื้นฐานการทำงานของเอฟยูพี (FUP based algorithm)

เอฟยูพีเป็นงานวิจัยแรกที่นำเสนอเกี่ยวกับเทคนิคการเพิ่มขยายการปรับปรุงที่พัฒนาขึ้นมาเพื่อบำรุงรักษากฎความสัมพันธ์ที่นำเสนอโดย Cheung et al [14] เมื่อมีการเพิ่มทรานแซคชันเข้าไปในฐานข้อมูลเดิม โดยลักษณะเด่นของอัลกอริทึมเอฟยูพีมีดังนี้คือ

1.1 นำความรู้ที่ได้จากค้นหากฎความสัมพันธ์โดยการไมน์นิ่งในฐานข้อมูลก่อนที่จะมีการเปลี่ยนแปลงมาใช้ เพื่อลดจำนวนทรานแซคชันที่จะต้องสแกนในฐานข้อมูลทั้งหมด จากนั้นจึงหาค่าสนับสนุนของข้อมูลต่างๆ ที่เกิดขึ้น ซึ่งแตกต่างจากอัลกอริทึมอะพริโอริที่จะต้องสแกนฐานข้อมูลทั้งหมด (ทั้งข้อมูลเก่าและข้อมูลใหม่ที่เปลี่ยนแปลง) เพื่อหากฎความสัมพันธ์ใหม่ โดยไม่นำความรู้ที่ได้จากการหากฎความสัมพันธ์ในฐานข้อมูลเดิมมาใช้ให้เกิดประโยชน์ ดังนั้นเอฟยูพีจึงเป็นแนวคิดแรกที่มีการนำความรู้ที่ได้จากการค้นหากฎความสัมพันธ์ก่อนหน้ามาใช้เพื่อลดการสแกนข้อมูลทั้งหมด

1.2 การหากฎความสัมพันธ์จะอยู่บนฐานของอัลกอริทึมอะพริโอริ คือมีการวนรอบหาข้อมูลในลักษณะแบบระดับ เช่น มีการค้นหาจาก 1 ไอเทมก่อนแล้วจึงไปค้นหาไอเทมระดับที่ 2 คือ 2 ไอเทมเซต จนถึง n ไอเทมเซต เป็นต้น และมีการหาฟรีเควนท์ไอเทมเซตจากแคนดิเดทไอเทมเซต

1.3 ใช้ค่าสนับสนุนน้อยที่สุด เดียวกันในการหากฎความสัมพันธ์เอฟยูพีจะใช้ค่าสนับสนุนน้อยที่สุดเดียวกันในการค้นหาฟรีเควนท์ k-ไอเทมเซตของฐานข้อมูลเก่าและใหม่

อัลกอริทึมของเอฟยูพีมีการทำงานอยู่บนพื้นฐานของอัลกอริทึม อะพริโอริโดยเป็นอัลกอริทึมที่กล่าวถึงเทคนิคการทำการเพิ่มขยายการปรับปรุงในกรณีการเพิ่มข้อมูลรายการทรานแซคชันเพียงกรณีเดียว โดยจะมีการวนรอบซ้ำเพื่อหาความสัมพันธ์ของข้อมูลเริ่มจาก 1 ไอเทมไปจนถึง k-ไอเทมเซต (k = 2, 3, …, n) ซึ่งแคนดิเดทไอเทมเซตในแต่ละรอบของการวนรอบซ้ำบนฐาน

ของฟรีเควนท์ไอเทมเซตที่พบในวนรอบซ้ำก่อนหน้า เอฟยูพีจะใช้ในการแก้ปัญหาด้านประสิทธิภาพในการปรับกฎความสัมพันธ์ หลังจากที่มีการเพิ่มทรานแซคชันใหม่เข้ามาในฐานข้อมูลจะนำกฎความสัมพันธ์ที่ได้ถูกค้นหาไว้ก่อนในฐานข้อมูลเดิมมาใช้ประโยชน์ เพื่อลดจำนวนข้อมูลที่จะต้องนำมาหากฎความสัมพันธ์ โดยจะทำการสแกนฐานข้อมูลใหม่ซึ่งมีจำนวนรานแซคชันที่เกิดขึ้นน้อยกว่าฐานข้อมูลเดิม แล้วจึงนำค่าสนับสนุนที่ได้จากการสแกนฐานข้อมูลใหม่มารวมกับฐานข้อมูลเดิม โดยจะทำการพิจารณาหาค่าของฟรีเควนท์ไอเทมเซตใหม่ (L^{UP}) ที่ได้จากการรวมกันของฐานข้อมูลเดิมและฐานข้อมูลที่เพิ่มเข้ามา (DB$\cup$db) ด้วยค่าสนับสนุนของ ไอเทม ต้องไม่น้อยกว่า $s \times (D+d)$ (โดย s หมายถึง ค่าสนับสนุน, D หมายถึง จำนวนของทรานแซคชันใน ฐานข้อมูลเดิม, d หมายถึง จำนวนของทรานแซคชันที่เพิ่มเข้าไปในฐานข้อมูลเดิม)

ข้อดี

กระบวนการค้นหากฎความสัมพันธ์ของอัลกอริทึมเอฟยูพีมีลักษณะเด่นในเรื่องการนำความรู้ที่ได้จากการค้นหาในฐานข้อมูลเดิมมาใช้ร่วมกับรายการข้อมูลที่เพิ่มเข้าไปใหม่ ทำให้สามารถลดจำนวนแคนดิเดทไอเทมเซตที่จะต้องทำสแกนในฐานข้อมูลเดิมลง โดยจะทำการสแกนฐานข้อมูลเดิมในกรณีที่แคนดิเดทไอเทมเซตที่ได้จากฐานข้อมูลใหม่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุดของฐานข้อมูลใหม่หรือเป็นฟรีเควนท์ k-ไอเทมเซตในฐานข้อมูลใหม่เท่านั้น กระบวนการค้นหากฎความสัมพันธ์ของอัลกอริทึมเอฟยูพีนั้นค่าสนับสนุนน้อยที่สุด สำหรับฐานข้อมูลเดิมและฐานข้อมูลใหม่คงที่คือใช้ค่าสนับสนุนน้อยที่สุดเท่ากันในการค้นหาฟรีเควนท์ k- ไอเทมเซตในฐานข้อมูลเดิมและฐานข้อมูลใหม่

ข้อด้อย

เนื่องจากในการไมน์นิ่งฐานข้อมูลเดิมจะเก็บเฉพาะฟรีเควนท์ k-ไอเทมเท่านั้น ดังนั้นการหาฟรีเควนท์ไอเทมเซตใหม่จะต้องทำการสแกนฐานข้อมูลเดิมทุกรอบของการหาฟรีเควนท์ k - ไอเทมเซต ถึงแม้จะเป็นการสแกนเฉพาะค่าของไอเทมเซตที่เป็นสมาชิกของแคนดิเดทไอเทมเซตที่พบว่าเป็นฟรีเควนท์ k-ไอเทมเซตที่ปรากฏในฐานข้อมูลใหม่ แต่การค้นหาค่าสนับสนุนในฐานข้อมูลเดิมนั้นจะต้องทำการสแกนทุกรอบของ k-ไอเทมเซต

2. ยูดับเบิลยูอีพี (Update with Early Prunning Algorithm: UWEP) [16]

งานวิจัยนี้จะนำเสนออัลกอริทึมที่ชื่อว่ายูดับเบิลยูอีพีสำหรับปรับปรุงฟรีเควนท์ไอเทมเซตเมื่อมีทรานแซคชันเพิ่มเข้ามาในฐานข้อมูล โดยจะต้องทำการค้นหากฎความสัมพันธ์ที่เกิดจากการปรับปรุงฐานข้อมูล โดยอัลกอริทึมนี้จะใช้กลยุทธ์ที่เรียกว่าในการตรวจสอบและลบไอเทมที่ไม่สามารถจะกลายเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงได้ออกไป ด้วยจำนวนครั้งที่ใช้ในการสแกนฐานข้อมูลเดิมเพียง 1 ครั้ง และทรานแซคชันที่เพิ่มเข้ามาใหม่ซึ่งในงานวิจัยนี้จะเรียกว่าฐานข้อมูลใหม่ (new database: db) เพียง 1 ครั้ง

ยูดับเบิลยูอีพีเป็นอัลกอริทึมที่นำวิธีการทำงานของอัลกอริทึมเอฟยูพี$_2$ [15] และพาร์ทิชันอัพเดท [17] มาปรับปรุงเพื่อใช้ในการสร้างและพิจารณาจำนวนของแคนดิเดทไอเทมเซตที่จะกลายมาเป็นฟรีเควนท์ไอเทมเซตใหม่และจะทำการตัดไอเทมเซตที่ไม่ใช่ฟรีเควนท์ไอเทมเซตออกไป ซึ่งจะมีผลให้มีแคนดิเดทไอเทมเซตจำนวนน้อยสำหรับนำไปหาฟรีเควนท์ไอเทมเซตใหม่ โดยในการทำงานของยูดับเบิลยูอีพีจะใช้ค่าสนับสนุนน้อยที่สุดคงที่ สำหรับการหาฟรีเควนท์ k - ไอเทมเซตในฐานข้อมูลเดิมและฐานข้อมูลปรับปรุง

ขั้นตอนการทำงานของอัลกอริทึมยูดับเบิลยูอีพีจะเริ่มจากส่วนรับข้อมูลเข้า (Input) สำหรับหาฟรีเควนท์ไอเทมเซตใหม่คือ ฐานข้อมูลเดิม (DB), ฐานข้อมูลใหม่ (db), ฟรีเควนท์ไอเทมเซตของฐานข้อมูลเดิม (L_{DB}), จำนวนทรานแซคชันของฐานข้อมูลเดิม (|DB|), จำนวนทรานแซคชันของฐานข้อมูลใหม่ (|db|) และค่าสนับสนุนน้อยที่สุด (min_sup) เพื่อนำมาหาค่าฟรีเควนท์ไอเทมเซตใหม่ที่เกิดในจากฐานข้อมูลปรับปรุง (DB+db)

อัลกอริทึมของยูดับเบิลยูอีพี ที่ใช้ในการปรับปรุงฟรีเควนท์ไอเทมเซต ซึ่งมีการทำงานเป็น 5 ขั้นตอนคือ

1. การนับ 1 ไอเทมเซต ที่เกิดขึ้นในฐานข้อมูลใหม่และสร้างรายการลำดับของทรานแซคชัน (Tidlist) สำหรับแต่ละไอเทมเซตในฐานข้อมูลใหม่

2. ตรวจสอบว่าฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิม ซึ่งไม่พบในฐานข้อมูลใหม่และมี ซูเปอร์เซตอยู่ในฐานข้อมูลเดิมและฐานข้อมูลใหม่ (DB+db)

3. ตรวจสอบฟรีเควนท์ไอเทมเซตในฐานข้อมูลใหม่ที่พบในฐานข้อมูลเดิมและฐานข้อมูลใหม่

4. ตรวจสอบฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิมที่ไม่ถูกนับในฐานข้อมูลใหม่ที่เป็นค่าที่พบในฐานข้อมูลเดิมและฐานข้อมูลใหม่

5. สร้างแคนดิเดทเซตจากเซตของฟรีเควนท์ไอเทมเซตที่ได้จากขั้นตอนก่อนหน้า

จากขั้นตอนของอัลกอริทึมยูดับเบิลยูอีพีทั้ง 5 ขั้นตอนข้างต้นสามารถแสดงสามารถสรุปได้ดังนี้

เริ่มจากสแกนฐานข้อมูลใหม่เพื่อหาแคนดิเดทไอเทมเซตที่เกิดขึ้นทั้งหมดในฐานข้อมูลใหม่ โดยจะนำแคนดิเดท 1 ไอเทมเซต ซึ่งประกอบด้วยค่าไอเทม X ที่มีค่าสนับสนุนมากกว่า 0 และสร้างรายการระบุหมายเลขทรานแซคชัน (Transaction identifier: TID) สำหรับแต่ละ 1 ไอเทมเซตในฐานข้อมูลใหม่ db

หาค่าเซตที่จะตัด (pruneset) ซึ่งหมายถึงไอเทม X ที่เป็น ฟรีเควนท์ 1-ไอเทมเซตในฐานข้อมูลเดิม DB แต่ไม่เป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลใหม่ db ไปตรวจสอบว่าค่าไอเทม X ที่อยู่ในเซตที่จะตัดนี้เป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงหรือไม่ โดยสามารถสรุปได้เป็น 2 กรณีคือ

กรณีที่ 1 ค่าไอเทม X ที่เป็นสมาชิกของเซตที่จะตัดเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุง (DB+db) จะทำการเพิ่มไอเทม X นี้ในฟรีเควนท์ไอเทมเซตของฐานข้อมูลปรับปรุง (L_{DB+db}) และนำค่าซูเปอร์เซตของไอเทม X เพิ่มเข้าไปเป็นสมาชิกของเซตที่จะตัดและลบค่าไอเทม X ออกจากฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิม L_{DB}

กรณีที่ 2 ค่าไอเทม X ที่เป็นสมาชิกของเซตที่จะตัดไม่เป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุง (DB+db) จะทำการลบไอเทม X และซูเปอร์เซตของไอเทม X ออกจากฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิม L_{DB} และ ลบไอเทม X และซูเปอร์เซตของไอเทม X ออกจากเซตที่จะตัด

จากการพิจารณาว่าถ้าซูเปอร์เซตทั้งหมดของ ไอเทมเซต X ในฐานข้อมูลเดิม D ไม่เป็นฟรีเควนท์ไอเทมเซตเพราะฉะนั้นไอเทม X จะไม่เป็นฟรีเควนท์ไอเทมเซตด้วย ดังนั้น จึงสามารถตัดไอเทมและซูเปอร์เซตของไอเทมเซตทั้งหมดจากเซตของฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิม DB ที่ไม่เป็นฟรีเควนท์ออกจากได้ การพิจารณาในขั้นตอนนี้ทำให้ยูดับเบิลยูอีพีแตกต่างจากอัลกอริทึมอื่นๆ

ซึ่งการตรวจสอบนี้จะเป็นการตรวจสอบค่าของแคนดิเดทไอเทมเซตที่เป็นสมาชิกของฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิม L_{DB} ก่อนที่จะมีการค้นหาฟรีเควนท์ไอเทมเซตของฐานข้อมูลปรับปรุงรอบแรก (k=1) ทำให้ลดจำนวนแคนดิเดทเซตที่จะต้องนำมาค้นหาให้มีจำนวนน้อยลง จากนั้นจึงนำไอเทมที่เป็นสมาชิกของแคนดิเดท k-ไอเทมเซตของฐานข้อมูลใหม่ (C_{db}^{k}) และฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิมที่ยังไม่อยู่ในเซตที่จะตัด ซึ่งจะเก็บไว้ในตัวแปรที่ชื่อว่า unchecked ไปตรวจสอบในกรณีต่างๆ ดังนี้คือ

กรณีที่ 1 ไอเทม X ไม่เป็นฟรีเควนท์ในฐานข้อมูลใหม่ แต่เป็นฟรีเควนท์ในฐานข้อมูลเดิม DB จะทำการลบ ไอเทม X ออกจากฟรีเควนท์ในฐานข้อมูลเดิม และจะทำการตรวจสอบต่อไปดังนี้คือ

1.1 ถ้าไอเทม X ไม่เป็นฟรีเควนท์ในฐานข้อมูลปรับปรุงจะลบซูเปอร์เซตทั้งหมดของไอเทม X ออกจากตัวแปรชื่อว่า unchecked

1.2 ถ้าไอเทม X เป็นฟรีเควนท์ในฐานข้อมูลปรับปรุงจะเพิ่มไอเทม X ในฐานข้อมูลปรับปรุง

กรณีที่ 2 ถ้าไอเทม X เป็นฟรีเควนท์ในฐานข้อมูลเดิมและฐานข้อมูลใหม่จะลบไอเทม X ออกจากค่าตัวแปร unchecked และเพิ่มไอเทม X ในฐานข้อมูลปรับปรุงฐานข้อมูลปรับปรุง L_{DB+db}

กรณีที่ 3 ถ้าไอเทม X เป็นฟรีเควนท์ในฐานข้อมูลใหม่ แต่ไม่เป็นฟรีเควนท์ในฐานข้อมูลเดิม จะหาค่าค่าสนับสนุนของไอเทม X จากการสแกนฐานข้อมูล โดยจะสร้างรายการระบุหมายเลขทรานแซคชันหรือทีไอดีลิสท์ (tidlist) ขึ้นมาเพื่อใช้ในการหาค่าสนับสนุนของไอเทม

X ในฐานข้อมูลเดิมและถ้าพบว่าฟรีเควนท์ในฐานข้อมูลปรับปรุง จะทำเพิ่มไอเทม X นี้ไปเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุง และฟรีเควนท์ไอเทมเซตในฐานข้อมูลใหม่

สุดท้ายจะทำการตรวจสอบค่าของไอเทม X ที่เป็นสมาชิกที่เก็บในตัวแปร unchecked อีกครั้งโดยจะทำการหาค่าสนับสนุนของ X ในฐานข้อมูลใหม่โดยใช้ทีไอดีลิสท์ ถ้าพบว่าไอเทม X ไม่เป็นฟรีเควนท์ในฐานข้อมูลปรับปรุง จะทำการลบซูเปอร์เซตทั้งหมดของไอเทม X ออกจากฟรีเควนท์ไอเทมเซตของฐานข้อมูลเดิมและถ้าพบว่าไอเทม X เป็นฟรีเควนท์ในฐานข้อมูลปรับปรุงจะเพิ่มไอเทม X ในฟรีเควนท์ไอเทมเซตของฐานข้อมูลปรับปรุง และทำการวนหา k+1 ไอเทมเซตต่อไป โดยหาค่าแคนดิเดท k-ไอเทมเซตของฐานข้อมูลใหม่ (C_{db}^{k}) จากฟรีเควนท์ (k-1)-ไอเทมเซตของฐานข้อมูลใหม่ (L_{db}^{k-1})

ขั้นตอนที่ 2-5 เป็นการตรวจสอบว่า แคนดิเดทไอเทมเซตในฐานข้อมูลใหม่ว่าจะเป็นฟรีเควนท์ในฐานข้อมูลปรับปรุงหรือไม่ ซึ่งในขั้นตอนนี้จะทำการปรับค่าสนับสนุนให้กับไอเทม X ในฐานข้อมูลปรับปรุง และเป็นการตรวจสอบว่าไอเทม X ใดๆ ที่เป็นฟรีเควนท์ไอเทมเซตของฐานข้อมูลเดิม แต่ไม่เป็นฟรีเควนท์ไอเทมเซตของฐานข้อมูลใหม่จะสามารถเป็นฟรีเควนท์ในฐานข้อมูลปรับปรุงหรือไม่ ซึ่งสามารถสรุปกรณีการเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงได้ดังนี้คือ

1. ถ้า $X \in L_{DB}$ และ $X \in L_{db}$ ดังนั้น $X \in L_{DB+db}$
2. ถ้า $X \notin L_{DB}$ ดังนั้น $X \in L_{DB+db}$ ได้ก็ต่อเมื่อ $X \in L_{db}$
3. ถ้า $X \notin L_{DB}$ และ $X \notin L_{db}$ ดังนั้น $X \notin L_{DB+db}$

สำหรับการหาค่าสนับสนุนของฐานข้อมูลปรับปรุง จะสามารถคำนวณได้จาก

$$\text{support}_{DB+db}(X) \geq \min\text{sup} \times |DB+db|$$

3. อัลกอริทึมที่มีพื้นฐานการทำงานของเนกาทีฟบอร์เดอร์ (Negative border based algorithm)

งานวิจัยนี้เป็นการศึกษาการบำรุงรักษากฎความสัมพันธ์ เมื่อมีการเพิ่มทรานแซคชันใหม่เข้าไปในฐานข้อมูลเดิม การหากฎความสัมพันธ์ในงานวิจัยนี้จะอยู่ในรูปแบบของอัลกอริทึมที่มีการค้นหาตามระดับเช่นเดียวกับอะพริโอริ แต่จะอาศัยหลักการของเนกาทีฟบอร์เดอร์ที่ได้นำเสนอโดย Toivonen [4]

สำหรับแนวคิดของของเนกาทีฟบอร์เดอร์จะเป็นการนำเสนอส่วนของซับเซตของไอเทมเซตที่เป็นสมาชิกของแคนดิเดท k-ไอเทมเซต (C_k) โดยส่วนของสมาชิกของแคนดิเดท k-ไอเทมเซตที่มีค่ามากกว่าหรือเท่ากับค่าค่าสนับสนุนน้อยที่สุด จะจัดเก็บไว้ในส่วนของฟรีเควนท์ k-ไอเทมเซต แต่ส่วนที่เหลือจะเรียกว่า บอร์เดอร์เซต (border set) ทั้งนี้แคนดิเดท k-ไอเทมเซตจะถูก

สร้างขึ้นมาด้วยหลักการเดียวกับอะพริโอริ ที่มีการนำฟรีเควนท์ (k-1) ไอเทมเซตมาสร้างแคนดิเดท k ไอเทมเซต ดังนั้นบอร์เดอร์เซตที่ได้จะมีซับเซตที่เป็นฟรีเควนท์ (k-1) ไอเทมเซต โดยที่สมาชิกของบอร์เดอร์เซตตัวนั้นไม่ได้เป็นฟรีเควนท์ k-ไอเทมเซต ความสัมพันธ์ระหว่างแคนดิเดท k-ไอเทมเซต(C_k) ,ฟรีเควนท์ k-ไอเทมเซต(L_k) และเนกาทีฟบอร์เดอร์ (NBd(L_k) สามารถแสดงได้ดังนี้

$$NBd(L_k) = C_k - L_k \quad \text{หรือ} \quad C_k = L_k \cup NBd(L_k)$$

ตัวอย่างเช่น

C_1 = { A, B, C, D, E, F}

L_1 = {A,B, C, F}

$NBd(L_1)$ = {D, E}

C_2 = { {A, B}, {A, C},{A,F},{B,C},{B,F},{C,F}}

L_2 = { {A, B}, {A,C},{A,F},{C,F}}

$NBd(L_2)$ = { {B,C},{B,F}}

แนวคิดนี้ได้มีผู้นำเสนอในส่วนของการเพิ่มขยายกฎความสัมพันธ์ 2 งานวิจัย [18][19] โดยทั้ง 2 งานวิจัยมีแนวคิดที่จะปรับปรุงการทำงานของอัลกอริทึมเอฟยูพีที่มีการนำความรู้ที่ได้จากการไมน์นิ่งฟรีเควนท์ k-ไอเทมเซต ที่ได้จากฐานข้อมูลเดิมมาใช้เพื่อลดการไมน์นิ่งฐานข้อมูลทั้งหมด ด้วยการลดจำนวนไอเทมเซตที่ต้องทำการไมน์นิ่ง แต่ยังคงต้องสแกนฐานข้อมูลทั้งฐานข้อมูลเดิม และส่วนของฐานข้อมูลที่เพิ่มในทุกรอบของการค้นหา ฟรีเควนท์ k-ไอเทมเซต ดังนั้นเพื่อเป็นการลดจำนวนการค้นหาฟรีเควนท์ k-ไอเทมเซต จึงได้มีการนำส่วนของบอร์เดอร์เซตเข้ามาช่วย ในงานวิจัยทั้ง 2 จึงมีการเก็บทั้งฟรีเควนท์ k-ไอเทมเซต และส่วนที่เป็นบอร์เดอร์ k-ไอเทมเซต

หลักการทำงานของอัลกอริทึมเนกาทีฟบอร์เดอร์ในส่วนของการเพิ่มขยายกฎความสัมพันธ์นั้นจะประกอบด้วย 2 คุณลักษณะหลัก ๆ คือ

1. ถ้าไม่มีฟรีเควนท์ไอเทมเซตใหม่ที่เกิดขึ้นจะไม่มีการเข้าถึงฐานข้อมูลเดิม ดังนั้นจึงสามารถรับประกันได้ว่าการสแกนฐานข้อมูลทั้งหมดจะเกิดกรณีเดียวเท่านั้นคือเมื่อมีของฟรีเควนท์ไอเทมเซตใหม่ โดยอัลกอริทึมที่ได้มีการนำเสนอก่อนหน้านี้ไม่สามารถรับประกันในเรื่องนี้ได้

2. ในกรณีที่ต้องสแกนฐานข้อมูลทั้งหมดใหม่นั้นจำนวนรอบของการสแกนฐานข้อมูลจะมีจำนวนไม่มากเนื่องจากจะทำการสร้างแคนดิเดท (k+1)ไอเทมเซต ให้ครอบคลุม

เฉพาะฟรีเควนท์ k-ไอเทมเซต ใหม่เท่านั้น ดังนั้นจึงมีแคนดิเดท (k+1)ไอเทมเซตเท่านั้นที่จะถูกสแกนเพื่อหาค่าสนับสนุน

ข้อดี

แนวคิดของเนกาทีฟบอร์เดอร์นั้นจะมีการเก็บข้อมูลทั้งในส่วนของไอเทมเซตที่เป็น ฟรีเควนท์ k-ไอเทมเซต และเนกาทีฟบอร์เดอร์ k-ไอเทมเซต ดังนั้นถ้าข้อมูลที่เกิดขึ้นในฐานข้อมูลที่ปรับปรุงไม่มีการเปลี่ยนแปลงในส่วนของฟรีเควนท์ k-ไอเทมเซตแล้ว การปรับปรุงค่าสนับสนุนของสมาชิกที่เก็บไว้จะสามารถทำได้อย่างรวดเร็วและสามารถลดจำนวนการสแกนฐานข้อมูลเดิมไปได้อย่างมาก

ข้อด้อย

จากแนวคิดที่จะลดการสแกนฐานข้อมูลเดิมของเนกาทีฟบอร์เดอร์นี้ทำให้ต้องมีการเก็บข้อมูลทั้งในส่วนของฟรีเควนท์ k-ไอเทมเซต และบอร์เดอร์ k-ไอเทมเซต ซึ่งเป็นสมาชิกของ แคนดิเดท k-ไอเทมเซต ทั้งหมดในพบในแต่ละ k-ไอเทมเซตทั้งของฐานข้อมูลเดิมและฐานข้อมูลปรับปรุง ดังนั้นจะต้องใช้พื้นที่จำนวนมากในการจัดเก็บไอเทมเซตทั้งหมด

นอกจากกรณีการใช้พื้นที่จำนวนมากแล้ว เมื่อฐานข้อมูลใหม่ที่เพิ่มเข้ามานั้นมีผลทำให้บอร์เดอร์ k-ไอเทมเซตที่พบในฐานข้อมูลเดิมกลายมาเป็นฟรีเควนท์ k-ไอเทมเซต ซึ่งฟรีเควนท์ไอเทมเซตใหม่นี้จะมีผลต่อการเกิดฟรีเควนท์ (k+1) ไอเทมเซตและบอร์เดอร์ (k+1) ไอเทมเซตบอร์เดอร์ (k+1) ไอเทมเซตในฐานข้อมูลปรับปรุง ดังนั้นเพื่อให้การค้นหากฎความสัมพันธ์ครอบคลุมไอเทมเซตใหม่ที่เกิดขึ้นนี้ทำให้ต้องทำการสแกนฐานข้อมูลทั้งหมดใหม่ ซึ่งอาจใช้เวลานานกว่าการทำไมน์นิ่งข้อมูลใหม่ทั้งหมดของอะพริโอริ ดังที่พบในงานวิจัย (Adnan M. et al. 2005)

ฐานข้อมูลโดยทั่วไปอาจมีการเพิ่มไอเทมใหม่ๆ เข้ามา ซึ่งไอเทมใหม่ๆ เหล่านี้อาจกลายมาเป็นฟรีเควนท์ไอเทมเซตได้ แต่ด้วยกระบวนการทำงานของอัลกอริทึมบนฐานของเนกาทีฟบอร์เดอร์นี้ไม่มีขั้นตอนการทำงานที่รองรับในส่วนของการค้นหาไอเทมเซตใหม่ที่อยู่นอกเหนือจากฟรีเควนท์ k-ไอเทมเซต และบอร์เดอร์ k-ไอเทมเซต ที่มีอยู่ทำให้ไม่สามารถค้นหากฎความสัมพันธ์สำหรับไอเทม ใหม่ๆ ได้

4. อัลกอริทึมที่มีพื้นฐานการทำงานของไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซต (Expected frequent itemset based algorithm) [7]

ไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตเป็นงานวิจัยที่มีการนำเสนอแนวคิดในการลดจำนวนครั้งในการสแกนฐานข้อมูลเดิม โดยนำความรู้ที่ได้จากการไมน์นิ่งในฐานข้อมูลเดิมทั้งส่วนที่เป็นฟรีเควนท์ไอเทมเซตและในส่วนที่คาดว่าจะกลายมาเป็นฟรีเควนท์ไอเทมเซตได้เมื่อมีการเพิ่มข้อมูลใหม่เข้ามาจำนวนหนึ่งเรียกไอเทมเซตเหล่านี้ว่าไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซต (Expected frequent itemset หรือ promising frequent itemset)

แนวคิดของไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตมี 3 งานวิจัย [7], [21], [22] ที่ได้นำเสนอวิธีการจัดเก็บไอเทมเซตที่แตกต่างจากอัลกอริทึมบอร์เดอร์เซต ซึ่งมีการเก็บไอเทมเซตจำนวนมาก ด้วยเกณฑ์การพิจารณาที่กำหนดเพิ่มขึ้นมาอีก 1 ค่าซึ่งถูกกำหนดโดยผู้ใช้ โดยค่าที่กำหนดขึ้นมานี้จะเป็นค่าที่น้อยกว่าค่าสนับสนุนน้อยที่สุด ในงานวิจัย [7] ได้นำเสนอเกณฑ์นี้ว่า ดีกรีที่ยอมรับหรือโทเลอแรนซ์ดีกรี (tolerance degree: t), $0 < t < s$ (s แทนค่าสนับสนุนน้อยที่สุด), งานวิจัย [21] ได้นำเสนออัลกอริทึมพรีลาจก์ที่มีเกณฑ์การพิจารณาไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตด้วยค่าสนับสนุน 2 ระดับคือค่าสนับสนุนระดับต่ำ (lower support) และค่าสนับสนุนระดับสูง (upper support) โดยค่าสนับสนุนระดับสูงจะมีค่าเท่ากับค่าสนับสนุนน้อยที่สุดงานวิจัยทั้งสองงานข้างต้นจะทำการค้นหากฎความสัมพันธ์ด้วยหลักการของเอฟยูพี โดยค่าโทเลอแรนซ์ดีกรีและค่าสนับสนุนระดับต่ำนั้นขึ้นกับการกำหนดค่าของผู้ใช้ ซึ่งจะมีค่าน้อยกว่าค่าสนับสนุนน้อยที่สุดส่วนงานวิจัยอีกงานคือที่ได้นำเสนออัลกอริทึมโพรมิสซิงฟรีเควนท์ไอเทมเซต ซึ่งใช้ค่าน้อยที่สุดที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซต (min_PL) ที่อาศัยหลักทางสถิติของค่าไอเทมเซตที่เกิดขึ้นด้วยแนวคิดที่ว่าไอเทมเซตที่เกิดขึ้นในฐานข้อมูลเดิมกับฐานข้อมูลใหม่ไม่แตกต่างกันหรือมีความแตกต่างกันเพียงเล็กน้อย [22] โดยจะนำค่าสนับสนุนของ 1 ไอเทมที่มีค่ามากที่สุดในฐานข้อมูลเดิมมาใช้ในการประมาณค่าให้กับไอเทมเซตที่คาดว่าจะกลายมาเป็นฟรีเควนท์ไอเทมเซตดังสมการที่ (2.4)

$$min\_sup_{DB} - \left(\frac{maxsupp}{total\ size} \right) \times inc\_size \leq min\_PL < min\_sup_{DB} \qquad (2.4)$$

โดย maxsupp หมายถึง ค่าสนับสนุนสูงสุดที่พบจากการไมน์นิ่งในฐานข้อมูลเดิม

total size หมายถึง ขนาดของฐานข้อมูลเดิม

inc_size หมายถึง ขนาดของฐานข้อมูลใหม่ที่จะเพิ่ม

การพิจารณาไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตนั้นจะพิจารณาจากไอเทมเซตที่มีค่าสนับสนุนน้อยกว่าค่าสนับสนุนน้อยที่สุด แต่มีค่ามากกว่าค่าโทเลอแรนซ์ดีกรี [7] หรือค่าสนับสนุนระดับต่ำหรือค่าที่คาดว่าไอเทมเจะเป็นฟรีเควนท์

สำหรับงานวิจัยทั้งสามมีแนวคิดที่จะหลีกเลี่ยงการสแกนฐานข้อมูลเดิมดังนี้

1. งานวิจัยได้นำโทเลอแรนซ์ดีกรีมาใช้ [7] จะมีการพิจารณาค่าสนับสนุนของไอเทมเซตที่เกิดขึ้นถ้าไม่สามารถกลายมาเป็นฟรีเควนท์ไอเทมเซตถ้าค่านั้นมีค่ามากกว่าหรือเท่ากับ $d^{+} \times (s-t)$ แล้วไอเทมนั้นๆคาดว่าอาจจะเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุง

โดย d+ แทนขนาดของฐานข้อมูลใหม่ที่เพิ่มเข้ามา

s แทนค่าสนับสนุนน้อยที่สุด

t แทนค่าโทเลอแรนซ์ดีกรี

เมื่อมีการเพิ่มฐานข้อมูลใหม่เข้าค่าโทเลอแรนซ์ดีกรียังคงใช้เกณฑ์เดียวกับการหาในฐานข้อมูลเดิม

2. งานวิจัยพรีลาจก์ [21] จะพิจารณาขนาดของฐานข้อมูลที่ปรับปรุงทั้งหมด (ขนาดของฐานข้อมูลเดิมรวมกับฐานข้อมูลใหม่ที่เพิ่มเข้ามา) นำมาคำนวณเทียบกับค่าสนับสนุนระดับต่ำและค่าสนับสนุนระดับสูง ถ้าขนาดของฐานข้อมูลปรับปรุงน้อยกว่าค่าที่คำนวณได้จะไม่ทำการสแกนฐานข้อมูลทั้งหมด ดังนี้

$$t \le \frac{(S_u - S_l)}{1 - S_u} d$$

โดย t แทน ขนาดของฐานข้อมูลที่เพิ่มเข้ามาใหม่

S_u แทน ค่าสนับสนุนระดับสูง ที่มีค่าเท่ากับค่าสนับสนุนน้อยที่สุด

S_l แทน ค่าสนับสนุนระดับต่ำ

d แทนขนาดของฐานข้อมูลเดิม

เมื่อมีการเพิ่มฐานข้อมูลใหม่เข้าค่าสนับสนุนขั้นต่ำยังคงใช้เกณฑ์เดียวกับการหาในฐานข้อมูลเดิม

3. โพรมิสซิงฟรีเควนท์ไอเทมเซต [22] ได้นำเสนอการคำนวณขนาดของฐานข้อมูลใหม่ที่เพิ่มเข้ามา โดยถ้าขนาดของฐานข้อมูลใหม่ที่เพิ่มเข้ามาอยู่ในช่วงที่คำนวณได้จะทำให้ค้นหากฎความสัมพันธ์ใหม่ได้โดยไม่ต้องทำการสแกนฐานข้อมูลเดิม จากสมการที่ (2.4) สามารถคำนวณขนาดของฐานข้อมูลใหม่ที่เพิ่มเข้ามาได้ด้วยสมการ (2.5) ดังนี้

$$0 < min\_PL < min\_sup$$

$$0 < min\_PL = \frac{max\_sup\ p}{|DB| + inc\_size} * inc\_size < min\_sup$$

$$0 < inc\_size < \frac{min\_sup * |DB|}{max\_sup\ p - min\_sup} \qquad (2.5)$$

ขนาดของข้อมูลที่คำนวณได้จะช่วยในการประมาณค่าของฐานข้อมูลใหม่ที่เพิ่มเข้ามาทำให้สามารถรับประกันได้ว่าด้วยขนาดของส่วนของฐานข้อมูลที่เพิ่มเข้ามาและจากสมมติฐานของสถิติการเกิดของไอเทมไม่แตกต่างหรือแตกต่างกันน้อย ทำให้สามารถนำค่าน้อยที่สุดที่คาดว่าจะเป็นฟรีเควนท์ไอเทมที่ได้มาใช้ในการพิจารณาหาไอเทมเซตที่คาดว่าจะกลายมาเป็นฟรีเควนท์ k-ไอเทมเซตได้ และจะช่วยลดหรือหลีกเลี่ยงการสแกนฐานข้อมูลเดิมลงไปได้ เนื่องจากนำทั้งฟรีเควนท์ k-ไอเทมเซต และโพรมิสซิงฟรีเควนท์ k-ไอเทมเซตที่ได้มาสร้างแคนดิเดท (k+1) ไอเทมเซต เพื่อให้การค้นหาไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตครอบคลุมทุกไอเทมเซตที่คาดว่าจะเกิดเมื่อมีการเพิ่มฐานข้อมูลใหม่เข้ามา

แนวคิดด้านไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตจะขอแสดงตัวอย่างของอัลกอริทึม โพรมิสซิงฟรีเควนท์ไอเทมเซต ดังนี้

ตัวอย่างฐานข้อมูลเดิม ซึ่งประกอบด้วย 10 ทรานแซคชันโดยฐานข้อมูลจะประกอบด้วยไอเทม 5 ตัว คือ {A, B, C, D, E} ด้วยค่าสนับสนุนน้อยที่สุด= 40% ดังแสดงในรูปที่ 2.12

TID	List of item
1	A, B, E
2	B, D
3	B,C
4	A, B,D
5	A, C
6	B, C
7	A, C
8	A, B, C, E
9	A, B, E
10	A,C

Itemset	support
A	7
B	7
C	6
D	2
E	3

รูปที่ 2.12 ทรานแซคชันในฐานข้อมูลเดิมและแคนดิเดท 1 ไอเทมเซต

จากรูปที่ 2.12 พบว่า max_sup มีค่าเท่ากับ 7 จากสมการ (2.4) และ (2.5) สามารถคำนวณขนาดของข้อมูลที่เพิ่มเข้ามาได้คือ

$$\min\_sup = 0.4 * 10 = 4$$

$$inc\_size < \frac{4*10}{7-4} \approx 13$$

ถ้าขนาดของฐานข้อมูลที่เพิ่มใหม่คือ 3 ทรานแซคชัน ดังนั้นค่าน้อยที่สุดที่คาดว่าไอเทมจะกลายฟรีเควนท์สามารถคำนวณได้ดังนี้

$$min\_PL = 4 - \frac{7}{10} * 3 \approx 2$$

จากรูปที่ 2.12 จะได้ฟรีเควนท์ 1 ไอเทมเซต (L_1) คือ {A, B, C} และไอเทมที่คาดว่าจะเป็นฟรีเควนท์ 1 ไอเทมเซต ในที่นี้แทนด้วย PL_1 คือ {D,E} รูปที่ 2.13 แสดงแคนดิเดท 2 ไอเทมเซต (C_2) ที่ได้จากการเชื่อมฟรีเควนท์ 1 ไอเทมเซตและไอเทมที่คาดว่าจะเป็นฟรีเควนท์ 1 ไอเทมเซตเข้าด้วยกันด้วยหลักการเชื่อมของอะพริโอริ รูปที่ 2.13 แสดงฟรีเควนท์ 2 ไอเทมเซต (L_2) และไอเทมที่คาดว่าจะเป็นฟรีเควนท์ 2 ไอเทมเซต (PL_2) ที่ได้หลังจากทำการสแกนฐานข้อมูลเดิมเรียบร้อย ด้วยหลักการเดียวกันกับการหาแคนดิเดท 2 ไอเทมเซต, ฟรีเควนท์ 2 ไอเทมเซตและไอเทมที่คาดว่าจะเป็นฟรีเควนท์ 2 ไอเทมเซต รูปที่2.14 แสดงถึง แคนดิเดท 3 ไอเทมเซต (C_3) และไอเทมที่คาดว่าจะเป็นฟรีเควนท์ 3 ไอเทมเซต (PL_3) ที่ได้โดยใน 3 ไอเทมเซตนี้ไม่ปรากฏว่ามีไอเทมเซตใดเป็นฟรีเควนท์ 3 ไอเทมเซตและสิ้นสุดขั้นตอนในการค้นหา เนื่องจากไม่สามารถสร้างแคนดิเดท 4 ไอเทมเซตได้

C2	support
AB	4
AC	4
AD	1
AE	3
BC	3
BD	2
BE	3
CD	0
CE	1
DE	0

L2	Support
AB	4
AC	4
PL2	support
AE	3
BC	3
BD	2
BE	3

รูปที่ 2.13 แสดง 2 ไอเทมเซตที่เป็นแคนดิเดท, ฟรีเควนท์และไอเทมที่คาดว่าจะเป็นฟรีเควนท์

C3	Support
ABC	1
ABE	3
ACE	1
BCD	0
BCE	0
BDE	0

PL3	support
ABE	3

รูปที่ 2.14 แสดง 3 ไอเทมเซตที่เป็นแคนดิเดท, ฟรีเควนท์และไอเทมที่คาดว่าจะเป็นฟรีเควนท์

เมื่อมีการเพิ่มส่วนของฐานข้อมูลใหม่เข้ามาจะทำการปรับปรุงค่าสนับสนุนของฟรีเควนท์ k-ไอเทมเซต และไอเทมที่คาดว่าจะเป็นฟรีเควนท์ k-ไอเทมเซต โดยจะทำการคำนวณค่า น้อยที่สุดที่คาดว่าจะเป็นฟรีเควนท์ min_PL ของฐานข้อมูลปรับปรุง ใหม่ ดังสมการที่ (2.6)

$$\min\_PL_{DB\cup db} = \min\_\sup_{DB\cup db} - \left(\frac{\max supp}{total\ size} \times inc\_size \right) \qquad (2.6)$$

ข้อดี

1. ลดจำนวนการสแกนฐานข้อมูลเดิม

2. เมื่อมีการเปลี่ยนแปลงจากไอเทมที่คาดว่าจะปีนฟรีเควนท์ไอเทมเซตมาเป็นฟรีเควนท์ไอเทมเซต จะสามารถลดการสแกนฐานข้อมูลเดิมโดยไม่จำเป็นต้องเข้าไปทำการไมน์นิ่งใหม่

ข้อเสีย

1. ถ้าค่าขีดแบ่ง (Threshold) ที่ใช้พิจารณาไอเทมที่คาดว่าจะปีนฟรีเควนท์มีค่าน้อยมาก จะมีผลต่อจำนวนไอเทมเซตที่จัดเก็บและปรับปรุงเมื่อมีข้อมูลใหม่เพิ่มเข้ามาทำให้เปลืองหน่วยความจำในการประมวลผลได้

2.4 ทฤษฎีเบอร์นูลลี (Bernoulli Theorem)

ทฤษฎีทางสถิติที่ใช้ในการพิจารณาการทดลอง ซึ่งผลการทดลอง 1 ครั้งมีผลอย่างใดอย่างหนึ่งใน 2 แบบเท่านั้น คือ ความสำเร็จ (success) หรือ ความไม่สำเร็จ (failure) [38] เช่น โยนเหรียญ 1 อัน จะปรากฏผลเป็นหัวหรือก้อย ซึ่งอาจถือว่าการขึ้นหัวเป็นผลสำเร็จ ส่วนการขึ้นก้อยเป็นความไม่สำเร็จ สำหรับตัวอย่างของการโยนเหรียญนี้ ความน่าจะเป็นที่จะขึ้นหัวเท่ากับ $\frac{1}{2}$ และความน่าจะเป็นที่จะขึ้นก้อยเท่ากับ $\frac{1}{2}$ โดยความน่าจะเป็นที่จะเกิดความสำเร็จเท่ากับ p และ ความน่าจะเป็นที่จะเกิดผลความไม่สำเร็จเท่ากับ q ซึ่ง p+q = 1

การค้นหากฎความสัมพันธ์ได้มีการนำเบอร์นูลลีมาประยุกต์ใช้ในการพิจารณาความน่าจะเป็นที่ไอเทมเซตจะกลายมาเป็นฟรีเควนท์ไอเทมเซต ซึ่งเป็นเหตุการณ์ของความสำเร็จ (p=1) ในขณะที่ความน่าจะเป็นที่ไอเทมเซตไม่สามารถกลายมาเป็นฟรีเควนท์ไอเทมเซตได้จะเป็นความไม่สำเร็จ (q = 1-p) เมื่อทำการเพิ่มรายการข้อมูลใหม่เข้ามาในฐานข้อมูลเดิมด้วยสูตรการคำนวณดังต่อไปนี้

$$P(x) = \binom{n+m}{x} \cdot p^x \cdot (1-p)^{n+m-x}$$

โดย P หมายถึงค่าความน่าจะเป็น

(n+m) หมายถึงขนาดของข้อมูลที่จะทำการทดลอง

p หมายถึงค่าความน่าจะเป็นที่เกิดความสำเร็จจากการทดลอง

(1-p) หมายถึงค่าความน่าจะเป็นที่การทดลองไม่มีความสำเร็จ

x หมายถึงจำนวนครั้งที่การทดลองจะเกิดความสำเร็จ

ในส่วนของการไมน์นิ่งกฎความสัมพันธ์โดยนำการกระจายแบบเบอร์นูลลี (Bernoulli distribution) มาใช้ในการประมาณค่าไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซต โดยไอเทมเซตที่เป็นฟรีเควนท์ไอเทมเซตจะเป็นเหตุการณ์ที่สำเร็จ (success) ส่วนไอเทมเซตใดที่ไม่สามารถเป็นฟรีเควนท์ไอเทมเซตได้ จะเรียกว่า เป็นเหตุการณ์ที่ไม่สำเร็จ (failure)

นอกจากนำเบอร์นูลลีมาใช้ในการทำนายการเกิดไอเทมเซตแล้วยังมีการนำเบอร์นูลลีมาใช้ในการประมาณค่าการเกิดไอเทมเซตด้วย วิธีที่นำมาใช้ในการประมาณค่าที่ได้รับความนิยมมี 2 วิธีคือ เชอร์นอฟฟ์บาวนด์ (Chernoff bound) และทฤษฎีการเข้าสู่ศูนย์กลาง (Central limit theorem)

สำหรับเทคนิคเชอร์นอฟฟ์บาวนด์ได้ถูกนำมาประยุกต์ใช้ในการไมน์นิ่งฐานข้อมูลขนาดใหญ่ [3][4] นอกจากนี้ [3] ได้มีการประยุกต์ทฤษฎีการเข้าสู่ศูนย์กลางมาใช้ในการสุ่มเลือกประชากรสุ่ม เพื่อใช้ไมน์นิ่งหากฎความสัมพันธ์ในฐานข้อมูลขนาดใหญ่ ซึ่งจะช่วยให้สามารถลด

จำนวนขนาดของฐานข้อมูลและสามารถข้อมูลสุ่มที่มีจำนวนเล็กกว่าฐานข้อมูลเดิมมาใช้ในการค้นหากฎความสัมธ์ได้อย่างมีประสิทธิภาพ

แนวคิดนี้พัฒนาขึ้นมาภายใต้ข้อจำกัดของเวลา เนื่องจากเวลามีผลต่อการตัดสินใจเป็นอย่างมาก ดังนั้นข้อมูลที่ใช้จึงเป็นลักษณะของการประมาณค่าที่คาดว่าจะเกิดขึ้นโดยไม่จำเป็นต้องใช้ข้อมูลที่มีความถูกต้อง 100 เปอร์เซ็นต์ เช่น การลงทุนในตลาดหุ้น ถ้านักลงทุนสามารถ ที่จะประมาณค่าของฟรีเควนท์ไอเทมเซต จากข้อมูลในฐานข้อมูลหุ้น (Stock database) ได้อย่างรวดเร็วจะช่วยให้การตัดสินใจที่จะลงทุนได้รวดเร็วและถูกต้อง แทนที่จะต้องทำการค้นหาฟรีเควนท์ไอเทมเซต จากฐานข้อมูลทั้งหมด

การเพิ่มขยายการค้นหากฎความสัมพันธ์ของงานวิทยานิพนธ์นี้ สามารถค้นหากฎความสัมพันธ์ของฐานข้อมูลได้อย่างมีประสิทธิภาพในกรณีของการเพิ่มข้อมูลใหม่เข้าในฐานข้อมูลเดิม โดยทั่วไปฐานข้อมูลเดิมมักจะมีจำนวนทรานแซคชันจำนวนมากกว่าส่วนของข้อมูลใหม่ที่เพิ่มเข้ามา ดังนั้นถ้าข้อมูลที่เกิดขึ้นในฐานข้อมูลอยู่ในสมมติฐานที่ว่าไอเทมเซตที่เกิดในฐานข้อมูลเดิมมีความแตกต่างกันน้อยมากกับการเกิดขึ้นของไอเทมเซตในส่วนของข้อมูลใหม่ที่เพิ่มขึ้นมา ซึ่งภายใต้สมมติฐานนี้เราสามารถคำนวณหาค่าความน่าจะเป็นที่ไอเทมเซตจะเกิดขึ้นเมื่อมีการเพิ่มข้อมูลใหม่เข้ามาได้ด้วยหลักการทฤษฎีของเบอร์นูลลี

จากแนวคิดนี้ทำให้สามารถหาไอเทมเซตที่คาดว่าจะกลายมาเป็นฟรีเควนท์ไอเทมเซตได้โดยอาศัยค่าความน่าจะเป็นที่เกิดขึ้นของไอเทมเซตที่เกิดขึ้นในฐานข้อมูลเดิมซึ่งจะแทนด้วย p เมื่อนำมาคำนวณด้วยจำนวนของข้อมูลที่คาดว่าจะเพิ่มเข้ามาจะทำให้สามารถประมาณค่าความน่าจะเป็นของไอเทมเซตที่จะเกิดขึ้นได้ ซึ่งในส่วนของค่าความน่าจะเป็นของไอเทมเซตที่เกิดในฐานข้อมูลเดิมนี้จะเป็นค่าเฉลี่ยที่เกิดขึ้นในฐานข้อมูลเดิม ที่ตรงกับกฎว่าด้วยจำนวนมาก (Law of large number) ที่กล่าวว่า “ค่าเฉลี่ยตัวแปรสุ่มของตัวอย่างประชากรจำนวนมากจะมีค่าเข้าใกล้ค่าเฉลี่ยของประชากรทั้งหมด” จากที่กล่าวข้างต้นว่าจำนวนรายการข้อมูลของฐานข้อมูลเดิมโดยปกติจะมีจำนวนมาก ดังนั้นในงานวิจัยนี้จึงได้นำค่าความน่าจะเป็นของไอเทมเซตที่เกิดในฐานข้อมูลเดิมมาเป็นค่าความน่าจะเป็นที่คาดว่าไอเทมเซตนั้นๆ จะเกิดขึ้นโดยอาศัยหลักการคำนวณของเบอร์นูลลี ผลที่ได้จากแนวคิดนี้จะทำให้สามารถลดจำนวนไอเทมเซตที่จะต้องนำไปสแกนในฐานข้อมูลเดิมและสามารถหาฟรีเควนท์ไอเทมเซตใหม่ได้อย่างมีประสิทธิภาพโดยนำทฤษฎีเบอร์นูลลีมาประยุกต์ใช้ดังจะกล่าวรายละเอียดต่างๆ ในบทต่อไป

บทที่ 3
อัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น

ปัจจุบันข้อมูลและจำนวนข้อมูลที่จัดเก็บในฐานข้อมูลมีขนาดใหญ่มากและมีการขยายตัวเพิ่มจำนวนขึ้นตลอดเวลา การทำดาต้าไมน์นิ่งเป็นการสกัดสารสนเทศจากฐานข้อมูลมาใช้ทำให้ทราบถึงความรู้ที่ซ่อนอยู่ที่สามารถนำมาใช้ให้เกิดประโยชน์ในด้านการบริหาร ช่วยในการตัดสินใจ วางแผนกลยุทธ์ และ อื่นๆ ซึ่งเป็นการเพิ่มคุณค่าให้กับฐานข้อมูลที่มีอยู่

การค้นหากฎความสัมพันธ์เป็นเทคนิคหนึ่งในการทำดาต้าไมน์นิ่งที่ใช้ในการวิเคราะห์หรือทำนายปรากฏการณ์ต่างๆ ของข้อมูลในฐานข้อมูลขนาดใหญ่ที่เกิดขึ้นร่วมกัน โดยการค้นหากฎความสัมพันธ์ในฐานข้อมูลนั้นจะเริ่มจากการหาฟรีเควนท์ไอเทมเซต ซึ่งหมายถึง เซตของไอเทมที่เกิดขึ้นร่วมกันและมีค่าสนับสนุนที่มากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุดที่ผู้ใช้กำหนด จากนั้นจึงนำฟรีเควนท์ไอเทมเซตมาสร้างเป็นกฎความสัมพันธ์

เมื่อรายการข้อมูลต่างๆ มีการปรับเปลี่ยน เช่นการเพิ่ม, ลบหรือแก้ไขข้อมูลในฐานข้อมูล ทำให้ต้องทำการปรับฐานข้อมูลใหม่ให้มีความเป็นปัจจุบัน ซึ่งทำให้เกิดการเปลี่ยนแปลงกฎความสัมพันธ์ที่ได้ค้นหาไว้แล้ว

การเพิ่มขยายการค้นหากฎความสัมพันธ์เป็นอัลกอริทึมที่ใช้ในปรับปรุงกฎความสัมพันธ์ของข้อมูลในฐานข้อมูลขนาดใหญ่ เมื่อมีการเพิ่มทรานแซคชันใหม่เข้าไปในฐานข้อมูลเดิม (DB) ในที่นี้จะเรียกส่วนของรายการข้อมูลใหม่ที่เพิ่มเข้าไปนี้ว่า ฐานข้อมูลใหม่ (db) ซึ่งทรานแซคชันที่เพิ่มเข้ามานี้จะมีผลต่อกฎความสัมพันธ์ที่ได้จากการทำดาต้าไมน์นิ่งในฐานข้อมูลเดิม โดยอาจทำให้กฎที่มีอยู่ไม่มีความถูกต้องเมื่อทำการค้นหากฎความสัมพันธ์ที่ปรากฏในฐานข้อมูลที่ได้รับการปรับปรุงใหม่ให้เป็นปัจจุบันทั้งหมด (updated database: UP)

อัลกอริทึมในการค้นหากฎความสัมพันธ์เมื่อฐานข้อมูลได้รับการปรับปรุงใหม่ให้เป็นปัจจุบันทั้งหมดประกอบด้วยการทำงานหลักๆ 3 เฟสด้วยกันคือ

1. การปรับปรุงค่าสนับสนุนให้กับฟรีเควนท์ไอเทมเซตของฐานข้อมูลเดิมทั้งหมด
2. การพรุนหรือตัดฟรีเควนท์ไอเทมเซตที่มีค่าสนับสนุนน้อยกว่าค่าสนับสนุนน้อยที่สุดเมื่อทำการปรับปรุงฐานข้อมูลใหม่ออกไป
3. การค้นหาฟรีเควนท์ไอเทมเซตใหม่ที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุดเมื่อทำการปรับปรุงฐานข้อมูลใหม่

สำหรับเฟสที่ 1 และ 2 ซึ่งเป็นการปรับปรุงค่าสนับสนุนของฟรีเควนท์ไอเทมเซตที่ปรากฏในฐานข้อมูลเดิมนั้นรวมถึงการตัดไอเทมเซตที่ไม่สามารถเป็นฟรีเควนท์ไอเทมเซตออกไป นั้นเป็นส่วนที่สามารถทำได้เนื่องจากทราบค่าสนับสนุนของฟรีเควนท์ไอเทมเซตที่พบจากการไมน์นิ่งฐานข้อมูลเดิม ดังนั้นเมื่อมีส่วนของข้อมูลใหม่เพิ่มเข้ามา หลังจากที่มีการสแกนส่วนของข้อมูลใหม่แล้วจะทราบค่าสนับสนุนที่นำมาบวกเพิ่มทำให้ทราบค่าสนับสนุนที่ผ่านการปรับปรุงได้อย่างถูกต้อง

แต่ในเฟสที่ 3 ซึ่งเป็นการส่วนของฟรีเควนท์ไอเทมเซตใหม่ที่ไม่ได้เป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิม การจะทำการค้นหาฟรีเควนท์ไอเทมเซตใหม่นี้นอกจากจะต้องสแกนฐานข้อมูลใหม่เพื่อหาค่าฟรีเควนท์ไอเทมเซตใหม่ที่พบแล้วยังต้องทำการสแกนฐานข้อมูลเดิมซึ่งมีจำนวนมากเพื่อหาค่าสนับสนุนสำหรับฟรีเควนท์ไอเทมเซตที่เกิดใหม่สำหรับฐานข้อมูลปรับปรุงอีกด้วย ดังนั้นงานในส่วนนี้เป็นส่วนที่มีความสำคัญอย่างมากสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์

เมื่อฐานข้อมูลที่ได้จากการปรับปรุงได้ผ่านกระบวนการค้นหากฎความสัมพันธ์แล้ว อาจทำให้เกิดการสแกนฐานข้อมูลเดิมอีกครั้ง โดยทั่วไปขนาดของฐานข้อมูลเดิมมักจะมีขนาดใหญ่กว่าฐานข้อมูลใหม่ที่เพิ่มเข้ามา ดังนั้นการสแกนฐานข้อมูลเดิมจะมีผลต่อความถูกต้องของกฎความสัมพันธ์, เวลาที่ใช้ในการประมวลผล รวมถึงทำให้เกิดการสิ้นเปลืองทรัพยากรในการค้นหาฟรีเควนท์ไอเทมเซตที่พบในฐานข้อมูลใหม่แต่เป็นสมอลไอเทมเซตในฐานข้อมูลเดิม

โดยทั่วไปอัลกอริทึมต่างๆ ที่ใช้ในการเพิ่มขยายการค้นหากฎความสัมพันธ์จะเก็บเฉพาะไอเทมเซตและค่าสนับสนุนของไอเทมเซตที่พบว่าเป็นฟรีเควนท์ไอเทมเซตเท่านั้น ดังนั้นเพื่อให้ทราบค่าสนับสนุนที่แท้จริงของไอเทมเซตใหม่ทำให้ต้องสแกนฐานข้อมูลเดิมซึ่งเป็นการเสียเวลาในการค้นหาและสิ้นเปลืองทรัพยากร

อัลกอริทึมอะพริโอริเป็นอัลกอริทึมที่ได้รับความนิยมในการค้นหากฎความสัมพันธ์ของข้อมูลในฐานข้อมูลขนาดใหญ่ การไมน์นิ่งเป็นลักษณะการค้นหาแบบทีละระดับ (Levelwise search) ดังนั้นในแต่ละ k-ไอเทมเซตจะเก็บฟรีเควนท์ไอเทมเซตที่เกิดจากการไมน์นิ่ง เพื่อใช้ในการสร้างกฎความสัมพันธ์ที่ได้จากการดึงความรู้ที่ซ่อนอยู่ในฐานข้อมูล

การที่ข้อมูลต่างๆ ในฐานข้อมูลไม่คงที่สามารถเกิดการเปลี่ยนแปลงได้ตลอดเวลา ในลักษณะที่เรียกว่า ฐานข้อมูลแบบไดนามิค (Dynamic database) นั้น ซึ่งทำให้ต้องไมน์นิ่งฐานข้อมูลใหม่เพื่อให้ได้กฎความสัมพันธ์ที่มีความถูกต้องสมบูรณ์

การไมน์นิ่งฐานข้อมูลด้วยหลักการทำงานของอัลกอริทึมอะพริโอริถึงแม้ผลการค้นหากฎความสัมพันธ์ที่ได้จะมีความถูกต้องสมบูรณ์แต่กระบวนการค้นหากฎความสัมพันธ์นั้นจะต้องเริ่มค้นหากฎความสัมพันธ์ใหม่ทั้งหมดในฐานข้อมูลที่ปรับปรุง โดยไม่ได้นำความรู้เดิมที่ได้จากการไมน์นิ่งฐานข้อมูลเดิมมาใช้ ทำให้เสียเวลาในการค้นหาฟรีเควนท์ไอเทมเซตใหม่ทั้งหมด

จากปัญหาการไมน์นิ่งฐานข้อมูลทั้งหมดของอัลกอริทึมอะพริโอริที่เกิดขึ้น ทำให้มีงานวิจัยต่างๆ หาวิธีที่จะปรับปรุงให้การเพิ่มขยายการค้นหากฎความสัมพันธ์ให้มีประสิทธิภาพมากขึ้น อัลกอริทึมแรกที่นำเสนอวิธีการแก้ปัญหาด้วยการนำความรู้ที่ได้จากการไมน์นิ่งฐานข้อมูลเดิมมาใช้ให้เกิดประโยชน์ คือ อัลกอริทึมเอฟยูพี (FUP) โดยนำฟรีเควนท์ k-ไอเทมเซตเดิมที่ได้จากการไมน์นิ่งมาทำการปรับปรุงค่าสนับสนุนที่ปรากฏในฐานข้อมูลใหม่ ซึ่งแนวคิดนี้สามารถลดการค้นหาฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิมได้ และเมื่อพบฟรีเควนท์ k-ไอเทมเซตในส่วนของฐานข้อมูลใหม่จะมีการนำฟรีเควนท์ kไอเทมเซตใหม่นี้มาทำการสแกนหาค่าสนับสนุนในฐานข้อมูลเดิม เพื่อให้ได้ฟรีเควนท์ k-ไอเทมที่ถูกต้องสำหรับสร้างกฎความสัมพันธ์

ความรู้ที่ได้จากการไมน์นิ่งในฐานข้อมูลในอัลกอริทึมอะพริโอริและเอฟยูพีนั้นส่วนใหญ่จะค้นหาเฉพาะฟรีเควนท์ไอเทมเซตเท่านั้น แต่เมื่อมีการเพิ่มข้อมูลใหม่เข้ามาอาจทำให้ไอเทมเซตที่ไม่เป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิมกลายมาเป็นฟรีเควนท์ไอเทมเซตได้ Thomas et. al. [18] และFeldman et. al. [19] ได้จึงได้นำแนวคิดของเนกาทีฟบอร์เดอร์ (Negative border) มาพัฒนาอัลกอริทึมโดยจะทำการเก็บข้อมูลของฟรีเควนท์ k-ไอเทมเซตและส่วนของแคนดิเดทไอเทมเซตที่ไม่ใช่ฟรีเควนท์ไอเทมเซตไว้ เรียกไอเทมเซตนี้ว่าเนกาทีฟบอร์เดอร์ k-ไอเทมเซต ซึ่งสามารถลดการสแกนฐานข้อมูลเดิม อัลกอริทึมนี้จะทำงานได้ดีในกรณีที่ไม่พบว่ามีฟรีเควนท์ k-ไอเทมเซตใหม่เกิดขึ้น แต่ในทางกลับกันถ้าพบว่ามีฟรีเควนท์ k-ไอเทมเซตใหม่เกิดขึ้นจะต้องสแกนฐานข้อมูลปรับปรุงเพื่อค้นหาฟรีเควนท์และเนกาทีฟบอร์เดอร์ k-ไอเทมเซตทั้งหมด งานวิจัย [29] ได้ทำการเปรียบเทียบประสิทธิภาพการทำงาน ระหว่างอัลกอริทึมเนกาทีฟบอร์เดอร์เทียบกับอัลกอริทึมอะพริโอริ และพบว่าการทำงานของอัลกอริทึมเนกาทีฟบอร์เดอร์เหมาะกับไอเทมที่มีจำนวนไม่มากและเวลาในการประมวลผล (Execution time) ของเนกาทีฟบอร์เดอร์ในกรณีที่พบฟรีเควนท์ k-ไอเทมเซตใหม่และต้องสแกนฐานข้อมูลทั้งหมดนั้นใช้เวลาไม่แตกต่างหรืออาจจะนานกว่าการค้นหากฎความสัมพันธ์ด้วยอัลกอริทึมอะพริโอริ นอกจากนี้ทั้งงานวิจัยของ Thomas et. al. และFeldman et. al. ไม่มีการนำเสนอในส่วนของการค้นหาฟรีเควนท์ k-ไอเทมเซตใหม่ซึ่งอาจเกิดขึ้นได้ในฐานข้อมูลใหม่ ในกรณีที่ค่าสนับสนุนน้อยที่สุดมีค่าน้อย ซึ่งอาจทำให้เกิดไอเทมเซตใหม่ขึ้นได้

เพื่อลดปัญหาการจัดเก็บข้อมูลและการสแกนฐานข้อมูลใหม่ทั้งหมดในกรณีที่พบฟรีเควนท์ไอเทมเซตใหม่ Hong et. al. [21] และ Amornchewin และ Kreesuradej [22] ได้นำเสนอแนวคิดในการค้นหาไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ เพื่อลดปัญหาการจัดเก็บข้อมูลทุกตัวที่เป็นสมาชิกของแคนดิเดท k-ไอเทมเซตที่ไม่เป็นฟรีเควนท์ k-ไอเทมเซตแนวคิดนี้จะมีการนำเสนอการหาไอเทมเซตที่มีโอกาสจะกลายเป็นฟรีเควนท์เมื่อมีฐานข้อมูลใหม่จำนวนหนึ่งเพิ่มเข้ามา ทั้งนี้เพื่อลดหรือหลีกเลี่ยงการสแกนฐานข้อมูลเดิมงานวิจัยทั้งสองได้นำเสนอการคำนวณหาขนาดของฐานข้อมูลใหม่ที่สามารถใช้ในการประมาณค่าของไอเทมที่คาดว่าจะเป็นฟรีเควนท์

ไอเทมเซต โดยจะให้ความสำคัญกับไอเทมที่คาดว่าจะเป็นฟรีเควนท์ k-ไอเทมเซต เช่นเดียวกับฟรีเควนท์ k-ไอเทมเซตยกเว้นในส่วนของการสร้างกฎความสัมพันธ์ที่จะใช้ฟรีเควนท์ k-ไอเทมเซต ($k \geq 2$) มาสร้างกฎความสัมพันธ์ ดังนั้นไอเทมที่คาดว่าจะเป็นฟรีเควนท์ k-ไอเทมเซต จะถูกนำมาใช้ในการสร้าง แคนดิเดท k-ไอเทมเซตในขั้นตอนของการเชื่อมด้วยหลักการเดียวกับอัลกอริทึมอะพริโอริ ซึ่งทำให้จำนวนไอเทมเซตที่ต้องจัดเก็บมีจำนวนมาก

สำหรับงานวิจัยนี้จะเป็นการนำเสนอวิธีการแก้ปัญหาการทำดาต้าไมน์นิ่งในการหากฎความสัมพันธ์ที่ได้จากฐานข้อมูลที่ปรับปรุงใหม่จากการเพิ่มรายการข้อมูลเข้าไปในฐานข้อมูลเดิมเพื่อลดจำนวนครั้งและจำนวนไอเทมเซตที่จะต้องสแกนในฐานข้อมูลเดิม ตารางที่3.1แสดงสัญลักษณ์และความหมายของสัญลักษณ์ที่ใช้ในอัลกอริทึม โดยขั้นตอนการทำงานของอัลกอริทึมจะนำเสนอในส่วนต่อไป

ตารางที่ 3.1 แสดงรายการสัญลักษณ์ที่ใช้สำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์เมื่อมีฐานข้อมูลใหม่เพิ่มเข้ามา

สัญลักษณ์	ความหมาย
DB	ฐานข้อมูลเดิม (Original database)
db	ฐานข้อมูลใหม่ที่เพิ่ม (Increment database)
UP	ฐานข้อมูลปรับปรุง (Updated database)
$\|DB\|$	ขนาดของฐานข้อมูลเดิม (original database size)
$\|db\|$	ขนาดของฐานข้อมูลใหม่ที่เพิ่ม (increment database size)
k	จำนวนไอเทมเซต (Number of itemset)
σ; min_sup	ค่าสนับสนุนน้อยที่สุด (minimum support)
ρ	ค่าคาดหวังน้อยที่สุดที่ไอเทมจะกลายเป็นฟรีเควนท์ (Minimum expected frequent)
$prob_{EF}$	ค่าความน่าจะเป็นน้อยที่สุดที่คาดว่าไอเทมเซตจะกลายเป็นฟรีเควนท์(Minimum expected frequent probability)
C_k	แคนดิเดท k- ไอเทมเซต (Candidate k- itemset)
F_k	ฟรีเควนท์ k-ไอเทมเซต (Frequent k -itemset)
EF_k	ไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ (Expected frequent k-itemset)

3.1 การประมาณค่าไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ไอเทมเซตโดยใช้หลักความน่าจะเป็น

งานวิจัยนี้เป็นการนำเสนอแนวคิดของการใช้ค่าความน่าจะเป็นมาช่วยในการประมาณค่าไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ไอเทมเซตโดยนำทฤษฎีทางสถิติของเบอร์นูลลีที่ใช้ใน

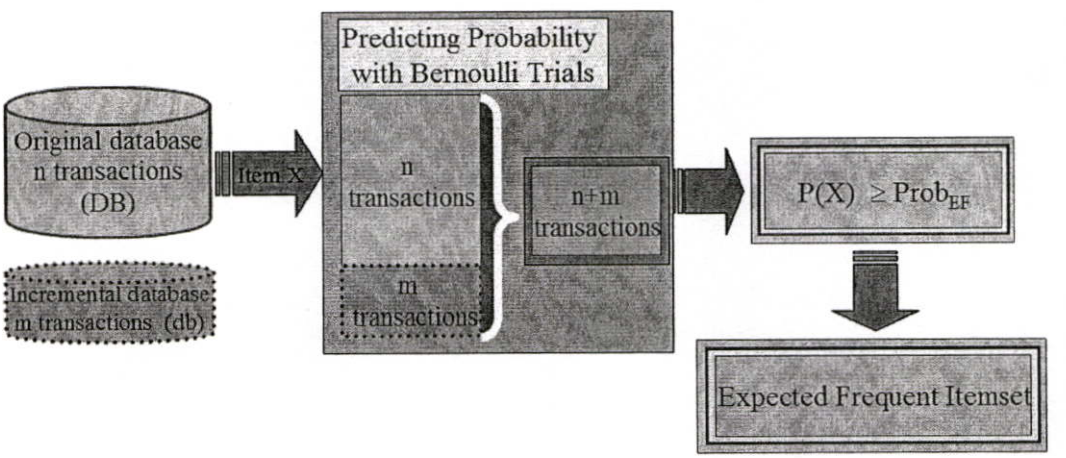
การพิจารณาการทดลองมาประยุกต์ใช้ในการพิจารณาการเกิดของฟรีเควนท์ไอเทมเซตซึ่งในการทดลองจะประกอบด้วย 2 ผลการทดลองอย่างใดอย่างหนึ่งใน 2 เหตุการณ์เท่านั้นคือ เหตุการณ์คือที่ประสบความสำเร็จหรือความไม่สำเร็จ ในที่นี้ถ้าพบว่าไอเทมเซตใดที่เป็นฟรีเควนท์ไอเทมเซต หมายถึงเหตุการณ์ที่เกิดความสำเร็จ และเหตุการณ์ที่ไอเทมเซตใดเป็นสมอลล์ไอเทมเซตหรือไม่ผ่านค่าสนับสนุนน้อยที่สุดเป็นเหตุการณ์ที่ ไม่สำเร็จ

ด้วยแนวคิดของกฎว่าด้วยจำนวนมาก (Law of large number) ของเบอร์นูลลีที่ได้กล่าวว่า "ค่าเฉลี่ยตัวแปรสุ่มของตัวอย่างประชากรจำนวนมากจะมีค่าเข้าใกล้ค่าเฉลี่ยของประชากรทั้งหมด" สำหรับในงานวิจัยนี้ได้นำแนวคิดนี้มาประยุกต์ใช้ในการประมาณค่าของไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์โดยนำค่าสถิติการเกิดของไอเทมเซตต่างๆ ที่ปรากฏในฐานข้อมูลเดิม ซึ่งโดยปกติจะประกอบด้วยทรานแซคชันจำนวนมาก มาเป็นค่าเฉลี่ยของไอเทมเซตที่ปรากฏในฐานข้อมูลแทนด้วยค่า p ซึ่งสามารถคำนวณ ได้ดังนี้

$$p=\frac{support\ count}{|DB|}$$

โดย p หมายถึง ค่าความน่าจะเป็นที่ไอเทมเซตนั้นปรากฏในฐานข้อมูล

$|DB|$ หมายถึงขนาดฐานข้อมูลเดิม ในที่นี้คือ m ทรานแซคชัน



รูปที่ 3.1 การทำนายค่าคาดหวังของไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ด้วยการทดลองแบบเบอร์นูลลี (Bernoulli trials)

ถ้าข้อมูลที่เกิดขึ้นในฐานข้อมูลอยู่ในสมมติฐานที่ว่าไอเทมเซตที่เกิดในฐานข้อมูลเดิมมีความแตกต่างกันน้อยมากกับการเกิดขึ้นของไอเทมเซตในส่วนของข้อมูลใหม่ที่เพิ่มขึ้นมา ซึ่งภายใต้สมมติฐานนี้เราสามารถคำนวณหาค่าความน่าจะเป็นที่ไอเทมเซตจะเกิดขึ้นเมื่อมีการเพิ่มข้อมูลใหม่เข้ามาได้ด้วยหลักการทฤษฎีของเบอร์นูลลี

จากทฤษฎีของเบอร์นูลลีนี้ภายใต้สมมติฐานที่ว่าไอเทมเซตที่เกิดในฐานข้อมูลเดิมมีความแตกต่างกันน้อยมากกับการเกิดขึ้นของไอเทมเซตในส่วนของข้อมูลใหม่ที่เพิ่มขึ้นมาสามารถนำมาใช้ในการประมาณค่าของการเกิดของไอเทมเซตต่างๆ ที่มีโอกาสจะกลายเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงได้ ดังนั้นเมื่อมีส่วนของข้อมูลใหม่ที่เพิ่มเข้ามาขนด n ทรานแซคชันเพิ่มเข้าไปในฐานข้อมูลเดิม ขนาด m ทรานแซคชัน ด้วยจำนวนค่าสนับสนุน x จำนวนที่จะทำให้ไอเทมนี้เป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุง ดังแสดงในรูปที่ 3.1 ในที่นี้ ค่าสนับสนุน x สามารถคำนวณได้ดังนี้

$$x = (min\_sup * (|DB|+|db|) - support\ count_{DB})$$

จากค่าต่างๆ ข้างต้นสามารถนำมาคำนวณค่าความน่าจะเป็นที่ไอเทม A จะเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงขนาด n+m ทรานแซคชัน ได้ดังสมการ (3.1)

$$P(A)=\binom{n+m}{x}\cdot p^{x}\cdot(1-p)^{n+m-x} \qquad (3.1)$$

การพิจารณาว่าไอเทมเซตใดจะเป็นฟรีเควนท์ไอเทมเซตได้นั้นจะดูได้จากค่าสนับสนุนของไอเทมเซตนั้นที่ปรากฏในฐานข้อมูลมีค่ามากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุดที่ผู้ใช้กำหนด เนื่องจากฐานข้อมูลมีขนาดใหญ่มากและส่วนของข้อมูลใหม่ที่เพิ่มเข้ามามักมีขนาดที่เล็กกว่ามากทำให้ไอเทมเซตที่ปรากฏในฐานข้อมูลใหม่นั้นจะมีความแตกต่างหรือมีความเปลี่ยนแปลงน้อยมากจากฐานข้อมูลเดิม ดังนั้นด้วยขนาดของฐานข้อมูลเดิมที่มีขนาดใหญ่และมีจำนวนมากที่จะทำให้เราสามารถนำค่าความน่าจะเป็นของไอเทมเซตที่ปรากฏในฐานข้อมูลเดิมนี้มาใช้ในการประมาณค่าให้กับไอเทมเซตที่มีค่าสนับสนุนที่ปรากฏในฐานข้อมูลเดิมน้อยกว่าค่าสนับสนุนน้อยที่สุดในกรณีที่มีการเพิ่มข้อมูลใหม่ขนาดหนึ่งเข้ามา ซึ่งอาจมีผลทำให้ไอเทมเซตนั้นมีโอกาสที่จะกลายเป็นฟรีเควนท์ไอเทมเซตได้ ค่าความน่าจะเป็นที่ใช้ในการประมาณค่าไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์นี้จะเป็นค่าที่ผู้ใช้กำหนดขึ้นเพิ่มอีก 1 ค่านอกเหนือจากค่าสนับสนุนน้อยที่สุดและค่าความเชื่อมั่นน้อยที่สุด ในที่นี้ค่าความน่าจะเป็นที่ผู้ใช้กำหนดเรียกว่า ค่าความน่าจะเป็นน้อยที่สุดที่คาดว่าไอเทมเซตจะกลายเป็นฟรีเควนท์ ($Prob_{EF}$) ซึ่งเป็นค่าประมาณการที่ผู้ใช้ยอมรับได้ว่าไอเทมเซตนี้มีโอกาสที่เกิดขึ้นอย่างน้อย k-ค่า เมื่อมีส่วนของข้อมูลใหม่ที่เพิ่มเข้าขนาด n ทรานแซคชัน เพิ่มเข้ามาในฐานข้อมูลเดิม ขนาด m ทรานแซคชัน ด้วยความน่าจะเป็น p ซึ่งเป็นค่าความน่าจะเป็นที่เกิดจริงในฐานข้อมูลเดิม โดยในแต่ละไอเทมเซตสามารถคำนวณหาค่าความน่าจะเป็นที่จะเกิดในฐานข้อมูลปรับปรุงอย่างน้อย k-ค่าได้ดังสมการที่ (3.2)

$$P(x<k)_{item} = \sum_{x=0}^{k-1} \binom{n+m}{x} p^{x} (1-p)^{n+m-x} \quad (3.2)$$

(n+m) หมายถึง ขนาดของข้อมูลของฐานข้อมูลปรับปรุง ด้วยขนาดของฐานข้อมูลเดิม (|DB|) ขนาด m ทรานแซคชัน และ ส่วนของฐานข้อมูลใหม่ (|db|) ขนาด n ทรานแซคชัน

p หมายถึง ค่าความน่าจะเป็นที่เกิดความสำเร็จจากการทดลอง ในที่นี้หมายถึงค่าความน่าจะเป็นที่พบไอเทมเซตนั้นๆในฐานข้อมูลซึ่งคำนวณได้ดังนี้

$$p = \frac{support\ count}{|DB|}$$

(1-p) หมายถึง ค่าความน่าจะเป็นที่การทดลองไม่มีความสำเร็จในที่นี้หมายถึงค่าความน่าจะเป็นที่ไม่พบไอเทมเซตนั้นๆ ในฐานข้อมูล

k หมายถึง จำนวนค่าสนับสนุนที่เพิ่มเข้าแล้วจะทำให้เกิดความสำเร็จ คำนวณได้ดังนี้

$$k = min\_sup_{UP} - support\ count_{DB}$$

สำหรับค่าความน่าจะเป็นที่คำนวณได้จากสมการ (3.2) จะเป็นค่าความน่าจะเป็นอย่างน้อยที่จะเกิดด้วยค่าสนับสนุนอย่างน้อย k ค่า เมื่อมีการเพิ่มฐานข้อมูลใหม่ขนาด n ทรานแซคชันเข้าไป ดังนั้นค่าความน่าจะเป็นของไอเทมเซตนี้จะกลายเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงด้วยค่าความน่าจะเป็นเท่าใด สามารถคำนวณได้ดังสมการ (3.3)

$$P(x \geq k)_{item} = 1 - P(x<k)_{item} \quad (3.3)$$

โดย $P(x \geq k)_{item}$ หมายถึง ความน่าจะเป็นที่ไอเ ทมจะมีโอกาสที่ค่าสนับสนุน x มีค่ามากกว่าค่าสนับสนุน k

k หมายถึง ค่าสนับสนุนอย่างน้อยที่สุดที่จะทำให้ไอเทมเซตนั้นๆ มีกลายมาเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุง ซึ่งคำนวณได้จากค่าสนับสนุนน้อยที่สุดของฐานข้อมูลปรับปรุง (min_sup_{UP}) ลบกับค่าสนับสนุนของไอเทมเซตที่เกิดขึ้นจริงในฐานข้อมูล ($support\ count_{item}$) โดยค่ากรณีที่ค่า $k < 0$ จะถูกปรับให้เป็น 0

ตัวอย่างค้นหาฟรีเควนท์ไอเทมเซตและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ด้วยอัลกอริทึมการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นในฐานข้อมูลเดิมซึ่งประกอบไปด้วยรายการข้อมูล 10 ทรานแซคชน ด้วยค่าสนับสนุนน้อยที่สุดคือ 40 เปอร์เซ็นต์ดังนั้นไอเทมเซตที่มีค่าสนับสนุนมากกว่าหรือเท่ากับ 4 จะเป็นฟรีเควนท์ไอเทมเซต และกำหนดให้จำนวนฐานข้อมูลใหม่ที่คาดว่าจะเพิ่มเข้ามาคือ 5 ทรานแซคชัน ค่าความน่าจะเป็นน้อยที่สุดที่คาดว่า

ไอเทมเซตจะกลายเป็นฟรีเควนท์คือ 0.10 ($Prob_{EF}$ = 0.10) กระบวนการในการค้นหาฟรีเควนท์และไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ประกอบด้วยส่วนสำคัญ 2 เฟสคือ

เฟสที่ 1 เริ่มจากการค้นหาแคนดิเดท 1- ไอเทมเซตและค่าสนับสนุนของแคนดิเดท 1-ไอเทมเซตโดยการสแกนฐานข้อมูลดังรูปที่ 3.2

TID	List of item
1	A, B, E
2	B, D
3	B,C
4	A, B,D
5	A, C
6	B, C
7	A, C
8	A, B, C, E
9	A, B, E
10	A,C

Itemset	support
A	7
B	7
C	6
D	2
E	3

รูปที่ 3.2 ตัวอย่างของฐานข้อมูลเดิม และแคนดิเดท 1-ไอเทมเซตของฐานข้อมูลเดิม

เฟสที่ 2 คำนวณค่าความน่าจะเป็นของไอเทมเซตแต่ละตัว โดยนำค่าสนับสนุนที่พบในฐานข้อมูลมาใช้ในการคำนวณค่าความน่าจะเป็น ในที่นี้กำหนดฐานข้อมูลใหม่ที่จะเพิ่มเข้ามาคือ 5 ทรานแซคชัน โดยการคำนวณจะใช้ทฤษฎีของเบอร์นูลลีในสมการที่ (3.3) มาหาค่าความน่าจะเป็นดังนี้

$$P(x \geq k) = 1 - P(x < k)$$

$$P(x < k) = \sum_{x=0}^{k-1} \binom{n}{x} p^x (1-p)^{n-x}$$

โดยค่าที่นำมาใช้ในการคำนวณมีดังนี้คือ

- ค่าความน่าจะเป็นที่พบไอเทมเซต (p) ในฐานข้อมูล = ค่าสนับสนุนหารด้วยขนาดของฐานข้อมูล
- ค่าความน่าจะเป็นที่ไม่พบไอเทมเซต (q) ในฐานข้อมูล = 1- p ดังแสดงค่า p และ q ในตารางที่ 3.2
- ขนาดของฐานข้อมูล n ที่นำมาคำนวณ = |DB| + |db| = 15
- ค่าสนับสนุนน้อยที่สุดสำหรับฐานข้อมูลปรับปรุง คำนวณได้คือ = $\frac{40}{100} * 15 = 6$

- ค่า k ในที่นี้หมายถึง ค่าสนับสนุนที่อย่างน้อยที่จะทำให้ไอเทมเซตนั้นๆ เป็นฟรีเควนท์ไอเทมเซต ในที่นี้คือ สนับสนุนน้อยที่สุดสำหรับฐานข้อมูลปรับปรุง ซึ่งคำนวณได้ = 6
- ค่า x หมายถึง ค่าสนับสนุนที่น้อยกว่าค่า k ในที่นี้คือ ค่า 0 – 5

ตารางที่ 3.2 แสดงค่าความน่าจะเป็นที่พบไอเทมเซตและค่าความน่าจะเป็นที่ไม่พบไอเทมเซต

ไอเทมเซต	ค่าความน่าจะเป็นที่พบไอเทมเซต (p)	ค่าความน่าจะเป็นที่ไม่พบไอเทมเซต (q)
A	7/10	3/10
B	7/10	3/10
C	6/10	4/10
D	2/10	8/10
E	3/10	7/10

ตัวอย่างการคำนวณค่าความน่าจะเป็นของแต่ละไอเทมเซตถ้ามีข้อมูลใหม่จำนวน 5 ทรานแซคชันเพิ่มเข้ามาด้วยค่าสนับสนุนน้อยที่สุด 40เปอร์เซ็นต์แสดงดังรูปที่ 3.3

$$P(x\geq 6)_A = 1 - \sum_{x=0}^{5} \binom{15}{x} \frac{7}{10}^{x} \frac{3}{10}^{15-x} = 1$$

$$P(x\geq 6)_B = 1 - \sum_{x=0}^{5} \binom{15}{x} \frac{7}{10}^{x} \frac{3}{10}^{15-x} = 1$$

$$P(x\geq 6)_C = 1 - \sum_{x=0}^{5} \binom{15}{x} \frac{6}{10}^{x} \frac{4}{10}^{15-x} = 1$$

$$P(x\geq 6)_D = 1 - \sum_{x=0}^{5} \binom{15}{x} \frac{2}{10}^{x} \frac{8}{10}^{15-x} = 0.06$$

$$P(x\geq 6)_E = 1 - \sum_{x=0}^{5} \binom{15}{x} \frac{3}{10}^{x} \frac{7}{10}^{15-x} = 0.28$$

รูปที่ 3.3 ตัวอย่างการคำนวณหาค่าความน่าจะเป็นของแคนดิเดท 1-ไอเทมเซต

สำหรับงานวิจัยนี้จะเป็นการนำเสนออัลกอริทึมที่มีประสิทธิภาพในการไมน์นิ่งฐานข้อมูล โดยสามารถลดการสแกนฐานข้อมูลใหม่ทั้งหมด จากแนวคิดที่มีการนำความรู้จากการไมน์นิ่งฐานข้อมูลเดิม ด้วยการจัดเก็บทั้งส่วนที่เป็นฟรีเควนท์ k-ไอเทมเซตและส่วนที่คาดว่าจะกลายเป็นฟรีเควนท์ k-ไอเทมเซตโดยอาศัยหลักการของความน่าจะเป็นเข้ามาช่วยในการกรองข้อมูลที่คาดว่า

จะกลายเป็นฟรีเควนท์ k-ไอเทมเซตจริงเมื่อมีการเพิ่มฐานข้อมูลใหม่เข้ามา นอกจากนี้ยังสามารถค้นหากฎความสัมพันธ์ที่เกิดจากไอเทมใหม่ที่เกิดขึ้นในฐานข้อมูลใหม่ได้อย่างถูกต้องสมบูรณ์

3.2 อัลกอริทึมในการค้นหากฎความสัมพันธ์ในฐานข้อมูลเดิม

การค้นหากฎความสัมพันธ์ระหว่างข้อมูลที่นำเสนอในงานวิจัยนี้จะใช้หลักการการหากฎความสัมพันธ์ของอัลกอริทึมอะพริโอริ ซึ่งเป็นการค้นหาข้อมูลตามลำดับจำนวนสมาชิกของไอเทมเซตจากน้อยไปมาก (k = 1, 2...n) โดยนำไอเทมเซตที่เป็นฟรีเควนท์ k-1 ไอเทมเซตมาใช้ในการสร้างและค้นหาฟรีเควนท์ k-ไอเทมเซต ด้วยขั้นตอนการจอยน์และตัดไอเทมเซตที่ไม่สามารถเป็นฟรีเควนท์ไอเทมเซตออกไป ในงานวิจัยนี้นอกจากจะทำการค้นหาฟรีเควนท์ไอเทมเซตในแต่ละรอบของ k- ไอเทมเซตแล้วยังสามารถค้นหาไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ในแต่ละ k-ไอเทมเซตเพิ่มขึ้นมาด้วยโดยใช้หลักการหาความน่าจะเป็นด้วยทฤษฎีเบอร์นูลลีดังที่ได้กล่าวในหัวข้อ 3.1

Algorithm1 : Main Algorithm for original database

Input : $DB, \sigma^{DB}, prob_{EF}$

Output : $F_k^{UP}, EF_k^{UP}, \rho^{DB}$

1. $k = 1$
2. *if* $k = 1$ *then*
3. *Scan DB and find count* $c(X, DB)$ *for all* $X \in 1-itemset$
4. *Calculate Probability for all X*
5. $\rho^{DB} = min(c(X, DB) \mid prob_X \geq prob_{EF})$
6. $F_k^{DB} = \{X \mid c(X, DB) \geq \sigma^{DB}\}$
7. $EF_k^{DB} = \{X \mid c(X, DB) \geq \rho^{DB}\}$
8. $k = k + 1$
9. *else*
10. *for* $(k = 2; F_{k-1}^{DB} \neq \varphi; k++)$ *do*
11. $C_k^{DB} = apriori\_gen(F_{k-1}^{DB} \cup EF_{k-1}^{DB}, \sigma^{DB})$
12. *Scan DB and find count* $c(X, DB)$ *for all* $X \in C_k^{DB}$
13. *Calculate Probability for all X*
14. $\rho^{DB} = min(c(X, DB) \mid prob_X \geq prob_{EF})$
15. $F_k^{DB} = \{X \mid c(X, DB) \geq \sigma^{DB}\}$
16. $EF_k^{DB} = \{X \mid c(X, DB) \geq \rho^{DB}\}$
17. $k = k + 1$
18. *end do*
19. *end if*

รูปที่ 3.4 อัลกอริทึมหลักสำหรับการค้นหาฟรีเควนท์และไอเทมที่คาดว่าจะเป็นฟรีเควนท์ในฐานข้อมูลเดิม

การค้นหาฟรีเควนท์ไอเทมเซต ซึ่งโดยทั่วไปจะพิจารณาจากค่าสนับสนุนของไอเทมเซตในแต่ละk-ไอเทมเซตที่มีค่ามากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุด แต่สิ่งสำคัญที่งานวิจัยนี้นำเสนอคือการค้นหาไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ (Expected frequent itemset) โดยแทนด้วย EF

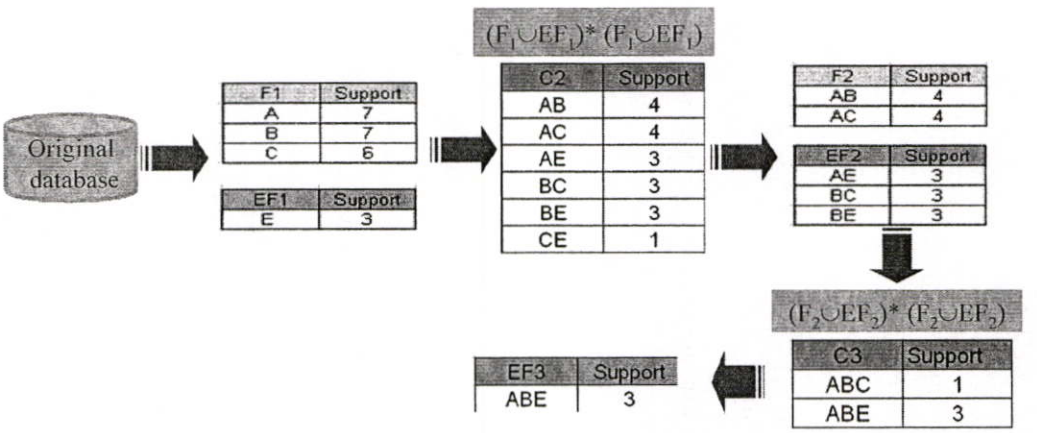
การค้นหาไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์นั้นจะพิจารณาได้จากค่าความน่าจะเป็นที่ผู้ใช้กำหนดคือ ค่าความน่าจะเป็นน้อยที่สุดที่คาดว่าไอเทมเซตจะกลายเป็นฟรีเควนท์ ($prob_{EF}$) ซึ่งเป็นค่าที่ผู้ใช้กำหนดว่าค่าความน่าจะเป็นอย่างน้อยที่สุดของไอเทมเซตที่ไม่เป็นฟรีเควนท์ไอเทมเซตแต่เมื่อนำมาคำนวณค่าความน่าจะเป็นที่คาดว่าจะเกิดเมื่อมีฐานข้อมูลใหม่ที่เพิ่มเข้ามาจำนวนหนึ่งด้วยหลักการของเบอร์นูลลีแล้วอาจพบว่าไอเทมเซตนั้นมีความน่าจะเป็นที่คาดว่าจะกลายเป็นฟรีเควนท์ไอเทมเซตได้ เช่น ถ้ากำหนดค่าความน่าจะเป็นน้อยที่สุดที่คาดว่าไอเทมเซตจะกลายเป็นฟรีเควนท์ ($prob_{EF}$) = 0.06 หมายความว่า จะยอมรับว่าไอเทมเซตใดๆ ที่คาดว่าจะกลายเป็นฟรีเควนท์ ก็ต่อเมื่อผลลัพธ์ที่ได้จากการคำนวณด้วยสมการ (3.3) แล้วมีค่าความน่าจะเป็นมากกว่าหรือเท่ากับ 0.06

แนวคิดในการการค้นหาฟรีเควนท์ไอเทมเซตและไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์มีจุดเด่นนอกเหนือจากเก็บไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์แล้วยังเป็นการกรองเพื่อเลือกเก็บเฉพาะไอเทมเซตที่มีโอกาสหรือคาดว่าจะกลายเป็นฟรีเควนท์ไอเทมเซตเมื่อมีฐานข้อมูลใหม่จำนวนหนึ่งเพิ่มเข้ามาเท่านั้น ซึ่งจะช่วยลดจำนวนการเก็บไอเทมเซตที่ไม่มีโอกาสเป็นฟรีเควนท์ไอเทมเซตจำนวนหนึ่งลงไป ทำให้มีจำนวนไอเทมเซตที่ต้องเก็บน้อยกว่าอัลกอริทึมเนกาทีฟบอร์เดอร์และอัลกอริทึมในด้านของไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ที่ได้ทำการวิจัยมาก่อนหน้านี้[12] [16] นอกจากนี้ยังช่วยทำให้การปรับปรุงข้อมูลต่างๆ ใช้เวลาน้อยลงเนื่องจากสามารถลดขั้นตอนการหาไอเทมเซตใหม่ที่มีโอกาสกลายเป็นฟรีเควนท์ไอเทมเซตและลดการสแกนฐานข้อมูลเดิมในกรณีที่มีการปรับปรุงฐานข้อมูลใหม่เป็นปัจจุบัน ขั้นตอนการหาฟรีเควนท์ k-ไอเทมเซตและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ k-ไอเทมเซตแสดงดังรูปที่ 3.4

การค้นหาฟรีเควนท์และไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์จะเริ่มจาก 1 ไอเทมเซต ดังแสดงในบรรทัดที่ 1 – 7 ในส่วนของ 1-ไอเทมจะมีการคำนวณค่าความน่าจะเป็นสำหรับทุก 1 ไอเทม ภายหลังจากทราบค่าความน่าจะเป็นของทุกไอเทมแล้ว สามารถหาค่าคาดหวังน้อยที่สุดที่ไอเทมเซตจะกลายเป็นฟรีเควนท์ หรือแทนด้วย ρ^{DB} ได้จากการหาค่าสนับสนุนน้อยที่สุดที่มีค่าความน่าจะเป็นของไอเทม มากกว่าหรือเท่ากับค่าความน่าจะเป็นน้อยที่สุดที่คาดว่าไอเทมเซตจะกลายเป็นฟรีเควนท์ (minimum expected value) ซึ่งเป็นค่าความน่าจะเป็นที่กำหนดโดยผู้ใช้

ในตัวอย่างการคำนวณค่าความน่าจะเป็น ในรูปที่ 3.3 ถ้าผู้ใช้กำหนดค่าความน่าจะเป็นน้อยที่สุดที่คาดว่าไอเทมเซตจะกลายเป็นฟรีเควนท์ด้วยค่า $prob_{EF}$ =0.10 หมายความว่า ผู้ใช้ยอมรับค่าความน่าจะเป็นที่จะเกิดขึ้น ว่าในการเพิ่มข้อมูลครั้งละ 1 ชุดจำนวน 100 ชุดจะพบว่าไอเทมเซตนี้เป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงอย่างน้อย 10 ชุดข้อมูล ถ้าค่าความน่าจะเป็นผ่านเกณฑ์นี้จะถูกเก็บไว้ในส่วนของไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์

จากตัวอย่างไอเทมเซตที่มีค่าสนับสนุนตั้งแต่ 4 ค่าขึ้นไปหรือมีค่าความน่าจะเป็นเท่ากับ 1 เป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลแน่นอน ดังในภาพที่ 3.3 จะพบว่าฟรีเควนท์ 1-ไอเทมเซต (F_1) คือ เซตของไอเทม A,B และ C { A,B,C} สำหรับไอเทมเซตที่มีค่าสนับสนุนที่น้อยกว่า 4 ค่า แต่อาจมีค่าความน่าจะเป็นมากกว่าหรือเท่ากับค่าความน่าจะเป็นน้อยที่สุดที่คาดว่าไอเทมเซตจะกลายเป็นฟรีเควนท์ในที่นี้จะพบว่าไอเทม E ซึ่งมีค่าสนับสนุนเท่ากับ 3 มีค่าความน่าจะเป็นที่คาดว่าจะกลายเป็นฟรีเควนท์คือ 0.28 ซึ่งมีค่ามากกว่า $prob_{EF}$ ดังนั้นจะจัดเก็บไอเทมเซตที่คาดว่าจะกลาย เป็นฟรีเควนท์ คือ ไอเทม E (EF_1 = {E})ในขณะที่ไอเทม F ซึ่งมีค่าสนับสนุนคือ 2 มีค่าความน่าจะเป็นเท่ากับ 0.06 ซึ่งน้อยกว่า $prob_{EF}$ ดังนั้นไอเทม F จะไม่ถูกจัดเก็บไว้ จากการคำนวณนี้ทำให้สามารถเทียบค่าความน่าจะเป็นน้อยที่สุดที่คาดว่าไอเทมจะกลายเป็นฟรีเควนท์เป็นค่าสนับสนุนอย่างน้อยที่คาดว่าไอเทมเซตจะกลายเป็นฟรีเควนท์ได้คือ ไอเทมนั้นๆ จะต้องมีค่าสนับสนุนอย่างน้อย 3 ค่าในฐานข้อมูลเดิม เราจะเรียกค่าสนับสนุนนี้ว่าค่าคาดหวังน้อยที่สุด ดังแสดงในภาพที่ 3.3 ในส่วนการคำนวณค่าความน่าจะเป็นของไอเทม {E} ซึ่งจะถูกนำไปใช้ในส่วนของฐานข้อมูลปรับปรุงเพื่อลดการสแกนฐานข้อมูลในหัวข้อ 3.3 ต่อไป



รูปที่ 3.5 แสดงการหาฟรีเควนท์ไอเทมเซตและไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ในฐานข้อมูลเดิม

ในส่วนของการค้นหาไอเทมเซตตั้งแต่ 2 ไอเทมเซตขึ้นไปแสดงในรูปที่ 3.4 บรรทัดที่ 10-16 จะทำการค้นหาเช่นเดียวกับการหา 1-ไอเทมเซต โดยในส่วนของแคนดิเดท 2-ไอเทมเซตจะสร้างได้จากการนำ F และ EF 1-ไอเทมเซตมาเชื่อมและตัดตามหลักของอะพริโอริในบรรทัดที่ 11 จากนั้นจะนำแคนดิเดท 2-ไอเทมเซตไปสแกนในฐานข้อมูลเพื่อหาค่าสนับสนุนและพิจารณาหาฟรีเควนท์ 2-ไอเทมเซต(F_2) และไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ 2- ไอเทมเซต (EF_2) ดังแสดงในบรรทัดที่ 12 -14 รูปที่ 3.4 ตัวอย่างการค้นหาฟรีเควนท์และไอเทมที่คาดว่าจะเป็นฟรีเควนท์แสดงในรูปที่ 3.5

3.3 อัลกอริทึมในการปรับปรุงค่าฟรีเควนท์ไอเทมเซตและไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์เมื่อฐานข้อมูลใหม่เพิ่มเข้ามาในฐานข้อมูลเดิม

การปรับปรุงค่าฟรีเควนท์ไอเทมเซตและไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ในฐานข้อมูลปรับปรุง เมื่อมีฐานข้อมูลใหม่เพิ่มเข้ามาในฐานข้อมูลเดิม ซึ่งนอกจากจะมีผลต่อฟรีเควนท์ไอเทมเซตและไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์แล้วยังมีผลต่อการค้นหาไอเทมเซตใหม่ที่ปรากฏเมื่อฐานข้อมูลถูกปรับปรุงให้เป็นปัจจุบันอีกด้วย ดังนั้นขั้นตอนการปรับปรุงฐานข้อมูลนี้จึงเป็นขั้นตอนที่สำคัญ

ในงานวิจัยส่วนใหญ่จะต้องมีการสแกนทั้งฐานข้อมูลเดิม และฐานข้อมูลใหม่ดังเช่นอัลกอริทึมเอฟยูพีที่จะมีการปรับปรุงค่าสนับสนุนที่ปรากฏในฐานข้อมูลใหม่ให้กับฟรีเควนท์ไอเทมเซตในขณะเดียวกันเมื่อพบว่ามีฟรีเควนท์ไอเทมเซตใหม่เกิดขึ้นในฐานข้อมูลใหม่จะต้องสแกนฐานข้อมูลเดิม เพื่อหาค่าสนับสนุนให้กับฟรีเควนท์ไอเทมเซตใหม่ที่พบ เนื่องจากส่วนใหญ่ฐานข้อมูลใหม่จะมีขนาดเล็ก ดังนั้น อาจพบจำนวนไอเทมเซตที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุดของฐานข้อมูลใหม่จำนวนมาก ดังนั้น การสแกนฐานข้อมูลเดิมเพื่อหาค่าสนับสนุนให้กับฟรีเควนท์ไอเทมเซตใหม่จำนวนมากจะทำให้ใช้เวลานาน โดยที่ฟรีเควนท์ไอเทมเซตใหม่นี้อาจไม่สามารถจะกลายเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงได้เลย

ในบางอัลกอริทึม เช่น เนกาทีฟบอร์เดอร์ เมื่อพบว่ามีการเปลี่ยนแปลงจากไอเทมเซตมาเป็นฟรีเควนท์ไอเทมเซตต้องทำการสแกนฐานข้อมูลใหม่ คือสแกนฐานข้อมูลเดิมและฐานข้อมูลใหม่ที่เพิ่ม) ซึ่งเป็นการไมน์นิ่งใหม่ทั้งหมดเพื่อค้นหาฟรีเควนท์ไอเทมเซตและสร้างกฎความสัมพันธ์ใหม่ที่เกิดขึ้น

สำหรับฟรีเควนท์ k-ไอเทมเซต ที่พบใหม่นี้หมายถึงไอเทมเซตที่ปรากฏในฐานข้อมูลเดิมแต่มีค่าสนับสนุนน้อยกว่าค่าสนับสนุนน้อยที่สุดของฐานข้อมูลเดิม และมีค่าสนับสนุนน้อยกว่าค่าคาดหวังน้อยที่สุดที่ไอเทมจะกลายเป็นฟรีเควนท์หรือเรียกว่าเป็นสมอลไอเทมเซตดังนั้นจึงไม่ถูกจัดเก็บในส่วนของฟรีเควนท์ k-ไอเทมเซตและไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์

ในงานวิจัยนี้ได้แบ่งการปรับปรุงค่าฟรีเควนท์ไอเทมเซตและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ดังรูปที่ 3.4 สามารถแบ่งการทำงานได้เป็น 3 ส่วนหลัก ดังนี้คือ

3.3.1 การปรับปรุงค่าฟรีเควนท์ไอเทมเซตและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ 1 - ไอเทมเซต

เมื่อมีฐานข้อมูลใหม่เพิ่มเข้ามาจะเริ่มจากการปรับปรุงสนับสนุนของฟรีเควนท์ไอเทมเซตและไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์สำหรับ 1-ไอเทมเซตดังแสดงในบรรทัดที่ 3 รูปที่ 3.6 และส่วนของการปรับปรุง 1-ไอเทมเซต เริ่มจากการสแกนฐานข้อมูลใหม่เพื่อหาค่าสนับสนุนสำหรับแคนดิเดท 1-ไอเทมเซตของฐานข้อมูลเดิม และแคนดิเดท 1-ไอเทมเซตของฐานข้อมูลใหม่ที่เพิ่มเข้ามา

จากการทำงานในส่วนนี้จะทำให้ทราบค่าของฟรีเควนท์ 1-ไอเทมเซตและไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลที่ผ่านการปรับปรุงแล้ว ดังแสดงในบรรทัดที่ 1- 6 ของรูปที่ 3.7 เพื่อปรับค่าสนับสนุนสำหรับไอเทมเซตที่เป็นสมาชิกของฟรีเควนท์ไอเทมเซตและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ให้เป็นปัจจุบัน

Algorithm1 : Main Algorithm

Input : $DB, db, k, \sigma^{UP}, \rho^{UP}, \rho^{DB}, C_l^{DB}, F_l^{DB}, EF_l^{DB}$ *and their count*

Output : F_k^{UP}, EF_k^{UP}

1. $k = 1$
2. *if* $k = 1$ *then*
3. *Update* $1-itemset$
4. $k = k + 1$
5. *else*
6. *for* $\left(k = 2; F_{k-1}^{UP} \neq \phi; k++\right)$ *do*
7. *Generate Candidate Itemset*
8. *Update* $k-itemset$ (*return m, Temp _ scanDB*)
9. *// m is the* max *imum itemset of Temp _ scanDB*
10. $k = k + 1$
11. *end do*
12. *end if*
13. $k = 2$
14. *while* $(Temp\_scanDBl_k \neq \phi$ *and* $(k \leq m)$ *do*
15. *Scan Original Database*$(Temp\_scanDB_k)$
16. $k = k + 1$
17. *end do*
18. *clear Temp _ scanDB*

รูปที่ 3.6 อัลกอริทึมหลักในการเพิ่มขยายการค้นหาโดยอาศัยหลักความน่าจะเป็น

จากนั้นจะนำค่าสนับสนุนที่ได้มาคำนวณหาค่าความน่าจะเป็นสำหรับไอเทมเซตทที่ผ่านการปรับปรุงและสามารถหาค่าคาดหวังน้อยที่สุดที่ไอเทมเซตจะกลายเป็นฟรีเควนท์ $\left(\rho^{DB}\right)$ สำหรับไอเทมเซตที่ผ่านการปรับปรุงในบรรทัดที่ 4-5 รูปที่ 3.7 ไอเทมเซตที่มีค่าสนับสนุนหลังจากการปรับปรุงฐานข้อมูลมากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุดสำหรับฐานข้อมูลปรับปรุงจะจัดเก็บไว้ในส่วนของฟรีเควนท์ไอเทมเซต แต่ถ้ามีค่าสนับสนุนน้อยกว่าค่าสนับสนุนน้อยที่สุดแต่มีค่ามากกว่าหรือเท่ากับค่าคาดหวังน้อยที่สุดที่ไอเทมเซตจะกลายเป็นฟรีเควนท์จะจัดเก็บไว้ในส่วนของไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ดังแสดงในบรรทัดที่ 7-8 รูปที่ 3.7

Algorithm 2 : Updating 1-itemsets

Input : $DB, db, \sigma^{UP}, \rho^{UP}, C_1^{DB}, F_1^{DB}, EF_1^{DB}, C_1^{db}$ *and their count*

Output : $F_1^{UP}, EF_1^{UP}, C_1^{UP}$ *and their count*

1. *Scan db and find count* $c(X, db)$ *for all* $X \in C_1^{DB} \cup C_1^{db}$
2. *for all* $X \in C_1^{DB} \cup C_1^{db}$ *do*
3. $\quad c(X, UP) = c(X, DB) + c(X, db)$
4. $\quad$ *Calculate Probability for all X*
5. $\quad \rho^{UP} = min(c(X, UP) \mid prob_X \geq prob_{EF})$
6. *end do*
7. $F_1^{UP} = \left\{ X \in C_1^{UP} \mid c(X, UP) \geq \sigma^{UP} \right\}$
8. $EF_1^{UP} = \left\{ X \in C_1^{UP} \mid \rho^{UP} \leq c(X, UP) < \sigma^{UP} \right\}$

รูปที่ 3.7 อัลกอริทึมการปรับปรุง 1 ไอเทมเซต

3.3.2 การปรับปรุงค่าฟรีเควนท์ไอเทมเซตและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ตั้งแต่ 2 ไอเทมเซตขึ้นไป

เมื่อค่าไอเทมเซตของ 1 ไอเทมเซตถูกปรับปรุงแล้วจะทำการปรับปรุงฟรีเควนท์ไอเทมเซตและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ที่มีสมาชิกมากกว่า 1 ไอเทมเซต ถ้าพบว่ามีฟรีเควนท์ 1-ไอเทมเซตของฐานข้อมูลปรับปรุงเกิดขึ้นใหม่จะทำการสร้างแคนดิเดท 2-ไอเทมเซตโดยจะทำการเชื่อมฟรีเควนท์ k-ไอเทมเซตของฐานข้อมูลปรับปรุงเข้าด้วยกันดังแสดงในรูปที่ 3.8 บรรทัดที่ 3 แต่ถ้าไอเทมเซตมากกว่า 2 ไอเทมเซตขึ้นไปจะทำการเชื่อมฟรีเควนท์ k-ไอเทมเซต (k>2) ด้วยและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ k-ไอเทมเซต ใหม่ที่ปรากฏขึ้นในฐานข้อมูลใหม่ทั้ง k = 2 และ k > 2 ไอเทมเซตจะถูกนำมาพิจารณาว่ามีฟรีเควนท์ k-1 ไอเทมเซตเป็นสมาชิกของฐานข้อมูลปรับปรุงหรือไม่ ถ้าไม่ใช่ไอเทมเซตนั้นจะถูกตัดออกไป

เมื่อได้แคนดิเดท k-ไอเทมเซต (k>=2) แล้วรูปที่ 3.9 จะสแกนฐานข้อมูลใหม่ที่เพิ่มเข้ามาเพื่อหาค่าสนับสนุนของแคนดิเดทไอเทมเซตและเพื่อปรับค่าสนับสนุนสำหรับไอเทมเซตที่เป็นสมาชิกของฟรีเควนท์ไอเทมเซตและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ให้เป็นปัจจุบันดังแสดงในบรรทัดที่ 1-7 จากนั้นจะนำค่าสนับสนุนที่ได้มาคำนวณหาค่าความน่าจะเป็นสำหรับไอเทมเซตที่ผ่านการปรับปรุงและสามารถหาค่าคาดหวังน้อยที่สุดที่ไอเทมเซตจะกลายเป็นฟรีเควนท์ (ρ^{DB}) สำหรับไอเทมเซตที่ผ่านการปรับปรุง ในบรรทัดที่ 11-12 ไอเทมเซตที่มีค่าสนับสนุนหลังจากการปรับปรุงฐานข้อมูลมากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุดสำหรับฐานข้อมูลปรับปรุงจะจัดเก็บไว้ในส่วนของฟรีเควนท์ไอเทมเซต แต่ถ้ามีค่าสนับสนุนน้อยว่าค่าสนับสนุนน้อยที่สุดแต่มีค่ามากกว่าหรือเท่ากับค่าคาดหวังน้อยที่สุดที่ไอเทมเซตจะกลายเป็นฟรีเควนท์จะจัดเก็บไว้ในส่วนของไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ดังแสดงในบรรทัดที่ 14-15

Algorithm 3: Generating Candidate k- itemsets

Input : $F_1^{UP}, F_{k-1}^{UP}, F_{k-1}^{db}, k$

Output : C_k^{new}

1. *if* $k = 2$ *then*
2. *if* $\left(length\left(F_1^{UP}\right) \geq 2\right)$ *then*
3. $C_2^{db} = F_1^{UP} * F_1^{UP}$
4. *for all* $X \in C_2^{db}$ *do*
5. $C_2^{new} = \left\{X \in C_2^{db} \middle| X \notin \left(F_2^{DB} \cup EF_2^{DB}\right)\right\}$
6. *end do*
7. *end if*
8. *else if* $k > 2$ *then*
9. $C_k^{db} = F_{k-1}^{db} * EF_{k-1}^{db}$
10. *for all* $X \in C_k^{db}$ *do*
11. $C_k^{new} = \left\{ X \in C_k^{db} \middle| X \in F_{k-1}^{UP} \text{ and } X \notin \left(F_k^{DB} \cup EF_k^{DB}\right)\right\}$
12.. *end do*
13. *end if*

รูปที่ 3.8 อัลกอริทึมสำหรับสร้างแคนดิเดท k-ไอเทมเซต

Algorithm 4 : Update ($k \geq 2$) itemset

Input: $DB, db, \sigma^{UP}, \rho^{UP}, \rho^{DB}, F_k^{DB}, EF_k^{DB}$ *and their count*

Ouiput: F_k^{UP} *and* EF_k^{UP}, F_k^{db}, *Temp_scanDB and their count, m*

1. *Scan db and find count* $c(X, db)$ *and* $c(Y, db)$
2. $\backslash\backslash\, X \in \left(F_k^{DB} \cup EF_k^{DB}\right)$ *and* $Y \in C_k^{new}$
3. *for all* $X \in \left(F_k^{DB} \cup EF_k^{DB} \cup C_k^{new}\right)$ *do*
4. *if* $X \in \left(F_k^{DB} \cup EF_k^{DB}\right)$ *and* $X \in C_k^{new}$ *then*
5. $c\left(X, UP\right) = c\left(X, DB\right) + c\left(X, db\right)$
6. *else if* $X \in \left(F_k^{DB} \cup EF_k^{DB}\right)$ *and* $X \notin C_k^{new}$ *then*
7. $c\left(X, UP\right) = c\left(X, DB\right)$
8. *else if* $X \notin \left(F_k^{DB} \cup EF_k^{DB}\right)$ *and* $X \in C_k^{new}$ *then*
9. $Temp\_scanDB_k = \{X \mid \left(c\left(X, db\right) + \left(\rho^{DB} - 1\right)\right) \geq \sigma^{UP}\}$
10 *end if*
11. *Calculate Probability for all X*
12. $\rho^{UP} = min(c(X, UP) \mid prob_X \geq prob_{EF})$
13.. *end do*
14. $F_k^{UP} = \left\{X \middle| c\left(X, UP\right) \geq \sigma^{UP}\right\}$
15. $EF_k^{UP} = \left\{X \middle| \rho^{UP} \leq c\left(X, UP\right) < \sigma^{UP}\right\}$

รูปที่ 3.9 อัลกอริทึมสำหรับปรับปรุง ($k \geq 2$) ไอเทมเซต

สำหรับแคนดิเดทไอเทมเซตเราจะทราบค่าสนับสนุนที่ไอเทมเซตนั้นปรากฏในฐานข้อมูลใหม่ ซึ่งถ้าแคนดิเดทไอเทมเซตนี้มีค่าสนับสนุนที่มากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุดของฐานข้อมูลใหม่ (min_sup^{db}) แต่เนื่องจากแคนดิเดทไอเทมเซตนี้ไม่พบว่าเป็นสมาชิกของฟรีเควนท์ไอเทมเซตหรือไอเทมที่คาดว่าจะเป็นฟรีเควนท์ ในงานวิจัยนี้จะทำการตรวจสอบว่าแคนดิเดทไอเทมเซตนี้มีโอกาสที่จะกลายมาเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงหรือไม่โดยจะยังไม่ทำการสแกนเพื่อหาค่าสนับสนุนที่แท้จริงในฐานข้อมูลเดิม แต่จะใช้ค่าคาดหวังที่ได้จากการไมน์นิ่งฐานข้อมูลเดิม ในการประมาณค่าสนับสนุนสูงสุดที่จะเกิดขึ้นในที่นี้คือ ค่าคาดหวังน้อยที่สุดลบหนึ่ง (Expected value -1 หรือ $(\rho^{DB} - 1)$) โดยถ้าผลรวมของสนับสนุนและค่าคาดหวังลบหนึ่งของแคนดิเดทไอเทมเซตมีค่ามากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุดของฐานข้อมูลปรับปรุงจะเก็บแคนดิเดทไอเทมเซตนั้นๆ เพื่อนำไปหาค่าสนับสนุนที่แท้จริงด้วยการสแกนฐานข้อมูลเดิม ในขั้นตอนสุดท้ายดังรูปที่ 3.10

3.3.3 การสแกนฐานข้อมูลเดิม

เป็นขั้นตอนสุดท้ายที่นำฟรีเควนท์ไอเทมเซตใหม่ตั้งแต่ 2 ไอเทมเซตขึ้นไป ที่พบได้จากส่วนที่ 2 มาทำการสแกนในฐานข้อมูลเดิมเพื่อหาค่าสนับสนุนที่แท้จริงให้กับไอเทมเซตเหล่านั้น ผลลัพธ์ที่ได้ในส่วนนี้คือค่าฟรีเควนท์ k-ไอเทมเซตและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตของฐานข้อมูลที่ปรับปรุงให้เป็นปัจจุบันทั้งหมด ดังแสดงในรูปที่ 3.10

Algorithm 5 : Scanning an original database

Input : $Temp\_scanDB_k, \sigma^{UP}, \rho^{UP}, F_k^{UP}, EF_k^{UP}$ *and their count*

Output : F_k^{UP}, EF_k^{UP} *and their count*

1. *Scan DB and obtain count* $c(X, DB)$ *for all* $Temp\_scanDB_k$
2. *for all* $X \in Temp\_scanDB_k$ *do*
3. $\quad c(X, UP) = c(X, DB) + c(X, db)$
4. *end do*
5. $F_k^{new} = \{X \mid X \in Temp\_scanDB_k \text{ and } c(X, UP) \geq \sigma^{UP}\}$
6. $EF_k^{new} = \{X \mid X \in Temp\_scanDB_k \text{ and } \rho^{UP} \leq c(X, UP) < \sigma^{UP}\}$
7. $F_k^{UP} = F_k^{UP} \cup F_k^{new}$
8. $EF_k^{UP} = EF_k^{UP} \cup EF_k^{new}$

รูปที่ 3.10 อัลกอริทึมสำหรับสแกนฐานข้อมูลเดิม

จากนั้นจะนำฟรีเควนท์ k-ไอเทมเซต ซึ่ง k > 1 มาคำนวณเพื่อหากฎความสัมพันธ์ที่เข้มแข็งและเขียนให้อยู่ในรูปกฎความสัมพันธ์ต่อไป

จากตัวอย่างการไมน์นิ่งฐานข้อมูลเดิมในรูปที่ 3.2 และ 3.5 เมื่อมีการเพิ่มส่วนของฐานข้อมูลใหม่เข้ามาจำนวน 5 ทรานแซคชันดังรูปที่ 3.11 จะเริ่มทำการปรับปรุงข้อมูลในลักษณะของตามลำดับ k-ไอเทมเซต โดยเริ่มจากงานส่วนแรกคือการปรับปรุงค่า 1 ไอเทมเซต เพื่อการปรับปรุงค่าเป็นปัจจุบันและมีความถูกต้องในงานวิจัยนี้จะทำการปรับปรุงค่าแคนดิเดท 1-ไอเทมเซตของฐานข้อมูลเดิมด้วยแคนดิเดท 1 ไอเทมเซตของฐานข้อมูลใหม่ ภายหลังจากการปรับปรุงค่าจะได้ฟรีเควนท์ 1-ไอเทมเซตและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ 1 –ไอเทมเซตของฐานข้อมูลที่เป็นปัจจุบัน

หลังจากปรับปรุงค่า 1ไอเทมเซตให้เป็นปัจจุบันเรียบร้อยแล้ว ในส่วนของ 2-ไอเทมเซต จะนำฟรีเควนท์1ไอเทมเซตที่ได้มาสร้างแคนดิเดท 2-ไอเทมเซตด้วยการเชื่อมและตัดโดยใช้หลักการเช่นเดียวกับอะพริโอริดังแสดงในเฟสที่ 2 ของรูปที่ 3.12 เนื่องจากเป็นไอเทมเซตที่มีความเป็นปัจจุบันแล้ว ดังนั้นในส่วนของ 2 -ไอเทมเซตนี้จะไม่นำไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์จากการปรับปรุงค่ามาเชื่อมเนื่องจากไม่สามารถเป็นฟรีเควนท์ไอเทมเซตในส่วนของ 2-ไอเทมเซตได้ นอกจากนี้แคนดิเดท 2-ไอเทมเซตนี้จะต้องไม่เป็นสมาชิกของฟรีเควนท์และไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ที่ได้จากการไมน์นิ่งในฐานข้อมูลเดิม ทำให้ได้แคนดิเดท 2 ไอเทมเซตจริงๆที่จะต้องทำการค้นหาและช่วยให้ลดจำนวนไอเทมเซตที่ต้องสแกนในฐานข้อมูลใหม่ลงไป

จากนั้นจะนำแคนดิเดท 2-ไอเทมเซตที่สร้างขึ้นนี้พร้อมด้วยฟรีเควนท์และไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ที่ได้จากการไมน์นิ่งในฐานข้อมูลเดิมไปสแกนในส่วนของฐานข้อมูลใหม่เพื่อหาค่าสนับสนุนและปรับปรุงค่า เมื่อได้ค่าสนับสนุนจากการสแกนฐานข้อมูลใหม่เรียบร้อยแล้วจะทำการพิจารณาไอเทมเซตเป็น 2 ส่วนคือ

ส่วนที่ 1 จะพิจารณาฟรีเควนท์ 2-ไอเทมเซตและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ 2-ไอเทมเซตสำหรับไอเทมเซตที่ได้จากการไมน์นิ่งฐานข้อมูลเดิม ว่ามีไอเทมเซตใดที่เป็นฟรีเควนท์ 2-ไอเทมเซตและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ 2-ไอเทมเซตสำหรับฐานข้อมูลปรับปรุง

ส่วนที่ 2 จะพิจารณาแคนดิเดท 2-ไอเทมเซตที่เป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลใหม่ นั่นคือแคนดิเดท 2-ไอเทมเซตที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุดในฐานข้อมูลใหม่ ($min\_sup^{db} * |db|$) เพื่อหาไอเทมเซตที่จะนำไปสแกนในฐานข้อมูลเดิมเพื่อหาค่าสนับสนุนที่แท้จริง โดยจะนำค่าที่ได้จากการประมาณค่าไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ในฐานข้อมูลเดิม ในที่นี้คือค่าคาดหวังน้อยที่สุดที่มีค่าความน่าจะเป็นที่ไอเทมเซตจะกลายเป็นฟรีเควนท์ในฐานข้อมูลเดิมจากรูปที่3.5 ไอเทมเซตที่พบในฐานข้อมูลเดิมอย่างน้อย 3 ทรานแซคชันและมีค่าสนับสนุนน้อยกว่า 4 (min_sup^{DB})

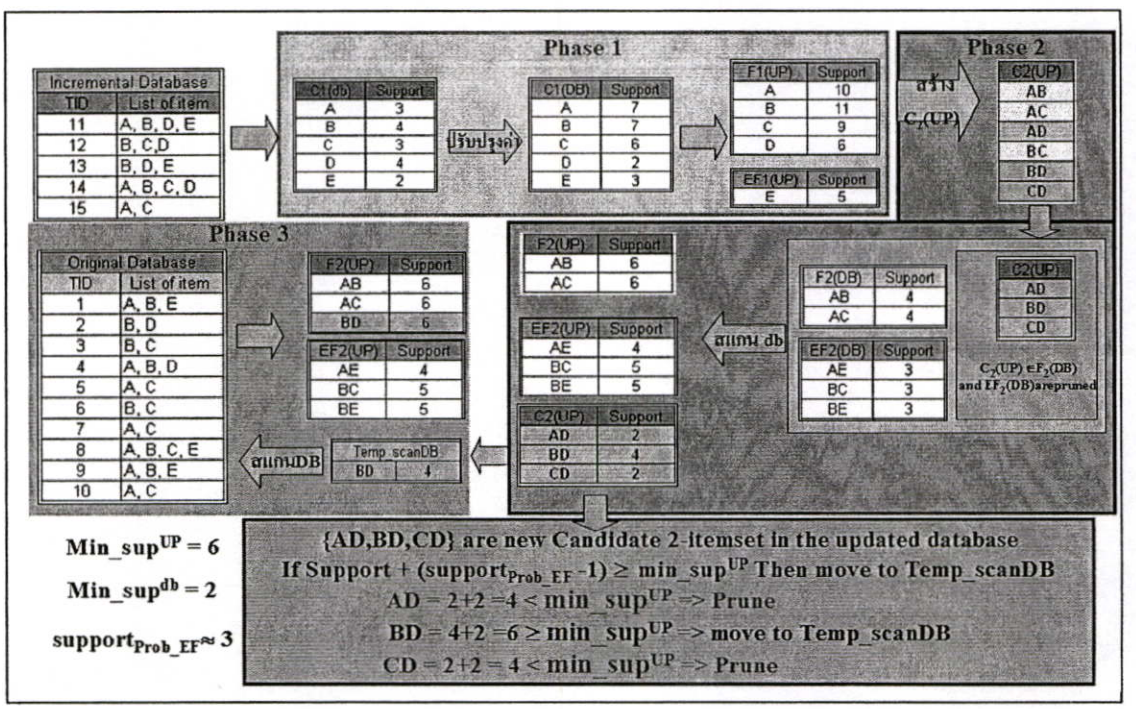
ไอเทมเซตใดที่มีค่าสนับสนุนมากกว่าหรือเท่ากับ 3 จะจัดเก็บไว้ในส่วนของไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ ดังนั้นแคนดิเดท 2-ไอเทมเซตที่พบในฐานข้อมูลใหม่ไม่เป็นสมาชิกของฟรีเควนท์ 2-ไอเทมเซตและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ 2-ไอเทมเซตในฐานข้อมูลเดิมดังนั้นค่าสนับสนุนสูงสุดที่แคนดิเดท 2-ไอเทมเซตนี้จะเป็นได้คือ 2 ซึ่งได้มาจากค่าค่าสนับสนุนน้อยที่สุดที่

มีค่าความน่าจะเป็นที่ไอเทมเซตจะกลายเป็นฟรีเควนท์ในฐานข้อมูลเดิมลบหนึ่ง $(\rho^{DB} - 1)$ เมื่อนำค่า $(\rho^{DB} - 1)$ มารวมกับค่าสนับสนุนของฟรีเควนท์ 2 -ไอเทมเซตของฐานข้อมูลใหม่แล้วพบว่าไอเทมเซตใดมีค่าผลรวมมากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุดของฐานข้อมูลปรับปรุงแล้วจะนำไอเทมเซตนี้ไปเก็บไว้ในส่วนของ Temp_scanDB เพื่อนำไปสแกนในฐานข้อมูลเดิมในขั้นตอนสุดท้าย ในตัวอย่างรูปที่ 3.12 จะพบว่ามีไอเทมเซต 1 ตัวคือ {BD} ส่วนของ {AD} และ {CD} จะถูกตัดออกเนื่องจากไม่มีโอกาสที่จะกลายเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงได้

สำหรับกรณี k-ไอเทมเซต ที่ k มีค่าตั้งแต่ 3 ไอเทมเซตขึ้นไปจะนำฟรีเควนท์ k 1 ที่พบในฐานข้อมูลใหม่มาเชื่อมและการตัดไอเทมออกด้วยไอเทมเซตที่ได้จากฐานข้อมูลปรับปรุงจากนั้นจะนำไปสแกนหาค่าสนับสนุนพร้อมกับฟรีเควนท์ k-ไอเทมเซตและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ k-ไอเทมเซตในฐานข้อมูลเดิมจากนั้นจะทำการพิจารณาหาไอเทมเซตที่จะนำไปสแกนในฐานข้อมูลเดิมเช่นเดียวกับที่ทำในส่วนของ 2-ไอเทมเซต

Incremental Database	
TID	List of item
11	A, B, D, E
12	B, C,D
13	B, D, E
14	A, B, C, D
15	A, C

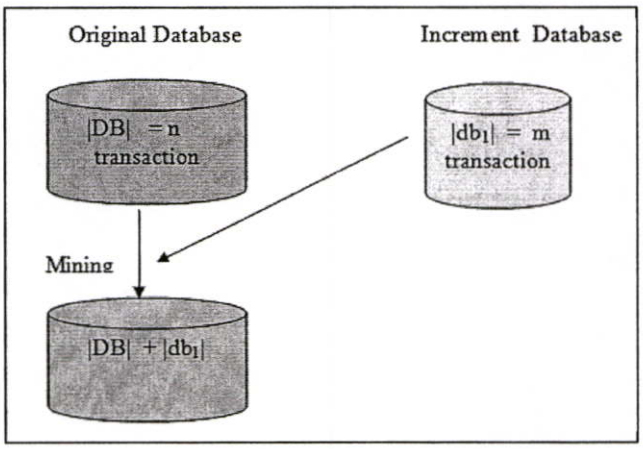
รูปที่ 3.11 ฐานข้อมูลใหม่ที่เพิ่ม



รูปที่ 3.12 ตัวอย่างขั้นตอนการเพิ่มขยายการค้นหากฎความสัมพันธ์

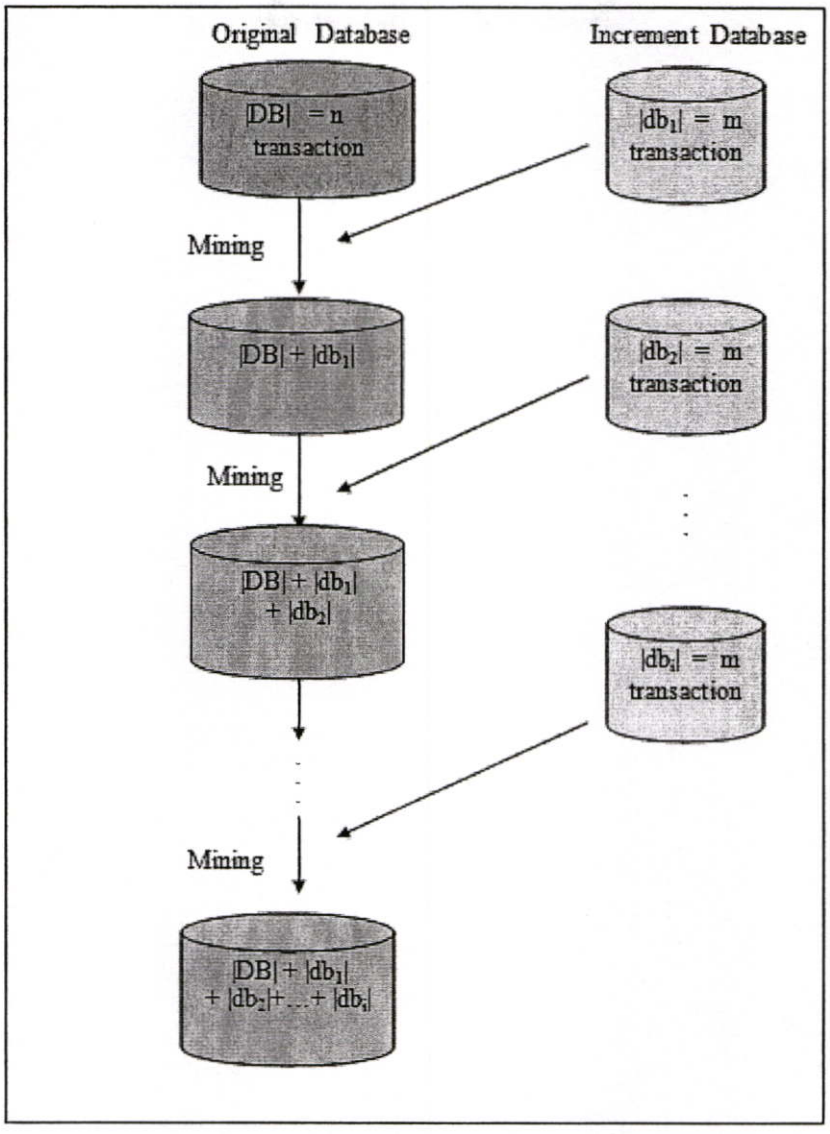
ขั้นตอนสุดท้ายในการเพิ่มขยายการค้นหากฎความสัมพันธ์ในงานวิจัยนี้คือนำไอเทมเซต ที่อยู่ในตัวแปร Temp_scanDB ไปสแกนในฐานข้อมูลเดิม เพื่อปรับปรุงค่าสนับสนุนให้กับไอเทม เซตที่อยู่ในตัวแปร Temp_scanDB จากนั้นจะพิจารณาหาฟรีเควนท์ไอเทมเซตและไอเทมเซตที่คาด ว่าจะเป็นฟรีเควนท์สำหรับ k-ไอเทมเซต (k >1) ดังรูปที่ 3.12 จะนำไอเทมเซต {BD} ไปสแกนใน ฐานข้อมูลเดิมแล้วเมื่อรวมค่าสนับสนุนที่พบในฐานข้อมูลปรับปรุงจะพบว่าไอเทมเซต {BD} มีค่า มากกว่าค่าสนับสนุนน้อยที่สุดของฐานข้อมูลปรับปรุงจะนำไอเทมเซต {BD} ไปเก็บในส่วนของ ฟรีเควนท์ 2-ไอเทมเซต

การเพิ่มขยายการค้นหากฎความสัมพันธ์ในรูปที่ 3.12 เป็นการเพิ่มขยายการค้นหากฎ ความสัมพันธ์ที่ได้จากการไมน์นิ่งฐานข้อมูลเดิมด้วยฐานข้อมูลใหม่ที่เพิ่มเข้ามาเพียงฐานข้อมูลเดียว โดยจำนวนทรานแซคชันจากฐานข้อมูลใหม่จะถูกเพิ่มเข้าไปในฐานข้อมูลเดิม ดังรูปที่ 3.13



รูปที่ 3.13 แสดงการเพิ่มขยายการค้นหากฎความสัมพันธ์ด้วยฐานข้อมูลใหม่ 1 ครั้ง

โดยทั่วไปฐานข้อมูลมักมีการเปลี่ยนแปลงอยู่ตลอดเวลาดังนั้นเมื่อเวลาผ่านไป การเพิ่ม ข้อมูลใหม่เข้ามาในฐานข้อมูลเดิมเพื่อทำการเพิ่มขยายการค้นหากฎความสัมพันธ์สามารถเพิ่มได้ หลายครั้ง เมื่อทำการเพิ่มขยายกฎความสัมพันธ์ในแต่ละครั้งข้อมูลใหม่ที่เพิ่มเข้ามาจะถูกรวมเข้าไป ในฐานข้อมูลเดิมทำให้ขนาดของฐานข้อมูลเดิมมีจำนวนทรานแซคชันมากขึ้น ดังรูปที่ 3.14 แสดง การเพิ่มข้อมูลใหม่ขนาด m ทรานแซคชันชัน (|db| = m ทรานแซคชัน) เข้าไปในฐานข้อมูลเดิม ขนาด n ทรานแซคชัน (|DB| = n ทรานแซคชัน) เมื่อทำการเพิ่มขยายการค้นหากฎความสัมพันธ์ใน แต่ละครั้งเรียบร้อยแล้ว ขนาดของฐานข้อมูลเดิมจะเท่ากับ n+m ทรานแซคชัน หรือเท่ากับ |DB|+ $|db_i|$ โดย i = 1,2,3,..., n



รูปที่ 3.14 แสดงการเพิ่มขยายการค้นหากฎความสัมพันธ์ด้วยฐานข้อมูลใหม่มากกว่า 1 ครั้ง

จากขั้นตอนการค้นหาฟรีเควนท์ k-ไอเทมเซตของงานวิจัยนี้เป็นการค้นหากฎความสัมพันธ์ที่มีประสิทธิภาพในกรณีที่มีการเพิ่มฐานข้อมูลใหม่เข้าไปในฐานข้อมูลเดิม โดยสามารถที่จะปรับปรุงค่าสนับสนุนสำหรับฟรีเควนท์ k-ไอเทมเซตและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตและจะตัดไอเทมเซตที่ไม่อาจกลายเป็นฟรีเควนท์ k-ไอเทมเซตออกไปทำให้ไอเทมเซตต่างๆ ได้มีการปรับปรุงให้มีความเป็นปัจจุบันอยู่เสมอ และเมื่อมีฟรีเควนท์ k-ไอเทมเซตเกิดขึ้นสามารถทำการค้นหาได้อย่างถูกต้องและครบถ้วน และสามารถลดจำนวนไอเทมเซตที่จะนำไปสแกนค้นหาค่าสนับสนุนในฐานข้อมูลเดิมได้อย่างมีประสิทธิภาพ ดังแสดงการทดลองและผลการทดลองในบทที่ 4

บทที่ 4
การทดลองและวิเคราะห์ผลการทดลอง

การค้นหากฎความสัมพันธ์เป็นเทคนิคหนึ่งที่ได้รับความสนใจอย่างมากสำหรับการทำดาต้าไมน์นิ่ง โดยเฉพาะการค้นหากฎความสัมพันธ์ที่เกิดขึ้นกับการซื้อสินค้าที่สามารถนำมากฎความสัมพันธ์ที่ได้มาใช้ในการวิเคราะห์ว่ามีสินค้าใดที่มักจะถูกซื้อไปด้วยกัน เพื่อให้เกิดความได้เปรียบในเชิงการแข่งขันและสามารถนำมาใช้ในการวางแผนกลยุทธ์การตลาดได้อย่างมีประสิทธิภาพ ซึ่งในการค้นหากฎความสัมพันธ์นั้นจะประกอบไปด้วยขั้นตอนหลักๆ 2 ขั้นตอนคือ

4.1.1 การหาฟรีเควนท์ไอเทมเซต ซึ่งหมายถึง เซตของไอเทมที่ปรากฏในทรานแซคชันมากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุด

4.1.2 การนำฟรีเควนท์ไอเทมเซตที่ได้มาสร้างเป็นกฎความสัมพันธ์ โดยกฎความสัมพันธ์ที่มีความเข้มแข็งจะต้องมีค่ามากกว่าหรือเท่ากับค่าความเชื่อมั่นน้อยที่สุด

โดยทั่วไปฐานข้อมูลมักจะมีการเปลี่ยนแปลงเกิดขึ้นอยู่เสมอเรียกฐานข้อมูลในลักษณะนี้ว่า ฐานข้อมูลไดนามิก ซึ่งข้อมูลที่เปลี่ยนแปลงนี้อาจทำให้กฎความสัมพันธ์ที่ได้ค้นหาไว้เกิดการเปลี่ยนแปลง คือ ไอเทมเซตที่พบว่าเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิมอาจไม่สามารถเป็นฟรีเควนท์ไอเทมเซตได้อีกต่อไปเมื่อมีข้อมูลใหม่เพิ่มเข้ามาจำนวนหนึ่ง ในทางกลับกันอาจพบว่าไอเทมเซตที่ไม่ใช่ฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิมกลายมาเป็นฟรีเควนท์ไอเทมเซตเมื่อมีข้อมูลใหม่เพิ่มเข้ามาได้เช่นกัน

การที่ฟรีเควนท์ไอเทมเซตมีการเปลี่ยนแปลงจะมีผลโดยตรงต่อกฎความสัมพันธ์ที่ได้เคยทำการไมน์นิ่งไว้ก่อนหน้า ในหลายๆ งานวิจัยได้คิดค้นเทคนิคที่ใช้ในการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยมีนำเสนอวิธีการในการค้นหาไอเทมเซตใหม่ๆ เพื่อให้การปรับปรุงกฎความสัมพันธ์มีความถูกต้องและมีประสิทธิภาพ อัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น เป็นอีกแนวทางหนึ่งที่มีการนำค่าความน่าจะเป็นของไอเทมเซตที่เกิดในฐานข้อมูลเดิมมาใช้ในการทำนายไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงเมื่อมีการเพิ่มข้อมูลใหม่จำนวนหนึ่งเข้ามาในฐานข้อมูลเดิม ซึ่งนอกจากจะสามารถช่วยให้ค้นหาฟรีเควนท์ไอเทมเซตได้อย่างถูกต้องและมีประสิทธิภาพ แล้วยังสามารถลดจำนวนไอเทมเซตที่ต้องนำไปสแกนในฐานข้อมูลเดิมได้อีกด้วย

ในบทนี้จะกล่าวถึงการทดลองเพื่อเปรียบเทียบประสิทธิภาพในด้านความถูกต้องและเวลาที่ใช้ในการทำงานของอัลกอริทึมในการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นเทียบกับการทำงานของ 4 อัลกอริทึมได้แก่ อัลกอริทึมอะพริโอริ, อัลกอริทึมเอฟยูพี, อัลกอริทึมบอร์เดอร์ และอัลกอริทึมพรีลาจก์

4.1 วัตถุประสงค์การทดลอง

เพื่อแสดงให้เห็นถึงประสิทธิภาพการทำงานของอัลกอริทึมสำหรับการเพิ่มขยายค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นเมื่อมีการเพิ่มข้อมูลใหม่เข้ามาในฐานข้อมูลเดิม โดยมีวัตถุประสงค์ในการทดลองเปรียบเทียบประสิทธิภาพการทำงานกับอัลกอริทึมที่ใช้ในการเพิ่มขยายการค้นหากฎความสัมพันธ์ต่างๆ ดังนี้คือ

1. เพื่อทดสอบความถูกต้องที่ได้จากการเพิ่มฐานข้อมูลใหม่เข้าไปในฐานข้อมูลเดิมด้วยขนาดข้อมูลต่างๆกัน อัลกอริทึมการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นการอัลกอริทึมที่มีการทำงานอยู่บนพื้นฐานของอัลกอริทึมอะพริโอริ ดังนั้นในการทดสอบความถูกต้องนี้จะทำการเปรียบเทียบผลที่ได้จากการค้นหากฎความสัมพันธ์ที่ได้กับอัลกอริทึมอะพริโอริและอัลกอริทึมที่มีรูปแบบการทำงานอยู่บนพื้นฐานของอัลกอริทึมอะพริโอริซึ่งอัลกอริทึมอะพริโอริเป็นอัลกอริทึมที่ได้รับความนิยม ลักษณะของการทำงานของอัลกอริทึมจะเป็นแบบวนรอบซ้ำเพื่อค้นหาฟรีเควนท์ไอเทมเซตในฐานข้อมูล โดยมีการใช้ความรู้หรือคุณสมบัติของฟรีเควนท์ k-ไอเทมเซตมาใช้ในการค้นหาฟรีเควนท์ (k+1)-ไอเทมเซตในลักษณะที่เรียกว่า การค้นหาแบบทีละระดับ ทำให้การค้นหาไอเทมเซตแคบลง (Search space)

นอกจากงานวิจัยนี้จะทำการทดสอบเปรียบเทียบความถูกต้องจากการค้นหากฎความสัมพันธ์จากการเพิ่มข้อมูลใหม่เข้าไปในฐานข้อมูลเดิมของอัลกอริทึมการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นเทียบกับอัลกอริทึมอะพริโอริแล้วยังมีการทดสอบประสิทธิภาพเทียบกับอีก 3 อัลกอริทึม ที่มีรูปแบบการทำงานอยู่บนพื้นฐานของอัลกอริทึมอะพริโอริ โดยอัลกอริทึมทั้ง 4 มีหลักการทำงานของอัลกอริทึมพอสังเขป ดังนี้คือ

1.1 อัลกอริทึมอะพริโอริ

เมื่อมีการเพิ่มฐานข้อมูลใหม่เข้าไปในฐานข้อมูลเดิมอัลกอริทึมอะพริโอริจะต้องทำการค้นหากฎความสัมพันธ์ของไอเทมเซตใหม่ทั้งหมด โดยไม่นำความรู้ที่ได้จากการไมน์นิ่งในฐานข้อมูลเดิมมาใช้ ดังนั้นอัลกอริทึมอะพริโอริจะทำการวนรอบซ้ำเพื่อค้นหากฎความสัมพันธ์ของข้อมูลทั้งหมดคือข้อมูลที่มีในฐานข้อมูลเดิมรวมกับฐานข้อมูลใหม่ที่เพิ่มเข้ามา (Re-mining)

1.2 อัลกอริทึมเอฟยูพี (FUP)

เอฟยูพีเป็นอัลกอริทึมแรกที่มีการนำหลักการของอัลกอริทึมอะพริโอริมาปรับให้สามารถเพิ่มขยายการค้นหากฎความสัมพันธ์ โดยการนำฟรีเควนท์ k-ไอเทมเซตที่ได้จากการไมน์นิ่งฐานข้อมูลเดิมมาใช้เพื่อลดการค้นหาฟรีเควนท์ไอเทมเซตใหม่ทั้งหมด

อัลกอริทึมเอฟยูพีจะทำการไมน์นิ่งฐานข้อมูลใหม่เพื่อปรับปรุงฟรีเควนท์ไอเทมเซตที่ได้จากการไมน์นิ่งก่อนหน้า และหาฟรีเควนท์ไอเทมเซตใหม่ที่เกิดขึ้นจากฐานข้อมูลใหม่ที่เพิ่มเข้ามา โดยมีการวนรอบซ้ำเพื่อค้นหาฟรีเควนท์ k-ไอเทมเซตเช่นเดียวกับอะพริโอริแต่ลด

จำนวนไอเทมเซตที่ใช้ในการค้นหาลงไปด้วยการรวมค่าสนับสนุนที่ได้จากฐานข้อมูลใหม่ให้กับฟรีเควนท์ไอเทมเซตที่ได้จากการไมน์นิ่งฐานข้อมูลเดิม

1.3 บอร์เดอร์อัลกอริทึม(Borders algorithm)

บอร์เดอร์เป็นอัลกอริทึมที่พัฒนาขึ้นจากอัลกอริทึมเอฟยูพีที่ยังคงต้องสแกนฐานข้อมูลทุก k-ไอเทมเซต โดยนำแนวคิดของเนกาทีฟบอร์เดอร์มาประยุกต์ใช้ด้วยการเก็บข้อมูลทั้งส่วนเป็นฟรีเควนท์ไอเทมเซตและส่วนของแคนดิเดทไอเทมเซตที่ไม่ได้เป็นฟรีเควนท์ไอเทมเซตที่เรียกว่า บอร์เดอร์หรือเนกาทีฟบอร์เดอร์ เมื่อบอร์เดอร์ไอเทมเซตในฐานข้อมูลเดิมเปลี่ยนเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุง อัลกอริทึมบอร์เดอร์จะมีการสแกนฐานข้อมูลใหม่ทั้งหมด (ฐานข้อมูลเดิมรวมกับฐานข้อมูลใหม่) เพื่อค้นหาฟรีเควนท์ไอเทมเซตที่มีการเปลี่ยนแปลงทั้งหมดในฐานข้อมูล

1.4 พรีลาร์จอัลกอริทึม (Pre-large algorithm)

พรีลาร์จเป็นอัลกอริทึมที่นำเสนอค่าสนับสนุน 2 ระดับคือ ค่าสนับสนุนระดับต่ำ (Lower threshold) และค่าสนับสนุนระดับสูง (Upper threshold) เพื่อนำมาใช้ในการพิจารณาฟรีเควนท์ไอเทมเซตและไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ เมื่อกำหนดให้ค่าค่าสนับสนุนระดับสูง ค่าสนับสนุนระดับสูงมีค่าเท่ากับค่าสนับสนุนน้อยที่สุด โดยในงานวิจัยของพรีลาร์จไม่ได้มีการกำหนดเกณฑ์สำหรับค่าสนับสนุนระดับต่ำ ในการทดสอบนี้จะกำหนดให้ค่าสนับสนุนระดับมีค่าน้อยกว่าค่าสนับสนุนน้อยที่สุดเท่ากับ 0.25 แนวคิดของอัลกอริทึมพรีลาจก์ได้นำเสนอการลดการสแกนฐานข้อมูลเดิมโดยคำนวณขนาดของฐานข้อมูลใหม่ที่เพิ่มเข้ามาถ้าไม่เกินค่าที่คำนวณได้จะไม่มีการสแกนฐานข้อมูลเดิม

ในงานวิจัยนี้จะทำการเปรียบเทียบความถูกต้องที่ได้จากการค้นหากฎความสัมพันธ์โดยเปรียบเทียบกับฟรีเควนท์ไอเทมเซตที่ได้จากการปรับปรุงฐานข้อมูลกับอัลกอริทึมดังกล่าวข้างต้น

2. เพื่อวัดประสิทธิภาพในการเพิ่มขยายการค้นหากฎความสัมพันธ์ของอัลกอริทึมในกรณีของการเพิ่มรายการข้อมูลใหม่เข้าไปในฐานข้อมูลเดิม

การทดสอบประสิทธิภาพของอัลกอริทึมในที่นี้เป็นการทดสอบเพื่อวัดประสิทธิภาพการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้เวลาจากการกระทำการ (Execution time) ที่ได้จากค้นหากฎความสัมพันธ์ในฐานข้อมูลปรับปรุงเปรียบเทียบกับอัลกอริทึมที่ใช้ในการเพิ่มขยายกฎความสัมพันธ์ต่างๆ เมื่อทำการทดลองเพิ่มฐานข้อมูลใหม่ขนาดที่เพิ่มด้วยเปอร์เซ็นต์ของฐานข้อมูลเดิมขนาดต่างๆ กัน โดยในงานวิจัยนี้จะทำการทดสอบเปรียบเทียบประสิทธิภาพการทำงานของอัลกอริทึมการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นเทียบกับ 4 อัลกอริทึมดังได้กล่าวข้างต้น

4.2 วิธีการทดลอง

การทดลองเพื่อเพิ่มขยายการค้นหากฎความสัมพันธ์เมื่อฐานข้อมูลมีการเปลี่ยนแปลงจากการเพิ่มข้อมูลใหม่เข้าไปในฐานข้อมูลเดิม ซึ่งทำให้ฟรีเควนท์ไอเทมเซตที่ได้จากการค้นหาในฐานข้อมูลเดิมเปลี่ยนไป ในการทดลองสำหรับงานวิจัยนี้จะเป็นการทดลองกรณีของการเพิ่มข้อมูลขนาดต่างๆ กันเข้าไปในฐานข้อมูลเดิม นอกจากอัลกอริทึมอะพริโอริแล้ว อัลกอริทึมเอฟยูพี, อัลกอริทึมบอร์เดอร์, อัลกอริทึมพรีลาจก์และอัลกอริทึมในการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักค่าน่าจะเป็น มีแนวคิดในการนำความรู้ที่ได้จากการไมน์นิ่งฐานข้อมูลเดิมมาใช้ คือส่วนของฟรีเควนท์ k-ไอเทมเซต และส่วนที่ไม่เป็นฟรีเควนท์ k–ไอเทมเซต ยกเว้นอัลกอริทึมเอฟยูพีที่ไม่มีส่วนของไอเทมเซตที่ไม่เป็นฟรีเควนท์ k -ไอเทมเซต

วิธีการทดลองจึงได้รับการออกแบบมาเพื่อทดลองภายใต้สมมติฐานที่แตกต่างกัน ดังนั้นการทดลองจะประกอบด้วยชุดข้อมูลจำนวน 3 ชุดที่สร้างด้วยชุดข้อมูลสังเคราะห์ที่กำหนดค่าพารามิเตอร์ในการสร้างให้เหมาะกับสมมติฐานการทดลองที่แตกต่างกัน ดังนี้

4.2.1 การทดลองชุดข้อมูลที่ 1: การทดลองเพิ่มขยายการค้นหากฎความสัมพันธ์ในกรณีค่าทางสถิติของฐานข้อมูลเดิมและฐานข้อมูลใหม่ไม่แตกต่างกัน

การทดลองชุดข้อมูลที่ 1 นี้ต้องการทดสอบความถูกต้องและประสิทธิภาพการทำงานของอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น ในกรณีที่ค่าทางสถิติของการเกิดไอเทมเซตในฐานข้อมูลเดิมและฐานข้อมูลใหม่ไม่แตกต่างกัน ซึ่งค่าความน่าจะเป็นในการเกิดไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลเดิมที่มีทรานแซคชันจำนวนมากเมื่อนำมาใช้ในการคำนวณจะทำให้ได้ค่าความน่าจะเป็นเข้าใกล้กับค่าการเกิดไอเทมเซตทั้งหมด” ซึ่งสอดคล้องกับแนวคิดของกฎว่าด้วยจำนวนมาก คือ “ค่าเฉลี่ยของไอเทมเซตที่ปรากฏในฐานข้อมูลเดิมซึ่งโดยปกติมีจำนวน ทรานแซคชันมากกว่าจำนวนทรานแซคชันของฐานข้อมูลใหม่ที่เพิ่มเข้ามาจะมีค่าเข้าใกล้ค่าเฉลี่ยของไอเทมเซตทั้งหมด”

เนื่องจากการทดลองนี้ต้องการศึกษาการค้นหาฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงที่มีค่าสถิติของฐานข้อมูลเดิมเทียบกับฐานข้อมูลใหม่ไม่แตกต่างกัน ดังนั้นเพื่อให้ทราบว่าการทำงานของอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์สามารถทำงานได้อย่างถูกต้องและมีประสิทธิสำหรับข้อมูลหลายรูปแบบภายใต้ค่าสถิติของฐานข้อมูลเดิมเทียบกับฐานข้อมูลใหม่ไม่แตกต่างกัน จึงได้สร้างชุดข้อมูลสังเคราะห์โดยกำหนดให้มีค่าเฉลี่ยของขนาดทรานแซคชัน | T | และค่าเฉลี่ยของขนาดสูงสุดที่จะเป็นฟรีเควนท์ไอเทมเซตได้ (| I |) ที่แตกต่างกัน ซึ่งการกำหนดเช่นนี้จะมีผลต่อขนาดของทรานแซคชันและจำนวนของฟรีเควนท์ไอเทมเซตที่ได้จากการไมน์นิ่งที่แตกต่างกัน โดยกำหนดค่าสถิติของการเกิดจำนวนที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้สูงสุด (| L |) และจำนวนไอเทม (N) เดียวกัน จำนวน 3 ชุด ดังนี้คือ

ชุดข้อมูลที่ 1.1 T10I4i200N100

เป็นการสร้างข้อมูลจากไอเทมจำนวน 100 ตัว นำไปสุ่มสร้างเป็นจำนวนที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้สูงสุด จำนวน 200 ชุด ด้วยค่าเฉลี่ยของขนาดทรานแซคชัน เท่ากับ 10 และค่าเฉลี่ยของขนาดสูงสุดที่จะเป็นฟรีเควนท์ไอเทมเซตได้คือ 4 ไอเทมเซต ซึ่งจะพบว่าใน 1 ทรานแซคชันของชุดข้อมูลการทดลองนี้ จะประกอบด้วยชุดของจำนวนที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้สูงสุด มากกว่า 1 ชุด ซึ่งมีผลต่อฟรีเควนท์ไอเทมเซตที่ผ่านการไมน์นิ่งเพื่อหากฎความสัมพันธ์

ชุดข้อมูลที่ 1.2 T10I10i200N100

เป็นการสร้างข้อมูลจากไอเทมจำนวน 100 ตัว นำไปสร้างเป็น MFI ขนาดเฉลี่ย = 10 ไอเทม จำนวน 200 ชุด และกำหนดค่าเฉลี่ยของทรานแซคชันเท่ากับ10 ซึ่งจะพบว่าใน 1 ทรานแซคชันจะประกอบด้วยชุดของจำนวนที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้สูงสุดมากกว่า 1 ชุด ซึ่งมีผลต่อฟรีเควนท์ไอเทมเซตที่ผ่านการไมน์นิ่งเพื่อหากฎความสัมพันธ์

ชุดข้อมูลที่ 1.3 T4I10i200N100

เป็นการสร้างข้อมูลจากไอเทมจำนวน 100 ตัว นำไปสร้างเป็นจำนวนที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้สูงสุดด้วยขนาดเฉลี่ย = 10 ไอเทม จำนวน 200 ชุด และกำหนดค่าเฉลี่ยของทรานแซคชันเท่ากับ 10 ซึ่งจะพบว่าใน 1 ทรานแซคชัน จะประกอบด้วยชุดของจำนวนที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้สูงสุดขนาด 200 ชุดเพียงชุดเดียว ซึ่งมีผลต่อฟรีเควนท์ไอเทมเซตที่ผ่านการไมน์นิ่งเพื่อหากฎความสัมพันธ์

แต่ละชุดข้อมูลที่ใช้ในการทดลองจะประกอบด้วยฐานข้อมูลเดิม (DB) จำนวน 10000 ทรานแซคชัน และฐานข้อมูลใหม่ที่เพิ่มด้วยขนาดที่เป็นเปอร์เซ็นต์ของฐานข้อมูลเดิมเข้าไปจำนวน 3 ชุดคือ 20%, 50% และ 100% หรือด้วยขนาด 2000, 5000 และ 10000 ทรานแซคชันตามลำดับที่สร้างด้วยค่าพารามิเตอร์เดียวกัน และมีการสุ่มจำนวนสูงสุดที่สามารถเป็นฟรีเควนท์ไอเทมเซตชุดเดียวกันคือ 200 ชุด มาสร้างเป็นทรานแซคชันในฐานข้อมูล

การไมน์นิ่งฐานข้อมูลเดิมด้วยค่าสนับสนุนน้อยที่สุดที่1%, 3% และ 5% สำหรับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์, พรีลาจก์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้ค่าความน่าจะเป็นแล้วนอกจากอัลกอริทึมเอฟยูพีที่จะสนใจฟรีเควนท์ไอเทมเซตที่ได้แล้วอัลกอริทึมบอร์เดอร์จะเก็บส่วนของแคนดิเดทไอเทมเซตที่ไม่เป็นฟรีเควนท์ไอเทมเซตไว้ในขณะที่อัลกอริทึมพรีลาจก์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้ค่าความน่าจะเป็นจะเก็บในส่วนของไอเทมที่คาดว่าจะกลายเป็นฟรีเควนท์ไอเทมเซตดังแสดงในตารางที่ 4.1

ตารางที่ 4.1 แสดงจำนวนไอเทมเซตที่ได้จากการไมน์นิ่งฐานข้อมูลเดิมด้วยค่าสนับสนุนน้อยที่สุด เท่ากับ 1%, 3% และ 5%

ชุดการทดลองที่	ค่าสนับสนุนน้อยที่สุด	Frequent itemset	Border itemset	Prelarge itemset	$Prob_{EF} = 0.01$			$Prob_{EF} = 0.03$			$Prob_{EF} = 0.05$		
					ขนาดของข้อมูลที่เพิ่ม (|db|)			ขนาดของข้อมูลที่เพิ่ม (|db|)			ขนาดของข้อมูลที่เพิ่ม (|db|)		
					2,000	5,000	10,000	2,000	5,000	10,000	2,000	5,000	10,000
1.1	1%	3,833	18,054	1,282	508	626	626	343	450	451	266	343	344
	3%	528	4,269	108	68	80	84	54	68	68	47	68	68
	5%	153	3,708	19	14	19	19	10	13	17	8	11	13
1.2	1%	11,774	14,514	10,021	3,950	3950	4,569	2,638	3,532	3,532	2,435	2,638	2,637
	3%	348	4,542	103	45	58	64	38	45	45	29	45	45
	5%	113	4,359	3	2	2	3	2	2	2	2	2	2
1.3	1%	369	2,216	261	40	57	57	23	30	30	17	23	23
	3%	55	1,067	0	0	0	0	0	0	0	0	0	0
	5%	38	702	1	3	0	0	0	0	0	0	0	0

4.2.2 การทดลองชุดข้อมูลที่ 2: การทดลองผลที่ได้จากการทำนายฟรีเควนท์ไอเทมเซตของไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงที่ได้จากการคำนวณโดยใช้หลักความน่าจะเป็นของเบอร์นูลลี

ชุดข้อมูลที่ 2 นี้ ต้องการทดสอบผลการทำนายไอเทมที่คาดว่าจะเป็นฟรีเควนท์ k ไอเทมเซตที่ได้จาการคำนวณโดยใช้หลักความน่าจะเป็นของเบอร์นูลลีจากฐานข้อมูลเดิมว่าสามารถกลายเป็นฟรีเควนท์ k-ไอเทมเซตจริงในฐานข้อมูลปรับปรุงได้อย่างถูกต้อง เมื่อมีข้อมูลจำนวนหนึ่งเพิ่มเข้ามาในฐานข้อมูลเดิม ภายใต้ค่าทางสถิติของฐานข้อมูลเดิมและฐานข้อมูลใหม่ไม่แตกต่างกัน โดยการทดลองเพิ่มข้อมูลขนาด 10 เปอร์เซ็นต์หรือ 1,000 ทรานแซคชัน จำนวน 100 ชุดเข้าไปในฐานข้อมูลเดิมขนาด 10000 ทรานแซคชัน ($DB + db_1$, $DB+db_2$, …,$DB+db_{100}$) โดยนอกจากเปรียบเทียบความถูกต้องในการทำนายไอเทมที่คาดว่าจะกลายเป็นฟรีเควนท์ k-ไอเทมเซตที่กลายเป็นฟรีเควนท์ k-ไอเทมเซตในฐานข้อมูลปรับปรุงแล้วการทดลองนี้ยังเปรียบเทียบความถูกต้องและประสิทธิภาพการทำงานกับอัลกอริทึมทั้ง 4 อัลกอริทึมดังได้กล่าวไว้ข้างต้นด้วย

4.2.3 การทดลองชุดข้อมูลที่ 3 : การทดลองในกรณีค่าทางสถิติของฐานข้อมูลเดิมและฐานข้อมูลใหม่แตกต่างกัน

นอกจากการทดลองภายใต้ค่าสถิติของฐานข้อมูลเดิมและฐานข้อมูลใหม่ที่เหมือนกันแล้ว อัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นยังสามารถค้นหากฎความสัมพันธ์ของฐานข้อมูลเดิมที่มีค่าทางสถิติแตกต่างจากฐานข้อมูลใหม่ด้วย โดยในชุดข้อมูลที่ 3 นี้ได้ออกแบบมาเพื่อใช้ในการทดลองการทำงานของอัลกอริทึมว่าสามารถค้นหาฟรีเควนท์ไอเทมเซตจากฐานข้อมูลปรับปรุงได้ครบถ้วนถูกต้อง เมื่อสมมติฐานของความน่าจะเป็นในการเกิดฟรีเควนท์ไอเทมเซตของฐานข้อมูลใหม่แตกต่างกันฐานข้อมูลเดิม ซึ่งมีผลต่อการเกิดฟรีเควนท์ไอเทมเซตใหม่ในฐานข้อมูลปรับปรุง

การสร้างชุดข้อมูลสังเคราะห์สำหรับการทดลองนี้จะสร้างชุดข้อมูลสังเคราะห์โดยกำหนดให้ฐานข้อมูลเดิมและฐานข้อมูลใหม่สร้างจากค่าสถิติของการเกิดจำนวนที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้สูงสุด ($|L|$) ที่แตกต่างกัน 3 ค่าดังนี้คือ

ชุดข้อมูลที่ 3.1 กำหนดให้ฐานข้อมูลใหม่ที่จะเพิ่มเข้าไปในฐานข้อมูลเดิมสร้างจากค่าสถิติของการเกิดจำนวนที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้สูงสุด ($|L|$) ที่แตกต่างจากฐานข้อมูลเดิม 20 เปอร์เซ็นต์

ชุดข้อมูลที่ 3.2 กำหนดให้ฐานข้อมูลใหม่ที่จะเพิ่มเข้าไปในฐานข้อมูลเดิมสร้างจากค่าสถิติของการเกิดจำนวนที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้สูงสุด ($|L|$) ที่แตกต่างจากฐานข้อมูลเดิม 60 เปอร์เซ็นต์

ชุดข้อมูลที่ 3.3 กำหนดให้ฐานข้อมูลใหม่ที่จะเพิ่มเข้าไปในฐานข้อมูลเดิมสร้างจากค่าสถิติของการเกิดจำนวนที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้สูงสุด (| L |) ที่แตกต่างจากฐานข้อมูลเดิม 100 เปอร์เซ็นต์

จากชุดข้อมูลที่สร้างทั้ง 3 ชุด จะทำให้ความน่าจะเป็นในการเลือกไอเทมที่ใส่เข้าไปในทรานแซคชันแตกต่างกันซึ่งมีผลต่อความน่าจะเป็นในการเกิดฟรีเควนท์ไอเทมเซตที่ต่างกัน โดยกำหนดค่าพารามิเตอร์ที่เหลือได้แก่ค่าเฉลี่ยของขนาดทรานแซคชัน | T | และค่าเฉลี่ยของขนาดสูงสุดที่จะเป็นฟรีเควนท์ไอเทมเซตได้ (| I |) และจำนวนไอเทม (N) เดียวกัน คือ | T | = 10, | I | =4 และ N =100

การทดลองจะใช้ฐานข้อมูลเดิมขนาด 10000 ทรานแซคชันตัวเดียวกับการทดลองชุดข้อมูลที่ 1.1 ดังแสดงในตารางที่ 4.1 และทำการเพิ่มฐานข้อมูลใหม่ที่สร้างจากค่าสถิติของการเกิดจำนวนสูงสุดที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้ที่แตกต่างกันจำนวน 3 ชุด ฐานข้อมูลใหม่แต่ละชุดจะมีขนาดเท่ากับเปอร์เซ็นต์ของฐานข้อมูลเดิมจำนวน 3 ขนาดคือ 20%, 50% และ 100% หรือด้วยขนาด 2000, 5000 และ 10000 ทรานแซคชันตามลำดับ เพื่อทดสอบว่าภายหลังจากการเพิ่มข้อมูลใหม่ขนาดต่างๆ ดังกล่าวข้างต้นจากแต่ละชุดข้อมูลย่อยเข้าไปในฐานข้อมูลเดิมที่มีค่าสถิติการเกิดของฟรีเควนท์ k ไอเทมเซตและไอเทมที่คาดว่าจะกลายเป็นฟรีเควนท์ k ไอเทมเซตที่แตกต่างกัน อัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นสามารถค้นหาฟรีเควนท์ k-ไอเทมเซตสำหรับฐานข้อมูลปรับปรุงได้ถูกต้องเทียบกับอัลกอริทึมอะพริโอริ เอฟยูพี บอร์เดอร์และพรีลาจก์หรือไม่

นอกจากสมมติฐานในการทดลองข้างต้นแล้ว วิธีการทดลองชุดข้อมูลแต่ละชุดมีการกำหนดค่าพารามิเตอร์สำหรับการทดลองต่างๆ กัน ดังนี้คือ

1. ค่าสนับสนุนน้อยที่สุด (min_sup) ในชุดข้อมูลที่ 1 และ 3 จะทำการทดลองที่ค่าสนับสนุนน้อยที่สุด 3 ค่า คือ 1%, 3% และ 5% สำหรับชุดข้อมูลที่ 2 จะทำการทดลองที่ค่าสนับสนุนน้อยที่สุด 1 ค่าคือ 1%

2. ขนาดฐานข้อมูลใหม่ที่เพิ่มเข้าไปในฐานข้อมูลเดิมในชุดข้อมูลที่ 1 และ 3 จะเพิ่มข้อมูลด้วยขนาดของเปอร์เซ็นต์ของขนาดฐานข้อมูลเดิมต่างๆ กัน เช่น 20%, 50%, 100% เป็นต้น สำหรับชุดข้อมูลที่ 2 จะทำการเพิ่มข้อมูลขนาด 10% ของขนาดฐานข้อมูลเดิมคือ จำนวน 1000 ทรานแซคชัน จำนวน 100 ชุด

3. ค่าพารามิเตอร์เฉพาะอัลกอริทึมได้แก่

3.1 อัลกอริทึมพรีลาจก์ ได้กำหนดค่าสนับสนุนขั้นต่ำ (Lower support: s_lower) ในขณะที่ค่าสนับสนุนขั้นสูง (Upper support: s_upper) ดังนี้

ค่าสนับสนุนขั้นสูง = ค่าสนับสนุนน้อยที่สุด เช่นถ้ากำหนดค่าสนับสนุนน้อยที่สุดเท่ากับ 3% ค่าสนับสนุนขั้นสูงจะมีค่าเท่ากับ 3% เช่นกัน

ค่าสนับสนุนขั้นต่ำ = ค่าสนับสนุนน้อยที่สุด – 0.25 ในที่นี้จะหมายถึงการกำหนดค่าสนับสนุนขั้นต่ำให้มีค่าน้อยกว่าค่าสนับสนุนน้อยที่สุดเท่ากับ 0.25 เช่นถ้ากำหนดค่าสนับสนุนน้อยที่สุดเท่ากับ 3% จะกำหนดค่าสนับสนุนขั้นต่ำเท่ากับ 2.75% เป็นต้น ทั้งนี้การกำหนดค่าสนับสนุนขั้นต่ำนี้ขึ้นกับผู้ใช้เป็นผู้กำหนด

3.2 อัลกอริทึมการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นได้กำหนดค่าความน่าจะเป็นที่คาดว่าไอเทมเซตจะกลายเป็นฟรีเควนท์ ($Prob_{EF}$) ในชุดข้อมูลที่ 1 และ 3 จำนวน 3 ค่าคือ $Prob_{EF} = 0.01$, $Prob_{EF} = 0.03$ และ $Prob_{EF} = 0.05$ สำหรับชุดข้อมูลที่ 2 จะกำหนดค่า $Prob_{EF} = 0.01$ โดยทำการเพิ่มข้อมูล

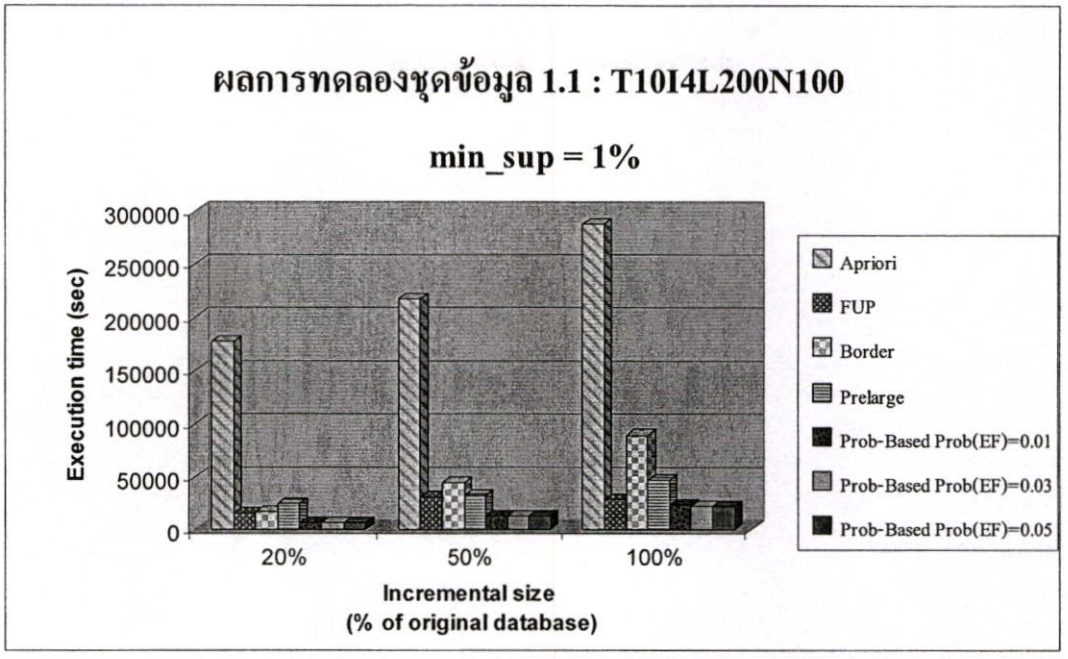
การทดลองสำหรับงานวิจัยนี้ทำการทดลองด้วยเครื่องไมโครคอมพิวเตอร์ อินเทลคอร์ทูดูโอ (Core 2 duo) ซีพียู E7400@ 2.8 GHz ซึ่งมีหน่วยความจำหลัก (main memory) 2 GB

4.3 ผลการทดลองและการวิเคราะห์ผลการทดลอง

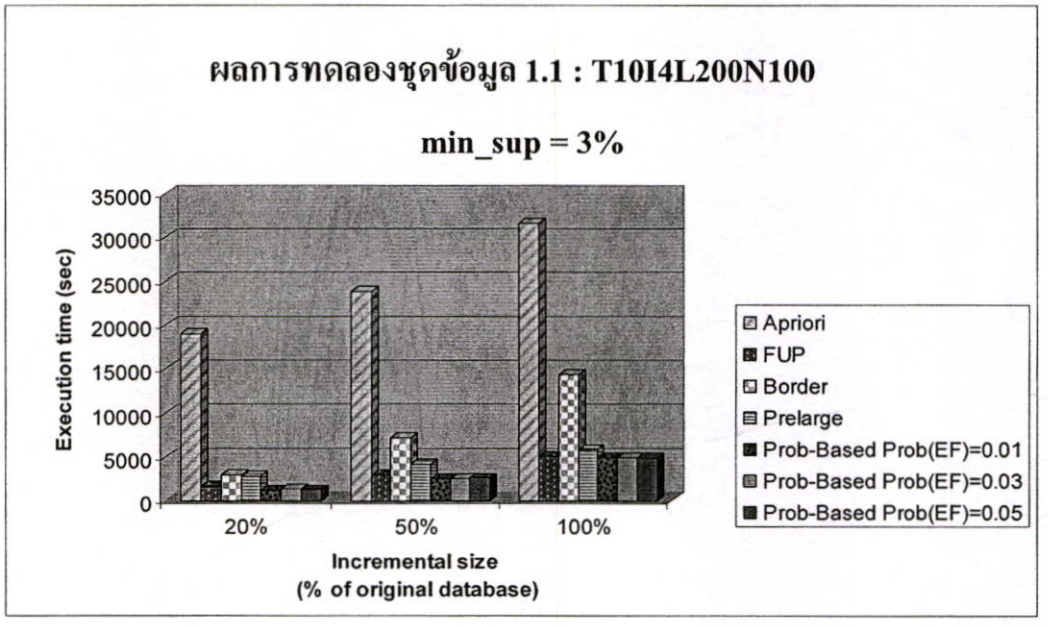
เมื่อนำชุดข้อมูลต่างๆ ข้างต้นมาทำการทดลองด้วยโปรแกรม MATLAB 7.0 ได้ผลการทดลองดังนี้

4.3.1 ผลการทดลองชุดข้อมูลที่ 1: ผลที่ได้จากการเพิ่มขยายการค้นหากฎความสัมพันธ์ในกรณีค่าทางสถิติของฐานข้อมูลเดิมและฐานข้อมูลใหม่ไม่แตกต่างกัน

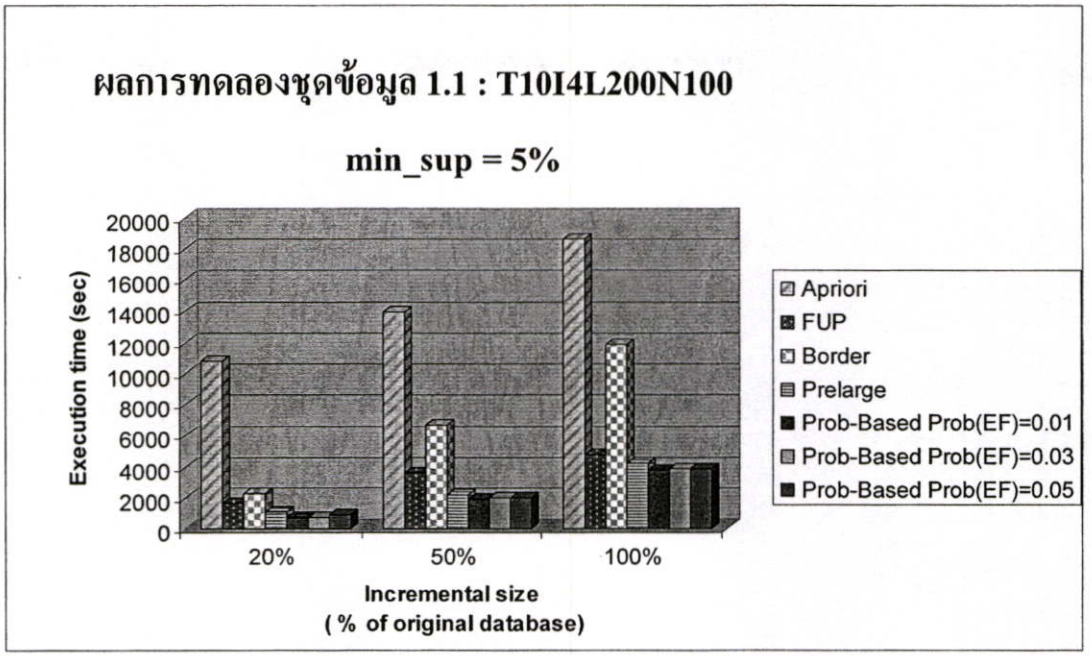
การทดลองชุดข้อมูลที่ 1 จากสมมติฐานในการทดลองที่ว่าการเกิดของไอเทมเซตที่ปรากฏในฐานข้อมูลเดิมและฐานข้อมูลใหม่มีค่าความน่าจะเป็นในการเกิดของไอเทมเซตไม่แตกต่างกัน ด้วยชุดข้อมูลที่สร้างด้วยค่าเฉลี่ยของทรานแซคชัน และค่าเฉลี่ยสูงสุดที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้แตกต่างกันโดยมีค่าพารามิเตอร์เดียวกันคือ จำนวนที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้สูงสุด คือ 200 ชุด และจำนวนไอเทมที่นำมาสร้าง คือ 100 ตัวพบว่าการทดลองด้วยชุดข้อมูลทั้ง 3 ชุดมีเวลากระทำการ (Execution time) สำหรับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และพรีลาจก์ที่ใช้ในการค้นหาการเพิ่มขยายกฎความสัมพันธ์เมื่อมีการเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 2000, 5000 และ 10000 ด้วยค่าสนับสนุนน้อยที่สุดคือ 1%, 3% และ 5% รูปที่ 4.1, รูปที่ 4.2, รูปที่ 4.3 และตารางที่ 4.2 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับทั้ง 5 อัลกอริทึมด้วยชุดข้อมูลที่ 1.1



รูปที่ 4.1 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับชุดข้อมูลที่ 1.1 เมื่อเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 20%, 50% และ 100% ด้วยค่าสนับสนุนน้อยที่สุดคือ 1%



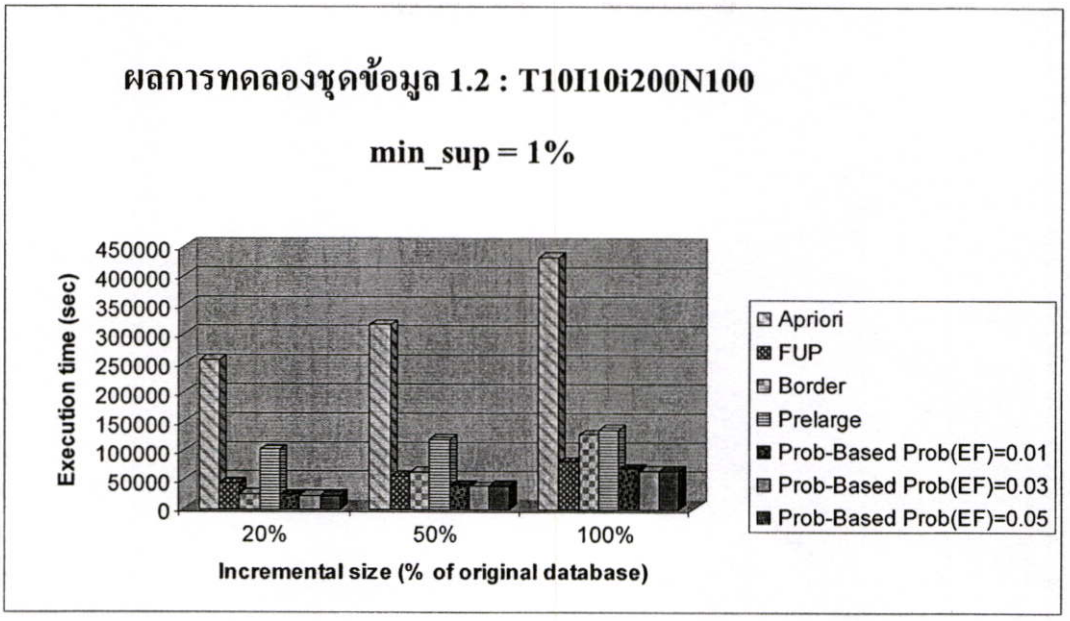
รูปที่ 4.2 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับชุดข้อมูลที่ 1.1 เมื่อเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 20%, 50% และ 100% ด้วยค่าสนับสนุนน้อยที่สุดคือ 3%



รูปที่ 4.3 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับชุดข้อมูลที่ 1.1 เมื่อเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 20%, 50% และ 100% ด้วยค่าสนับสนุนน้อยที่สุดคือ 5%

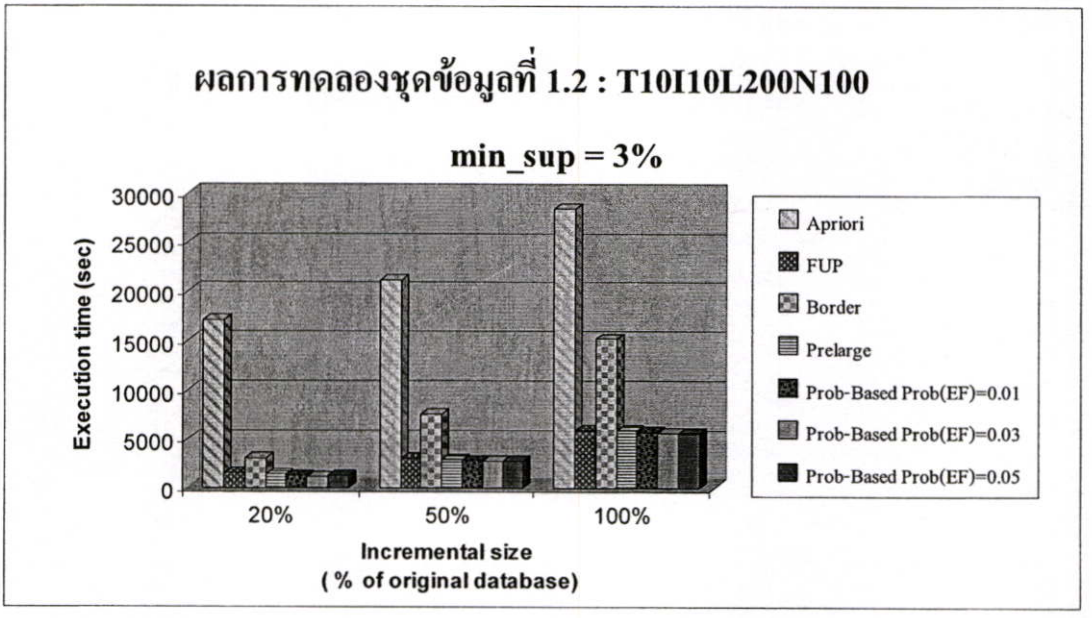
ตารางที่ 4.2 แสดงเวลากระทำของอัลกอริทึมสำหรับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และพรีลาจก์ที่ใช้ในการค้นหาการเพิ่มขยายกฎความสัมพันธ์ด้วยชุดข้อมูล 1.1

เวลาในการทำงาน (วินาที) ของชุดข้อมูลที่ 1.1 : T10I4L200N100								
ค่าสนับสนุนน้อยที่สุด (min_sup)	ขนาดข้อมูลใหม่ |db|	Algorithm						
		Apriori	FUP	Border	Prelarge	Probability-Based		
						$Prob_{EF}$		
						0.01	0.03	0.05
1%	20%	17,6935	15,562	18,004	25,884	7,566	7,574	7,592
	50%	217,735	32,052	45,168	32,714	13,976	13,524	13,450
	100%	288,255	29,554	89,594	47,317	23,387	23,237	23,068
3%	20%	18,978	1,651	2,962	2,861	1,189	1,320	1,223
	50%	23,858	2,963	7,148	4,181	2,467	2,561	2,589
	100%	31,626	5,104	14,311	5,784	4,921	4,916	4,954
5%	20%	10,788	1,620	2,249	1,090	756	761	760
	50%	13,929	3,584	6,687	2,174	1,894	1,999	2,001
	100%	18,631	4,805	11,850	4,260	3,782	3,842	3,899

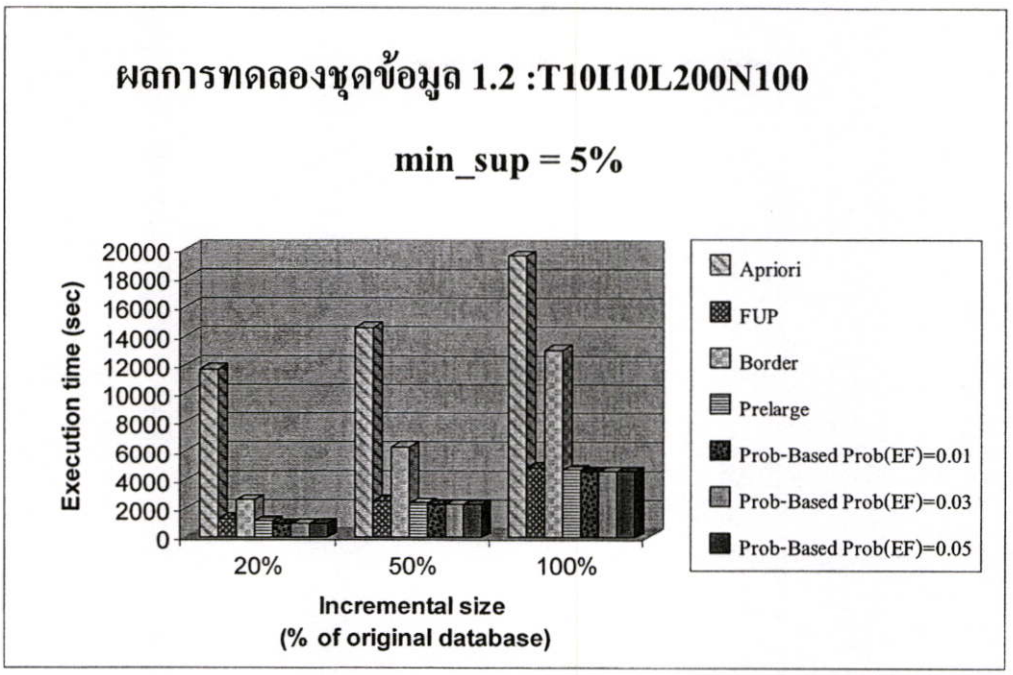


รูปที่ 4.4 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับชุดข้อมูลที่ 1.2 เมื่อเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 20%, 50% และ 100% ด้วยค่าสนับสนุนน้อยที่สุดคือ 1%

รูปที่ 4.4, รูปที่ 4.5, รูปที่ 4.6 และตารางที่ 4.3 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับทั้ง 5 อัลกอริทึมสำหรับชุดข้อมูลที่ 1.2 และรูปที่ 4.7, รูปที่ 4.8, รูปที่ 4.9 และตารางที่ 4.4 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับทั้ง 5 อัลกอริทึมสำหรับชุดข้อมูลที่ 1.3



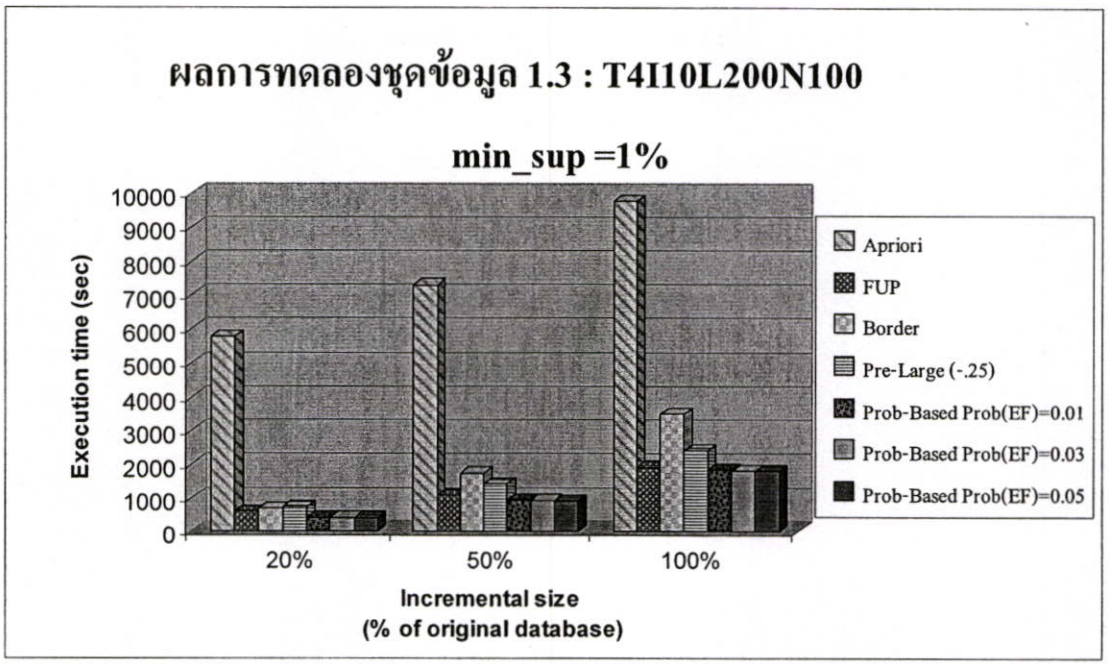
รูปที่ 4.5 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับชุดข้อมูลที่ 1.2 เมื่อเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 20%, 50% และ 100% ด้วยค่าสนับสนุนน้อยที่สุดคือ 3%



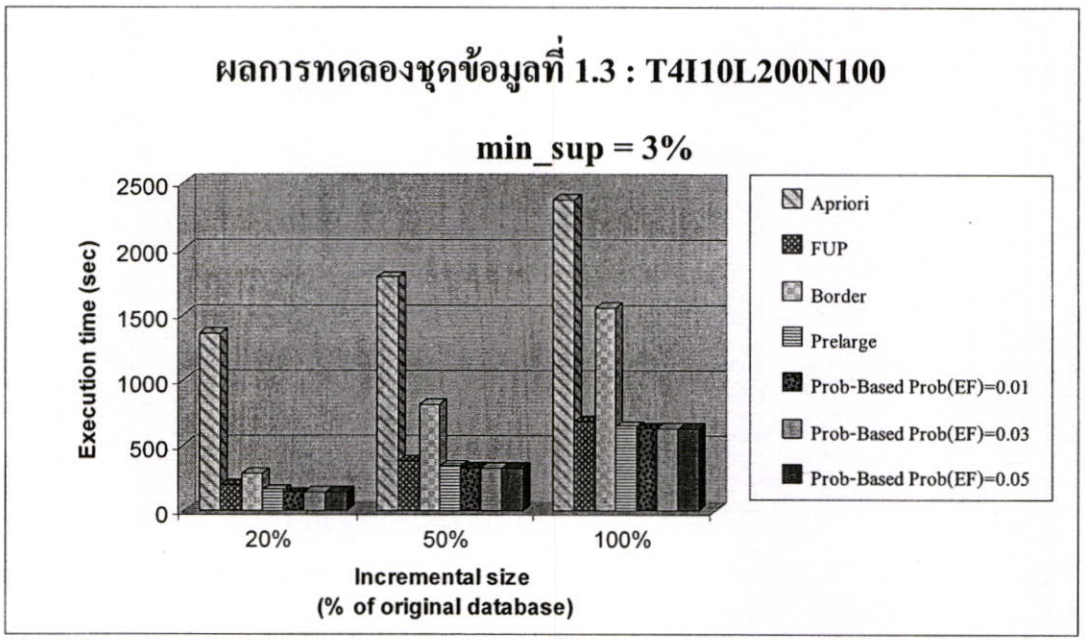
รูปที่ 4.6 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับชุดข้อมูลที่ 1.2 เมื่อเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 20%, 50% และ 100% ด้วยค่าสนับสนุนน้อยที่สุดคือ 5%

ตารางที่ 4.3 แสดงเวลากระทำของอัลกอริทึมสำหรับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และพรีลาจก์ที่ใช้ในการค้นหาการเพิ่มขยายกฎความสัมพันธ์เมื่อมีการเพิ่มข้อมูลใหม่ด้วยชุดข้อมูล 1.2

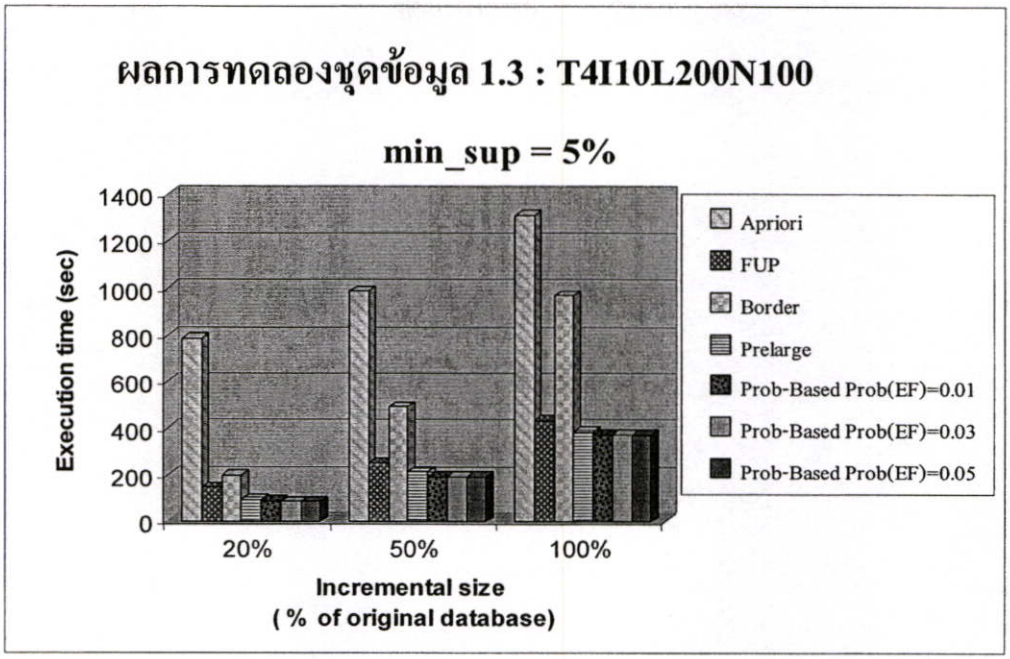
เวลาในการทำงาน (วินาที) ของชุดข้อมูลที่ 1.2 : T10I10L200N100								
ค่าสนับสนุนน้อยที่สุด (min_sup)	ขนาดข้อมูลใหม่ |db|	Algorithm						
		Apriori	FUP	Border	Prelarge	Probability-Based		
						$Prob_{EF}$		
						0.01	0.03	0.05
1%	20%	25,5801	43,996	25,727	102,995	24,353	22,607	22,921
	50%	316,048	56,828	63,631	120,335	41,537	39,122	39,769
	100%	432,063	81,460	129,159	137,798	69,992	65,601	66,295
3%	20%	17,134	1,589	3,126	1,454	1,242	1,183	1,248
	50%	21,167	3,112	7,543	3,006	2,902	2,811	2,861
	100%	28,430	5,893	15,198	5,930	5,744	5,623	5,659
5%	20%	11,673	1,273	2,593	1,099	932	938	936
	50%	14,559	2,525	6,269	2,391	2,241	2,233	2,235
	100%	19,506	4,827	13,008	4,658	4,535	4,527	4,560



รูปที่ 4.7 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับชุดข้อมูลที่ 1.3 เมื่อเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 20%, 50% และ 100% ด้วยค่าสนับสนุนน้อยที่สุดคือ 1%



รูปที่ 4.8 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับชุดข้อมูลที่ 1.3 เมื่อเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 20%, 50% และ 100% ด้วยค่าสนับสนุนน้อยที่สุดคือ 3%



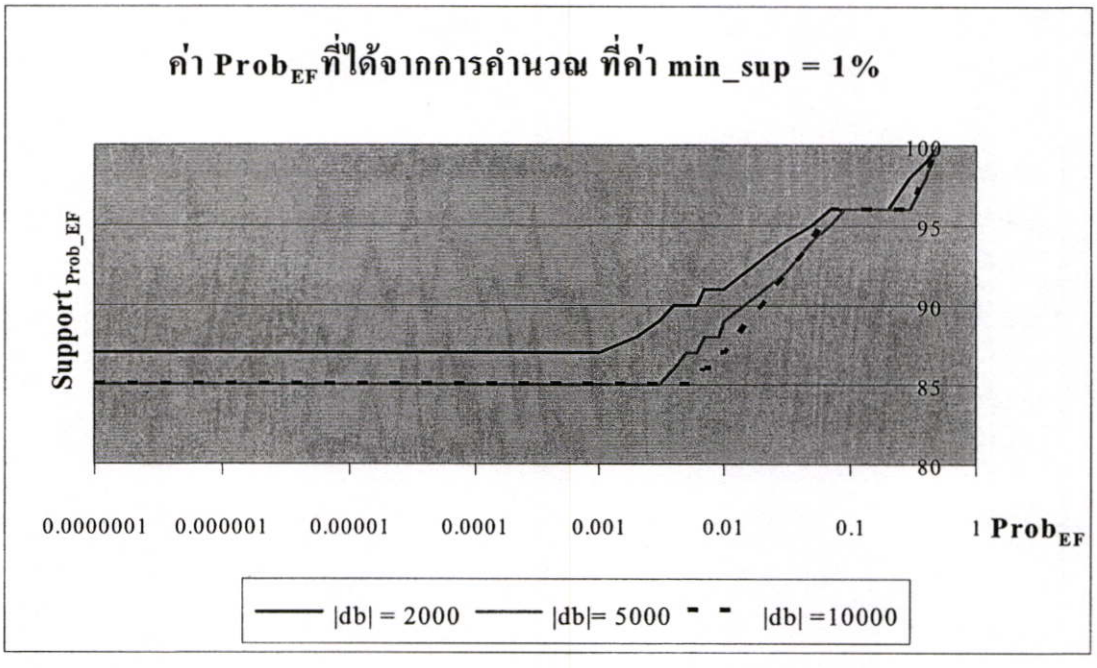
รูปที่ 4.9 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับชุดข้อมูลที่ 1.3 เมื่อเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 20%, 50% และ 100% ด้วยค่าสนับสนุนน้อยที่สุดคือ 5%

ตารางที่ 4.4 แสดงเวลากระทำของอัลกอริทึมสำหรับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และพรีลาจก์ที่ใช้ในการค้นหาการเพิ่มขยายกฎความสัมพันธ์เมื่อมีการเพิ่มข้อมูลใหม่ด้วยชุดข้อมูล 1.3

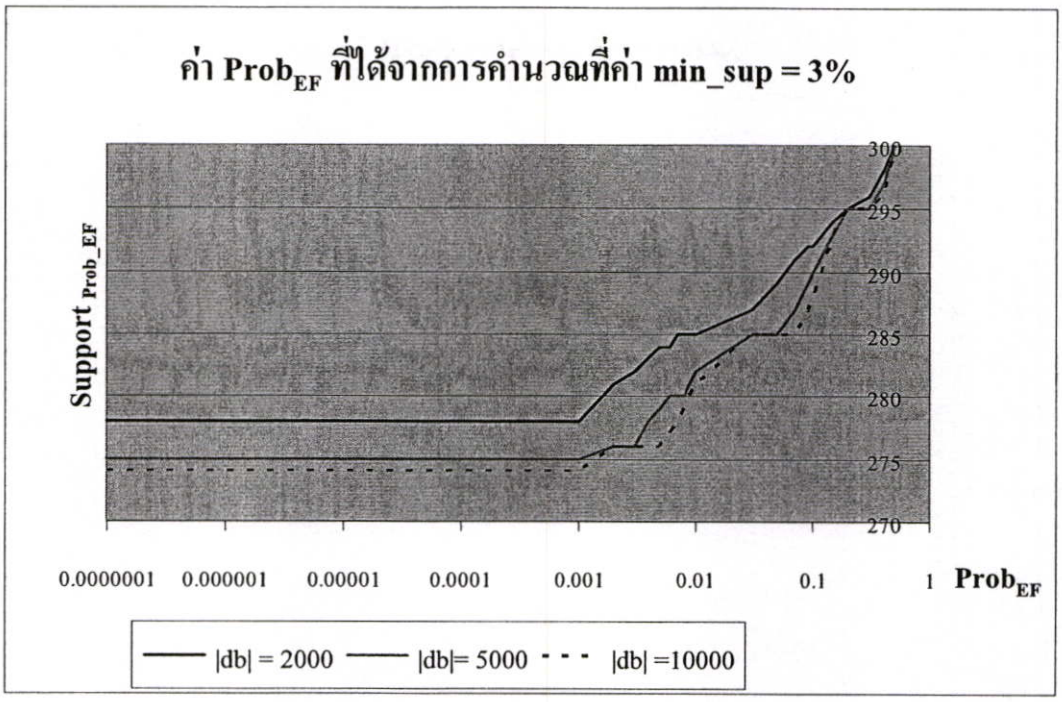
เวลาในการทำงาน (วินาที) ของชุดข้อมูลที่ 1.3 : T4I10L200N100								
ค่าสนับสนุนน้อยที่สุด (min_sup)	ขนาดข้อมูลใหม่ |db|	Algorithm						
		Apriori	FUP	Border	Prelarge	Probability-Based		
						$Prob_{EF}$		
						0.01	0.03	0.05
1%	20%	5,778	602	694	742	419	416	415
	50%	7,304	1,070	1,756	1,434	946	934	926
	100%	9,777	1,925	3,484	2,430	1,847	1,832	1,812
3%	20%	1,357	196	288	159	139	143	140
	50%	1,786	385	819	343	329	330	329
	100%	2,378	686	1,557	648	634	634	632
5%	20%	785	146	199	94	88	86	85
	50%	986	254	495	213	194	194	194
	100%	1,311	433	970	384	373	373	373

การกำหนดค่า $Prob_{EF}$ ผู้ใช้สามารถเพิ่มหรือลดค่า $Prob_{EF}$ ได้ตามความต้องการของผู้ใช้ โดยค่า $Prob_{EF}$ ที่ผู้ใช้กำหนดจะนำไปคำนวณดังสมการที่ 3.2 และ3.3 ที่ได้กล่าวไว้ในบทที่ 3 รูปที่ 4.10, 4.11, 4.12 แสดงค่า $Prob_{EF}$ ระดับต่างๆ คือ ค่า $Prob_{EF}$ ที่มีค่ามากกว่าหรือเท่ากับ 0.0000001 และค่า $Prob_{EF}$ ที่น้อยกว่าหรือเท่ากับ 0.5 ที่ได้จากการคำนวณเมื่อมีข้อมูลใหม่ขนาด 2000, 5000 และ 10000 ทรานแซคชันเข้าไปในฐานข้อมูลเดิมขนาด 10000 ทรานแซคชัน ที่ค่าสนับสนุนน้อยที่สุด 1%, 3% และ 5% ซึ่งจะพบว่าค่าของ $Prob_{EF}$ ที่คำนวณได้จะมีค่าสนับสนุนที่นำมาใช้เป็นเกณฑ์ในการพิจารณาว่าไอเทมเซตใดในฐานข้อมูลเดิมที่สามารถจะกลายมาเป็นฟรีเควนท์ไอเทมเซตเมื่อมีการเพิ่มข้อมูลใหม่จำนวนหนึ่งเข้ามาในฐานข้อมูลเดิม ค่า $Prob_{EF}$ ที่คำนวณได้จะขึ้นกับขนาดของข้อมูลที่เพิ่มและค่าสนับสนุนน้อยที่สุดที่ใช้ในการค้นหากฎความสัมพันธ์

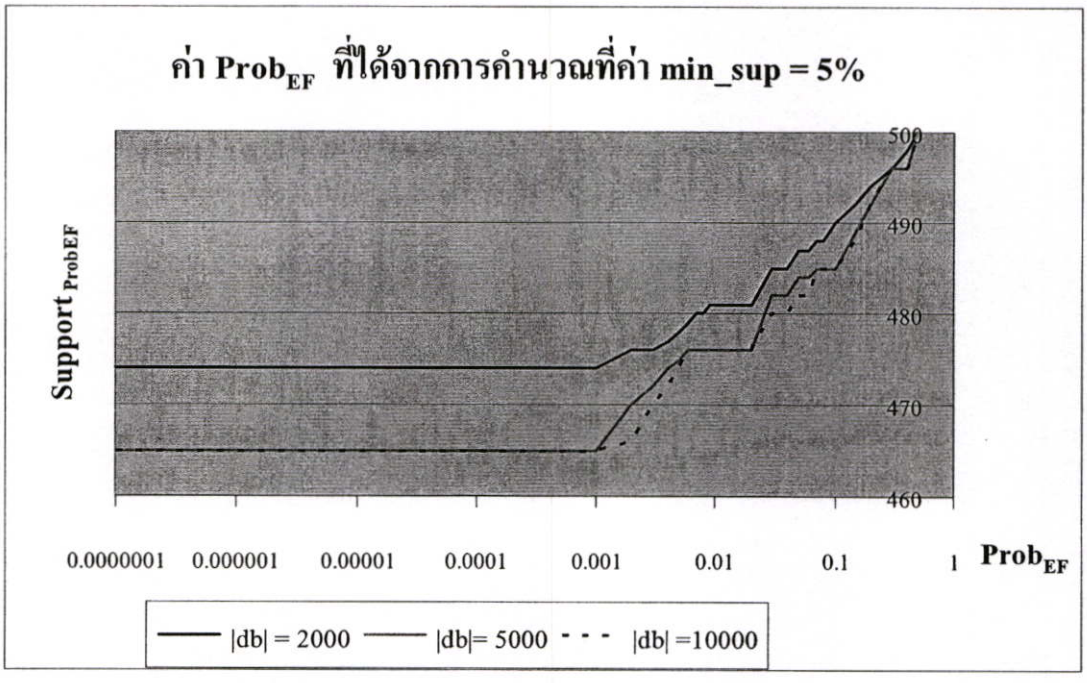
จากกราฟที่แสดงในรูปที่ 4.10, 4.11, 4.12 จะพบว่าค่า $Support_{Prob\_EF}$ ที่คำนวณได้แตกต่างกันในแต่ละค่า $Prob_{EF}$ ที่กำหนด สำหรับการทดลองในบทนี้ได้กำหนดค่า $Prob_{EF}$ = 0.01, 0.03 และ 0.05 เพื่อใช้ในการวัดประสิทธิภาพสำหรับการทดลองชุดข้อมูลที่ 1 และชุดข้อมูลที่ 3 จากกราฟจะเห็นได้ว่า ค่า $Support_{Prob\_EF}$ ที่เข้าใกล้ $Prob_{EF}$ = 0.01 แทบไม่มีความแตกต่างกันดังนั้นจึงอนุมานให้ $Prob_{EF}$ = 0.01 ใช้สำหรับการทดลองค่า $Support_{Prob\_EF}$ ที่มีค่าน้อย ส่วน $Prob_{EF}$ = 0.03 และ $Prob_{EF}$ = 0.05 ใช้สำหรับการทดลองค่า $Support_{Prob\_EF}$ ที่มีค่าปานกลางและสูงตามลำดับ



รูปที่ 4.10 แสดงการคำนวณค่า $Prob_{EF}$ ที่ค่าสนับสนุนน้อยที่สุด 1% เมื่อเพิ่มข้อมูลขนาด 2000, 5000 และ 10000 ทรานแซคชันเข้าไปในฐานข้อมูลเดิม



รูปที่ 4.11 แสดงการคำนวณค่า Prob$_{EF}$ ที่ค่าสนับสนุนน้อยที่สุด 3% เมื่อเพิ่มข้อมูลขนาด 2000, 5000 และ 10000 ทรานแซคชันเข้าไปในฐานข้อมูลเดิม



รูปที่ 4.12 แสดงการคำนวณค่า Prob$_{EF}$ ที่ค่าสนับสนุนน้อยที่สุด 3% เมื่อเพิ่มข้อมูลขนาด 2000, 5000 และ 10000 ทรานแซคชันเข้าไปในฐานข้อมูลเดิม

ค่า $Prob_{EF}$ ที่เพิ่มจะมีผลต่อจำนวนไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ไอเทมเซตที่ต้องเก็บ เมื่อกำหนดค่า $Prob_{EF}$ น้อยจะทำให้มีจำนวนไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ไอเทมเซตจำนวนมาก ในขณะที่ค่ากำหนดค่า $Prob_{EF}$ มากจะทำให้มีจำนวนไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ไอเทมเซตจำนวนน้อยดังแสดงตัวอย่างในตารางที่ 4.5

จำนวนไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ไอเทมเซตจะมีผลต่อเวลาที่ใช้ในการค้นหากฎความสัมพันธ์สำหรับการทำงานของอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์ โดยใช้หลักความน่าจะเป็นนี้จะสามารถทำงานได้ดีภายใต้สมมติฐานที่ค่าสถิติของฐานข้อมูลเดิมและฐานข้อมูลใหม่ที่เพิ่มเข้ามาไม่แตกต่างกันหรือแตกต่างกันน้อยมาก ซึ่งในกรณีนี้จะพบว่าเวลาที่ใช้ในการเพิ่มขยายการค้นหากฎความสัมพันธ์สามารถแบ่งได้เป็น 2 ส่วนคือ ส่วนที่ใช้ในการปรับปรุงค่าสนับสนุนที่พบในฐานข้อมูลใหม่ให้กับฟรีเควนท์และไอเทมที่คาดว่าจะกลายเป็นฟรีเควนท์และเวลาที่ใช้ในการสแกนฐานข้อมูลเดิมเมื่อพบว่ามีไอเทมเซตใหม่ที่เกิดขึ้นซึ่งไอเทมเซตใหม่นี้ไม่เป็นสมาชิกของฟรีเควนท์และไอเทมที่คาดว่าจะกลายเป็นฟรีเควนท์ แต่พบว่าไอเทมเซตใหม่นี้มีปรากฏในฐานข้อมูลเดิม

ในกรณีที่มีกำหนดค่า $Prob_{EF}$ น้อยจะมีจำนวนของฟรีเควนท์และไอเทมที่คาดว่าจะกลาย เป็นฟรีเควนท์ไอเทมเซตมีจำนวนมากกว่าค่า $Prob_{EF}$ ที่มากกว่า ซึ่งในกรณีนี้จะทำให้สามารถลดเวลาในการสแกนฐานข้อมูลเดิมไปได้เนื่องจากสามารถเก็บไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ได้ครอบคลุม ในขณะเดียวกันก็จะใช้เวลาในการปรับปรุงค่าสนับสนุนให้กับฟรีเควนท์ไอเทมเซตและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ สำหรับกรณีที่กำหนดค่า $Prob_{EF}$ สูงจะทำให้มีจำนวนไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ไอเทมเซตจำนวนน้อยทำให้ใช้เวลาในการปรับปรังค่าสนับสนุนให้กับฟรีเควนท์ไอเทมเซตและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์น้อย แต่ถ้าพบไอเทมเซตใหม่ที่มีค่าน้อยกว่า $Prob_{EF}$ ที่กำหนดจะต้องใช้เวลาในการสแกนฐานข้อมูลใหม่เพื่อหาค่าสนับสนุนให้กับไอเทมเซตใหม่นี้ ดังเช่นข้อมูลที่แสดงในตารางที่ 4.5 การทดลองชุดข้อมูลที่ 1.1 จะพบว่าที่การเพิ่มข้อมูลขนาดใหญ่เช่นที่ขนาด 5000 และ 10000 ทรานแซคชันในการเพิ่มขยายการค้นหากฎความ สัมพันธ์ที่ค่าสนับสนุนน้อยที่สุด 1% พบว่าที่ค่า $Prob_{EF} = 0.01$ จะใช้เวลาในการปรับปรุงค่าสนับสนุนให้กับฟรีเควนท์ไอเทมเซตในขณะที่ค่า $Prob_{EF}$ ที่กำหนดเท่ากับ 0.05 จะใช้เวลาในการปรับปรุงน้อยกว่า ค่า $Prob_{EF} = 0.01$ และ 0.03 เช่นเดียวกับข้อมูลที่พบในการเพิ่มขยายการค้นหากฎความสัมพันธ์ในตารางที่ 4.6 การทดลองชุดข้อมูลที่ 1.2 และตารางที่ 4.7 การทดลองชุดข้อมูลที่ 1.3 นอกจากนี้จะพบกรณีพิเศษที่พบในตารางที่ 4.6 และ 4.7 คือ เวลาที่ใช้ในการสแกนฐานข้อมูลเป็นศูนย์ หมายความว่าในการเพิ่มขยายการค้นหากฎความ สัมพันธ์โดยใช้หลักความน่าจะเป็นใช้เวลาในการปรับปรุงค่าสนับสนุนให้กับฟรีเควนท์และไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์เท่านั้น ซึ่งสามารถครอบคลุมการเพิ่มขยายการค้นหากฎความสัมพันธ์ได้โดยไม่ต้องทำการสแกนฐานข้อมูล เดิมและมีผลทำให้อัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดย

ใช้หลักความน่าจะเป็นทำงานได้เร็วกว่าอัลกอริทึมอื่นๆ เช่น อะพริโอริ, เอฟยูพี, บอร์เดอร์และพรีลาจก์ แต่มีความถูกต้องเช่นเดียวกันกับการเพิ่มขยายการค้นหากฎความสัมพันธ์ที่ได้จากอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และพรีลาจก์

สรุปผลการทดลองชุดข้อมูลที่ 1 : ผลที่ได้จากการเพิ่มขยายการค้นหากฎความสัมพันธ์ในกรณีค่าทางสถิติของฐานข้อมูลเดิมและฐานข้อมูลใหม่ไม่แตกต่างกัน

จากผลการทดลองเพิ่มข้อมูลขนาด 20%, 50% และ 100% ของฐานข้อมูลเดิมด้วยค่าสนับสนุนน้อยที่สุดคือ 1%, 3%และ 5% สำหรับชุดข้อมูลสังเคราะห์ที่สร้างโดยกำหนดค่าพารามิเตอร์คือ $|T| = 10, |I| = 4, |L| = 200$ และ $N = 100$ พบว่าจากการทดลองชุดข้อมูลที่ 1 ทั้ง 3 ชุดคือชุดข้อมูลที่ 1.1, 1.2, 1.3 พบว่าอัลกอริทึมการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้ค่าความน่าจะเป็น ซึ่งได้กำหนดค่าคาดหวังไว้ 3 ค่า คือ $Prob_{EF}$ = 0.01, 0.03 และ 0.05 สามารถทำงานได้เร็วกว่าอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์และพรีลาจก์ เนื่องมาจากอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้ค่าความน่าจะเป็น ได้นำค่าความน่าจะเป็นของการเกิดไอเทมเซตในฐานข้อมูลเดิมมาช่วยในทำนายไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ k- ไอเทมเซตทำให้สามารถลดเวลาในการสแกนฐานข้อมูลได้ ดังตารางที่ 4.1 จะพบว่าจำนวนไอเทมที่คาดว่าจะกลายเป็นฟรีเควนท์ไอเทมเซตของอัลกอริทึมสำหรับพิ่มขยายการค้นหาโดยใช้หลักความน่าจะเป็นนั้นมีจำนวนน้อยกว่าจำนวนบอร์เดอร์ไอเทมเซตของอัลกอริทึมบอร์เดอร์และจำนวน ไอเทมที่คาดว่าจะกลายเป็นฟรีเควนท์ไอเทมเซตของอัลกอริทึมพรีลาจก์ เนื่องจากค่าความน่าจะเป็นที่เกิดขึ้นตรงกับสมมติฐานที่ว่าการเกิดของไอเทมเซตที่ปรากฏในฐานข้อมูลเดิมและฐานข้อมูลใหม่มีค่าความน่าจะเป็นในการเกิดของไอเทมเซตไม่แตกต่างกัน ทำให้อัลกอริทึมเพิ่มขยายการค้นหากฎความสัมพันธ์ในฐานข้อมูลโดยใช้หลักความน่าจะเป็นสามารถนำค่าฟรีเควนท์ไอเทมเซตและไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ในฐานข้อมูลเดิมมาช่วยให้การปรับปรุงและค้นหาฟรีเควนท์ไอเทมเซตใหม่ทำได้เร็วขึ้น

เมื่อนำเวลาที่ได้จากการทดลองชุดข้อมูลที่ 1 ซึ่งเป็นการเพิ่มขยายการค้นหากฎความสัมพันธ์ในกรณีค่าทางสถิติของฐานข้อมูลเดิมและฐานข้อมูลใหม่ไม่แตกต่างกัน เพื่อทดสอบความถูกต้องและประสิทธิภาพการทำงานของอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นมาวิเคราะห์เพื่อเปรียบเทียบเวลาในการทำงานกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์และพรีลาจก์ โดยการวิเคราะห์ความแปรปรวนแบบจำแนกสองทางโดยใช้วิธีน็อนพาราเมตริกแยกตามค่าสนับสนุนน้อยที่สุดที่ทดลอง คือ 1%, 3% และ 5% ผลการทดสอบพบว่าเวลาเฉลี่ยระหว่างอัลกอริทึมอะพริโอริ,เอฟยูพี, บอร์เดอร์ และพรีลาจก์แตกต่างกับเวลาเฉลี่ยของอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นที่ค่าสนับสนุนน้อยที่สุด 1% 3% และ 5% อย่างมีนัยสำคัญ .05 ดังแสดงรายละเอียดในภาคผนวก ข.

ตารางที่ 4.5 แสดงเวลาที่ใช้ในการปรับปรุงข้อมูลและเวลาในการสแกนฐานข้อมูลเดิมของอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นชุดข้อมูลที่ 1.1

การทดลองชุดข้อมูลที่ 1.1										
ค่าสนับสนุนน้อยที่สุด	ขนาดข้อมูล	$Prob_{EF}$=0.01			$Prob_{EF}$=0.03			$Prob_{EF}$=0.05		
		เวลาในการปรับปรุง	เวลาในการสแกน	เวลารวม	เวลาในการปรับปรุง	เวลาในการสแกน	เวลารวม	เวลาในการปรับปรุง	เวลาในการสแกน	เวลารวม
1%	2000	7,559.39	7.11	7,566.50	7,542.47	31.53	7,574.00	7,546.39	46.58	7,592.97
	5000	13,975.32	1.08	13,976.40	13,487.38	36.67	13,524.05	13,396.41	54.27	13,450.67
	10000	23,367.20	20.14	23,387.34	23,143.34	93.47	23,236.81	22,934.63	134.08	23,068.70
3%	2000	1,188.45	0.67	1,189.13	1,319.90	0.63	1,320.53	1,198.86	24.28	1,223.14
	5000	2,467.17	0.02	2,467.19	2,548.55	12.74	2,561.28	2,563.59	26.09	2,589.69
	10000	4,900.36	20.77	4,921.13	4,882.60	33.86	4,916.46	4,920.00	34.58	4,954.58
5%	2000	755.64	0.55	756.19	759.88	1.42	761.3	757.14	3.157	760.3
	5000	1894	0.016	1894.01	1999.235	0.015	1999.25	2000.93	0.73	2001.66
	10000	3781.28	1.17	3782.45	3841.34	1.502	3842.84	3895.44	3.58	3899.02

ตารางที่ 4.6 แสดงเวลาที่ใช้ในการปรับปรุงข้อมูลและเวลาในการสแกนฐานข้อมูลเดิมของอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นชุดข้อมูลที่ 1.2

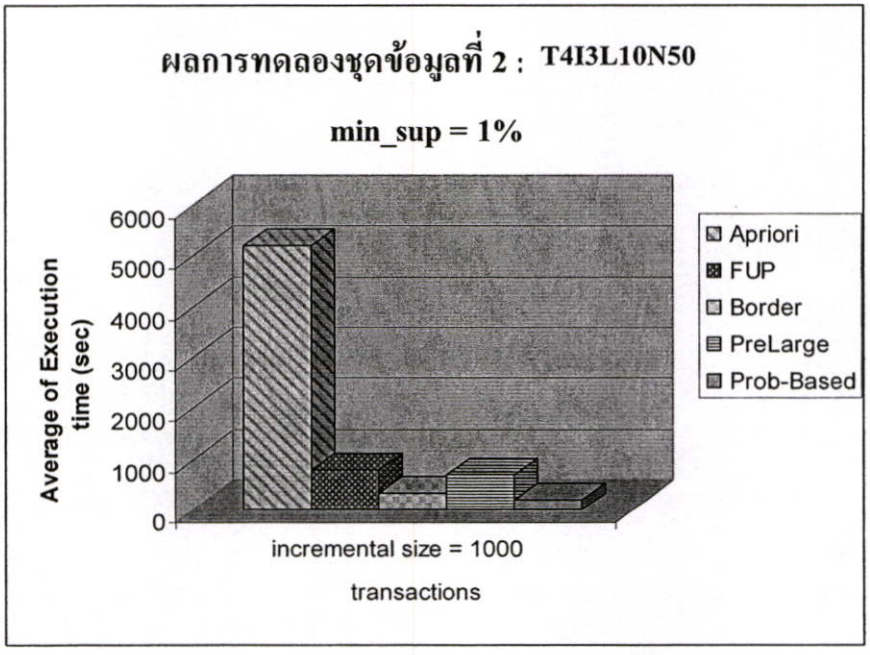
การทดลองชุดข้อมูลที่ 1.2										
ค่าสนับสนุนน้อยที่สุด	ขนาดข้อมูล	$Prob_{EF}$ =0.01			$Prob_{EF}$ =0.03			$Prob_{EF}$ =0.05		
		เวลาในการปรับปรุง	เวลาในการสแกน	เวลารวม	เวลาในการปรับปรุง	เวลาในการสแกน	เวลารวม	เวลาในการปรับปรุง	เวลาในการสแกน	เวลารวม
1%	2000	24,349.63	4.33	24,353.95	22,591.06	16.33	22,607.39	22,877.36	43.77	22,921.13
	5000	41,535.56	1.78	41,537.34	39,118.16	4.81	39,122.97	39,764.34	5.13	39,769.47
	10000	69,951.27	41.20	69,992.47	65,522.58	79.38	65,601.96	66,202.28	93.28	66,295.56
3%	2000	1,242.91	0	1,242.91	1,183.16	0	1,183.16	1,248.59	0	1,248.59
	5000	2,901.69	0.70	2,902.39	2,808.98	2.34	2,811.33	2,859.47	2.30	2,861.77
	10000	5,744.38	0	5,744.38	5,623.13	0	5,623.13	5,659.59	0	5,659.59
5%	2000	932.38	0	932.38	938.14	0.02	938.16	936.52	0.03	936.55
	5000	2,241.31	0	2,241.31	2,232.77	0.64	2,233.41	2,234.69	0.66	2,235.34
	10000	4,535.30	0.02	4,535.31	4,526.53	0.92	4,527.45	4,557.29	3.20	4,560.50

ตารางที่ 4.7 แสดงเวลาที่ใช้ในการปรับปรุงข้อมูลและเวลาในการสแกนฐานข้อมูลเดิมของอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นชุดข้อมูลที่ 1.3

การทดลองชุดข้อมูลที่ 1.3										
ค่าสนับสนุนน้อยที่สุด	ขนาดข้อมูล	Prob : PL =0.01			Prob : PL =0.03			Prob : PL =0.05		
		เวลาในการปรับปรุง	เวลาในการสแกน	เวลารวม	เวลาในการปรับปรุง	เวลาในการสแกน	เวลารวม	เวลาในการปรับปรุง	เวลาในการสแกน	เวลารวม
1%	2000	419.36	0	419.36	414.86	2.12	416.98	412.78	2.88	415.66
	5000	946.94	0	946.94	934.08	0.28	934.36	925.53	0.49	926.02
	10000	1,847.63	0	1,847.63	1,832.16	0	1,832.16	1,811.44	0.84	1,812.28
3%	2000	139.67	0.02	139.69	143.39	0	143.39	140.38	0	140.38
	5000	329.36	0	329.36	330.92	0	330.92	329.30	0	329.30
	10000	634.44	0	634.44	634.61	0	634.61	632.55	0.02	632.56
5%	2000	88.33	0	88.33	86.93	0	86.93	85.21	0	85.21
	5000	194.66	0	194.66	194.31	0	194.31	194.03	0	194.03
	10000	373.16	0	373.16	373.42	0	373.42	373.41	0	373.41

4.3.2 ผลการทดลองชุดข้อมูลที่ 2 : ผลที่ได้จากการทำนายฟรีเควนท์ไอเทมเซตของไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงที่ได้จากการคำนวณโดยใช้หลักความน่าจะเป็นด้วยทฤษฎีเบอร์นูลลี

จากผลการทดลองเพื่อวัดความถูกต้องและประสิทธิภาพของอัลกอริทึมด้วยชุดข้อมูลที่ 2 ซึ่งมีสมมติฐานของการทดลองในการหาว่าค่าความน่าจะเป็นที่ได้จากการนำทฤษฎีเบอร์นูลลีมาประยุกต์ใช้ในการคำนวณหาไอเทมที่คาดว่าจะกลายเป็นฟรีเควนท์ไอเทมเซตของอัลกอริทึมในการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้ค่าความน่าจะเป็น สามารถนำมาใช้ในการทำนายการเกิดฟรีเควนท์ไอเทมเซตเมื่อมีข้อมูลจำนวนหนึ่งเพิ่มเข้ามาในฐานข้อมูลเดิมได้จริง โดยกำหนดค่าคาดหวังในการทดลองสำหรับอัลกอริทึมการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้ค่าความน่าจะเป็น เท่ากับ 0.01 ซึ่งมีความหมายว่าถ้าไอเทมเซตใดๆ ที่มีค่าความน่าจะเป็นในการเกิดฟรีเควนท์ไอเทมเซต 1 ครั้งในการทดลองเพิ่มข้อมูลใหม่ขนาดเท่าๆกันจำนวนหนึ่งเข้ามา 100 ครั้ง จะจัดเป็นไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซต ดังนั้นในการทดลองนี้จึงทดลองโดยการเพิ่มข้อมูลขนาด 1000 ทรานแซคชันจำนวน 100 ชุด โดยทดลองเพิ่มข้อมูลทีละ 1 ชุดเข้าไปในฐานข้อมูลเดิม ด้วยค่าสนับสนุนน้อยที่สุด เท่ากับ 1% เพื่อทดสอบสมมิตฐานดังกล่าวข้างต้น ตารางที่ 4.8 และรูปที่ 4.13 แสดงเวลาเฉลี่ยที่ได้จากการเพิ่มข้อมูลจำนวน 100 ชุดสำหรับอัลกอริทึม อะพริโอริ เอฟยูพี บอร์เดอร์ พรีลาจก์และอัลกอริทึมการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้ค่าความน่าจะเป็น



รูปที่ 4.13 แสดงผลการเปรียบเทียบเวลาเฉลี่ยที่ได้จากการทดลองข้อมูลชุดที่ 2

ตารางที่ 4.8 แสดงการเปรียบเทียบค่าเฉลี่ยของเวลาสำหรับการทดลองเพิ่มข้อมูล 100 ชุด

	ค่าเฉลี่ยของเวลาสำหรับการทดลองเพิ่มข้อมูล 100 ชุด				
min_sup(%)	**Apriori**	**FUP**	**Border**	**Prelarge**	**Prob-Based**
1%	5,210	789	335	691	200

สรุปผลการทดลองชุดข้อมูลที่ 2: ผลที่ได้จากการทำนายฟรีเควนท์ไอเทมเซตของไอเทมที่คาดว่าจะเป็น ฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงที่ได้จากการคำนวณโดยใช้หลักความน่าจะเป็นด้วยทฤษฎีเบอร์นูลลี

จากผลการทดลองชุดข้อมูลที่ 2 ที่แสดงในรูปที่ 4.13 และตารางที่ 4.8 พบว่าอัลกอริทึมการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นใช้เวลาในการเพิ่มขยายการค้นหาฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงมีค่าเฉลี่ยน้อยกว่าอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และพรีลาจก์ การที่อัลกอริทึมการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นสามารถทำงานได้เร็วกว่าอัลกอริทึมอื่น ส่วนหนึ่งมีสาเหตุ มาจากไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตกลายมาเป็นฟรีเควนท์ไอเทมเซตจริงตามค่าความจะเป็นที่คำนวณไว้ทำให้สามารถลดเวลาในการสแกนฐานข้อมูลได้ ตารางที่ 4.9 แสดงค่าไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตจากฐานข้อมูลเดิมทั้งหมดจำนวน 55 ไอเทมเซต ที่มีค่าความน่าจะเป็นในฐานข้อมูลเดิมมากกว่าค่าคาดหวังที่กำหนดไว้คือ 0.01 เมื่อนำมาค่าความน่าจะเป็นที่ได้จากการคำนวณด้วยหลักของเบอร์นูลลีมาเปรียบเทียบกับการที่ไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ไอเทมเซตที่ได้กลายเป็นฟรีเควนท์ไอเทมเซตจริงจากการเพิ่มข้อมูลจำนวน 100 ชุด พบว่าค่าความน่าจะเป็นที่ได้จากการคำนวณด้วยทฤษฎีเบอร์นูลลีนั้นจะมีบางค่าที่ได้จากคำนวณด้วยทฤษฎีเบอร์นูลลีให้ผลที่แตกต่างจากค่าที่มีการเกิดจริงที่เป็นเช่นนี้พบว่ามีผลมาจากค่าทางสถิติที่ใช้ในการสร้างข้อมูลโดยไอเทมที่นำมาสร้างในทรานแซคชันต่างๆ นั้นจะถูกสร้างโดยการสุ่มไอเทมแต่ละตัวจากค่าสถิติของการเกิดจำนวนสูงสุดที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้ที่มีการกำหนดค่าน้ำหนักให้กับชุดของจำนวนสูงสุดที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้แต่ละชุดแตกต่างกัน และค่าน้ำหนักนี้จะมีพบต่อการสุ่มหยิบข้อมูลลงในทรานแซคชัน โดยมีปัจจัยอื่นๆ ที่เกี่ยวข้องเช่น จำนวนไอเทมที่กำหนด ค่าเฉลี่ยของขนาด ทรานแซคชัน | T | และค่าเฉลี่ยของขนาดสูงสุดที่จะเป็นฟรีเควนท์ไอเทมเซตได้ (| I |) ที่แตกต่างกัน ที่มีผลต่อการสร้างชุดข้อมูลสังเคราะห์สำหรับทดลองโดยสามารถแบ่งได้เป็น 2 กรณีคือ

กรณีที่ 1 ถ้ากำหนดค่า | T | น้อยกว่าค่า | I | อาจพบว่าใน 1 ไอเทมเซตอาจประกอบด้วยจำนวนสูงสุดที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้มาจากชุดเดียวกันซึ่งอาจไม่มีความหลากหลายของไอเทมเซต

กรณีที่ 2 กำหนด | T | มากกว่าหรือเท่ากับค่า | I | อาจพบว่าใน 1 ทรานแซคชันประกอบด้วยหลายชุดของไอเทมเซตที่มีความหลากหลาย ทำให้ฟรีเควนท์ไอเทมเซตใหม่ๆ ที่อยู่นอกเหนือจากค่าสถิติของการเกิดจำนวนสูงสุดที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้

สำหรับในการทดลองชุดที่ 2 นี้กำหนดค่า | T | = 4 และ | I | =3 จึงมีความเป็นไปได้ที่อาจพบชุดของไอเทมเซตที่มีความคลาดเคลื่อนจากการสุ่มหยิบไอเทมเพื่อนำมาสร้างทรานแซคชัน ดังนั้นเมื่อนำค่าความน่าจะเป็นที่ได้จากการคำนวณโดยใช้หลักของเบอร์นูลลีมาหาค่าผลต่างเทียบกับค่าที่เกิดจริงจากการทดลอง 100 ครั้งจะได้ค่าเฉลี่ยของผลต่างที่มีค่าความคลาดเคลื่อนจำนวนหนึ่ง

เพื่อทดสอบค่าความคลาดเคลื่อนที่เกิดจากค่าเฉลี่ยของผลต่างระหว่างค่าที่ได้จากการคำนวณความน่าจะเป็นไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตโดยใช้หลักของเบอร์นูลลีกับค่าที่ไอเทมเซตเกิดจริงในฐานข้อมูลปรับปรุง จึงได้นำกระบวนการทางสถิติ t-test มาใช้สำหรับเปรียบเทียบค่าเฉลี่ยทั้ง 2 ค่าข้างต้นที่ได้มาจากไอเทมที่คาดว่าจะกลายเป็นฟรีเควนท์ k-ไอเทมเซต จำนวน 55 ไอเทมเซต โดยมีสมมติฐาน คือ ค่าเฉลี่ยของไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท k-ไอเทมเซตจากการคำนวณโดยใช้ทฤษฎีเบอร์นูลลี ไม่แตกต่างจากค่าเฉลี่ยของการเกิดจริงที่ได้การนับจำนวนครั้งที่ไอเทมที่คาดว่าจะเป็นฟรีเควนท์ k-ไอเทมเซตในฐานข้อมูลปรับปรุงจำนวน 100 ชุด หรือค่าผลต่างระหว่างค่าเฉลี่ยที่ได้จากการคำนวณค่าความน่าจะเป็นด้วยหลักของเบอร์นูลลีและค่าเฉลี่ยไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตกลายเป็นฟรีเควนท์ k-ไอเทมเซตที่เกิดจริงมีค่าเท่ากับศูนย์ดังแสดงในสมมติฐานทางสถิติต่อไปนี้

สมมติฐานทางสถิติ :

H_0 = ค่าเฉลี่ยของ (ค่าที่ได้จากการคำนวณ - ค่าที่เกิดจริง) = 0

H_1 = ค่าเฉลี่ยของ (ค่าที่ได้จากการคำนวณ - ค่าที่เกิดจริง) $\neq$ 0

$$t = \frac{\sum D}{\sqrt{\frac{N\sum D^2 - (\sum D)^2}{N-1}}}$$

$$df = n\text{-}1$$

จากรูปที่ 4.14 แสดงผลการทดสอบทางสถิติ t-test โดยใช้โปรแกรม SPSS แสดงค่าแตกต่างระหว่างค่าที่ได้จากการคำนวณค่าความน่าจะเป็นของไอเทมที่คาดว่าจะกลายเป็นฟรีเควนท์ไอเทมเซตโดยใช้ทฤษฎีเบอร์นูลลี แสดงด้วยตัวแปรชื่อ theory และผลลัพธ์ของการที่ไอเทมเซตเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ไอเทมเซตเกิดขึ้นจริงในฐานข้อมูลปรับปรุงจำนวน 100 ครั้ง แสดงด้วยตัวแปรชื่อ result โดยในชุดการทดลองที่ 2 นี้พบว่ามีไอเทมที่คาดว่าจะกลายเป็นฟรีเควนท์

ไอเทมเซตจำนวน 55 ไอเทมเซต เมื่อนำมาหาค่าความแตกต่างพบว่ามีค่าเฉลี่ยของค่าแตกต่าง คือ 0.02382 $\left(\overline{D} = 0.02382\right)$ ค่าเบี่ยงเบนมาตรฐานของความแตกต่าง (Std. Deviation) มีค่าเท่ากับ 0.11143 ค่าสถิติทดสอบ t-test ได้เท่ากับ 1.585, df = 54 มีนัยสำคัญทางสถิติที่ .119 ซึ่งมากกว่า 0.05 สรุปได้ว่าค่าเฉลี่ยของค่าที่ได้จากการคำนวณหาค่าความน่าจะเป็นของไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตโดยใช้ทฤษฎีเบอร์นูลลีไม่แตกต่างกับค่าเฉลี่ยของการที่ไอเทมเหล่านั้นกลายเป็นฟรีเควนท์จริงจากการทดลองเพิ่มข้อมูลจำนวน 100 ชุด เข้าไปในฐานข้อมูลอย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05

ตารางที่ 4.9 แสดงการเปรียบเทียบผลที่ไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์กลายเป็นฟรีเควนท์ในฐานข้อมูลปรับปรุงจากการคำนวณกับจำนวนที่พบว่าเป็นฟรีเควนท์ไอเทมเซตจาก 100 ชุดข้อมูล

ลำดับ	ไอเทมที่คาดว่าจะเป็นฟรีเควนท์ k-ไอเทมเซตจากฐานข้อมูลเดิม	ค่าสนับสนุนในฐานข้อมูลเดิม	ค่าความน่าจะเป็นที่คำนวณด้วยทฤษฎีเบอร์นูลลี	จำนวนที่พบว่าเป็นฟรีเควนท์ไอเทมเซตจาก 100 ชุดข้อมูล
1	2 6	97	0.21	0.21
2	2 28	93	0.01	0.01
3	2 45	95	0.08	0.02
4	5 42	98	0.3	0.13
5	25 36	99	0.42	0
6	29 42	99	0.42	0.56
7	32 35	94	0.02	0.1
8	2 10 40	98	0.3	0.43
9	2 11 12	95	0.08	0.2
10	2 20 47	95	0.08	0.01
11	2 20 49	98	0.3	0.31
12	2 24 32	97	0.21	0.23

ตารางที่ 4.9 แสดงการเปรียบเทียบผลที่ไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์กลายเป็นฟรีเควนท์ในฐานข้อมูลปรับปรุงจากการคำนวณกับจำนวนที่พบว่าเป็นฟรีเควนท์ไอเทมเซตจาก 100 ชุด (ต่อ)

ลำดับ	ไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตจากฐานข้อมูลเดิม	ค่าสนับสนุนในฐานข้อมูลเดิม	ค่าความน่าจะเป็นที่คำนวณด้วยทฤษฎีเบอร์นูลลี	จำนวนที่พบว่าเป็นฟรีเควนท์ไอเทมเซตจาก 100 ชุดข้อมูล
13	2 44 47	96	0.13	0.03
14	4 12 35	94	0.02	0.04
15	4 14 35	98	0.3	0.26
16	4 16 23	93	0.01	0.02
17	4 35 42	98	0.3	0.21
18	10 16 42	98	0.3	0.16
19	11 12 25	96	0.13	0.16
20	11 12 35	99	0.42	0
21	11 14 21	95	0.08	0.02
22	11 14 35	95	0.08	0.02
25	11 32 44	96	0.13	0.13
26	11 44 49	94	0.02	0.03
27	12 14 21	94	0.02	0.01
28	12 14 23	95	0.08	0.02
29	12 21 25	96	0.13	0.13
30	16 23 40	93	0.01	0
31	21 24 49	93	0.01	0.01
32	21 44 49	95	0.06	0.05
33	24 25 32	96	0.11	0.06
34	25 32 44	94	0.03	0.03
35	2 11 32 42	97	0.18	0.13
36	2 23 25 50	94	0.03	0.08

ตารางที่ 4.9 แสดงการเปรียบเทียบผลที่ไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์กลายเป็นฟรีเควนท์ในฐานข้อมูลปรับปรุงจากการคำนวณกับจำนวนที่พบว่าเป็นฟรีเควนท์ไอเทมเซตจาก 100 ชุดข้อมูล (ต่อ)

ลำดับ	ไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตจากฐานข้อมูลเดิม	ค่าสนับสนุนในฐานข้อมูลเดิม	ค่าความน่าจะเป็นที่คำนวณด้วยทฤษฎีเบอร์นูลลี	จำนวนที่พบว่าเป็นฟรีเควนท์ไอเทมเซตจาก 100 ชุดข้อมูล
37	2 25 42 50	97	0.18	0.44
38	4 11 32 50	93	0.01	0.01
39	4 12 42 50	94	0.03	0.03
40	4 21 32 50	96	0.11	0.07
41	4 21 42 49	95	0.06	0.07
42	4 23 42 47	94	0.03	0.02
43	4 23 49 50	98	0.28	0
44	4 42 44 50	93	0.01	0.02
45	11 23 32 49	95	0.06	0.1
46	11 23 32 50	94	0.03	0.03
47	11 25 32 42	95	0.06	0.02
48	12 23 42 50	96	0.11	0.08
49	23 25 42 49	94	0.03	0
50	2 4 11 42 50	98	0.28	0
51	2 4 11 21 23	96	0.11	0.06
52	2 4 21 42 50	97	0.18	0.17
53	2 4 21 23 50	94	0.03	0.03
54	2 4 21 23 25	93	0.01	0.01
55	4 11 21 25 32	94	0.03	0

T-Test

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	theory - result	.02382	.11143	.01503	-.00631	.05394	1.585	54	.119

รูปที่ 4.14 แสดงการทดสอบค่าทางสถิติ t-test เพื่อหาผลต่างของค่าเฉลี่ยที่ได้จากการคำนวณและค่าเฉลี่ยการเกิดขึ้นจริงของไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์ไอเทมเซต

4.3.3 ผลการทดลองชุดข้อมูล3: การทดลองในกรณีค่าทางสถิติของฐานข้อมูลเดิมและฐานข้อมูลใหม่แตกต่างกัน

การทดลองเพิ่มขยายการค้นหาฟรีเควนท์ไอเทมเซตสำหรับชุดข้อมูลที่ 3 เป็นการทดลองเพื่อหาความถูกต้องในการค้นหาฟรีเควนท์ k-ไอเทมเซตจากฐานข้อมูลปรับปรุงภายใต้สมมติฐานของฐานข้อมูลใหม่ที่มีความแตกต่างกันฐานข้อมูลเดิม จากการทดลองนี้จะพบว่ามีฟรีเควนท์ k- ไอเทมเซตใหม่เกิดขึ้นแตกต่างจากฟรีเควนท์ k-ไอเทมเซตที่พบในการไมน์นิ่งฐานข้อมูลเดิม ซึ่งทำให้อัลกอริทึมต่างๆ ต้องสแกนฐานข้อมูลเดิมเพื่อค้นหาค่าสนับสนุนสำหรับฟรีเควนท์ k-ไอเทมเซตใหม่

สรุปผลการทดลองชุดข้อมูลที่ 3: ผลที่ได้จากการทดลองในกรณีค่าทางสถิติของฐานข้อมูลเดิมและฐานข้อมูลใหม่แตกต่างกัน

ความถูกต้องในการค้นหาฟรีเควนท์ k-ไอเทมเซตที่ได้จากการปรับปรุงฐานข้อมูลของอัลกอริทึมต่างๆ จะนำมาเปรียบเทียบฟรีเควนท์ k-ไอเทมเซตที่ได้จากการไมน์นิ่งเพื่อค้นหาฟรีเควนท์ k-ไอเทมเซตจากฐานข้อมูลรวมทั้งหมดของอัลกอริทึมอะพริโอริ และอัลกอริทึมเอฟยูพี ซึ่งมีการนำฟรีเควนท์ k-ไอเทมเซตจากฐานข้อมูลเดิมมาปรับปรุงและนำฟรีเควนท์ k-ไอเทมเซตที่พบในฐานข้อมูลใหม่ที่เพิ่มเข้ามาไปสแกนหาค่าสนับสนุนในฐานข้อมูลเดิมเป็นหลัก

ในขณะที่อีก 3 อัลกอริทึมได้แก่ บอร์เดอร์, พรีลาจก์และการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น ซึ่งมีการจัดเก็บทั้งฟรีเควนท์ k-ไอเทมเซตและไอเทมเซตที่ไม่ใช่ฟรีเควนท์ k-ไอเทมเซตในฐานข้อมูลเดิม พบว่าฐานข้อมูลใหม่ที่เพิ่มเข้ามาในแต่ละชุดจะประกอบด้วยฟรีเควนท์ k-ไอเทมเซตใหม่ที่แตกต่างจากฟรีเควนท์ k-ไอเทมเซตและไอเทมเซตที่ไม่ใช่ฟรีเควนท์ k-ไอเทมเซตในฐานข้อมูลเดิมดังนั้นการค้นหาฟรีเควนท์ไอเทมเซตใหม่สำหรับทั้ง 3 อัลกอริทึมจึงต้องมีการสแกนฐานข้อมูลเดิมเพื่อหาค่าสนับสนุนสำหรับฟรีเควนท์

ไอเทมเซตใหม่ๆ เช่น อัลกอริทึมบอร์เดอร์จะพบโปรโมทไอเทมเซต ซึ่งเป็นไอเทมเซตที่เป็นสมาชิกของแคนดิเดทไอเทมเซตที่ไม่เป็นฟรีเควนท์ไอเทมเซตหรือที่เรียกว่าบอร์เดอร์ไอเทมเซตในฐานข้อมูลเดิมได้กลายมาเป็นฟรีเควนท์ไอเทมเซตใหม่ในฐานข้อมูลปรับปรุง ซึ่งทำให้มีการสแกนฐานข้อมูลทั้งหมดเพื่อหาฟรีเควนท์ไอเทมเซตที่เกิดจากการปรับปรุงทั้งหมดทำให้ใช้เวลาในการปรับปรุงฟรีเควนท์ k-ไอเทมเซตและบอร์เดอร์ k-ไอเทมเซต และเวลาในการค้นหาฟรีเควนท์ k ไอเทมเซตใหม่ที่เกิดขึ้นมากกว่าอัลกอริทึมพรีลาจก์และอัลกอริทึมในการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น

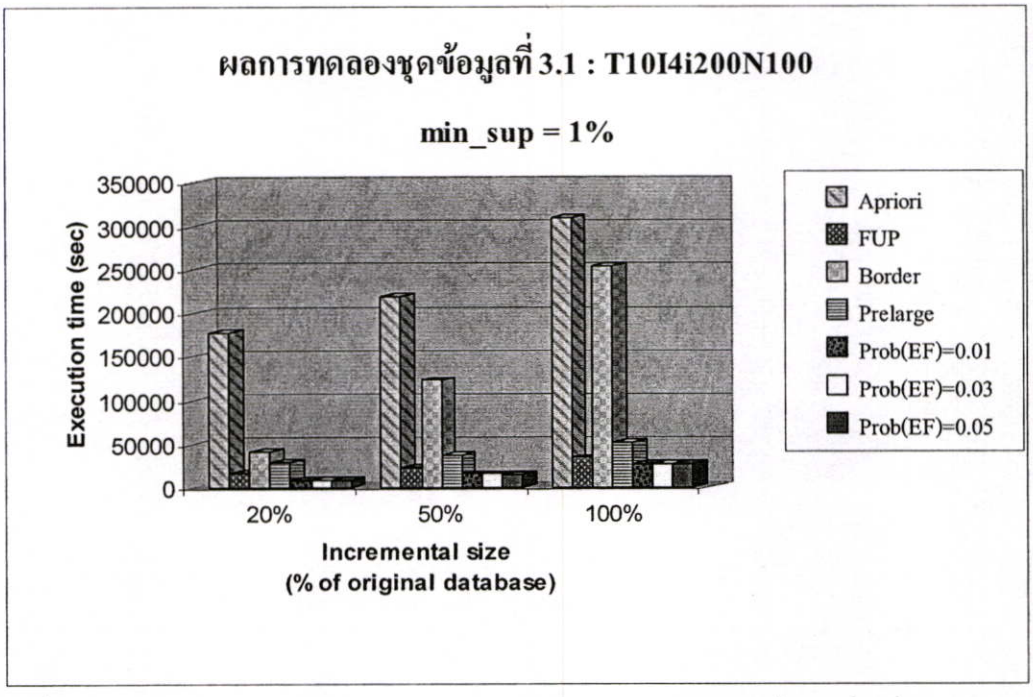
สำหรับอัลกอริทึมพรีลาจก์ซึ่งมีการเก็บไอเทมที่ไม่ใช่ฟรีเควนท์ไอเทมเซตที่พิจารณาจากค่าสนับสนุนที่มีค่าน้อยกว่าค่าสนับสนุนขั้นสูง หรือเท่ากับค่าสนับสนุนน้อยที่สุดแต่มีค่ามากกว่าค่าสนับสนุนขั้นต่ำ เรียกค่านี้ว่าพรีลาจก์ไอเทมเซต จากการทดลองทั้ง 3 ชุดของการทดลองชุดที่ 3 พบว่าพรีลาจก์อัลกอริทึมมีการปรับปรุงฟรีเควนท์ k-ไอเทมเซตและพรีลาจก์ k-ไอเทมเซตที่ได้จากการไมน์นิ่งในฐานข้อมูลเดิม โดยพบว่าพรีลาจก์บางไอเทมเซตเมื่อได้รับการปรับปรุงจากฐานข้อมูลใหม่ที่เพิ่มเข้ามาแล้วกลายเป็นฟรีเควนท์ไอเทมเซต ในขณะเดียวกันได้ค้นพบฟรีเควนท์ k-ไอเทมเซตใหม่เกิดขึ้น ซึ่งทำให้ต้องใช้เวลาในการสแกนฐานข้อมูลเดิมเพื่อทำการค้นหาฟรีเควนท์ k-ไอเทมเซตได้ครอบคลุมทั้งหมด

เช่นเดียวกับอัลกอริทึมเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น พบว่านอกจากฟรีเควนท์ k-ไอเทมเซตเดิมที่ยังคงเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงและมีไอเทมที่คาดว่าจะกลายเป็นฟรีเควนท์ k-ไอเทมเซตในฐานข้อมูลเดิมบางไอเทมเซตได้กลายมาเป็นฟรีเควนท์ k-ไอเทมเซตในฐานข้อมูลปรับปรุงแล้ว ยังพบว่ามีไอเทมเซตใหม่ที่อาจกลายมาเป็นฟรีเควนท์ไอเทมเซตได้หลังจากมีการรวมค่าสนับสนุนจากฐานข้อมูลเดิมที่ได้จากนำไอเทมเซตเหล่านี้ไปสแกนในฐานข้อมูล

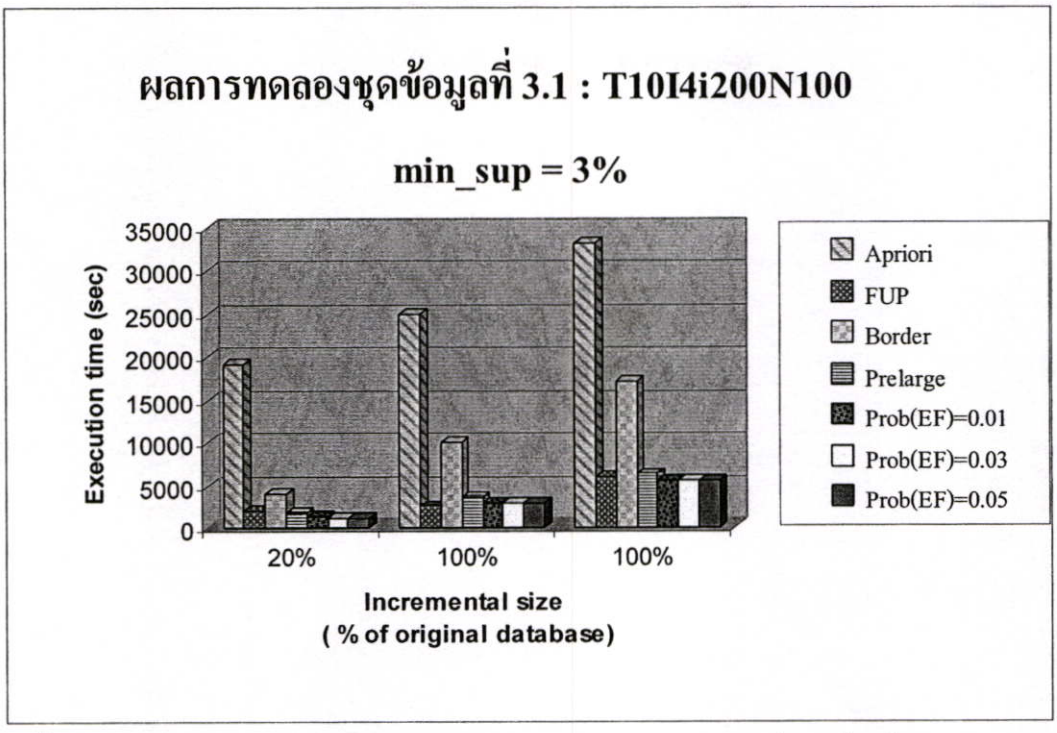
จำนวนฟรีเควนท์ k-ไอเทมเซตใหม่ที่ปรากฏจากการไมน์นิ่งฐานข้อมูลใหม่เข้าไปนี้มีจำนวนมากที่จะต้องสแกนในฐานข้อมูลเดิมเมื่อมีการวนรอบซ้ำในการค้นหาสำหรับ k-ไอเทมเซต สำหรับอัลกอริทึมเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นสามารถลดจำนวน k-ไอเทมเซตที่ต้องการสแกนในฐานข้อมูลเดิม โดยการนำค่าคาดหวังที่พบในฐานข้อมูลเดิมมาช่วยในการประมาณค่าสนับสนุนที่อาจเกิดได้สูงสุดสำหรับแต่ละ k-ไอเทมเซต ถ้าไอเทมเซตใดมีค่าสนับสนุนที่เกิดจากฐานข้อมูลใหม่บวกกับค่าสนับสนุนที่ได้จากการนำค่าคาดหวังในฐานข้อมูลเดิมมาประมาณมีค่ามากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุดของฐานข้อมูลปรับปรุงจึงจะนำไอเทมเซตเหล่านี้ไปสแกนในฐานข้อมูลเดิม จากแนวคิดนี้ทำให้สามารถลดจำนวนไอเทมเซตที่จะต้องนำไปสแกนฐานข้อมูลเดิมน้อยลง ดังนั้นเวลาที่ได้จากการทดลองชุดข้อมูลที่ 3 ที่แสดงในตารางที่ 4.13, 4.14 และ 4.15 จะพบว่าอัลกอริทึมเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นใช้เวลาในการค้นหาฟรีเควนท์ k-ไอเทมเซตได้เร็วกว่า

อัลกอริทึมที่มีการเก็บทั้ง ฟรีเควนท์ k-ไอเทมเซตและไอเทมเซตที่ไม่เป็นฟรีเควนท์ k-ไอเทมเซต อย่างเช่นอัลกอริทึม บอร์เดอร์และพรีลาจก์อัลกอริทึมดังแสดงในรูปที่ 4.15– 4.23 และสามารถทำการค้นหาฟรีเควนท์ k-ไอเทมเซตทั้งหมดในฐานข้อมูลปรับปรุงได้อย่างถูกต้องครบถ้วน เช่นเดียวกับอัลกอริทึมอะพริโอริ เอฟยูพี บอร์เดอร์และพรีลาจก์

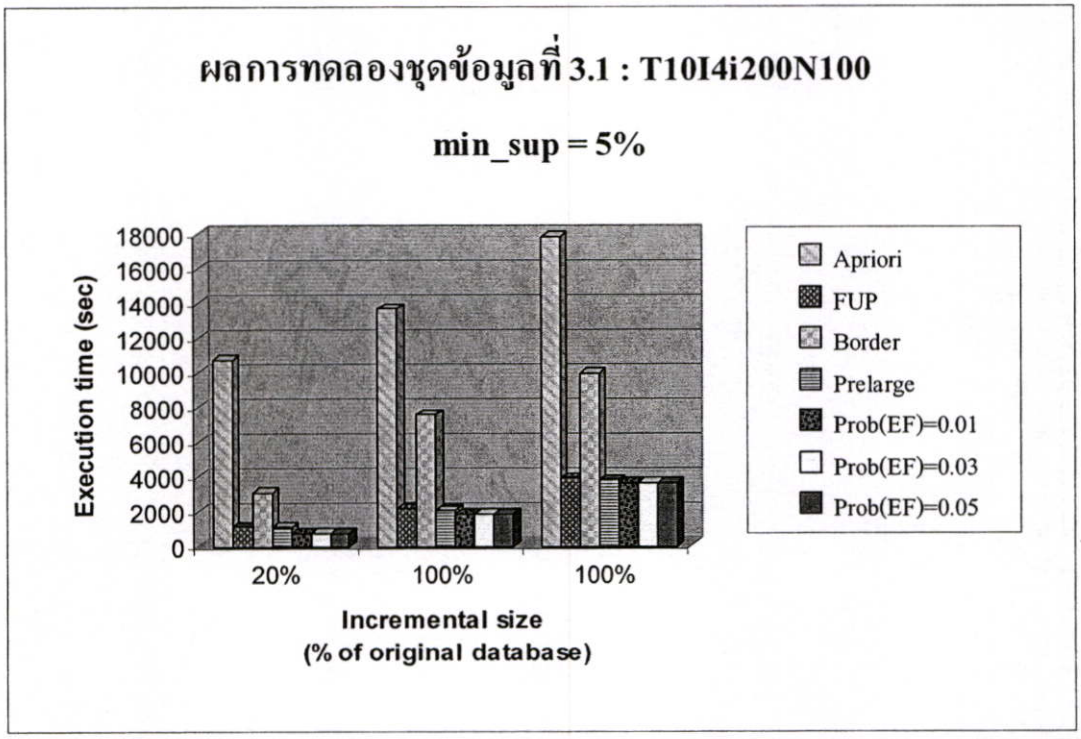
เมื่อนำเวลาที่ได้จากการทดลองชุดข้อมูลที่ 3 ซึ่งเป็นการทดลองเพื่อหาความถูกต้องในการค้นหาฟรีเควนท์ไอเทมเซตจากฐานข้อมูลปรับปรุงภายใต้สมมติฐานของฐานข้อมูลใหม่ที่มีความแตกต่างกันฐานข้อมูลเดิม มาวิเคราะห์เพื่อเปรียบเทียบเวลาในการทำงานกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์และพรีลาจก์ โดยการวิเคราะห์ความแปรปรวนแบบจำแนกสองทางโดยใช้วิธีน็อนพาราเมตริก แยกตามค่าสนับสนุนน้อยที่สุดที่ทดลอง คือ 1%, 3% และ 5% ผลการทดสอบพบว่าเวลาเฉลี่ยระหว่างอัลกอริทึมอะพริโอริ,เอฟยูพี, บอร์เดอร์ และพรีลาจก์แตกต่างกับเวลาเฉลี่ยของอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นที่ค่าสนับสนุนน้อยที่สุด 1% 3% และ 5% อย่างมีนัยสำคัญ .05 ดังแสดงรายละเอียดในภาคผนวก ข.



รูปที่ 4.15 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับชุดข้อมูลที่ 3.1 เมื่อเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 20%, 50% และ 100% ด้วยค่าสนับสนุนน้อยที่สุดคือ 1%



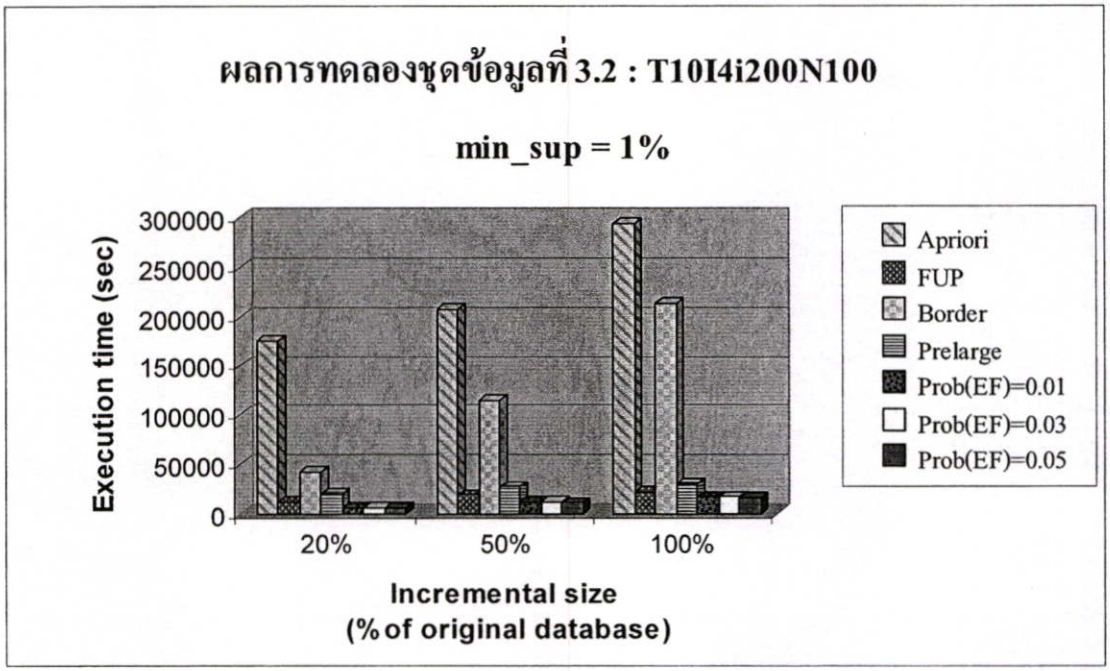
รูปที่ 4.16 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับชุดข้อมูลที่ 3.1เมื่อเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 20%, 50% และ 100% ด้วยค่าสนับสนุนน้อยที่สุดคือ 3%



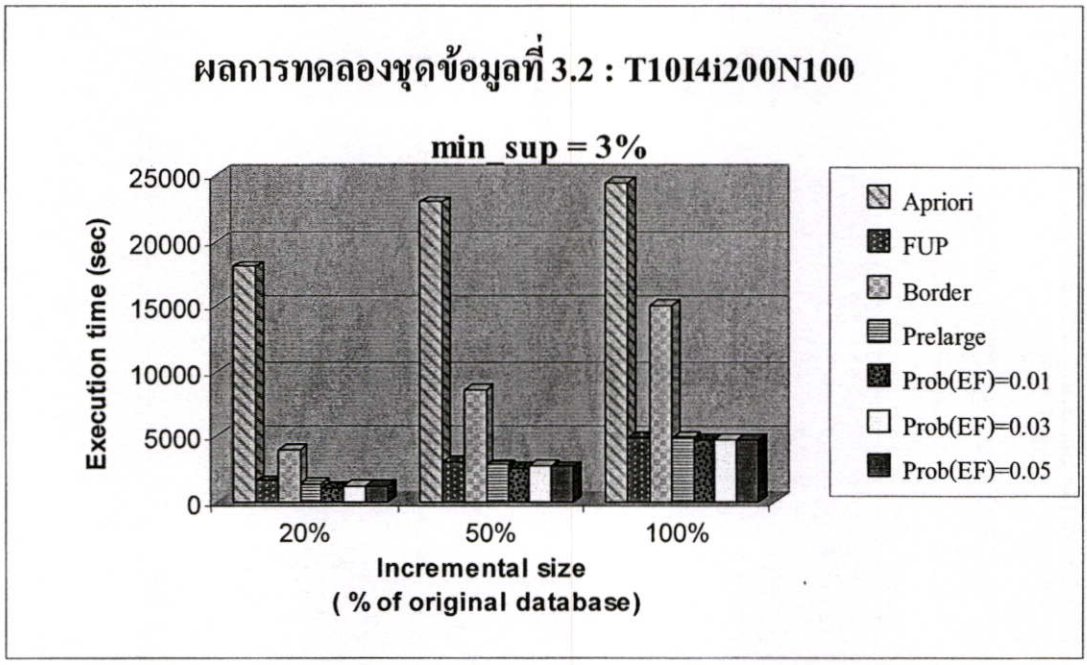
รูปที่ 4.17 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับชุดข้อมูลที่ 3.1เมื่อเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 20%, 50% และ 100% ด้วยค่าสนับสนุนน้อยที่สุดคือ 5%

ตารางที่ 4.10 แสดงเวลากระทำของอัลกอริทึมสำหรับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และ พรีลาจก์ที่ใช้ในการค้นหาการเพิ่มขยายกฎความสัมพันธ์เมื่อมีการเพิ่มข้อมูลใหม่ด้วยชุดข้อมูล 3.1

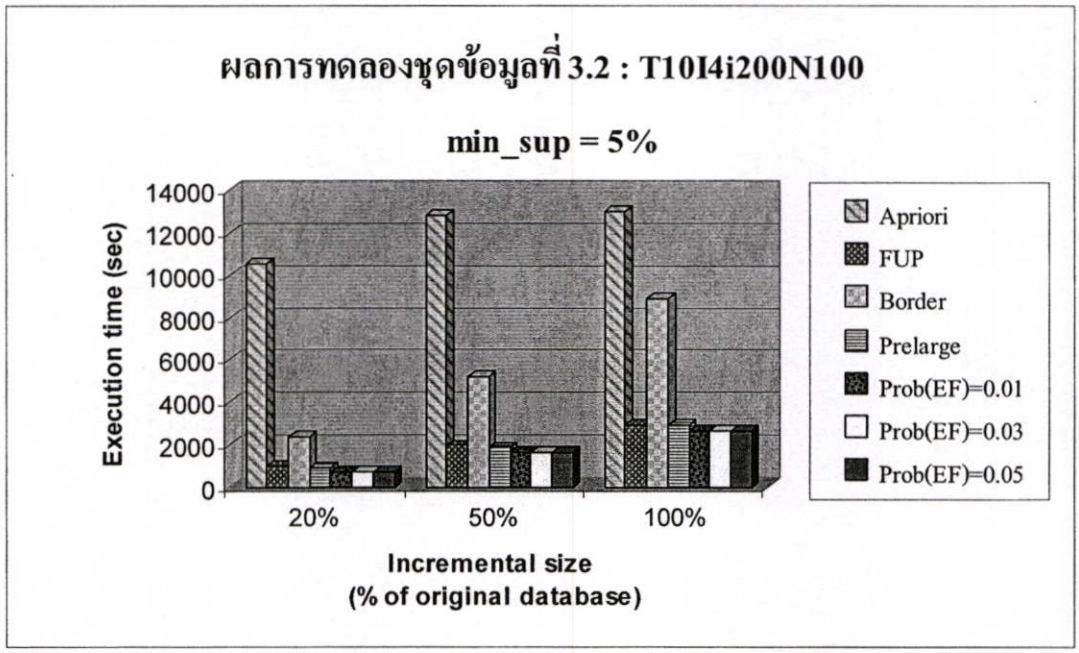
เวลาในการทำงาน (วินาที) ของชุดข้อมูลที่ 3.1 : T10I4L200N100								
ค่าสนับสนุนน้อยที่สุด (min_sup)	ขนาดข้อมูลใหม่ |db|	Algorithm						
		Apriori	FUP	Border	Prelarge	Probability-Based		
						$Prob_{EF}$		
						0.01	0.03	0.05
1%	20%	182,042	19,249	45,139	49,504	16,034	17,824	18,006
	50%	177,210	16,390	41,340	27,830	8,364	8,517	8,657
	100%	218,009	22,676	122,566	36,222	14,891	14,935	14,849
3%	20%	308,530	33,941	252,870	51,634	26,409	26,417	26,245
	50%	19,107	1,922	3,957	1,827	1,274	1,095	1,095
	100%	24,909	2,546	9,993	3,592	2,906	2,849	2,849
5%	20%	33,300	6,057	17,179	6,338	5,653	5,527	5,527
	50%	10,692	1,125	3,096	1,074	771	776	776
	100%	13,640	2,179	7,581	2,100	1,852	1,866	1,866



รูปที่ 4.18 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับชุดข้อมูลที่ 3.2 เมื่อเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 20%, 50% และ 100% ด้วยค่าสนับสนุนน้อยที่สุดคือ 1%



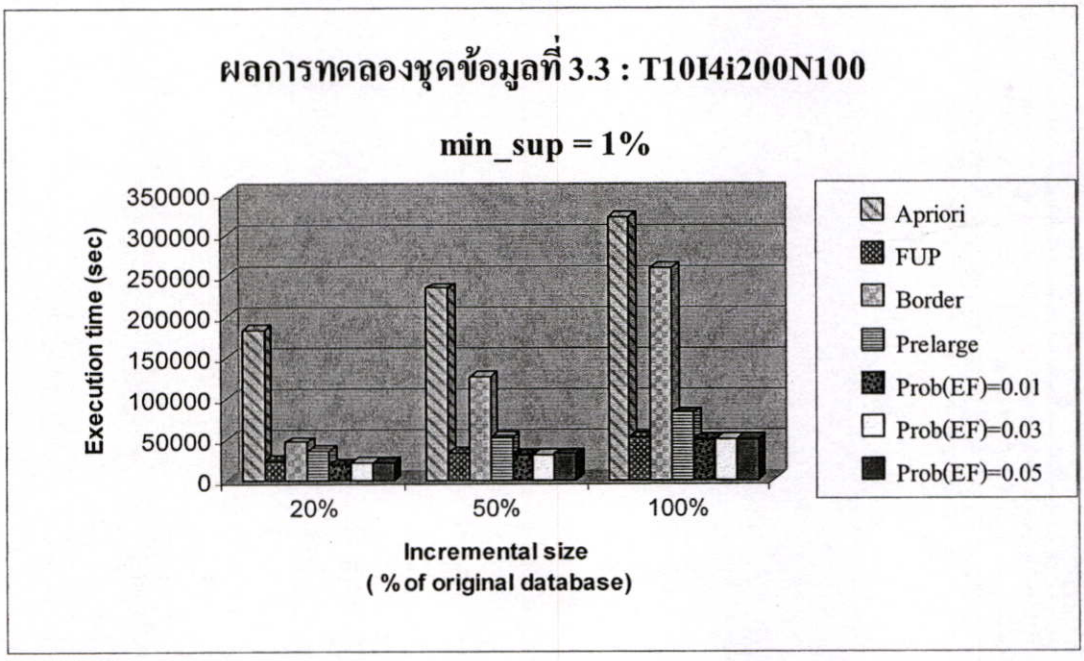
รูปที่ 4.19 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับชุดข้อมูลที่ 3.2 เมื่อเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 20%, 50% และ 100% ด้วยค่าสนับสนุนน้อยที่สุดคือ 3%



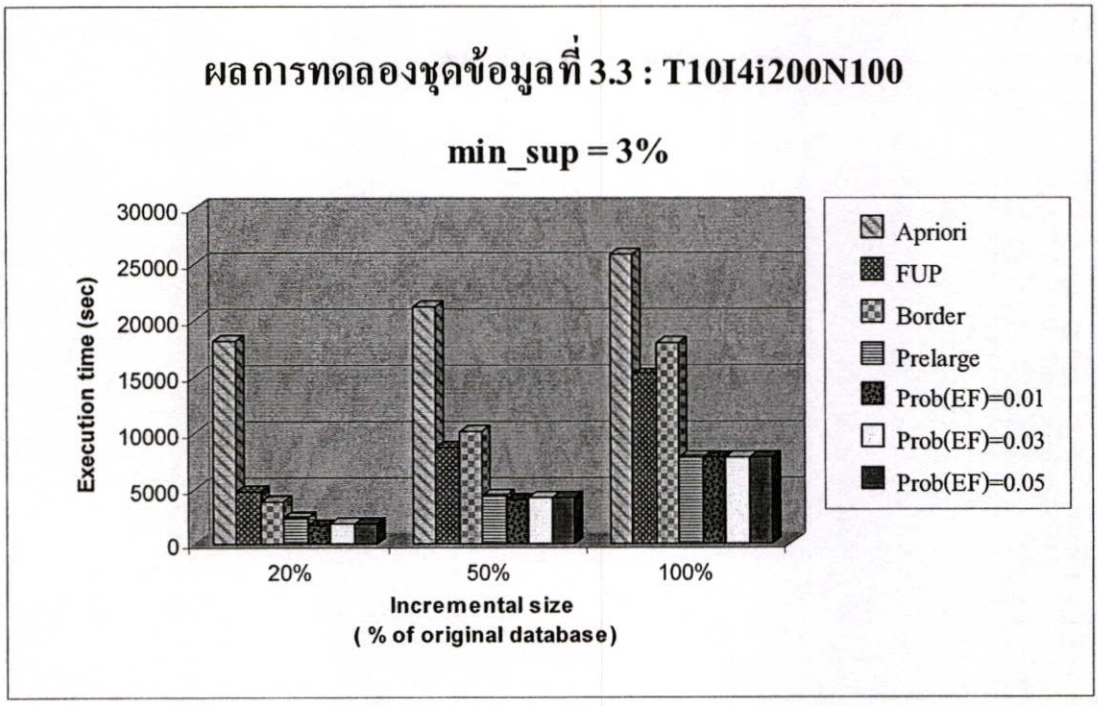
รูปที่ 4.20 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับชุดข้อมูลที่ 3.2เมื่อเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 20%, 50% และ 100% ด้วยค่าสนับสนุนน้อยที่สุดคือ 5%

ตารางที่ 4.11 แสดงเวลากระทำของอัลกอริทึมสำหรับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และ พรีลาจก์ที่ใช้ในการค้นหาการเพิ่มขยายกฎความสัมพันธ์เมื่อมีการเพิ่มข้อมูลใหม่ด้วยชุดข้อมูล 3.2

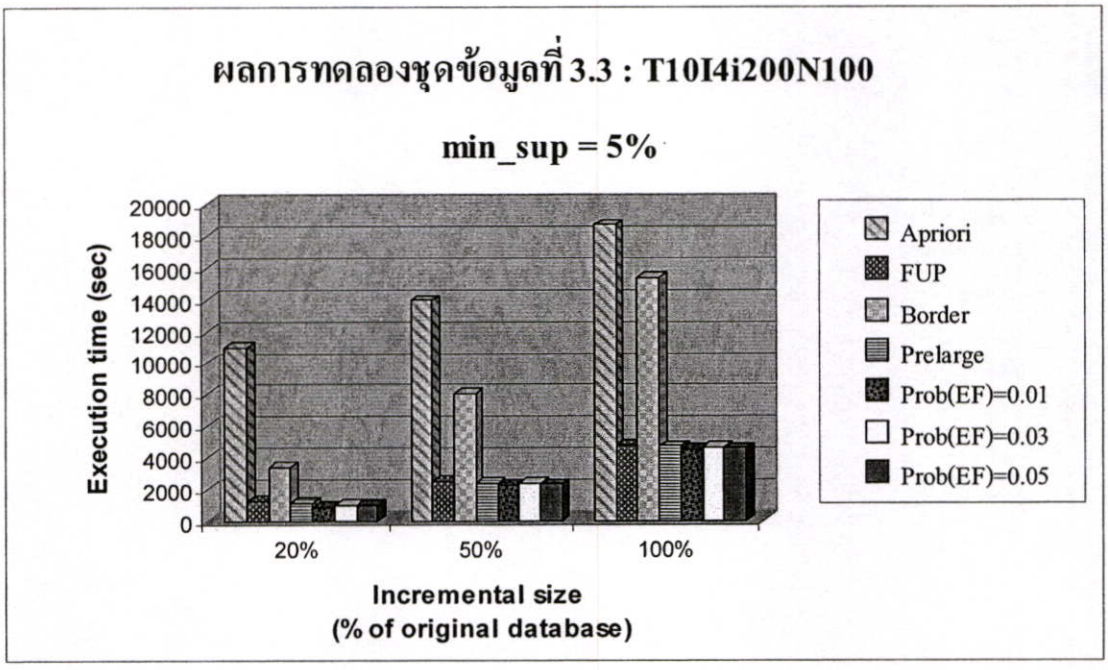
เวลาในการทำงาน (วินาที) ของชุดข้อมูลที่ 3.2 : T10I4L200N100								
ค่าสนับสนุนน้อยที่สุด (min_sup)	ขนาดข้อมูลใหม่ |db|	Algorithm						
		Apriori	FUP	Border	Prelarge	Probability-Based		
						$Prob_{EF}$		
						0.01	0.03	0.05
1%	20%	175,294	12,281	41,834	20,056	6,545	6,595	6,341
	50%	207,850	19,135	114,304	27,489	11,785	11,917	11,504
	100%	293,290	22,489	213,906	30,236	17,176	17,289	16,611
3%	20%	18,027	1,603	3,988	1,431	1,071	1,171	1,165
	50%	22,922	3,044	8,538	2,843	2,628	2,718	2,695
	100%	24,350	4,885	15,004	4,870	4,699	4,763	4,709
5%	20%	10,496	1,025	2,389	896	704	705	702
	50%	12,818	1,964	5,198	1,875	1,667	1,667	1,662
	100%	13,025	2,896	8,852	2,906	2,675	2,674	2,664



รูปที่ 4.21 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับชุดข้อมูลที่ 3.3 เมื่อเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 20%, 50% และ 100% ด้วยค่าสนับสนุนน้อยที่สุดคือ 1%



รูปที่ 4.22 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับชุดข้อมูลที่ 3.3 เมื่อเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 20%, 50% และ 100% ด้วยค่าสนับสนุนน้อยที่สุดคือ 3%



รูปที่ 4.23 แสดงผลการเปรียบเทียบเวลากระทำการสำหรับชุดข้อมูลที่ 3.3 เมื่อเพิ่มข้อมูลใหม่ด้วยขนาดข้อมูล 20%, 50% และ 100% ด้วยค่าสนับสนุนน้อยที่สุดคือ 5%

ตารางที่ 4.12 แสดงเวลากระทำของอัลกอริทึมสำหรับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และพรีลาจก์ที่ใช้ในการค้นหาการเพิ่มขยายกฎความสัมพันธ์เมื่อมีการเพิ่มข้อมูลใหม่ด้วยชุดข้อมูล 3.3

เวลาในการทำงาน (วินาที) ของชุดข้อมูลที่ 3.3 : T10I4L200N100								
ค่าสนับสนุนน้อยที่สุด (min_sup)	ขนาดข้อมูลใหม่ |db|	Algorithm						
		Apriori	FUP	Border	Prelarge	Probability-Based		
						Prob(EF)		
						0.01	0.03	0.05
1%	20%	183,748	24,173	46,498	34,727	18,175	20,919	20,853
	50%	234,722	32,409	125,878	52,771	30,392	31,878	32,019
	100%	321,362	53,249	258,738	82,237	49,303	50,304	49,460
3%	20%	18,054	4,669	3,854	2,292	1,606	1,771	1,771
	50%	21,159	8,598	10,009	4,270	3,934	4,094	4,126
	100%	25,904	15,179	17,958	7,719	7,692	7,780	7,689
5%	20%	10,983	1,224	3,398	1,143	914	986	994
	50%	13,893	2,564	8,075	2,416	2,285	2,366	2,340
	100%	18,664	4,796	15,368	4,714	4,573	4,631	4,579

4.4 สรุปผลการทดลอง

จากการทดลองข้อมูลชุดต่างๆ เพื่อทดสอบความถูกต้องและประสิทธิภาพในการเพิ่มขยายการค้นหากฎความสัมพันธ์ของอัลกอริทึม ในกรณีของการเพิ่มข้อมูลใหม่เข้าไปใน ฐานข้อมูลเดิมในการทำงานของอัลกอริทึมการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นเปรียบเทียบกับอัลกอริทึมอะพริโอริ เอฟยูพี บอร์เดอร์ พรีลาจก์พบว่าอัลกอริทึมการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นสามารถทำการค้นหาการเพิ่มขยายการค้นหากฎความสัมพันธ์ได้อย่างถูกต้องและมีประสิทธิภาพตามวัตถุประสงค์ในการทดลองดังแสดงในผลการทดลองของชุดข้อมูลที่ 1, 2 และ 3

ผลทดลองชุดข้อมูลที่ 1 แสดงให้เห็นว่าภายใต้สมมติฐานของการเกิดของไอเทมเซตที่ปรากฏในฐานข้อมูลเดิมและฐานข้อมูลใหม่มีค่าความน่าจะเป็นในการเกิดของไอเทมเซตไม่แตกต่างกันการทำงานของอัลกอริทึมเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นสามารถทำงานได้เร็วกว่าอัลกอริทึมอื่นๆ ภายใต้การทดลองข้อมูลที่มีค่าความน่าจะเป็นในการเกิดของไอเทมเซตในฐานข้อมูลเดิมและฐานข้อมูลใหม่ไม่แตกต่างกันด้วยจำนวนไอเทมที่คาดว่าจะ

กลายเป็นฟรีเควนท์ไอเทมเซตที่เก็บไว้จำนวนน้อยกว่าบอร์เดอร์ไอเทมเซตและพรีลาจก์ไอเทมเซต ของอัลกอริทึมบอร์เดอร์และอัลกอริทึมพรีลาจก์ตามลำดับและจากผลการทดสอบการวิเคราะห์ความแปรปรวนแบบสองทางระหว่างขนาดของข้อมูลที่เพิ่มและอัลกอริทึมแยกตามค่าสนับสนุนน้อยที่สุดที่ 1%,3% และ 5% พบว่าอัลกอริทึมมีอิทธิพลต่อเวลาในการทำงานในการเพิ่มขยายการค้นหากฎความสัมพันธ์ โดยขนาดข้อมูลจะมีอิทธิพลกับการทดลองที่ค่าสนับสนุนน้อยที่สุดที่ 3% และ 5% สำหรับชุดข้อมูลที่ 2 ที่ผ่านการทดลองและทดสอบด้วยค่าทางสถิติ t-test แสดงให้เห็นว่าค่าเฉลี่ยของค่าที่ได้จากการคำนวณหาค่าความน่าจะเป็นของไอเทมที่คาดว่าจะเป็นฟรีดเวนท์ไอเทมเซตโดยใช้ทฤษฎีเบอร์นูลลีกับการที่ไอเทมเหล่านั้นกลายเป็นฟรีเควนท์จริงจากการทดลองเพิ่มข้อมูลจำนวน 100 ชุด เข้าไปในฐานข้อมูลไม่แตกต่างกัน และจากชุดข้อมูลที่ 3 สามารถยืนยันได้ว่าอัลกอริทึมในการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นสามารถค้นหาฟรีเควนท์ k- ไอเทมเซตได้ครอบคลุมฐานข้อมูลที่มีค่าความน่าจะเป็นในการเกิดของไอเทมเซตแตกต่างกันได้อย่างถูกต้องและมีประสิทธิภาพ

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

การค้นหากฎความสัมพันธ์เป็นเทคนิคหนึ่งของการทำดาต้าไมน์นิ่งที่สามารถนำไปประยุกต์ใช้ได้จริงกับงานต่างๆ เช่น นำไปใช้ในการพิจารณาการจัดวางสินค้าในร้าน การวางแผนกลยุทธ์ทางการตลาด การวิเคราะห์การตัดสินใจ และการจัดการทางธุรกิจ เป็นต้น หลักการทำงานของการค้นหากฎความสัมพันธ์คือการค้นหาความสัมพันธ์ที่มีอยู่ของข้อมูลจากฐานข้อมูลขนาดใหญ่เพื่อนำไปใช้ในการวิเคราะห์หรือทำนายปรากฏการณ์ต่าง แต่ที่เป็นที่นิยมคือการนำไปใช้ในการค้นหาและวิเคราะห์พฤติกรรมการซื้อสินค้าของลูกค้าว่ามักซื้อสินค้าใดร่วมกัน ผลที่ได้จากการวิเคราะห์จะถูกนำเสนอในรูปของกฎความสัมพันธ์ “ถ้า ... แล้ว...” (IF...THEN ...) เพื่อแสดงความสัมพันธ์ระหว่างแอททริบิวต์ที่แตกต่างกัน โดยจะมีการอ้างถึงเซตของไอเทมที่เรียกว่าไอเทมเซต และจากข้อมูลทรานแซคชันในฐานข้อมูลขนาดใหญ่จะสามารถทราบถึงความสัมพันธ์ที่เกิดขึ้นระหว่างรายการข้อมูลได้ด้วยการนับจำนวนของไอเทมเซตที่ปรากฏในแต่ละ ทรานแซคชัน ถ้าไอเทมเซตปรากฏในทรานแซคชันมากกว่าหรือเท่ากับค่าสนับสนุนน้อยที่สุด จะเรียกว่าฟรีเควนท์ไอเทมเซต (L_k)

อัลกอริทึมอะพริโอริเป็นอัลกอริทึมที่นำเสนอวิธีในการค้นหากฎความสัมพันธ์ที่ได้รับความนิยม โดยจะทำการค้นหาข้อมูลที่เป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูล ด้วยการสแกนข้อมูลแต่ละทรานแซคชันในฐานข้อมูลผ่านการวนรอบซ้ำหลายครั้งและในการวนรอบซ้ำแต่ละครั้งจะค้นหาจำนวนสมาชิกของฟรีเควนท์ไอเทมเซตเพิ่มขึ้นทีละ 1 ระดับ คือมีจำการค้นหาจำนวนไอเทมเซตเพิ่มทีละ 1 ตัวการลดจำนวนของทรานแซคชันที่จะถูกสแกนในรอบถัดไป ด้วยแนวคิดที่ว่า “ถ้าสับเซต (k-1) ใด ๆ ของแคนดิเดทไอเทมเซต ไม่ได้เป็นสมาชิก L_{k-1} ดังนั้น แคนดิเดทไอเทมเซตนั้นๆ ไม่สามารถเป็น ฟรีเควนท์ไอเทมเซตในระดับต่อไปได้ ดังนั้นจะลบแคนดิเดทไอเทมเซตนั้นๆ ออกไป”

โดยทั่วไปแล้วฐานข้อมูลที่นำมาใช้ค้นหากฎความสัมพันธ์มักมีการเปลี่ยนแปลงอยู่เสมอ ซึ่งการเปลี่ยนแปลงนี้จะมีผลต่อกฎความสัมพันธ์ที่ได้ทำการค้นหาไว้แล้ว เนื่องจากอาจพบว่าฟรีเควนท์ไอเทมเซตเดิมที่มีอยู่ไม่ใช่สิ่งที่น่าสนใจอีกต่อไป ในขณะเดียวกันอาจทำให้พบฟรีเควนท์ไอเทมเซตใหม่ที่เกิดจากการเปลี่ยนแปลงนี้ การทำการค้นหากฎความสัมพันธ์ในฐานข้อมูลที่ได้รับการปรับปรุงแล้วจะทำให้ได้กฎความสัมพันธ์ใหม่ที่น่าสนใจเกิดขึ้น เรียกการค้นหากฎความสัมพันธ์ใหม่ที่เกิดขึ้นนี้ว่าการเพิ่มขยายการค้นหากฎความสัมพันธ์

เมื่อฐานข้อมูลมีการเปลี่ยนแปลงการค้นหากฎความสัมพันธ์ของอัลกอริทึมอะพริโอริจะทำการค้นหากฎความสัมพันธ์ของฐานข้อมูลทั้งหมดที่ได้รับการปรับปรุงโดยไม่นำความรู้ที่ได้จากการไมน์นิ่งก่อนหน้ามาใช้ ดังนั้นในหลายๆ งานวิจัยจึงได้นำหลักการของอะพริโอริมาพัฒนาสำหรับค้นหาการเพิ่มขยายกฎความสัมพันธ์ โดยสามารถสรุปได้ 2 ลักษณะคือ

1.งานวิจัยที่มีการนำฟรีเควนท์ไอเทมเซตที่ได้จากการไมน์นิ่งก่อนหน้ามาช่วยลดการสแกนฐานข้อมูล ได้แก่

1.1 อัลกอริทึมเอฟยูพี (FUP) ที่ได้นำฟรีเควนท์ไอเทมเซตจากฐานข้อมูลเดิมมาปรับปรุงค่าสนับสนุน และเมื่อพบฟรีเควนท์ไอเทมเซตในส่วนของฐานข้อมูลที่เพิ่มเข้ามาจึงจะนำไปสแกนในฐานข้อมูลเดิม มีงานวิจัยต่างๆ ได้นำเสนอเทคนิคในการเพิ่มขยายการค้นหากฎความสัมพันธ์ ซึ่งช่วยลดการสแกนฐานข้อมูลเดิมซึ่งมีขนาดใหญ่ได้

2.งานวิจัยที่นำฟรีเควนท์ไอเทมเซตและไอเทมเซตที่ไม่เป็นฟรีเควนท์ไอเทมเซตจากการไมน์นิ่งฐานข้อมูลก่อนหน้ามาใช้ในการลดการสแกนฐานข้อมูลเดิม ได้แก่อัลกอริทึมต่อไปนี้

2.1 อัลกอริทึมบอร์เดอร์ เป็นอัลกอริทึมที่พัฒนาจากอัลกอริทึมเอฟยูพีเพื่อลดการสแกนฐานข้อมูลเดิมด้วยการค้นหาทั้งส่วนที่เป็นฟรีเควนท์ไอเทมเซตและส่วนของแคนดิเดทไอเทมเซตที่ไม่ได้เป็นฟรีเควนท์ไอเทมเซตที่เรียกว่า บอร์เดอร์หรือเนกาทีฟบอร์เดอร์ ในกรณีที่พบว่าบอร์เดอร์ไอเทมเซตที่พบในฐานข้อมูลเดิมกลายมาเป็นฟรีเควนท์ไอเทมเซตในฐานข้อมูลปรับปรุงเท่านั้น ซึ่งมีผลทำให้ต้องสแกนฐานข้อมูลทั้งหมด (ฐานข้อมูลเดิมรวมกับฐานข้อมูลใหม่) เพื่อค้นหาฟรีเควนท์ไอเทมเซตที่เกิดขึ้นในฐานข้อมูลปรับปรุง

2.2 อัลกอริทึมพรีลาจก์ เป็นอัลกอริทึมที่นำเสนอค่าการเก็บฟรีเควนท์ไอเทมเซตและไอเทมเซตที่ไม่เป็นฟรีเควนท์ไอเทมเซตจากการไมน์นิ่งฐานข้อมูลเดิมโดยมีการกำหนดค่าที่ใช้ในการพิจารณาฟรีเควนท์ไอเทมเซตด้วยค่าสนับสนุนระดับสูงซึ่งก็คือค่าเดียวกับค่าสนับสนุนน้อยที่สุดและอีกค่าที่นำมาใช้พิจารณาไอเทมที่ไม่เป็นฟรีเควนท์ไอเทมเซตด้วยค่าค่าสนับสนุนขั้นต่ำที่มีค่าน้อยกว่าค่าสนับสนุนน้อยที่สุด และด้วยวิธีนี้ทำให้อัลกอริทึมพรีลาจก์เก็บไอเทมที่ไม่เป็นฟรีเควนท์ไอเทมเซตน้อยกว่าบอร์เดอร์ที่เก็บส่วนที่เป็นแคนดิเดททีที่ไม่เป็นฟรีเควนท์ไอเทมเซต นอกจากนี้พรีลาจก์ได้นำเสนอการลดการสแกนฐานข้อมูลด้วยการคำนวนค่าสัดส่วนของฐานข้อมูลที่เพิ่มเข้ามาถ้าไม่เกินขนาดที่ได้จากการคำนวณขนาดของฐานข้อมูลที่ปรับปรุงทั้งหมด (ขนาดของฐานข้อมูลเดิมรวมกับฐานข้อมูลใหม่ที่เพิ่มเข้ามา) เทียบกับค่าค่าสนับสนุนขั้นต่ำและค่าสนับสนุนขั้นแล้วไอเทมเซตนั้นไม่สามารถเป็นฟรีเควนท์ไอเทมเซตได้ ทำให้ลดการสแกนฐานข้อมูลเดิมได้

2.3 อัลกอริทึมในการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นเป็นงานวิจัยที่มีการนำเสนอแนวคิดในการลดจำนวนครั้งในการสแกนฐานข้อมูลเดิม โดยนำความรู้ที่ได้จากการไมน์นิ่งในฐานข้อมูลเดิมทั้งส่วนที่เป็นฟรีเควนท์ไอเทมเซตและในส่วนที่คาดว่าจะกลายมาเป็นฟรีเควนท์ไอเทมเซตได้เมื่อมีการเพิ่มข้อมูลใหม่เข้ามาจำนวนหนึ่งภายใต้สมมติฐานที่ว่า

ไอเทมเซตในฐานข้อมูลกับส่วนของฐานข้อมูลที่เพิ่มเข้ามาไม่แตกต่างกัน ทำให้สามารถนำค่าความน่าจะเป็นในการเกิดของไอเทมเซตที่ไม่เป็นฟรีเควนท์ในฐานข้อมูลเดิมมาใช้ในการทำนายความน่าจะเป็นฟรีเควนท์ไอเทมเซตเมื่อมีข้อมูลจำนวนหนึ่งเพิ่มเข้าโดยใช้ทฤษฎีเบอร์นูลลีเรียกไอเทมเซตเหล่านี้ว่าไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์

ด้วยแนวคิดของเบอร์นูลลีที่กล่าวถึงกฎว่าด้วยจำนวนมาก (Law of large number) ว่า "ค่าเฉลี่ยตัวแปรสุ่มของตัวอย่างประชากรจำนวนมากจะมีค่าเข้าใกล้ค่าเฉลี่ยของประชากรทั้งหมด" สำหรับในงานวิจัยนี้ได้นำแนวคิดนี้มาประยุกต์ใช้ในการประมาณค่าของไอเทมเซตที่คาดว่าจะกลายเป็นฟรีเควนท์โดยนำค่าสถิติการเกิดของไอเทมเซตต่างๆ ที่ปรากฏในฐานข้อมูลเดิมซึ่งมีขนาดของข้อมูลจำนวนมากมาใช้ในการทำนายค่าความน่าจะเป็นของไอเทมเซตที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตเมื่อมีข้อมูลใหม่จำนวนหนึ่งเพิ่มเข้ามาซึ่งหลักการนี้ ทำให้มีการเก็บเฉพาะไอเทมที่คาดว่าจะเป็นฟรีเควนท์ไอเทมเซตที่คำนวณแล้วว่ามีค่าความน่าจะเป็นที่จะกลายเป็นฟรีเควนท์ไอเทมเซตเมื่อมีข้อมูลจำนวนหนึ่งเพิ่มเข้ามาเท่านั้นทำให้สามารถลดจำนวนไอเทมเซตที่ต้องเก็บและลดจำนวนไอเทมเซตที่ต้องปรับปรุงในฐานข้อมูลใหม่ลงไป นอกจากนี้ยังสามารถลดการสแกนฐานข้อมูลเดิมได้อีกด้วย ในกรณีที่มีไอเทมเซตใหม่เกิดขึ้นแตกต่างจากที่ปรากฏในฐานข้อมูลเดิม อัลกอริทึมการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นยังสามารถใช้ลดจำนวนไอเทมเซตที่จะนำไปสแกนฐานข้อมูลเดิมได้นำค่าคาดหวังน้อยที่สุดที่ได้จากการคำนวณในฐานข้อมูลเดิมมาประมาณค่าสนับสนุนของไอเทมเซตใหม่ที่คาดว่าจะเกิดขึ้นในฐานข้อมูลเดิมรวมกับค่าสนับสนุนที่เกิดขึ้นจริงจากฐานข้อมูลใหม่ แล้วพิจารณาว่าถ้าไม่สามารถเป็นฟรีเควนท์ไอเทมเซตได้จะไม่นำไอเทมเซตนั้นไปสแกนในฐานข้อมูลเดิม ทำให้ลดจำนวนไอเทมเซตที่จะสแกนฐานข้อมูลเดิมที่มีขนาดใหญ่มาก

จากการทดลองการค้นหาฟรีเควนท์ k ไอเทมเซตโดยมีวัตถุประสงค์ในการทดลองเพื่อวัดความถูกต้องและประสิทธิภาพการทำงานของอัลกอริทึมในการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นด้วยสมมติฐานการทดลองด้วยชุดข้อมูลต่างๆ ได้แก่

- สมมติฐานในการทดลองที่ว่าการเกิดของไอเทมเซตที่ปรากฏในฐานข้อมูลเดิมและฐานข้อมูลใหม่มีค่าความน่าจะเป็นในการเกิดของไอเทมเซตไม่แตกต่างกันจากการทดลองชุดข้อมูลที่ 1 ด้วยสมมติฐานนี้พบว่าอัลกอริทึมในการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นสามารถปรับปรุงและค้นหาฟรีเควนท์ k ไอเทมเซตใหม่ได้อย่างถูกต้องและมีประสิทธิภาพ

-สมมติฐานของค่าความน่าจะเป็นที่ได้จากการนำทฤษฎีเบอร์นูลลีมาประยุกต์ใช้ในการคำนวณหาไอเทมที่คาดว่าจะกลายเป็นฟรีเควนท์ไอเทมเซตของอัลกอริทึมในการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้ค่าความน่าจะเป็น สามารถนำมาใช้ในการทำนายการเกิดฟรีเควนท์ไอเทมเซตเมื่อมีข้อมูลจำนวนหนึ่งเพิ่มเข้ามาในฐานข้อมูลเดิมได้อย่างถูกต้อง ดังแสดงในการ

ทดลองด้วยชุดข้อมูลที่ 2 ด้วยการเพิ่มข้อมูลขนาด 1000 ทรานแซคชัน จำนวน 100 ชุดเข้าไป และผลที่ได้จากการทดลองนอกจากพบว่าอัลกอริทึมสามารถทำการค้นหาฟรีเควนท์ k ไอเทมเซตได้อย่างถูกต้องและมีประสิทธิภาพแล้วยังพบว่าค่าเฉลี่ยของความแตกต่างระหว่างค่าที่ได้จากการคำนวณและจำนวนการเกิดขึ้นจริงของไอเทมที่คาดว่าจะเป็นฟรีเควนท์ k ไอเทมเซตได้กลายมาเป็นฟรีเควนท์ k-ไอเทมเซตไม่แตกต่างกันจากผลการทดสอบด้วยค่าสถิติ t-test

- สมมติฐานที่ใช้ในการค้นหาความถูกต้องของฟรีเควนท์ k- ไอเทมเซตเมื่อความน่าจะเป็นในการเกิดฟรีเควนท์ k-ไอเทมเซตของฐานข้อมูลใหม่แตกต่างจากฐานข้อมูลเดิมจากการทดลองเพิ่มข้อมูลใหม่ที่มีค่าความน่าจะเป็นในการเกิดฟรีเควนท์ไอเทมเซตต่างจากฐานข้อมูลเดิมได้แสดงให้เห็นด้วยผลทดลองชุดข้อมูลที่ 3 ว่าอัลกอริทึมในการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นสามารถค้นหาฟรีเควนท์ k ไอเทมเซตได้ครอบคลุมฐานข้อมูลปรับปรุงได้อย่างถูกต้องและมีประสิทธิภาพเมื่อเปรียบเทียบกับอัลกอริทึมอะพริโอริ เอฟยูพี บอร์เดอร์ และพรีลาจก์

5.2 ข้อเสนอแนะ

1. แนวคิดของอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นไปนั้นมีพื้นฐานการทำงานมาจากอัลกอริทึมอะพริโอริที่มีการวนรอบซ้ำเพื่อหาฟรีเควนท์ k-ไอเทมเซต ด้วยการนำฟรีเควนท์ (k-1)-ไอเทมเซตและไอเทมที่คาดว่าจะกลายเป็นฟรีเควนท์(k-1) ไอเทมเซตมาใช้ในการสร้างแคนดิเดท k ไอเทมเซตที่ครอบคลุมทุกไอเทมเซต ขั้นตอนในการสร้างและสแกนแคนดิเดท k ไอเทมเซตเพื่อค้นหาฟรีเควนท์ k ไอเทมเซตจึงต้องใช้เวลาในการสแกนฐานข้อมูลเพื่อหาค่าสนับสนุนสำหรับแคนดิเดท k-ไอเทมเซต ถึงแม้ว่าอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นไปจะสามารถทำการค้นหา ฟรีเควนท์ k ไอเทมเซตในฐานข้อมูลปรับปรุงใหม่ภายใต้สถิติของฐานข้อมูลเดิมและฐานข้อมูลใหม่ที่ไม่แตกต่างกันได้อย่างถูกต้องและรวดเร็ว แต่ถ้ามีการนำเทคนิคต่างๆ มาช่วยในการลดการสแกนฐานข้อมูลจะทำให้ประสิทธิภาพในการทำงานของอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นไปเพิ่มมากขึ้นไปอีก เช่น การนำเทคนิคของแฮชซึ่งมีจุดเด่นในด้านการลดจำนวนแคนดิเดท 2-ไอเทมเซตมาช่วยจะทำให้สามารถลดจำนวนการสแกนฐานข้อมูลสำหรับแคนดิเดท 2 ไอเทมเซตเพื่อค้นหาฟรีเควนท์ 2 -ไอเทมเซตน้อยลง เป็นต้น

2. การประยุกต์ใช้แนวคิดของอัลกอริทึมในการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นไปกับการศึกษาพฤติกรรมของผู้ใช้ที่ใช้บริการของระบบการประมวลผลธุรกรรมแบบออนไลน์ ซึ่งระบบการประมวลผลธุรกรรมแบบออนไลน์ได้เข้ามามีบทบาทอย่างมากในการใช้ชีวิตในยุคปัจจุบันที่มีการนำระบบอินเทอร์เน็ตมาใช้ในการดำเนินธุรกิจที่มีเครือข่ายติดต่อไปได้ทั่วโลก ทำให้มีผู้สนใจบริการซื้อ-ขายสินค้าผ่านระบบประมวลธุรกรรมออนไลน์ โดยสามารถทำธุรกรรมต่างๆ

เช่น การโอนเงิน การจองตั๋วเครื่องบิน การจองที่พักในโรงแรมผ่านระบบอินเทอร์เน็ต ถ้าได้ทำการศึกษาข้อมูลบริการของธุรกรรมออนไลน์ที่เกิดขึ้นอาจพบว่ามีธุรกิจใหม่ๆที่น่าสนใจรอที่จะเกิดขึ้นอย่างมากมายจากรูปแบบของการทำธุรกรรมออนไลน์นี้ การค้นหาความสัมพันธ์ระหว่างข้อมูลธุรกรรมด้วยอัลกอริทึมในการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นไปประยุกต์ใช้กับข้อมูลด้านการประมวลผลธุรกรรมออนไลน์นี้จะทำให้สามารถค้นหารูปแบบของข้อมูลหรือแนวโน้มของข้อมูลที่มีความน่าสนใจได้ โดยสามารถแสดงผลลัพธ์ที่ได้จากการไมน์นิ่งซึ่งการคือความรู้ที่ซ่อนอยู่ในการทำธุรกรรมเหล่านั้นในรูปของกฎความสัมพันธ์ระหว่างข้อมูล ตัวอย่างระบบการประมวลผลธุรกรรมออนไลน์ได้แก่ การซื้อ-ขายสินค้าผ่านระบบพาณิชย์อิเล็กทรอนิกส์, ระบบธนาคารอิเล็กทรอนิกส์, ระบบการศึกษาทางไกลผ่านสื่ออิเล็กทรอนิกส์ เป็นต้น

เอกสารอ้างอิง

[1] Liao S. "Knowledge management technologies and applications—literature review from 1995 to 2002."**Expert system with Applications,** vol.25, Aug, 2003.

[2] Agrawal R., Imielinski T., and Swami A. "Mining association rules between sets of items in large databases." **In Proceeding of the ACM SIGMOD Int'l Conf. on Management of Data (ACM SIGMOD '93)**, Washington, USA,May 1993, pp. 207-216.

[3] Agrawal R. and Srikant R. "Fast algorithms for mining association rules." **In Proceedings of 20th Intl Conf. on Very Large Databases (VLDB'94)**, Santiago, Chile, September 1994, pp. 478 -499.

[4] Toivonen H. "Sampling Large Databases for Association Rules." **Proceeding of the 22th International conference on Very Large Data Bases**, September 1996,pp. 134-145.

[5] Agarwal R. C., Aggarwal C. C., and Prasad V. V. V. **"**A Tree Projection Algorithm for Generation of Frequent Itemsets**" Journal of Parallel and Distributed Computing (Special Issue on High Performance Data Mining)**, 2001. pp. 350–371.

[6] Cheung W. and Zaiane O. R. "Incremental Mining of Frequent Patterns without Candidate Generation or Support Constraint." **Proceedings of the 7th International Database Engineering and Application Symposium**, July 2003. pp. 111–116.

[7] Tsai P. S.M., Lee C.C., and Chen A. L.P,. "An efficient approach for incremental association rule mining." **Methodologies for Knowledge Discovery and Data Mining: Third Pacific-Asia Conference**, PAKDD-99, Beijing, China, April, 1999.

[8] Zhang S. ,Zhang C. and Yan X. "Post-mining: maintenance of association rules by weighting." **Information system 28**, 2004. pp. 691-707

[9] Dudek D. and Zgrzywa A. "The Incremental Method for Discovery of Association Rules." **Springer Berlin**, Heidelberg,2005.

[10] Teng W.-G. and Chen M.-S. "Incremental Mining on Association Rules." **Springer-Verlag Berlin Heidelberg,** 2005. pp. 125-162

[11] Lee C. H., Lin C. R., and Chen M. S. "Sliding-Window Filtering: An Efficient Algorithm for Incremental Mining." **Proceeding of the ACM 10th International Conference on Information and Knowledge Management** , November 2001.

[12] Chang C.-H. and Yang S.-H. "Enhancing SWF for Incremental Association Mining by Itemset Maintenance." **Proceedings of the 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining**, April 2003.

[13] Ezeife C. I. and Su Y. "Mining Incremental Association Rules with Generalized FP-Tree." **Proceedings of the 15th Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Artificial Intelligence**, May 2002. pp. 147–160.

[14] Cheung D., Han J., Ng, V. and Wong, C. Y. "Maintenance of discovered association rules in large databases: An incremental updating technique", **In 12th IEEE International Conference on Data Engineering**, February 1996, pp. 106-114.

[15] Cheung D., Lee S.D., Kao B., "A General incremental technique for maintaining discovered association rules" , **In Proceedings of the 5 th Intl. Conf. on Database Systems for Advanced Applications (DASFAA'97)**, Melbourne, Australia, April 1997, pp. 185-194.

[16] Ayan N. F., Tansel A.U., and Arun E. "An efficient algorithm to update large itemsets with early pruning." **Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, San Diego, August 1999, pp. 287-291.

[17] Omiecinski E. and Savasere A. "Efficient mining of association rules in large dynamic databases." **Proceedings of the 16th British National Conference on Databases: Advances in Databases**, 1998. pp. 49 – 63.

[18] Thomas S., Bodagala S., Alsabti, K. and Ranka S. "An efficient algorithm for the incremental updation of association rules in large databases" , **In Proceedings of the 3rd Intl. Conf. on Knowledge Discovery and Data Mining (KDD'97)**, New Port Beach, California, August 1997, pp. 263-266.

[19] Feldman R., Aumann Y., and Lipshtat O. "Borders: An efficient algorithm for association generation in dynamic databases" **Journal,Intelligent Information System**, 1999, pp. 61-73.

[20] Adnan M., Alhajj R., and Barker K. "Performance Analysis of Incremental Update of Association Rules Approaches." **In Proceeding of 9th IEEE International Conference on Intelligent Engineering System (INES '05) , IEEE,** 2005 ,16-19 Sept, 2005. pp. 129 – 134.

[21] Hong T.P., Wang C.Y. and Tao Y.H., "A new incremental data mining algorithm using pre-large itemsets", **Journal, Intelligent Data Analysis**, Vol. 5, No.2, pp. 111-129, 2001.

[22] Amornchewin R. and Kreesuradej W., "Incremental association rule mining using promising frequent itemset algorithm", **In Proceeding 6th International Conference on Information, Communications and Signal Processing**, Dec. 10-13 2007, pp.1-5.

[23] Amornchewin R. and Kreesuradej W. "Probability-Based Incremental Association Rule Discovery Algorithm" **International Symposium on Computer Science and Its Applications 2008,** 13-15 October 2008, pp.212-215.

[24] Amornchewin R. and Kreesuradej W. "Mining Dynamic Databases using Probability-Based Incremental Association Rule Discovery Algorithm." **Journal of Universal Computer Science**, Vol. 15, no.12, 2009. pp. 2409-2428.

[25] Hipp J., G¨untzer U. and Nakhaeizadeh G. "Algorithms for Association Rule Mining – A General Survey and Comparison." **ACM SIGKDD**, July 2000, pp.58-64.

[26] Chang C.C., Li Y.C. and Lee J.S. "An efficient algorithm for incremental mining of association rules." **Proceedings of the 15th international workshop on research issues in data engineering: stream data mining and applications (RIDE-SDMA'05), IEEE**, 2005.

[27] Veloso A. A. , Meira W. Jr, Carvalho de M.B., Pôssas B., Parthasarathy S. and Javeed Zaki M. "Mining frequent itemsets in evolving databases." **In Proc. 2nd SIAM Intl. Conf. on Data Mining**, Arlington, VA, April 2002.

[28] Sarda N. L. and Srinivas N. V., "An adaptive algorithm for incremental. mining of association rules" **In Proc. 9th Intl. Workshop on Database and Expert System Applications** , Vienna, Austria, Aug 1998, pp. 240-245.

[29] Adnan M., Alhajj R. and Barker K. "Performance Analysis of Incremental Update of Association Rules Mining Approaches." **In Proceeding of 9th IEEE International Conference on Intelligent Engineering System 2005**, Sept. 16-19, 2005, pp.129-134.

[30] Zhang C., Zhang S. and Webb G. I., "Identifying Approximate Itemsets of Interest in Large Databases" **Applied Intelligence** 18, 91–104, 2003.

[31] Su J. and Lin, W. "CBW: An Efficient Algorithm for Frequent Itemset Mining." **Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04), IEEE Computer Society**, vol. 3, January, 2004.

[32] Wang J., and Han J., "TFP: An Efficient Algorithm for Mining Top-K Frequent Closed Itersets." **Knowledge and Data Engineering, IEEE Transactions on Knowledge and Data Engineering**, Vol. 17, May, 2005. pp. 652-663.

[33] Brin S., Motwani R., Ullman J. D., and Tsur S., "Dynamic itemset counting and implication rules for market basket data." **In Proc. 1997 ACM SIGMOD Intl. Conf. on Management of Data**, Tucson, AZ, May, 1997. pp. 255-264.

[34] Yen S.J. and Chen A .L.P. "An Efficient Approach to Discovering Knowledge from Large Databases." **Proc. the IEEE/ACM International Conference on Parallel and Distributed Information System**, 1996.

[35] Ng K. and Lam, W., "Updating of association rules dynamically." **Proceedings 1999 International Symposium on Database Applications in Non-Traditional Environments (DANTE '99),**1999. pp. 84 – 91.

[36] Imberman S.P., Tansel A. U. and Pacuit E., "An efficient for finding emerging large itemset." **TMD'04**, Seattle, Washington, August 22, 2004.

[37] Chen M.S., Han J. , P.S. Yu, "Data Mining: An Overview from a Database Perspective", **IEEE Transactions on Knowledge and Data Engineering**, 1996.

[38] Feller W. **An Introduction to Probability Theory and its applications**, Third edition Volume1, New York ,1968.

[39] Han J. and Kamber M., **Data Mining: Concepts and Techniques**. Second edition. San Francisco, Morgan Kaufmann Publishers. 2006.

[40] กัลยา วานิชย์บัญชา. 2552. **การใช้ SPSS for Windows ในการวิเคราะห์ข้อมูล**. กรุงเทพฯ : จุฬาลงกรณ์มหาวิทยาลัย.

[41] ศิริชัย พงษ์วิชัย. 2552. การวิเคราะห์ข้อมูลทางสถิติด้วยคอมพิวเตอร์เน้นสำหรับงานวิจัย. กรุงเทพฯ : จุฬาลงกรณ์มหาวิทยาลัย.

ภาคผนวก

ภาคผนวก ก.

การสร้างชุดข้อมูล Synthetic Dataset

การสร้างชุดข้อมูล Synthetic Dataset

ชุดข้อมูลสังเคราะห์ (Synthetic dataset) เป็นชุดข้อมูลที่นำเสนอโดย Agrawal et.al. [2] ซึ่งได้นำเสนอวิธีการสร้างชุดสังเคราะห์เพื่อใช้ประเมินประสิทธิภาพ (performance) ในการทำงานของอัลกอริทึมที่ใช้ในการค้นหากฎความสัมพันธ์ในฐานข้อมูลขนาดใหญ่โดยใช้หลักการทางสถิติมาใช้ในการสร้างชุดข้อมูล

หลักการทางสถิติต่างๆ ที่นำมาใช้ได้แก่ การสุ่มค่าความน่าจะเป็นเพื่อหาขนาดของไอเทมเซต, ขนาดของทรานแซคชันรวมถึงการสุ่มค่าของไอเทมที่จะใส่ในทรานแซคชันด้วยการแจกแจงแบบต่างๆ เช่น การแจกแจงแบบปกติ, การแจกแจงแบบยูนิฟอร์ม, การแจกแจงแบบปัวส์ซองและการแจกแจงแบบเอ็กซ์โพแนนเชียล เป็นต้น โดยหลักการทางสถิติเหล่านี้จะถูกนำมาประยุกต์ในการสร้างชุดข้อมูลทรานแซคชันสังเคราะห์ (Synthetic transaction) เพื่อใช้ในการประเมินประสิทธิภาพของอัลกอริทึมที่ครอบคลุมลักษณะข้อมูลที่เป็นช่วงของข้อมูลขนาดใหญ่ที่เกิดขึ้นในทรานแซคชันที่เลียนแบบการซื้อ-ขายของร้านค้าปลีก โมเดลของชุดข้อมูลสังเคราะห์นี้จะแสดงแนวโน้มของเซตของไอเทมหรือเซตของสินค้าที่พบว่ามักมีการซื้อไปด้วยกัน แต่ละเซตของสินค้าที่ซื้อไปด้วยกันจะแสดงอยู่ในรูปของจำนวนที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้สูงสุด (Potentially a maximal frequent itemset) โดยจำนวนที่สามารถเป็นฟรีเควนท์ไอเทมเซตนี้อาจประกอบด้วยไอเทมสมาชิกจำนวนต่างๆ กันขึ้นอยู่กับสมาชิกของไอเทมที่เป็นฟรีเควนท์ไอเทมเซต ตัวอย่างเช่น เซตของการซื้อขนมปัง, กาแฟ, สบู่และแชมพู ประกอบด้วยไอเทม 4 ตัวที่เป็นฟรีเควนท์ไอเทมเซต ในขณะที่อาจพบว่ามีสินค้าเพียงบางตัวจากเซตที่มีสูงสุด (Maximal set) ปรากฏในทรานแซคชันต่างๆ เช่น อาจพบว่าลูกค้าซื้อขนมปังและกาแฟ ซึ่งเป็นส่วนหนึ่งที่ปรากฏในเซตของจำนวนที่สามารถเป็น ฟรีเควนท์ไอเทมเซตได้สูงสุด หรือลูกค้าบางคนอาจซื้อเพียงขนมปังอย่างเดียว เป็นต้น ดังนั้นในทรานแซคชันหนึ่งอาจจะมีมากกว่า 1 ฟรีเควนท์ไอเทมเซต เช่น ลูกค้าอาจจะซื้อ แยม และนม เมื่อซื้อขนมปังและกาแฟด้วย โดยในที่นี้อาจพบว่าแยม และนม มักถูกซื้อไปด้วยกันและเป็นสินค้าที่อยู่ในรูปของฟรีเควนท์ไอเทมเซตอีกชุดหนึ่งที่พบนอกเหนือจากขนมปังและกาแฟก็เป็นได้ ขนาดของทรานแซคชันที่สร้างนั้นจะได้มาจากการจัดกลุ่มของไอเทมต่างๆ ด้วยค่าเฉลี่ย (Mean) ทำให้พบว่าจำนวนฟรีเควนท์ไอเทมเซตสูงสุดของบางตัวจะประกอบด้วยขนาดของไอเทมที่แตกต่างกัน

การสร้างชุดข้อมูลสังเคราะห์ด้วยแนวคิดของ Agrawal et. al. จะประกอบด้วยค่าพารามิเตอร์สำคัญที่ใช้กำหนดเพื่อสร้างชุดข้อมูลดังแสดงในตารางที่ ก.1 ซึ่งมีรายละเอียดดังนี้

พารามิเตอร์ 1 จำนวนไอเทม (N) ในที่นี้แสดงถึงจำนวนของไอเทมหรือสินค้าที่มีในฐานข้อมูลทั้งหมด ในการสร้างชุดข้อมูลนี้จะใช้แทนด้วยตัวเลขเช่น กำหนด N = 100 ไอเทม

จะหมายถึงฐานข้อมูลนั้นๆ ประกอบด้วยสินค้าจำนวน 100 ตัว ได้แก่ ไอเทม 1, 2, 3... 99, 100 เป็นต้น

พารามิเตอร์ 2 จำนวนของทรานแซคชัน (| D |) ใช้ในการกำหนดขนาดของข้อมูลที่ต้องการสร้างว่าประกอบด้วยจำนวนทรานแซคชันเท่าใด เช่นกำหนดให้ | D | = 10000 คือกำหนดให้มีการสร้างทรานแซคชันจำนวน 10000 ทรานแซคชัน เป็นต้น

พารามิเตอร์ 3 จำนวนที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้สูงสุด (ในงานวิจัยของ Agrawal et. al.จะใช้แทนด้วย | L | หมายถึง จำนวนชุดของไอเทมเซตทั้งหมดที่สามารถเป็นฟรีเควนท์ไอเทมเซตได้ ซึ่ง| L | เหล่านี้ จะถูกนำไปสุ่มสร้างในขั้นตอนของการสร้างชุดของจำนวนที่สามารถเป็นฟรีเควนท์ไอเทมเซตได้สูงสุดตามขนาด | L | ที่ได้กำหนดในพารามิเตอร์ตัวนี้ เช่น กำหนดให้ | L | = 200 ชุด หมายถึง ในชุดข้อมูลสังเคราะห์ประกอบด้วย จำนวนไอเทมที่จะเป็นฟรีเควนท์ไอเทมเซตได้สูงสุดจำนวน 200 ชุด เป็นต้น

พารามิเตอร์ 4 ค่าเฉลี่ยขนาดสูงสุดที่จะเป็นฟรีเควนท์ไอทเมเซตได้ (| I |) หมายถึง ค่าเฉลี่ยสำหรับค่า | L | ที่จะสร้างมีค่าเฉลี่ยในการเกิดทั้งหมดเท่ากับเท่าใด เช่น กำหนด | L | = 200, | I | = 4 หมายความว่าค่าเฉลี่ยของ | L | จำนวน 200 ชุด จะมีค่าเฉลี่ยของ L มีขนาดเท่ากับ 4 ไอเทม เป็นต้น

พารามิเตอร์ 5 ค่าเฉลี่ยของขนาดทรานแซคชัน ในการสร้างชุดข้อมูลสังเคราะห์เพื่อเลียนแบบการซื้อขายจำนวนสินค้าในร้านค้าต่างๆ นั้นจะพบว่าจำนวนของไอเทมที่ปรากฏในทรานแซคชันต่างๆ จะมีขนาดแตกต่างกันไปตามความต้องการของลูกค้า ดังนั้นการกำหนดค่าเฉลี่ยของขนาดทรานแซคชันในชุดข้อมูลสังเคราะห์นี้จะได้จากการสุ่มขนาดของทรานแซคชันจากการแจกแจงแบบปัวส์ซอง ซึ่งใช้สุ่มจำนวนของทรานแซคชันที่จะเกิดขึ้นในฐานข้อมูลด้วยขนาดต่างๆ กัน โดยค่าที่ใช้ในการสุ่มจะกำหนดจากค่าเฉลี่ย μ เท่ากับขนาดของ | T |

ตารางที่ ก.1 ค่าพารามิเตอร์ที่ใช้สร้างชุดข้อมูลสังเคราะห์

พารามิเตอร์	สัญลักษณ์
จำนวนไอเทม (Number of items)	N
จำนวนของทรานแซคชัน (Number of transactions)	| D |
จำนวนที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้สูงสุด (Number of maximal potentially frequent itemsets)	| L |
ค่าเฉลี่ยของขนาดสูงสุดที่จะเป็นฟรีเควนท์ไอเทมเซตได้ (Average size of the maximal potentially frequent itemsets)	| I |
ค่าเฉลี่ยของขนาดทรานแซคชัน(Average size of the transactions)	| T |

โดยการสร้างชุดข้อมูลสังเคราะห์นี้จะนำค่าพารามิเตอร์ที่ได้กล่าวข้างต้นมาใช้ในการสร้างชุดข้อมูลสังเคราะห์ด้วยวิธีที่นำเสนอโดย Agrawal et.al. นี้ ดังสามารถสรุปได้เป็น 2 ขั้นตอนใหญ่ๆ ดังนี้คือ

1. ขั้นตอนการสร้างชุดของจำนวนที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้สูงสุด

ในการสร้างชุดข้อมูลสังเคราะห์จะเริ่มจากการสุ่มเพื่อสร้างชุดของไอเทมที่สามารถกลายเป็นฟรีเควนท์ได้ด้วยจำนวน | L | ที่ได้กำหนด ดังแสดงตัวอย่างในตารางที่ ก.2 ซึ่งได้กำหนดจำนวน | L | เท่ากับ 20 ชุด (| L | = 20) หมายความว่า จะสุ่มสร้างชุดของไอเทมที่สามารถเป็นฟรีเควนท์ได้สูงสุดคือ 20 ชุด โดยทั้ง 20 ชุดจะถูกสร้างภายใต้ขนาดที่ได้จากการสุ่มด้วยค่าเฉลี่ยของขนาดสูงสุดที่จะเป็นฟรีเควนท์ไอเทมเซตได้ | I | โดยค่าที่ปรากฏในช่องจำนวนที่สุ่มได้จากการแจกแจงปัวส์ซองคือขนาดของ L ที่จะถูกนำไปสุ่มสร้างชุดของ L ต่อไป

ตารางที่ ก.2 แสดงตัวอย่างการสุ่มขนาดของ | L | ด้วยการแจกแจงปัวส์ซอง

L	จำนวนที่สุ่มได้จากการแจกแจงปัวส์ซอง	L	จำนวนที่สุ่มได้จากการแจกแจงปัวส์ซอง	L	จำนวนที่สุ่มได้จากการแจกแจงปัวส์ซอง	L	จำนวนที่สุ่มได้จากการแจกแจงปัวส์ซอง
L_1	5	L_6	6	L_{11}	2	L_{16}	3
L_2	2	L_7	4	L_{12}	5	L_{17}	8
L_3	6	L_8	6	L_{13}	6	L_{18}	4
L_4	5	L_9	5	L_{14}	5	L_{19}	3
L_5	5	L_{10}	4	L_{15}	3	L_{20}	8

เมื่อได้ขนาดที่จะนำมาสร้างชุดของจำนวนที่สามารถจะเป็นฟรีเควนท์ไอเทมเซตได้สูงสุด 20 ชุด โดยจะแทนด้วย $L_1 - L_{20}$ แล้ว ในขั้นตอนต่อไปจะทำการสุ่มเลือกไอเทมสำหรับ L ทั้ง 20 ชุด

- การสุ่มเลือกไอเทมสำหรับ L จะเริ่มจาก L ชุดแรก ด้วยการสุ่มไอเทมทุกตัวใส่ไปใน L ชุดแรก สำหรับ L ตัวถัดไปจะทำการสุ่มไอเทมตัวแรก จากนั้นไอเทมเซตย่อยจะถูกเลือกจากไอเทมเซตที่ได้สร้างไว้ก่อนหน้า การพิจารณาส่วนของไอเทมสำหรับแต่ละไอเทมเซตจากตัวแปรสุ่มของการแจกแจงเอ็กซ์โพเนนเชียลด้วยค่าเฉลี่ยเท่ากับค่าระดับความสัมพันธ์ (correlation level) ซึ่งในฐานข้อมูลจะกำหนดให้ค่าระดับความสัมพันธ์มีค่าเท่ากับ 0.5 สำหรับไอเทมที่เหลือจะใช้การสุ่ม

ตัวอย่างจากตารางที่ ก.2 พบว่า L_1 ที่สุ่มได้มีขนาดเท่ากับ 5 ไอเทม ดังนั้นจะเริ่มการสุ่มไอเทมให้กับชุดของ L_1 จนถึง L20 ที่มีขนาดของไอเทมที่ต้องสุ่มจำนวน 8 ไอเทม โดยชุดของ L_1 จะได้จากการสุ่มทั้งชุดในขณะที่ชุดของ L_2- L_{20} จะมีการสุ่มด้วยโดยใช้เอ็กซ์โพแนนเชียล ด้วยค่าเฉลี่ยเท่ากับค่า correlation = 0.5 โดยใน L_2- L_{20} นี้จะประกอบด้วยไอเทมบางส่วนที่ได้จาก L ก่อนหน้า เช่น กรณีชุดที่ 1 จากตารางที่ ก.2 จะสุ่มแบบ random ขนาดเท่ากับ 5 ไอเทมได้มาดังนี้คือ

L{1} = {1 3 8 9 10}

ตั้งแต่ชุด L{2} จะทำการสุ่ม ค่า e จากการสุ่มแบบเอกซ์โพเนนเชียล (Exponential random) ด้วยค่าเฉลี่ยเท่ากับค่าสหสัมพันธ์ (Correlation) = 0.5 ในชุดที่ 2 ตัวแรกจะได้จากการสุ่มส่วนสมาชิกตัวที่ 2 –n จะทำการเลือกจากชุดก่อนหน้าในที่นี้คือชุดที่ 1 เช่น

L{2} = {3 4}

ซึ่งจากการสุ่มในตัวอย่างนี้ได้ทำการเรียงลำดับไอเทมจากน้อยไปมากแล้ว จะพบว่าใน L{2} จะประกอบด้วยไอเทม {3} ที่ได้มาจาก L{1} จากนั้นจะทำการสุ่ม L{3} เป็นชุดต่อไปพบว่าประกอบด้วยสมาชิกจำนวน 6 ไอเทมดังนี้คือ

L{3} = {1 4 5 7 8 9} จาก L{3} จะพบว่ามีไอเทม 4 ที่ได้จาก L{2} ส่วนที่เหลือจะได้จากการสุ่มจนครบ 6 ไอเทม เป็นต้น โดยจะทำการสุ่ม L ทั้งหมดจนครบ 20 ชุด ดังแสดงในตารางที่ ก.3 ในช่อง ไอเทมเซต | L |

แต่ละไอเทมเซตใน | L | จะมีการให้น้ำหนัก (weight) ที่สัมพันธ์กับชุดข้อมูลที่สร้างโดยค่าน้ำหนักนี้จะเกี่ยวข้องกับค่าความน่าจะเป็นที่ไอเทมเซตเหล่านี้จะถูกหยิบ ค่าน้ำหนักจะกำหนดได้ด้วยการแจกแจงเอ็กซ์โพเนนเชียลด้วยค่าเฉลี่ย= 1 และนำค่าน้ำหนักที่ได้มาทำให้เป็นบรรทัดฐาน (normalized) โดยการนำค่าของน้ำหนักของ L แต่ละชุดหารด้วย จำนวนรวมของค่าน้ำหนักของ L ทั้งหมด ซึ่งจากการทำให้น้ำหนักเป็นบรรทัดฐานนี้ผลรวมของน้ำหนักของไอเทมเซตทั้งหมดใน | L | จะมีค่าเท่ากับ 1 ค่าน้ำหนักของแต่ละชุดจะสัมพันธ์กับความน่าจะเป็นที่ชุดของ L นั้นๆ จะถูกสุ่มหยิบไปใส่ในทรานแซคชันต่างๆ

ในโมเดลที่เกิดขึ้นโดยธรรมชาตินั้นไอเทมในฟรีเควนท์ไอเทมเซตมักจะไม่ได้ถูกซื้อไปด้วยกันเสมอ ดังนั้นเราจะกำหนดแต่ละไอเทมเซตใน | L | ด้วยระดับค่าคอรัปชัน c (corruption level c) เมื่อมีการเพิ่มไอเทมเซตในทรานแซคชัน เราจะนำไอเทมจากไอเทมเซตที่ยาวเท่ากับค่าที่ได้จากการแจกแจงแบบยูนิฟอร์ม (Uniform distributed) ด้วยค่าสุ่มระหว่าง 0 และ 1 ที่น้อยกว่าค่าคอรัปชัน c ดังนั้นสำหรับไอเทมเซตของของ l เราจะเพิ่ม 1 ไอเทมเข้าไปในทรานแซคชัน *1-c* ของเวลา, l-*1* ไอเทม *c(1-c)* ของเวลา, $l-2$ ไอเทม $c^2(1-c)$ ของเวลา เป็นต้น ระดับคอรัปชันสำหรับไอเทมเซตจะถูกกำหนดเป็นค่าคงที่ซึ่งได้จากการสุ่มด้วยการแจกแจงปกติด้วยค่าเฉลี่ยเท่ากับ 0.5 และค่าความแปรปรวนเท่ากับ 0.1 ตัวอย่างค่าคอรัปชัน c ที่ได้จากการแจกแจงปกติด้วยค่าเฉลี่ยเท่ากับ0.5 และค่าความแปรปรวนเท่ากับ 0.1 สำหรับ | L | ทั้ง 20 ในตัวอย่างข้างต้นแสดงในตารางที่

ก.3 แสดงตัวอย่างของการให้ค่าน้ำหนัก, ค่าที่ได้จากการทำค่าน้ำหนักให้เป็นบรรทัดฐาน, ชุดของไอเทมเซตของ | L | และค่าระดับคอรัปชัน c ที่ได้จากการแจกแจง

ตารางที่ ก.3 แสดงตัวอย่างการให้ค่าน้ำหนัก, ค่าที่ได้จากการทำค่าน้ำหนักให้เป็นบรรทัดฐาน, ชุดของไอเทมเซตของ | L | และค่าระดับคอรัปชัน c ที่ได้จากการแจกแจง

ชุดที่	ค่าน้ำหนัก (weight)	ค่าบรรทัดฐาน (normalize)	ไอเทมเซต | L |	ค่าระดับคอรัปชัน (c)
1	0.0056	0.0056	{1 3 8 9 10}	[0.3632]
2	0.0861	0.0917	{3 4}	[0.5396]
3	0.0333	0.1249	{1 4 5 7 8 9}	[0.5910]
4	0.0738	0.1987	{1 3 4 5 7}	[0.1375]
5	0.0367	0.2354	{1 5 6 7 8}	[0.8766]
6	0.0282	0.2636	{1 2 3 5 7 8}	[0.8760]
7	0.0727	0.3363	{1 3 6 8}	[0.4881]
8	0.0695	0.4058	{1 3 5 7 8 9}	[0.6035]
9	0.0333	0.4391	{2 6 7 8 10}	[0.5552]
10	0.014	0.4531	{2 6 7 8}	[0.4410]
11	0.078	0.5311	{2 7}	[0.7295]
12	0.0673	0.5984	{2 4 5 6 7}	[0.3140]
13	0.0409	0.6393	{1 2 4 5 7 10}	[0.4569]
14	0.0449	0.6842	{1 2 4 5 7}	[0.5360]
15	0.0262	0.7104	{2 4 9}	[0.8373]
16	0.0528	0.7633	{6 9 10}	[0.5187]
17	0.0199	0.7832	{2 3 4 5 6 7 9 10}	[0.4698]
18	0.0563	0.8395	{2 6 9 10}	[0.2368]
19	0.1325	0.972	{4 6 8}	[0.5931]
20	0.028	1	{1 3 4 5 7 8 9 10}	[0.0775]

2. ขั้นตอนการเลือกไอเทมเข้าไปในทรานแซคชัน

เริ่มจากการพิจารณาขนาดของทรานแซคชันซึ่งจะสุ่มเลือกขนาดด้วยการแจกแจงแบบปัวส์ซองด้วยการกำหนดค่าเฉลี่ยเท่ากับขนาดของค่าเฉลี่ยของขนาดทรานแซคชัน |T| เช่น กำหนด |T| = 4 สังเกตได้ว่าถ้าแต่ละไอเทมถูกเลือกด้วยค่าความน่าจะเป็น p เดียวกัน เมื่อมีจำนวน N ไอเทม จะพบว่าสามารถประมาณขนาดของทรานแซคชันที่ได้ด้วยการแจกแจงแบบทวินามด้วยค่าพารามิเตอร์ N และ p และสามารถประมาณค่าได้ด้วยการแจกแจงของปัวส์ซองด้วยค่าเฉลี่ย Np ตัวอย่างขนาดของทรานแซคชันที่ได้จากการสุ่มด้วยการแจกแจงปัวส์ซองจำนวน 100 ทรานแซคชัน แสดงในตารางที่ ก.4

ตารางที่ ก.4 แสดงตัวอย่างขนาดของทรานแซคชันที่ได้การแจกแจงจำนวน 100 ทรานแซคชัน

ขนาดของทรานแซคชันที่ได้จากการสุ่มด้วยการแจกแจงปัวส์ซอง									
ทรานแซคชัน									
1-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80	81-90	91-100
5	6	9	10	7	7	8	3	6	10
6	6	6	6	4	9	7	10	10	10
6	9	10	9	7	8	7	4	10	10
9	10	7	9	9	8	9	7	9	7
7	8	10	8	8	9	9	10	10	10
10	7	8	7	9	7	10	6	4	9
7	9	8	10	4	7	5	9	10	8
7	8	8	6	8	5	10	9	10	7
7	5	9	6	10	7	7	10	7	8
9	8	7	4	8	7	8	8	9	10

เมื่อได้ขนาดของไอเทมในแต่ละทรานแซคชันแล้ว ขั้นตอนต่อไปจะเป็นการนำไอเทมต่างๆ ใส่ไปในทรานแซคชันตามขนาดที่ได้จากการสุ่มข้างต้น และไอเทมที่นำไปใส่ในแต่ละทรานแซคชันจะกำหนดจากชุดของ | L | โดยจะทำการสุ่มชุดของข้อมูล | L | มาใส่ในทรานแซคชันดังนี้

ขั้นตอนที่ 2.1 ทำการสุ่มหยิบค่าน้ำหนักด้วยการแจกแจงยูนิฟอร์ม โดยจะนำค่าน้ำหนักที่ได้ไปตรวจสอบว่าอยู่ในช่วงของ | L | ชุดใด จะนำสมาชิกของ | L | ชุดนั้นมาพิจารณาเพื่อหยิบใส่ในทรานแซคชัน เช่น ถ้าค่าน้ำหนักที่สุ่มได้คือ 0.1111 ซึ่งค่าน้ำหนักนี้มีค่ามากกว่าค่า

น้ำหนักของ |L| ชุดที่ 2 ที่มีค่าเท่ากับ 0.0917 และมีค่าน้อยกว่า |L| ชุดที่ 3 ที่มีค่าเท่ากับ 0.1249 จะนำค่าสมาชิกของ |L| ชุดที่ 3 ที่ประกอบด้วย {1 4 5 7 8 9}

ขั้นตอนที่ 2.2 ทำการสุ่มค่าความน่าจะเป็นด้วยการแจกแจงยูนิฟอร์มสำหรับสมาชิกแต่ละตัวของชุดของ |L| ที่สุ่มได้จากขั้นตอนที่ 2.1 แล้วนำมาเปรียบเทียบกับค่าคอรัปชัน c ถ้าค่าความน่าจะเป็นของสมาชิกแต่ละตัวมีค่าน้อยกว่าค่าคอรัปชัน c จะนำมาใส่ในทรานแซคชัน โดยทำการเปรียบเทียบกับสมาชิกทุกตัวจนกว่าจะครบเท่ากับขนาดของทรานแซคชัน

สำหรับกรณีที่ชุดที่สุ่มได้จากขั้นตอนที่ 2.1 ทำการเปรียบเทียบจนครบทุกตัวแล้วแต่ยังได้ไม่ครบจำนวนทรานแซคชันจะทำการสุ่มน้ำหนักในขั้นตอนที่ 2.1 ใหม่จากนั้นจึงมาทำในขั้นตอนที่ 2.2 วนรอบซ้ำจนกว่าจะได้จำนวนไอเทมเท่ากับขนาดของทรานแซคชันจะหยุดสุ่มและเปรียบเทียบกับคอรัปชัน c จากนั้นจึงไปทำการสุ่มไอเทมสำหรับทรานแซคชันถัดไปโดยทำซ้ำขั้นตอนที่ 2.1 และ 2.2 จนครบจำนวนทรานแซคชันที่กำหนด

ตัวอย่างเช่น จากตาราง ก.4 ทรานแซคชันที่ 1 มีจำนวน 5 ไอเทม จากขั้นตอนที่ 2.1 สุ่มได้| L |ชุดข้อมูลที่ 3 ที่ประกอบด้วย {1 4 5 7 8 9} จะทำการสุ่มค่าความน่าจะเป็นให้กับสมาชิกทุกตัวของ| L |ชุดข้อมูลที่ 3 ถ้าค่าของสมาชิกตัวใดมีค่าความน่าจะเป็นน้อยกว่า 0.5910 ซึ่งเป็นค่าระดับคอรัปชัน c ของ |L |ชุดข้อมูลที่ 3 จะไอเทมนั้นๆ ไปไว้ในทรานแซคชันที่ 1 สมมติได้ไอเทมต่อไปนี้ในทรานแซคชัน 1 {1 5 9} ซึ่งจำนวนไอเทมยังไม่ครบจำนวนทรานแซคชันที่กำหนดไว้จากการสุ่มจะทำในขั้นตอนที่ 2.1 สมมติได้ค่าน้ำหนัก เท่ากับ 0.9998 คือ |L |ชุดที่ 20 จะนำสมาชิกชุดนี้ได้แก่ {1 3 4 5 7 8 9 10} มาสุ่มค่าความน่าจะเป็นในขั้นตอนที่ 2.2 เพื่อเปรียบเทียบกับระดับคอรัปชัน c ที่เท่ากับ 0.0775 ถ้าไอเทมใดมีค่าความน่าจะเป็นที่ได้จากการสุ่มน้อยกว่า 0.0775 จะถูกนำไปใส่ต่อกัยไอเทมที่ได้ไว้ก่อนหน้านี้ โดยจะตรวจสอบว่าไอเทมที่นำมาใส่ในทรานแซคชันต้องไม่ซ้ำกัน สมมติสุ่มเพิ่มได้ไอเทมที่ใส่ในทรานแซคชันที่ 1 เพิ่มอีก 2 ตัวคือไอเทม 3 และ 8 จนครบ 5 ไอเทม คือ {1 3 5 8 9} จึงเริ่มนำไอเทมไปใส่ทรานแซคชันที่ 2 และทรานแซคชันอื่นๆ ต่อไปจนครบจำนวนที่ต้องการสร้างดังแสดงในตารางที่ ก.5

ตารางที่ ก.5 แสดงตัวอย่างทรานแซคชันที่ได้จากการสร้างชุดข้อมูลสังเคราะห์

TID	ขนาดของทรานแซคชัน	ชุดข้อมูลสังเคราะห์
1	5	{1 3 5 8 9}
2	6	{2 4 5 6 8 10}
3	6	{1 2 4 5 8 9}
...	...	...
100	10	{1 2 3 4 5 6 7 8 9 10}

ภาคผนวก ข.

การวิเคราะห์เปรียบเทียบเวลาการทำงานของอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และพรีลาจก์

การวิเคราะห์เปรียบเทียบเวลาการทำงานของอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และพรีลาจก์

จากการทดลองชุดข้อมูลที่ 1 และชุดข้อมูลที่ 3 ในบทที่ 4 ซึ่งประกอบด้วยการทดลองข้อมูลด้วยค่าสนับสนุนน้อยที่สุด 1%, 3% และ 5% และมีการเพิ่มข้อมูลขนาดต่างๆ จำนวน 3 ขนาด คือ 2000, 5000 และ 10000 ทรานแซคชัน โดยทดลองเปรียบเทียบเวลาการทำงานของอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และพรีลาจก์ ในส่วนของอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นได้ทำการทดลองโดยกำหนดค่าความน่าจะเป็นที่ไอเทมจะกลายมาเป็นฟรีเควนท์ไอเทมเซต ($Prob_{EF}$) จำนวน 3 ค่าคือ $Prob_{EF}$ คือ 0.01, 0.03, 0.05

การวิเคราะห์เปรียบเทียบเวลาการทำงานของอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และพรีลาจก์ จะทำการวิเคราะห์แบบน็อนพาราเมตริกแบบจำแนกสองทางมา เนื่องจากชุดข้อมูลที่นำมาทดลองมีค่าความแปรปรวนของกลุ่มข้อมูลโดยใช้ Levene [40][41] ด้วยสมมติฐานสำหรับการทดสอบดังนี้คือ

1. สมมติฐานในการทดลองที่ค่าสนับสนุนน้อยที่สุด x% โดย x = 1, 3 และ5 สำหรับชุดการทดลองที่ 1

H_0 : ความแปรปรวนของเวลาของอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎ ความสัมพันธ์โดยใช้หลักความน่าจะเป็นกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และพรีลาจก์ที่ทดลองด้วยค่าสนับสนุนน้อยที่สุด x% เท่ากัน

H_1 : ความแปรปรวนของเวลาของอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และพรีลาจก์ที่ทดลองด้วยค่าสนับสนุนน้อยที่สุด x% ไม่เท่ากัน

2. สมมติฐานในการทดลองที่ค่าสนับสนุนน้อย x% โดย x = 1, 3 และ5 สำหรับชุดการทดลองที่ 3

H_0 : ความแปรปรวนของเวลาของอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎ ความสัมพันธ์โดยใช้หลักความน่าจะเป็นกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และพรีลาจก์ที่ทดลองด้วยค่าสนับสนุนน้อยที่สุด x% เท่ากัน

H_1 : ความแปรปรวนของเวลาของอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และพรีลาจก์ที่ทดลองด้วยค่าสนับสนุนน้อยที่สุด x% ไม่เท่ากัน

จากตารางที่ ข.1 ซึ่งแสดงการทดสอบความแปรปรวนของเวลาของอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และพรีลาจก์ไม่เท่ากัน สำหรับชุดข้อมูลที่ 1 และ 3 ดังแสดงในตารางที่ ข.1 ซึ่งพบว่าค่า Sig. สำหรับชุดข้อมูลที่ 1 เมื่อทดสอบข้อมูลที่ค่าสนับสนุนน้อยที่สุด 1%, 3% และ 5% เท่ากับ .000 ซึ่งน้อยกว่าระดับนัยสำคัญ .05 จึงปฏิเสธสมมติฐาน H_0 ที่ระดับนัยสำคัญ .05 และสำหรับชุดข้อมูลที่ 3 เมื่อทดสอบข้อมูลที่ค่าสนับสนุนน้อยที่สุด 1%, 3% และ 5% มีค่า Sig. = .000,.000 และ .005 ซึ่งน้อยกว่าระดับนัยสำคัญ .05 จึงปฏิเสธสมมติฐาน H_0 ที่ระดับนัยสำคัญ .05

สรุปได้ว่าค่าความแปรปรวนของเวลาการทำงานของอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์ และพรีลาจก์ ไม่เท่ากันที่ระดับนัยสำคัญ .05 สำหรับชุดข้อมูลที่ 1 และ 3 เมื่อทดสอบข้อมูลที่ค่าสนับสนุนน้อยที่สุด 1%, 3% และ 5% ซึ่งเป็นการทดลองที่ไม่สามารถใช้วิธีการของพาราเมตริกได้

ตารางที่ ข.1 แสดงการทดสอบค่าความแปรปรวน

Test of Homogeneity of Variance : Based on Mean					
ชุดข้อมูล	ค่าสนับสนุนน้อยที่สุด	Levene Statistic	df1	df2	Sig.
1.1	1%	11.714	6	56	.000
1.2	3%	12.517	6	56	.000
1.3	5%	10.317	5	48	.000
3.1	1%	10.410	6	56	.000
3.2	3%	5.479	6	56	.000
3.3	5%	3.534	6	56	.005

สำหรับข้อมูลที่นำมาวิเคราะห์แบบน็อนพาราเมตริกแบบจำแนกสองทางมาใช้ในการทดสอบเกี่ยวกับค่าเฉลี่ยทั้ง 7 กลุ่ม คือ อัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์, พรีลาจก์ และอัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นได้ทำการทดลองโดยกำหนดค่าค่าความน่าจะเป็นที่ไอเทมจะกลายมาเป็นฟรีเควนท์ไอเทมเซต ($Prob_{EF}$) จำนวน 3 ค่าคือ $Prob_{EF}$ คือ 0.01, 0.03, 0.05 ผลการทดลองที่ได้คือเวลาที่ได้จากการค้นหากฎความสัมพันธ์สำหรับแต่ละอัลกอริทึมซึ่งเป็นอิสระต่อกัน ดังนั้นข้อมูลนี้จึงเหมาะกับข้อมูลที่ได้จากแผนแบบการทดลองแบบสุ่มในบล็อก (Randomized Block Design : RBD) โดยไม่มีการวัดซ้ำ โดยแต่ละกลุ่มของเวลาที่นำมาวิเคราะห์ต้องเป็นข้อมูลชุดเดียวกัน จึงต้องควบคุมคุณลักษณะทั้งทางด้านแถวหรือบล็อก (Block) และ ทางด้านคอลัมน์ (Treatment) โดยในที่นี้จะควบคุมเวลาที่ได้จากการค้นหากฎความสัมพันธ์ของแต่ละอัลกอริทึมในแต่ละชุดข้อมูลย่อยของชุดข้อมูลทดลองที่ 1 คือ ชุดข้อมูลที่ 1.1, 1.2 และ 1.3 และชุดข้อมูลทดลองที่ 3 คือชุดข้อมูลที่ 3.1, 3.2 และ 3.3 เมื่อมีการเพิ่มขนาดข้อมูลใหม่เข้าไปในฐานข้อมูลเดิมขนาดเดียวกันคือ 2000, 5000 และ 10000 ทรานแซคชัน แยกตามค่าสนับสนุนน้อยที่สุดที่ทดลอง คือ 1%, 3% และ 5% ในส่วนนี้จะนำค่าเวลาเฉลี่ยของชุดข้อมูลทดลองที่ 1 และชุดข้อมูลทดลองที่ 3 มาวิเคราะห์ค่าสถิติซึ่งในการวิเคราะห์จะเป็นการเปรียบเทียบอันดับเฉลี่ยที่ได้ของแต่ละอัลกอริทึมว่าแตกต่างกันหรือไม่ ดังนี้คือ

1. การวิเคราะห์ค่าทางสถิติสำหรับชุดข้อมูลที่ 1

ภายใต้สมมติฐานของข้อมูลชุดทดลองที่ 1 คือ การทดลองเพิ่มขยายการค้นหากฎความสัมพันธ์ในกรณีค่าทางสถิติของฐานข้อมูลเดิมและฐานข้อมูลใหม่ไม่แตกต่างกัน โดยการวิเคราะห์ค่าทางสถิตินี้จะนำเวลาที่ได้จากการทดลองชุดข้อมูลที่1 ทั้ง 3 ชุด มาแยกเปรียบเทียบตามค่าสนับสนุนน้อยที่สุดที่ได้ทดลองทีละค่าคือ ที่ 1%, 3% และ 5% มาทำการเปรียบเทียบกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์, พรีลาจก์ และ อัลกอริทึมสำหรับการเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น ผลการทดสอบที่ได้จากการวิเคราะห์จะเป็นการเปรียบเทียบอันดับเฉลี่ยที่ได้ของแต่ละอัลกอริทึม ถ้าพบว่าไม่แตกต่างกันก็อาจกล่าวได้ว่าเวลาที่ใช้ในการค้นหากฎความสัมพันธ์ของแต่ละอัลกอริทึมไม่แตกต่างกัน โดยสมมติฐานทางสถิติสำหรับการทดสอบกำหนดได้ดังนี้คือ

1. สมมติฐานในการทดลองที่ค่าสนับสนุนน้อยที่สุด 1% สำหรับชุดการทดลองที่ 1

H_0: เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด 1% จากแต่ละอัลกอริทึมไม่แตกต่างกัน

H_1: เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด 1% จากแต่ละอัลกอริทึมแตกต่างกัน

2. สมมติฐานในการทดลองที่ค่าสนับสนุนน้อยที่สุด 3% สำหรับชุดการทดลองที่ 1

H_0: เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด 3% จากแต่ละอัลกอริทึมไม่แตกต่างกัน

H_1 : เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด 3% จากแต่ละอัลกอริทึมแตกต่างกัน

3. สมมติฐานในการทดลองที่ค่าสนับสนุนน้อยที่สุด 5% สำหรับชุดการทดลองที่ 1

H_0: เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด 5% จากแต่ละอัลกอริทึมไม่แตกต่างกัน

H_1 : เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด 5% จากแต่ละอัลกอริทึมแตกต่างกัน

การทดสอบจะใช้วิธีการทดสอบของฟริดแมน ซึ่งเป็นส่วนขยายของการทดสอบของ Wilcoxon ที่ใช้กับข้อมูล 2 กลุ่ม ที่มีความสัมพันธ์กัน ค่าสถิติของฟริดแมนคำนวณได้จากสูตรดังนี้

$$Fr = \frac{12}{nk(k+1)}\left(\sum_{i=1}^{k} R_i^2\right) - 3 * n * (k+1)$$

โดย Fr คือ ค่าสถิติของฟริดแมน

R_i คือ ค่าผลของอันดับในกลุ่มที่ i

k คือ จำนวนกลุ่มที่นำมาทดสอบ

n คือ จำนวนตัวอย่างที่นำมาทดสอบ

เนื่องจากค่าสถิติฟริดแมนจะมีการแจกแจงใกล้เคียงกับ การแจกแจงของไคสแควร์ (Chi-square) ดังนั้นในการทดสอบสมมติฐานจะนำค่าที่คำนวณได้นี้ไปเปรียบเทียบกับค่าที่ได้จากตารางของไคสแควร์ โดยใช้ DF เท่ากับ k-1

ข้อมูลที่นำมาทดสอบจะแยกตามค่าสนับสนุนน้อยที่สุดสำหรับชุดการทดลองที่ 1 ซึ่งประกอบด้วยชุดข้อมูลที่นำมาทดสอบ 3 ชุด คือชุดข้อมูลที่ 1.1, ชุดข้อมูลที่ 1.2, ชุดข้อมูลที่ 1.3 ดังแสดงในตารางที่ ข.2 – ข. 4

ตารางที่ ข.2 เวลาในการทดลองชุดข้อมูลที่ 1 ที่ค่าสนับสนุนน้อยที่สุด 1%

เวลาในการทดลองชุดข้อมูลที่ 1 (วินาที) ที่ค่าสนับสนุนน้อยที่สุด 1%								
ชุดข้อมูล	ขนาดข้อมูล	Apriori	FUP	Border	Prelarge	Prob_Based		
						$Prob_{EF}$		
						0.01	0.03	0.05
1.1	20%	176,935	15,5622	18,004	25,884	7,566	7,574	7,592
1.2	20%	255,801	43,996	25,727	102,995	24,3535	22,607	22,921
1.3	20%	5,778	602	694	742	419	416	415
1.1	50%	217,735	32,052	45,168	32,714	13,9760	13,524	13,450
1.2	50%	316,048	56,828	63,631	120,335	41,5374	39,122	39,769
1.3	50%	7,304	1,070	1,756	1,434	946	934	926
1.1	100%	288,255	29,554	89,594	47,317	23,387	23,237	23,068
1.2	100%	432,063	81,460	129,159	137,798	69,992	65,601	66,295
1.3	100%	9,777	1,925	3,484	2,430	1,847	1,832	1,812

ตารางที่ ข.3 เวลาในการทดลองชุดข้อมูลที่ 1 ที่ค่าสนับสนุนน้อยที่สุด 3%

เวลาในการทดลองชุดข้อมูลที่1 (วินาที) ค่าสนับสนุนน้อยที่สุด 3%								
ชุดข้อมูล	ขนาดข้อมูล	Apriori	FUP	Border	Prelarge	Prob_Based		
						$Prob_{EF}$		
						0.01	0.03	0.05
1.1	20%	18,978	1,651	2,962	2,861	1,189	1,320	1,223
1.2	20%	17,134	1,589	3,126	1,454	1,242	1,183	1,248
1.3	20%	1,357	196	288	159	139	143	140
1.1	50%	23,858	2,963	7,148	4,181	2,467	2,561	2,589
1.2	50%	21,167	3,112	7,543	3,006	2,902	2,811	2,861
1.3	50%	1,786	385	819	343	329	330	329
1.1	100%	31,626	5,104	14,311	5,784	4,921	4,916	4,954
1.2	100%	28,430	5,893	15,198	5,930	5,744	5,623	5,659
1.3	100%	2,378	686	1,557	648	634	634	632

ตารางที่ ข.4 เวลาในการทดลองชุดข้อมูลที่ 1 ที่ค่าสนับสนุนน้อยที่สุด 5%

เวลาในการทดลองชุดข้อมูลที่1 (วินาที) ค่าสนับสนุนน้อยที่สุด 5%								
ชุดข้อมูล	ขนาดข้อมูล	Apriori	FUP	Border	Prelarge	Prob_Based		
						$Prob_{EF}$		
						0.01	0.03	0.05
1.1	20%	107,88	1,620	2,249	1,090	756	761	903
1.2	20%	11,673	1,273	2,593	1,099	932	938	936
1.3	20%	785	146	199	94	88	86	86
1.1	50%	13,929	3,584	6,687	2,174	1,894	1,999	2,001
1.2	50%	14,559	2,525	6,269	2,391	2,241	2,233	2,235
1.3	50%	986	254	495	213	194	194	194
1.1	100%	18,631	4,805	11,850	4,260	3,782	3,842	3,899
1.2	100%	19,506	4,827	13,008	4,658	4,535	4,527	4,560
1.3	100%	1,311	433	970	384	373	373	373

ผลลัพธ์ที่ได้จากการวิเคราะห์ความแปรปรวนแบบน็อนพาราเมตริกแบ่งได้ 3 ส่วนใหญ่ๆ โดยมีความหมายแสดงในตารางที่ ข.5, ตารางที่ ข.6 และตารางที่ ข.7 ซึ่งแสดงค่าสถิติเบื้องต้นของอัลกอริทึมทั้งหมดที่นำมาทดสอบความแตกต่างของค่าเฉลี่ย โดยประกอบด้วยค่าสถิติเบื้องต้นดังนี้คือ

ข้อมูลที่ปรากฏในตารางที่ ข.5 ตารางที่ ข.6 และ ตารางที่ ข.7 เป็นส่วนที่แสดงค่าเฉลี่ยของอันดับในแต่ละกลุ่ม จะพบว่าค่าเฉลี่ยของอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นมีค่าเฉลี่ยของเวลาในการค้นหากฎความสัมพันธ์น้อยที่สุดและอัลกอริทึมอะพริโอริมีค่าค่าเฉลี่ยของเวลาในการค้นหากฎความสัมพันธ์มากที่สุด สำหรับตารางที่ ข.8, ข.9 และ ข.10 เป็นส่วนที่แสดงค่าสถิติของฟริดแมนสำหรับทดสอบสมมติฐานคือ H_0 และH_1 ดังกล่าวข้างต้น ผลการทดสอบจะพิจารณาค่าฟริดแมนที่อยู่ในรูปของค่าสถิติ Chi-square หรือค่าความน่าจะเป็นในการยอมรับสมมติฐาน Asymp. Sig.จากผลที่ปรากฏในตารางที่ ข.11 ซึ่งเป็นผลการทดสอบสำหรับค่าสนับสนุนน้อยที่สุดที่ 1%, 3% และ 5% พบว่าค่า Asymp. Sig. มีค่าเท่ากับ .00 ซึ่งน้อยกว่าค่า α ที่กำหนดเท่ากับ .05 ดังนั้นจึงปฏิเสธ H_0

สรุปผลทดสอบได้ว่ามีอย่างน้อย 2 อัลกอริทึมที่มีเวลาเฉลี่ยแตกต่างกันที่ระดับนัยสำคัญ .05

ตารางที่ ข.5 การทดสอบแบบน็อนพาราเมตริกชุดข้อมูลที่1 สำหรับค่าสนับสนุนน้อยที่สุด 1%

Descriptive Statistics					
อัลกอริทึม	N	Mean	Std. Deviation	Minimum	Maximum
Apriori	9	1.8997E5	1.53786E5	5778.45	4.32E5
FUP	9	2.9228E4	27974.72460	602.69	81460.75
Border	9	4.1914E4	44706.86280	694.84	1.29E5
Prelarge	9	5.2406E4	54022.35767	742.45	1.38E5
Prob_Based($Prob_{EF}$ = 0.01)	9	2.0448E4	23072.00103	419.36	69992.47
Prob_Based($Prob_{EF}$ = 0.03)	9	1.9428E4	21601.39774	416.98	65601.96
Prob_Based($Prob_{EF}$ = 0.05)	9	1.9584E4	21867.45546	415.66	66295.56

ตารางที่ ข.6 การทดสอบแบบน็อนพาราเมตริกชุดข้อมูลที่ 1 สำหรับค่าสนับสนุนน้อยที่สุด 3%

Descriptive Statistics					
อัลกอริทึม	N	Mean	Std. Deviation	Minimum	Maximum
Apriori	9	1.6302E4	11717.87381	1357.84	31626.55
FUP	9	2.3982E3	2043.69967	196.58	5893.91
Border	9	5.8840E3	5634.35776	288.25	15198.58
Prelarge	9	2.7080E3	2236.87942	159.88	5930.05
Prob_Based($Prob_{EF}$ = 0.01)	9	2.1745E3	2020.49131	139.69	5744.38
Prob_Based($Prob_{EF}$ = 0.03)	9	2.1694E3	1986.63515	143.39	5623.12
Prob_Based($Prob_{EF}$ = 0.05)	9	2.1822E3	2006.28041	140.38	5659.59

ตารางที่ ข.7 การทดสอบแบบน็อนพาราเมตริกชุดข้อมูลที่1 สำหรับค่าสนับสนุนน้อยที่สุด 5%

Descriptive Statistics					
อัลกอริทึม	N	Mean	Std. Deviation	Minimum	Maximum
Apriori	9	1.0242E4	7463.66495	785.75	19506.61
FUP	9	2.1634E3	1871.58930	146.08	4827.53
Border	9	4.9248E3	4852.79874	199.31	13008.11
Prelarge	9	1.8187E3	1700.65108	94.11	4658.52
Prob_Based($Prob_{EF}$ = 0.01)	9	1.6442E3	1611.76689	88.33	4535.31
Prob_Based($Prob_{EF}$ = 0.03)	9	1.6619E3	1621.64097	86.93	4527.45
Prob_Based($Prob_{EF}$ = 0.05)	9	1.6879E3	1629.46875	86.93	4560.50

ตารางที่ ข.8 การทดสอบฟริดแมนสำหรับชุดข้อมูลที่ 1 ด้วยค่าสนับสนุนน้อยที่สุด 1%

Friedman Test : Ranks	
อัลกอริทึม	Mean Rank
Apriori	7.00
FUP	4.11
Border	5.33
Prelarge	5.56
Prob_Based($Prob_{EF}$ = 0.01)	2.78
Prob_Based($Prob_{EF}$ = 0.03)	1.67
Prob_Based($Prob_{EF}$ = 0.05)	1.56

ตารางที่ ข.9 การทดสอบฟริดแมนสำหรับชุดข้อมูลที่ 1 ด้วยค่าสนับสนุนน้อยที่สุด 3%

Friedman Test : Ranks	
อัลกอริทึม	Mean Rank
Apriori	7.00
FUP	4.56
Border	6.00
Prelarge	4.44
Prob_Based($Prob_{EF}$ = 0.01)	1.89
Prob_Based($Prob_{EF}$ = 0.03)	2.00
Prob_Based($Prob_{EF}$ = 0.05)	2.11

ตารางที่ ข.10 การทดสอบฟริดแมนสำหรับชุดข้อมูลที่ 1 ด้วยค่าสนับสนุนน้อยที่สุด 5%

Friedman Test : Ranks	
อัลกอริทึม	Mean Rank
Apriori	7.00
FUP	5.00
Border	6.00
Prelarge	4.00
Prob_Based($Prob_{EF}$ = 0.01)	1.78
Prob_Based($Prob_{EF}$ = 0.03)	1.83
Prob_Based($Prob_{EF}$ = 0.05)	2.39

ตารางที่ ข.11 การทดสอบทางสถิติสำหรับสำหรับชุดข้อมูลที่ 1 ที่ค่าสนับสนุน 1%, 3% และ 5%

Test Statistics[a]			
ค่าสนับสนุนน้อยที่สุด	1%	3%	5%
N	9	9	9
Chi-Square	50.381	49.238	50.886
df	6	6	6
Asymp. Sig.	.000	.000	.000
a. Friedman Test			

จากผลการทดสอบที่ทราบว่าเวลาเฉลี่ยของอัลกอริทึมแตกต่างกัน สามารถนำมาทดสอบแบบจับคู่โดยวิธี Wilcoxon Sign Rank test เพื่อให้ทราบว่าอัลกอริทึมใดแตกต่างกัน โดยในการทดสอบจะทำการเปรียบเทียบความแตกต่างของเวลาที่ได้จากการค้นหาของอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นเทียบกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์และพรีลาจก์ คือ โดยกำหนดค่าค่าความน่าจะเป็นที่ไอเทมจะกลายมาเป็นฟรีเควนท์ไอเทมเซต ($Prob_{EF}$) จำนวน 3 ค่าคือ $Prob_{EF}$ คือ 0.01, 0.03, 0.05 เมื่อทำการทดลองที่ค่าสนับสนุนน้อยที่สุด x% ซึ่ง x มีค่าเท่ากับ 1, 3 และ 5 โดยกำหนดสมมติฐานในการทดสอบดังนี้คือ

1. เปรียบเทียบอัลกอริทึมอะพริโอริและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น

H_0: เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด x% ของอัลกอริทึมอะพริโอริและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นไม่แตกต่างกัน

H_1: เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด x% ของอัลกอริทึมอะพริโอริและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นแตกต่างกัน

2. เปรียบเทียบอัลกอริทึมเอฟยูพีและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น

H_0: เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด x% ของอัลกอริทึมเอฟยูพีและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นไม่แตกต่างกัน

H_1: เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด x% ของอัลกอริทึมเอฟยูพีและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นแตกต่างกัน

3. เปรียบเทียบอัลกอริทึมบอร์เดอร์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น

H_0: เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด x% ของอัลกอริทึมบอร์เดอร์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นไม่แตกต่างกัน

H_1 : เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด x% ของอัลกอริทึมเอฟยูพีและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นแตกต่างกัน

4. เปรียบเทียบอัลกอริทึมพรีลาจก์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น

H_0: เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด x% ของอัลกอริทึมพรีลาจก์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นไม่แตกต่างกัน

H_1 : เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด x% ของอัลกอริทึมพรีลาจก์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นแตกต่างกัน

จากตารางที่ ข.12 - ข.15 เป็นการทดสอบเปรียบเทียบค่าเฉลี่ยระหว่างอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นเทียบกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์และพรีลาจก์ โดยกำหนดค่าค่าความน่าจะเป็นที่ไอเทมจะกลายมาเป็นฟรีเควนท์ไอเทมเซต ($Prob_{EF}$) จำนวน 3 ค่าคือ $Prob_{EF}$ คือ 0.01, 0.03, 0.05 ที่ค่าสนับสนุนน้อยที่สุด 1%พบว่าค่า Asymp. Sig. จากตารางที่ ข.12- ข.15 คือ .008 ซึ่งมีค่าน้อยกว่าค่า α ที่กำหนดไว้ คือ .05 ดังนั้นจึงปฏิเสธ H_0 และยอมรับH_1

สรุปผลการทดสอบได้ว่าเวลาเฉลี่ยระหว่างอัลกอริทึมอะพริโอริ,เอฟยูพี,บอร์เดอร์และพรีลาจก์แตกต่างจากอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นที่ค่า $Prob_{EF}$= 0.01, 0.03 และ 0.05 ที่ค่าสนับสนุนน้อยที่สุด 1% อย่างมีนัยสำคัญ .05

ตารางที่ ข.12 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมอะพริโอริและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นชุดข้อมูลที่ 1 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 1%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - Apriori		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.13 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมเอฟยูพีและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น ชุดข้อมูลที่ 1 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 1%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - FUP		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.14 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมบอร์เดอร์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะชุดข้อมูลที่ 1 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 1%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - Border		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.15 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมพรีลาจก์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น ชุดข้อมูลที่ 1 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 1%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - Prelarge		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

การวิเคราะห์ค่าทางสถิติสำหรับชุดข้อมูลที่ 1 ที่ทดลองด้วยค่าสนับสนุนน้อยที่สุด 3% และ 5% ทำเช่นเดียวกับที่ค่าสนับสนุนน้อยที่สุด 1% จากตารางที่ ข.11 พบว่าเวลาเฉลี่ยของอัลกอริทึมแตกต่างกัน สามารถนำมาทดสอบแบบจับคู่โดยวิธี Wilcoxon Sign Rank test เพื่อให้ทราบว่าอัลกอริทึมใดแตกต่างกัน โดยในการทดสอบจะทำการเปรียบเทียบความแตกต่างของเวลาที่ได้จากการค้นหาของอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นเทียบกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์และพรีลาจก์ที่ค่าสนับสนุนน้อยที่สุด 3% และ5% ดังแสดงในตารางที่ ข.16 – ข.23 พบว่าผลที่ได้จากการเปรียบเทียบเวลาเฉลี่ยระหว่างอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์, พรีลาจก์กับอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นที่ค่าสนับสนุนน้อยที่สุด 3% และ 5% พบว่าค่า Asymp. Sig. จากตารางที่ ข.16- ข.23 มีค่าน้อยกว่าค่า α ที่กำหนดไว้ คือ .05 ดังนั้นจึงปฏิเสธ H_0 และยอมรับ H_1

สรุปผลการทดสอบได้ว่า เวลาเฉลี่ยระหว่างอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์และพรีลาจก์แตกต่างจากอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นที่ค่า $Prob_{EF}$= 0.01, 0.03 และ 0.05 ที่ค่าสนับสนุนน้อยที่สุด 3% และ 5% อย่างมีนัยสำคัญ .05

ตารางที่ ข.16 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมอะพริโอริและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นชุดข้อมูลที่ 1 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 3%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - Apriori		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.17 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมเอฟยูพีและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น ชุดข้อมูลที่ 1 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 3%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - FUP		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.18 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมบอร์เดอร์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะชุดข้อมูลที่ 1 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 3%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - Border		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.19 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมพรีลาจก์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น ชุดข้อมูลที่ 1 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 3%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - Prelarge		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.20 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมอะพริโอริและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นชุดข้อมูลที่ 1 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 5%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - Apriori		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.21 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมเอฟยูพีและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น ชุดข้อมูลที่ 1 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 5%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - FUP		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asym	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.22 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมบอร์เดอร์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะชุดข้อมูลที่ 1 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 5%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - Border		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.23 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมพรีลาจก์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น ชุดข้อมูลที่ 1 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 5%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - Prelarge		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

2. การวิเคราะห์ค่าเวลาเฉลี่ยของข้อมูลชุดทดลองที่ 3

การทดลองในกรณีค่าทางสถิติของฐานข้อมูลเดิมและฐานข้อมูลใหม่แตกต่างกัน เพื่อใช้ในการทดลองการทำงานของอัลกอริทึมว่าสามารถค้นหาฟรีเควนท์ไอเทมเซตจากฐานข้อมูลปรับปรุงได้ครบถ้วนถูกต้องเมื่อเทียบผลที่ได้จากการค้นหากฎความสัมพันธ์ทียบกับอัลกอริทึมอะพริโอริ เอฟยูพี บอร์เดอร์และพรีลาจก์

การวิเคราะห์จะเป็นการเปรียบเทียบอันดับเวลาเฉลี่ยที่ได้ของแต่ละอัลกอริทึมเช่นเดียวกับชุดข้อมูลที่ 1 คือว่าแตกต่างกันหรือไม่ ถ้าไม่แตกต่างกันก็อาจกล่าวได้ว่า เวลาที่ใช้ในการค้นหากฎความสัมพันธ์ของแต่ละอัลกอริทึมไม่แตกต่างกัน ดังนั้น สมมติฐานทางสถิติสำหรับการทดสอบกำหนดดังนี้

1. สมมติฐานในการทดลองที่ค่าสนับสนุนน้อยที่สุด 1% สำหรับชุดการทดลองที่ 3

H_0: เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด 1% จากแต่ละอัลกอริทึมไม่แตกต่างกัน

H_1: เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด 1% จากแต่ละอัลกอริทึมแตกต่างกัน

2. สมมติฐานในการทดลองที่ค่าสนับสนุนน้อยที่สุด 3% สำหรับชุดการทดลองที่ 3

H_0: เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด 3% จากแต่ละอัลกอริทึมไม่แตกต่างกัน

H_1 : เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด 3% จากแต่ละอัลกอริทึมแตกต่างกัน

3. สมมติฐานในการทดลองที่ค่าสนับสนุนน้อยที่สุด 5% สำหรับชุดการทดลองที่ 3

H_0: เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด 5% จากแต่ละอัลกอริทึมไม่แตกต่างกัน

H_1 : เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด 5% จากแต่ละอัลกอริทึมแตกต่างกัน

การทดสอบจะใช้วิธีการทดสอบของฟริดแมนซึ่งเป็นส่วนขยายของการทดสอบของ Wilcoxon ที่ใช้กับข้อมูล 2 กลุ่มที่มีความสัมพันธ์กัน ข้อมูลที่นำมาทดสอบจะแยกตามค่าสนับสนุนน้อยที่สุดสำหรับชุดการทดลองที่ 1 ซึ่งประกอบด้วยชุดข้อมูลที่นำมาทดสอบ 3 ชุด คือชุดข้อมูลที่ 3.1, 3.2 และ 3.3 ดังแสดงในตารางที่ ข.24 – ข.26

ตารางที่ ข.24 เวลาในการทดลองชุดข้อมูลที่ 3 ที่ค่าสนับสนุนน้อยที่สุด 1%

เวลาในการทดลองชุดข้อมูลที่ 3 (วินาที) ที่ค่าสนับสนุนน้อยที่สุด 1%								
ชุดข้อมูล	ขนาดข้อมูล	Apriori	FUP	Border	Prelarge	Prob_Based $Prob_{EE}$		
						0.01	0.03	0.05
3.1	20%	177,210	16,390	41,340	27,830	8,364	8,517	8,657
3.2	20%	175,294	12,281	41,834	20,056	6,545	6,595	6,341
3.3	20%	183,748	24,173	46,498	34,727	18,175	20,919	20,853
3.1	50%	218,009	22,676	122,566	36,222	14,891	14,935	14,849
3.2	50%	234,722	32,409	125,878	52,771	30,392	31,878	32,019
3.3	50%	207,850	19,135	114,304	27,489	11,785	11,917	11,504
3.1	100%	293,290	22,489	213,906	30,236	17,176	17,289	16,611
3.2	100%	308,530	33,941	252,870	51,634	26,409	26,417	26,245
3.3	100%	321,362	53,249	258,738	82,237	49,303	50,304	49,460

ตารางที่ ข.25 เวลาในการทดลองชุดข้อมูลที่ 3 ที่ค่าสนับสนุนน้อยที่สุด 3%

เวลาในการทดลองชุดข้อมูลที่ 3 (วินาที) ที่ค่าสนับสนุนน้อยที่สุด 3%								
ชุดข้อมูล	ขนาดข้อมูล	Apriori	FUP	Border	Prelarge	Prob_Based		
						$Prob_{EF}$		
						0.01	0.03	0.05
3.1	20%	19,107	1,922	3,957	1,827	1,274	1,095	1,095
3.2	20%	18,027	1,603	3,988	1,431	1,071	1,171	1,165
3.3	20%	18,054	4,669	3,854	2,292	1,606	1,771	1,771
3.1	50%	24,909	2,546	9,993	3,592	2,906	2,849	2,849
3.2	50%	22,922	3,044	8,538	2,843	2,628	2,718	2,695
3.3	50%	21,159	8,598	10,009	4,270	3,934	4,094	4,126
3.1	100%	24,350	4,885	15,004	4,870	4,699	4,763	4,709
3.2	100%	33,300	6,057	17,179	6,338	5,653	5,527	5,527
3.3	100%	25,904	15,179	17,958	7,719	7,692	7,780	7,689

ตารางที่ ข.26 เวลาในการทดลองชุดข้อมูลที่ 3 ที่ค่าสนับสนุนน้อยที่สุด 5%

เวลาในการทดลองชุดข้อมูลที่ 3 (วินาที) ที่ค่าสนับสนุนน้อยที่สุด 5%								
ชุดข้อมูล	ขนาดข้อมูล	Apriori	FUP	Border	Prelarge	Prob_Based		
						ProbEF		
						0.01	0.03	0.05
3.1	20%	10,692	1,125	3,096	1,074	771	776	776
3.2	20%	10,496	1,025	2,389	896	704	705	702
3.3	20%	10,983	1,224	3,398	1,143	914	986	994
3.1	50%	13,640	2,179	7,581	2,100	1,852	1,866	1,866
3.2	50%	13,893	2,564	8,075	2,416	2,285	2,366	2,340
3.3	50%	12,818	1,964	5,198	1,875	1,667	1,667	1,662
3.1	100%	13,025	2,896	8,852	2,906	2,675	2,674	2,664
3.2	100%	17,810	3,908	9,955	3,789	3,627	3,652	3,652
3.3	100%	18,664	4,796	15,368	4,714	4,573	4,631	4,579

ผลลัพธ์ที่ได้จากการวิเคราะห์ความแปรปรวนแบบน็อนพาราเมตริกแบ่งได้ 3 ส่วนใหญ่ๆ โดยมีความหมายแสดงในตารางที่ ข.27, ข.28 และ ข.29 ซึ่งแสดงค่าสถิติเบื้องต้นของอัลกอริทึมทั้งหมดที่นำมาทดสอบความแตกต่างของค่าเฉลี่ย โดยประกอบด้วยค่าสถิติเบื้องต้นดังนี้คือ

ข้อมูลที่ปรากฏในตารางที่ ข.30, ข.31 และ ข.32 เป็นส่วนที่แสดงค่าเฉลี่ยของอันดับในแต่ละกลุ่มจะพบว่าค่าเฉลี่ยของอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นมีค่าเฉลี่ยของเวลาในการค้นหากฎความสัมพันธ์น้อยที่สุดและอัลกอริทึมอะพริโอริมีค่าค่าเฉลี่ยของเวลาในการค้นหากฎความสัมพันธ์มากที่สุด สำหรับตารางที่ ข.33 เป็นส่วนที่แสดงค่าสถิติของฟริดแมนสำหรับทดสอบสมมติฐานคือ H_0 และH_1 ดังกล่าวข้างต้น ผลการทดสอบจะพิจารณาค่าฟริดแมนที่อยู่ในรูปของค่าสถิติไคสแควร์หรือค่าความน่าจะเป็นในการยอมรับสมมติฐาน Asymp. Sig. จากผลที่ปรากฏในตารางที่ ข.33 ซึ่งเป็นผลการทดสอบสำหรับค่าสนับสนุนน้อยที่สุดที่ 1%, 3% และ 5% พบว่าค่า Asymp. Sig. มีค่าเท่ากับ .00 ซึ่งน้อยกว่าค่า α ที่กำหนดเท่ากับ .05 ดังนั้นจึงปฏิเสธ H_0

สรุปผลทดสอบพบว่า มีอย่างน้อย 2 อัลกอริทึมที่มีเวลาเฉลี่ยแตกต่างกันที่ระดับนัยสำคัญ .05

ตารางที่ ข.27 การทดสอบแบบน็อนพาราเมตริกชุดข้อมูลที่ 3 สำหรับค่าสนับสนุนน้อยที่สุด 1%

Descriptive Statistics					
Algorithm	N	Mean	Std.	Minimum	Maximum
Apriori	9	1.5528E4	3787.45689	10818.84	21111.78
FUP	9	3.7652E3	2638.31337	1155.45	9487.26
Border	9	9.5535E3	5967.35443	2896.12	18463.92
Prelarge	9	5.6617E3	5569.18837	1143.86	18678.02
Prob_Based($Prob_{EF}$ = 0.01)	9	2.7259E3	1679.48857	903.06	4924.59
Prob_Based($Prob_{EF}$ = 0.03)	9	3.0239E3	1910.08950	986.69	6354.40
Prob_Based($Prob_{EF}$ = 0.05)	9	2.0833E3	1617.87213	456.11	4811.59

ตารางที่ ข.28 การทดสอบแบบน็อนพาราเมตริกชุดข้อมูลที่ 3 สำหรับค่าสนับสนุนน้อยที่สุด 3%

Descriptive Statistics					
Algorithm	N	Mean	Std.	Minimum	Maximum
Apriori	9	2.3082E4	4843.45333	18027.52	33300.88
FUP	9	5.3899E3	4290.88292	1603.50	15179.70
Border	9	1.0054E4	5617.49285	3854.56	17958.14
Prelarge	9	3.9096E3	2115.37560	1431.27	7719.95
Prob_Based($Prob_{EF}$ = 0.01)	9	3.4964E3	2219.00519	1071.75	7692.88
Prob_Based($Prob_{EF}$ = 0.03)	9	3.5302E3	2224.75094	1095.26	7780.19
Prob_Based($Prob_{EF}$ = 0.05)	9	3.5145E3	2202.38621	1095.26	7689.69

ตารางที่ ข.29 การทดสอบแบบน็อนพาราเมตริกชุดข้อมูลที่ 3 สำหรับค่าสนับสนุนน้อยที่สุด 5%

Descriptive Statistics					
Algorithm	N	Mean	Std.	Minimu	Maximum
Apriori	9	1.3558E4	2943.53217	10496.92	18664.28
FUP	9	2.4094E3	1295.43233	1025.45	4796.48
Border	9	7.1018E3	4127.92038	2389.72	15368.50
Prelarge	9	2.3241E3	1296.84733	896.02	4714.59
Prob_Based($Prob_{EF}$ = 0.01)	9	2.1193E3	1331.48040	704.29	4573.89
Prob_Based($Prob_{EF}$ = 0.03)	9	2.1476E3	1340.54141	705.67	4631.28
Prob_Based($Prob_{EF}$ = 0.05)	9	2.1377E3	1327.28842	702.94	4579.22

ตารางที่ ข.30 การทดสอบฟริดแมนสำหรับชุดข้อมูลที่ 3 ด้วยค่าสนับสนุนน้อยที่สุด 1%

Friedman Test : Ranks	
Algorithm	Mean Rank
Apriori	7.00
FUP	3.33
Border	5.89
Prelarge	5.11
Prob_Based($Prob_{EF}$ = 0.01)	1.44
Prob_Based($Prob_{EF}$ = 0.03)	2.67
Prob_Based($Prob_{EF}$ = 0.05)	2.56

ตารางที่ ข.31 การทดสอบฟริดแมนสำหรับชุดข้อมูลที่ 3 ด้วยค่าสนับสนุนน้อยที่สุด 3%

Friedman Test : Ranks	
Algorithm	Mean Rank
Apriori	7.00
FUP	4.56
Border	5.89
Prelarge	4.11
Prob_Based($Prob_{EF}$ = 0.01)	1.89
Prob_Based($Prob_{EF}$ = 0.03)	2.50
Prob_Based($Prob_{EF}$ = 0.05)	2.00

ตารางที่ ข.32 การทดสอบฟริดแมนสำหรับชุดข้อมูลที่ 3 ด้วยค่าสนับสนุนน้อยที่สุด 5%

Friedman Test : Ranks	
Algorithm	Mean Rank
Apriori	7.00
FUP	4.89
Border	6.00
Prelarge	4.11
Prob_Based($Prob_{EF}$ = 0.01)	1.56
Prob_Based($Prob_{EF}$ = 0.03)	2.50
Prob_Based($Prob_{EF}$ = 0.05)	1.94

ตารางที่ ข.33 การทดสอบทางสถิติสำหรับสำหรับชุดข้อมูลที่ 3 ด้วยค่าสนับสนุนน้อยที่สุด 1%,3%, และ 5%

Test Statistics[a]			
ค่าสนับสนุนน้อยที่สุด	1%	3%	5%
N	9	9	9
Chi-Square	51.905	45.353	50.934
df	6	6	6
Asymp. Sig.	.000	.000	.000
a. Friedman Test			

จากผลการทดสอบที่ทราบว่าเวลาเฉลี่ยของอัลกอริทึมแตกต่างกัน สามารถนำมาทดสอบแบบจับคู่โดยวิธี Wilcoxon Sign Rank test เพื่อให้ทราบว่าอัลกอริทึมใดแตกต่างกัน โดยในการทดสอบจะทำการเปรียบเทียบความแตกต่างของเวลาที่ได้จากการค้นหาของอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นเทียบกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์และพรีลาจก์ คือ โดยกำหนดค่าค่าความน่าจะเป็นที่ไอเทมจะกลายมาเป็นฟรีเควนท์ไอเทมเซต ($Prob_{EF}$) จำนวน 3 ค่าคือ $Prob_{EF}$ คือ 0.01, 0.03, 0.05 เมื่อทำการทดลองที่ค่าสนับสนุนน้อยที่สุด x% ซึ่ง x = 1, 3 และ 5 โดยกำหนดสมมติฐานในการทดสอบดังนี้คือ

1. เปรียบเทียบอัลกอริทึมอะพริโอริและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น

H_0: เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด x% ของอัลกอริทึมอะพริโอริและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นไม่แตกต่างกัน

H_1 : เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด x% ของอัลกอริทึมอะพริโอริและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นแตกต่างกัน

2. เปรียบเทียบอัลกอริทึมเอฟยูพีและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น

H_0: เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด x% ของอัลกอริทึมเอฟยูพีและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นไม่แตกต่างกัน

H_1 : เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด x% ของอัลกอริทึมเอฟยูพีและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นแตกต่างกัน

3. เปรียบเทียบอัลกอริทึมบอร์เดอร์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น

H_0: เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด x% ของอัลกอริทึมบอร์เดอร์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นไม่แตกต่างกัน

H_1 : เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด x% ของอัลกอริทึมเอฟยูพีและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นแตกต่างกัน

4. เปรียบเทียบอัลกอริทึมพรีลาจก์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น

H_0: เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด x% ของอัลกอริทึมพรีลาจก์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นไม่แตกต่างกัน

H_1 : เวลาเฉลี่ยที่ได้จากการทดลองชุดข้อมูลที่ 1ที่ค่าสนับสนุนน้อยที่สุด x% ของอัลกอริทึมพรีลาจก์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นแตกต่างกัน

ผลการทดสอบเปรียบเทียบความแตกต่างของเวลาเฉลี่ยแสดงในตารางที่ ข.34 - ข.37 แสดงเปรียบเทียบเวลาเฉลี่ยในการค้นหากฎความสัมพันธ์ระหว่างอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นกับอัลกอริทึมต่างๆ คือ อัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์และพรีลาจก์เป็นคู่ โดยได้ทดสอบทำการเพิ่มขยายการค้นหาที่ค่าสนับสนุนน้อยที่สุด 1% ของชุดข้อมูลที่ 3 และกำหนดค่าความน่าจะเป็นที่ไอเทมจะกลายมาเป็นฟรีเควนท์ไอเทมเซตที่ 0.01 โดยผลที่ได้จากการทดสอบความแตกต่างของค่าเวลาเฉลี่ยโดยใช้ Asymp. Sig.(2-tailed) ซึ่งไม่ใช้ตารางสถิติ พบว่าค่า Asymp. Sig. คือ .008 ซึ่งมีค่าน้อยกว่าค่า α ที่กำหนดไว้ คือ .05 ดังนั้นจึงปฏิเสธ H_0 และยอมรับH_1

สรุปผลการทดสอบได้ว่าเวลาเฉลี่ยเป็นคู่ระหว่างอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์และพรีลาจก์เทียบกับอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นด้วยค่า $Prob_{EF}$= 0.01, 0.03 และ 0.05 ที่ค่าสนับสนุนน้อยที่สุด 1% แตกต่างกัน อย่างมีนัยสำคัญ .05

จากตารางที่ ข.38- ข.41 เป็นการทดสอบเปรียบเทียบค่าเฉลี่ยระหว่างอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นเทียบกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์และพรีลาจก์ คือ โดยกำหนดค่าค่าความน่าจะเป็นที่ไอเทมจะกลายมาเป็นฟรีเควนท์ไอเทมเซต ($Prob_{EF}$) จำนวน 3 ค่าคือ $Prob_{EF}$ คือ 0.01, 0.03, 0.05 ที่ค่าสนับสนุนน้อยที่สุด 3%พบว่าค่า Asymp. Sig. ของอัลกอริทึมอะพริโอริ, บอร์เดอร์และพรีลาจก์มีค่าเท่ากับ .008 และอัลกอริทึมเอฟยูพีมีค่า Asymp. Sig. เท่ากับ .015 ซึ่งมีค่าน้อยกว่าค่า α ที่กำหนดไว้ คือ .05 ดังนั้นจึงปฏิเสธ H_0 และยอมรับH_1สรุปผลได้ว่า เวลาเฉลี่ยระหว่างอัลกอริทึมอะพริโอริแตกต่างจากอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นที่ค่า $Prob_{EF}$= 0.01, 0.03 และ 0.05 ที่

ค่าสนับสนุนน้อยที่สุด 3% อย่างมีนัยสำคัญ .05 เช่นเดียวกับตารางที่ ข.42 – ข.45 ซึ่งเป็นการทดสอบเปรียบเทียบค่าเฉลี่ยระหว่างอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นเทียบกับอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์และพรีลาจก์ คือ โดยกำหนดค่าค่าความน่าจะเป็นที่ไอเทมจะกลายมาเป็นฟรีเควนท์ไอเทมเซต ($Prob_{EF}$) จำนวน 3 ค่าคือ $Prob_{EF}$ คือ 0.01, 0.03, 0.05 ที่ค่า สนับสนุนน้อยที่สุด 5%พบว่าค่า Asymp. Sig. ของอัลกอริทึมอะพริโอริ, เอฟยูพี,บอร์เดอร์และพรีลาจก์มีค่าเท่ากับ .008 ซึ่งมีค่าน้อยกว่าค่า α ที่กำหนดไว้คือ .05 ดังนั้นจึงปฏิเสธ H_0 และยอมรับH_1สรุปผลได้ว่า เวลาเฉลี่ยระหว่างอัลกอริทึมอะพริโอริแตกต่างจากอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นที่ค่า $Prob_{EF}$= 0.01, 0.03 และ 0.05 ที่ค่า สนับสนุนน้อยที่สุด 5% อย่างมีนัยสำคัญ .05

สรุปผลการทดสอบได้ว่าเวลาเฉลี่ยเป็นคู่ระหว่างอัลกอริทึมอะพริโอริ, เอฟยูพี, บอร์เดอร์และพรีลาจก์เทียบกับอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นด้วยค่า $Prob_{EF}$= 0.01, 0.03 และ 0.05 ที่ค่าสนับสนุนน้อยที่สุด 3% และ 5%แตกต่างกัน อย่างมีนัยสำคัญ .05

ตารางที่ ข.34 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมอะพริโอริและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นชุดข้อมูลที่ 3 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 1%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - Apriori		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.35 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมเอฟยูพีและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น ชุดข้อมูลที่ 3 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 1%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - FUP		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.36 เปรียบเทียบเวลาเฉลี่ยระหว่างอัลกอริทึมบอร์เดอร์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะชุดข้อมูลที่ 3 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 1%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - Border		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.37 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมพรีลาจก์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น ชุดข้อมูลที่ 3 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 1%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - Prelarge		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.38 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมอะพริโอริและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นชุดข้อมูลที่ 3 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 3%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - Apriori		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.39 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมเอฟยูพีและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น ชุดข้อมูลที่ 3 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 3%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - FUP		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.429[a]	-2.429[a]	-2.429[a]
Asymp. Sig. (2-tailed)	.015	.015	.015
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.40 เปรียบเทียบเวลาเฉลี่ยระหว่างอัลกอริทึมบอร์เดอร์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะชุดข้อมูลที่ 3 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 3%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - Border		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.41 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมพรีลาจก์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น ชุดข้อมูลที่ 3 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 3%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - Prelarge		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.42 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมอะพริโอริและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็นชุดข้อมูลที่ 3 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 5%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - Apriori		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.43 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมเอฟยูพีและอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น ชุดข้อมูลที่ 3 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 5%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - FUP		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.44 เปรียบเทียบเวลาเฉลี่ยระหว่างอัลกอริทึมบอร์เดอร์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะชุดข้อมูลที่ 3 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 5%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - Border		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

ตารางที่ ข.45 แสดงค่าทดสอบความแตกต่างระหว่างอัลกอริทึมพรีลาจก์และอัลกอริทึมสำหรับเพิ่มขยายการค้นหากฎความสัมพันธ์โดยใช้หลักความน่าจะเป็น ชุดข้อมูลที่ 3 ($Prob_{EF}$= 0.01, 0.03, 0.05) ที่ค่าสนับสนุนน้อยที่สุด 5%

Test Statistics[b]			
	Prob_Based ($Prob_{EF}$) - Prelarge		
	$Prob_{EF}$ = 0.01	$Prob_{EF}$ = 0.03	$Prob_{EF}$ = 0.05
Z	-2.666[a]	-2.666[a]	-2.666[a]
Asymp. Sig. (2-tailed)	.008	.008	.008
a. Based on positive ranks.			
b. Wilcoxon Signed Ranks Test			

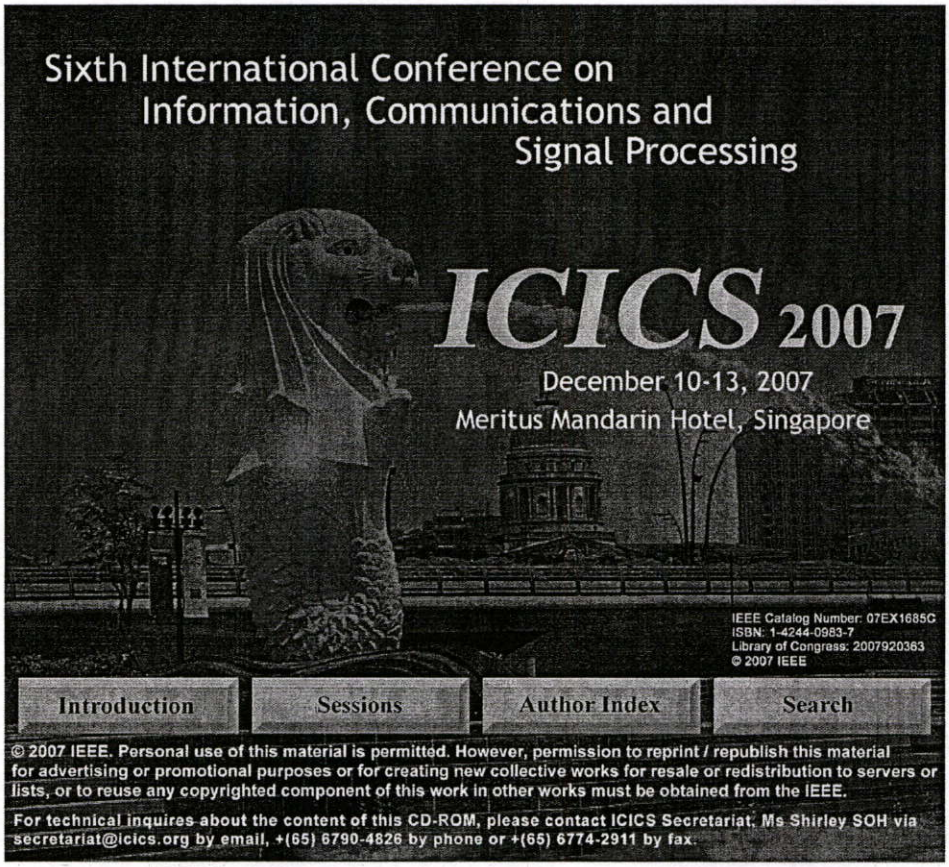
ภาคผนวก ค.

ผลงานวิจัยที่เกี่ยวข้องกับวิทยานิพนธ์และได้รับการตีพิมพ์

1. Amornchewin R. and Kreesuradej W., "Incremental association rule mining using promising frequent itemset algorithm", **In Proceeding 6th International Conference on Information, Communications and Signal Processing**, Dec. 10-13 2007, pp.1-5.

2. Amornchewin R. and Kreesuradej W. "Probability-Based Incremental Association Rule Discovery Algorithm" **International Symposium on Computer Science and Its Applications 2008,** 13-15 October 2008, pp.212-215.

3. Amornchewin R. and Kreesuradej W. "Mining Dynamic Databases using Probability-Based Incremental Association Rule Discovery Algorithm." **Journal of Universal Computer Science**, Vol. 15, no.12, 2009. pp. 2409-2428.



NANYANG TECHNOLOGICAL UNIVERSITY

IEEE

Incremental Association Rule Mining Using Promising Frequent Itemset Algorithm

Ratchadaporn Amornchewin
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok, 10520 Thailand
ramornchewin@yahoo.com

Worapoj Kreesuradej
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok, 10520 Thailand
worapoj@it.kmitl.ac.th

Abstract— **Association rule discovery is an important area of data mining. In dynamic databases, new transactions are appended as time advances. This may introduce new association rules and some existing association rules would become invalid. Thus, the maintenance of association rules for dynamic databases is an important problem. In this paper, promising frequent itemset algorithm, which is an incremental algorithm, is proposed to deal with this problem. The proposed algorithm uses maximum support count of 1-itemsets obtained from previous mining to estimate infrequent itemsets, called promising itemsets, of an original database that will capable of being frequent itemsets when new transactions are inserted into the original database. Thus, the algorithm can reduce a number of times to scan the original database. As a result, the algorithm has execution time faster than that of previous methods. This paper also conducts simulation experiments to show the performance of the proposed algorithm. The simulation results show that the proposed algorithm has a good performance.**

Keywords—**association rule, maintain association rule, incremental associatin rule**

I. INTRODUCTION

Data mining is one of the processes of Knowledge Discovery in Database (KDD) that is used for extracting information or pattern from large database. One major application area of data mining is association rule mining [1] that discovers hidden knowledge in database. The association rule mining problem is to find out all the rules in the form of X => Y, where X and Y $\subset$ I are sets of items, called itemsets. The association rule discovery algorithm is usually decomposed into 2 major steps. The first step is find out all large itemsets that have support value exceed a minimum support threshold and the second steps is find out all the association rules that have value exceed a minimum confidence threshold.

However, a database is dynamic when new transactions are inserted into the database. This may introduce new association rules and some existing association rules would become invalid. As a brute force approach, apriori may be reapplied to mining the whole dynamic database when the database has been changed. However, this approach is very costly even if small amount of new transactions is inserted into a database. Thus, the association rule mining for a dynamic database is an important problem. Several research works [7, 8, 9, 10, 11] have proposed several incremental algorithms to deal with this problem. Review of related works will be introduced in section 2.

In this paper, a new incremental algorithm, called promising frequent itemset algorithm, is introduced. The goal of this work is to solve the efficient updating problem of association rules after a nontrivial number of new records have been added to a database. Our approach introduces a promising frequent itemset for an infrequent itemset that has capable of being a frequent itemset after a number of new records have been added to a database. This can reduce a number of times to scan an original database. As a result, the algorithm has execution time faster than that of previous methods.

The remaining of this paper is organized as follows. We brief review of related works in Section 2. The Promising large itemset algorithm is described in Section 3. We evaluate the performance in Section 4. Finally, we conclude the work of this paper in section 5.

II. RELATED WORK

An influential algorithm for association rule mining is Apriori [2]. Apriori computes frequent itemsets in the large database through several iterations based on a prior knowledge. Each iteration has 2 steps. For each iteration with 2 steps, processes are join and prune step. For an frequent itemset, its support must be higher than a user-specified minimum support threshold. The association rule can be discoverd based on frequent itemsets that must be higher than user-specified minimum confidence.

For dynamic databases, several incremental updating techniques have been developed for mining association rules. One of the previous work for incremental association rule mining is FUP algorithm that was presented by Cheung et al [3]. FUP algorithm is the first incremental updating technique

1-4244-0983-7/07/$25.00 ©2007 IEEE

for maintenance association rules when new data are inserted to database. Based on the framework of Apriori algorithm, FUP computes frequent itemsets using large itemsets found at the previous iteration. The major idea of FUP is re-use frequent itemsets of previous mining to update with incremental database. The key performance of FUP is pruning technique to reduce the number of candidate set in update process. The extension algorithm of FUP is FUP2 [4] that is proposed to handle all update cases when database are added to, deleted from a database.

Ayan et al [5] present an algorithm called UWEP (Update With Early Pruning). UWEP follows the approaches of FUP and partition algorithm. It employs a dynamic look-ahead strategy in updating existing large itemsets by detecting and pruning superset of large itemsets in an original database that will no longer remain large in updated database. UWEP scans at most once in both original database and incremental database. UWEP generates smaller candidate set from the set of itemsets that are large both an original and incremental database.

Furthermore, negative border algorithm [6], an incremental mining algorithm based on FUP, reduces to scan original database and keeps track of large itemsets and negative border when transaction is added to or delete from database. Negative border consists of all itemsets that are candidates of the level-wise method. An itemset is in the negative border did not have enough support but all its subsets are frequent. If the negative border of large itemsets expands, this algorithm is required to full scan a whole database. This is the case because negative border algorithm does not cover all large itemsets in updated database.

III. PROMISING FREQUENT ITEMSET ALGORITHM

In an observation the itemset will be frequent itemset in updated database if it is member of large itemset in original database or incremental database. The main problem of incremental update is changing of frequent itemset that cause to re-execute from original database again.

In this paper we present the new idea to avoid scanning the original database. Then we compute not only frequent itemset but also compute itemset that may be potentially large in an incremental database called "Promising frequent Itemset".

An algorithm find all possible k-itemset of promising frequent itemset in original database. If member of frequent for each iteration is more than or equal to k-itemset. This idea is guarantee that promising frequent itemset algorithm are cover all frequent itemset that occur in updated database.Thus, updating the new transactions are quickly because it can use the information from the existing original database.

In this section we describe our algorithm into 2 subsection. In our approach, an original database is firstly mined and all frequent itemsets and promising frequent itemset. Secondly each incremental dataset in mined and updated to frequent and promising frequent itemset. The result of updating, some infrequent itemsets or new itemsets may be changed into frequent itemset.

A. Original database Discovery

A dynamic database may allow insert new transactions. This may not only invalidate existing association rules but also activate new association rules. Maintaining association rules for a dynamic database is an important issue. Thus, this paper proposes a new algorithm to deal with such updating situation. Our assumption for the new algorithm is that the statistics of new transactions slowly change from original transactions. According to the assumption, the statistics of old transactions, obtained from previous mining, can be utilized for approximating that of new transactions. Therefore, Support count of itemsets obtained from previous mining may slightly different from support count of itemsets after inserting new transactions into an original database that contains old transactions. The new algorithm uses maximum support count of 1-itemsets obtained from previous mining to estimate infrequent itemsets of an original database that will capable of being frequent itemsets when new transactions are inserted into the original database. With maximum support count and maximum size of new transactions that allow insert into an original database, support count for infrequent itemsets that will be qualified for frequent itemsets, i.e. min_pl, is shown in equation 1:

$$\min\_\sup_{DB} - \left(\frac{\max supp}{total\ size} \times inc\_size \right) \leq \min\_PL < \min\_\sup_{DB} \quad (1)$$

where $\min\_\sup_{(DB)}$ is minimum support count for an original database, maxsupp is maximum support count of itemsets, current size is a number of transaction of an original database and inc_size is a maximum number of new transactions.

Here, a promising frequent itemsets is defined as following definition:

Definition A promising frequent itemset is an infrequent itemset that satisfies the equation 1.

As an example, an original database shown in figure 1. has 10 transactions, i.e. |DB|=10. Then, three new transactions is inserted into the original database, i.e. |db| =3. Here, minimum support count for mining association rules is set to 4 (40 percent). From figure 1, maximum support count of 1-itemsets of the original database is 7. min_PL is computed as the follows:

$$\min\_PL = 4 - \left(\frac{7}{10} \times 3 \right) = 2 \quad (2)$$

According to equation 2, if any itemset has support count at least 2 but less than 4, then it will be promising frequent itemsets. Thus, the frequent 1- itemset is {A, B, C} and the promising frequent 1- itemset is {D, E}.

In this paper, apriori algorithm is applied to find all possible frequent k- itemsets and promising frequent k-itemsets. Apriori scans all transactions of an original database for each iteration with 2 steps processes are join and prune step. Unlike typical apriori algorithm, items in both frequent k- itemsets and promising frequent k-itemsets can be joined together in the join step. For a frequent item, its support count must be higher than

a user-specified minimum support count threshold and for a promising frequent item, its support count must be higher than min_PL but less than the user-specified minimum support count. As examples, figure 2. and 3. show the promising frequent and frequent 2- itemsets and the promising frequent and frequent 3- itemsets respectively.

TID	List of item
1	A, B, E
2	B, D
3	B,C
4	A, B,D
5	A, C
6	B, C
7	A, C
8	A, B, C, E
9	A, B, E
10	A,C

Itemset	support
A	7
B	7
C	6
D	2
E	3

Figure 1. Transaction data and candidate 1-itemsets

C2	support
AB	4
AC	4
AD	1
AE	3
BC	3
BD	2
BE	3
CD	0
CE	1
DE	0

L2	Support
AB	4
AC	4
PL2	support
AE	3
BC	3
BD	2
BE	3

Figure 2. candidate , frequent and promise frequent 2-itemset

C3	Support
ABC	1
ABE	3
ACE	1
BCD	0
BCE	0
BDE	0

PL3	support
ABE	3

Figure 3. candidate, frequent and promise frequent 3- itemset

B. Updating frequent and promising frequent itemsets

When new transactions are added to an original database, an old frequent k-item could become an infrequent k-item and an old promising frequent k-item could become a frequent k-item. This introduces new association rules and some existing association rules would become invalid. To deal with this problem, all k-items must be updated when new transactions are added to an original database. In this section, we explain how to update all old items.

The size of an updated database increases when new transactions are inserted into an original database. Thus, min_PL must be recalculated in order to associate with the new size of an updated database. min_PL $_{(update)}$ is computed as the follows:

$$\min\_PL_{DB\cup db} = \min\_\sup_{DB\cup db} - \left(\frac{\max supp}{total\ size} \times inc\_size \right) \quad (3)$$

Then, If any k-item has support count greater than or equal to min_sup$_{(DB\cup db)}$, this itemset is moved to a frequent k-item of an updated database. In the other case, if any k-item has support count less than min_sup$_{(DB\cup db)}$ but it is greater or equal to min_PL$_{(update)}$, this k-item is moved to a promise frequent itemset of an upated database. The following algorithms are developed to update frequent and promising frequent k-tems of an updated database.

The first algorithm, shown in figure 4, updates a frequent and a promising frequent 1-itemset. After obtaining support count of candidate 1-itemsets of an incremental database, the support count of the candidate 1-items is summed to that of the 1-items of an original database. Then, If any item has support count greater than or equal to min_sup$_{(DB\cup db)}$, this item is moved to a frequent 1-itemset of an updated database. In the other case, if any item has support count less than min_sup$_{(DB\cup db)}$ but it is greater or equal to min_PL$_{(update)}$, this item is moved to a promise frequent 1-itemset of an updated database. In addition, if any item is a new frequent item or a new promising frequent item, this item will be added to Temp1 set. Then, Temp1 is joined and pruned with both a promise frequent k-itemset and a frequent k-itemset to obtain Temp_newCk. The algorithm for joining and pruning items is shown in figure 5.

The k-itemsets of the Temp_newCk are scanned in an incremental database. A k-itemset of Temp_newCk can become a frequent itemset in an updated database only if the k-itemset of the Temp_newCk is a frequent itemset in an incremental database. Thus, if a itemset of the Temp_newCk has support count greater than or equal to min_sup(db), the item is moved to an estimated frequent k-itemset. Similarly, a k-itemset of Temp_newCk can become a promising frequent itemset in an updated database only if the k-itemset of the Temp_newCk is a frequent itemset in an incremental database. Thus, if a k-itemset of Temp_newCk has support count less than min_sup(db) but greater or equal to min_PL(update) or min_PL(DB), the k-itemset is moved to an estimated promising frequent k-itemset. Then, both the estimated promise frequent k-itemsets and the estimated frequent k-itemsets are added to Temp_scanDB. Figure 6 shows updating k-itemset algorithm for k>=2.

As the last phase for the incremental algorithm, the k-items of Temp_scanDB is scanned in an original database to update their support count. Like previous cases, If any k-item has support count greater than or equal to min_sup$_{(DB\cup db)}$, this k-item is moved to a frequent k-itemset and if any k-item has support count less than min_sup$_{(DB\cup db)}$ but it is greater or equal to min_PL$_{(update)}$, this k-item is moved to a promise frequent k-itemset. The algorithm for finding support count of k>=2 itemsets shows in figure 7.

Algorithm 1 Updating frequent and promising frequent 1-itemset

Input :
(1) L^1_{DB}: the set of all frequent 1-itemset in original database,
(2) PL^1_{DB} : the set of all promising frequent 1-itemset original database,
(3) C^1_{DB} : candidate 1-itemset of original database,
(4) C^1_{db}: candidate 1-itemset of incremental database.

Output :
(1) $L^1_{(DB \cup db)}$: frequent itemset in updated database,
(2) $PL^1_{(DB \cup db)}$: promising frequent itemset in updated database,
(3) new frequent itemsets : new frequent itemset in updated database ,
(4) new promising frequent itemsets : new promising frequent itemset in updated database
(5) Temp_newCk : new candidate 2-itemset in updated database

```
1   C^1_db = all 1-itemsets in db with support >0
2   k=1
3   While C^1_db > 0 do
4   For each X ∈ C^1_DB do
5      X.support_(DB∪db) = X.support_DB + X.support_db
6      If (X ∉ L^1_DB or X ∉ PL^1_DB ) and
7        (X.support_(DB∪db) ) >= min_sup_(DB∪db) Then
8         Add X to L^1_(DB∪db)
9         Add X to temp1
10     For each X ∈ L^1_DB do
11       If X.support_(DB∪db) >= min_sup_(DB∪db) Then
12         Add X to L^1_(DB∪db)
13       Else
14         If X.support_(DB∪db) >= min_PL_(DB∪db) Then
15           Add X to PL^1_(DB∪db)
16     For each X ∈ PL^1_DB do
17       If X.support_(DB∪db) >= min_sup_(DB∪db) Then
18         Add X to L^1_(DB∪db)
19         Add X to temp1
20       Else
21         If X.support_(DB∪db) >= min_PL_(DB∪db) Then
22           Add X to PL^1_(DB∪db)
23  For each X ∉ C^1_DB do
24     Add X to C^1_(DB∪db)
25     If X.support_db >= minsup_(DB∪db)  (new item in db) Then
26        Add X to L^1_(DB∪db)
27        Add X to temp1
28     Else
29       If X.support_(db) >= min_PL_(DB∪db) Then
30          Add X to PL^1_(DB∪db)
31          Add X to temp1
32   If temp1 ≠ {} Then
33     Y ∈ temp1
34     C^2_(DB∪db) (new) = gen_newcandidate (Y)
35     Clear temp1
36     Add C^2_(DB∪db) (new) to Temp_newCk
37  k=k+1
```

Figure 4. Updating frequent and promising frequent 1-itemset algorithm

Algorithm 2 Gen_newcandidate

Input :
(1)$L^k_{(DB \cup db)}$: frequent k-itemset in updated database,
(2) $PL^k_{DB \cup db)}$: promising k-itemset in updated database.
(3) Temp1 : new frequent k-itemset in updated database

Output :
(1) new C^{k+1}: new candidate k+1-itemset in updated database.

```
1   If k <= (length(L) + length(PL))
2     For each Y ∈ Temp1
3       C^(k+1) (new) = Y *(L^k_(DB ∪db))∪PL^k_(DB ∪db))
4     For c ∈ C^(k+1) (new)
5       Delete c from C^(k+1)_DB(new) if all subset of c is in L^k or PL^k
```

Figure 5. Generating new candidate itemset algorithm

Algorithm 3 Update frequent and promising frequent itemsets for k>= 2 itemset

Input :
(1) L^k_{DB} : frequent k-itemset in original database,
(2) $PL^k_{DB,}$: promising frequent k-itemset in original, database
(3) Temp_newCk : new candidate k-itemset in updated database.

Output :
(1) $L^k_{(DB \cup db)}$: frequent k-itemset in updated database,
(2) $PL^k_{DB \cup db)}$: Promising frequent k-itemset in updated database,
(3) Temp_scanDB : estimated frequent k-itemset and estimated promising frequent k-itemset in updated database
(4) Temp1 : new estimated frequent k-itemset and new estimated promising frequent k-itemset in updated database
(5) Temp_newCk : new candidate k+1-itemset in updated database.

```
1   k=2
2   While k <= (length(L^k) +length(PL^k)) do
3    Scan db for ∀(L^k) , ∀(PL^k) and ∀(items) ∈ Temp_newCk
4      X.support_(DB∪db) = X.support_DB + X.support_db
5    For each X ∈ L^k_DB do
6       If X.support_(DB∪db) >= min_sup_(DB∪db) Then
7        Add X to L^k_(DB∪db)
8       Else
9       If X.support_(DB∪db) >= min_PL_(DB∪db) Then
10        Add X to PL^k_(DB∪db)
11   For each X ∈ PL^k_DB do
12      If X.support_(DB∪db) >= min_sup_(DB∪db) Then
13        Add X to L^k_(DB∪db)
14        Add X to temp1
15      Else
16        If X.support_(DB∪db) >= min_PL_(DB∪db) Then
17          Add X to PL^1_(DB∪db)
18   For each Y ∈ Temp_newCk do
19    If Y.support_(db) >= min_sup_(db) Then
20       Add Y to Temp_scanDB(L^k_(DB∪db))
21       Add Y to Temp1(L^k_(DB∪db))
22    Else
23      If Y.support_(db) >= (min_PL_(DB∪db))
24        or Y.support_(db) >= min_PL_(DB) )Then
25         Add Y to Temp_scanDB(PL^k_(DB∪db))
26         Add Y to Temp1(PL^k_(DB∪db))
27   Clear Temp_newCk
28   If Temp1 ≠ {} Then
29      Y ∈ Temp1
30      C^(k+1)_(DB∪db) (new) = gen_newcandidate (Y)
31      Clear Temp1
32      Add C^(k+1)_(DB∪db) (new) to Temp_newCk
33   k=k+1
     If Temp_scanDB ≠ {} Then Find_SuppcountDB
```

Figure 6. Update k >= 2 itemset algorithm

Algorithm 4 Find_SuppcountDB

Input :

(1) $L^k_{(DB\cup db)} \in$ Temp_scanDB : Estimated frequent k-itemset,

(2) $PL^k_{(DB\cup db)} \in$ Temp_scanDB : Estimated promising frequent k-itemset

Output :

(1) $L_{(DB\cup db)}$: frequent k-itemset in updated database,

(2) $PL_{(DB\cup db)}$: promising frequent k-itemset in updated database

```
1   For each W ∈ Temp_scanDB
2     Scan DB for W
3     W.support(DB∪db) = W.supportDB + W.supportdb
4       If W.support(DB∪db) >= min_sup(DB∪db) Then
5         Add W to L^k(DB∪db)
6       Else
7         If W.support(DB∪db) >= min_PL(DB∪db) Then
8           Add W to PL^k(DB∪db)
9     Clear Temp_scamDB
```

Figure 7. Finding support count in an original database algorithm

IV. EXPERIMENT

To evaluate the performance of promising frequent algorithm, the algorithm is implemented and tested on a PC with a 2.8 GHz Pentium 4 processor, and 1 GB main memory. The experiments are conducted on a synthetic dataset, called T10I4D10K. The technique for generating the dataset is proposed by Agrawal and etc. [1]. The synthetic dataset comprises 20,000 transactions over 100 unique items, each transaction has 10 items on average and the maximal size itemset is 4

Firstly, the proposed algorithm is used to find association rules from an original database of 10,000 transactions. Then, several sizes of incremental databases, i.e. 10%, 20%, 30%, 40% and 50% of the original database, are added to the original database. For comparison purpose, FUP algorithm is also used to find association rules from the same original database and the same incremental databases. The experimental results with various minimum support thresholds are shown in Table1 and Figure 8. From the results, the proposed algorithm has better running time than that of FUP algorithm.

TABLE I. EXECUTION TIME WITH VARYING SIZE OF INCREMENTAL DATABASE

Min_sup	Algorithm	Execution time (sec.) Percent of Incremental database size				
		10%	20%	30%	40%	50%
4%	Promising Frequent Itemset	185.2	546	1047	1563.4	1935.5
	FUP	2521.5	5012	7110.7	9996.3	11539
5%	Promising Frequent Itemset	191.26	430.11	632.06	855.39	1048
	FUP	2023.6	4162.4	5970.4	8678.1	10681
6%	Promising Frequent Itemset	93.562	180.41	304.56	366.78	452.7
	FUP	1680.5	3868	5446.8	6889.5	9516.3
7%	Promising Frequent Itemset	52.985	98.296	463.22	1808.4	2147.1
	FUP	1350.6	2723.7	4873.3	5972.3	8856.3

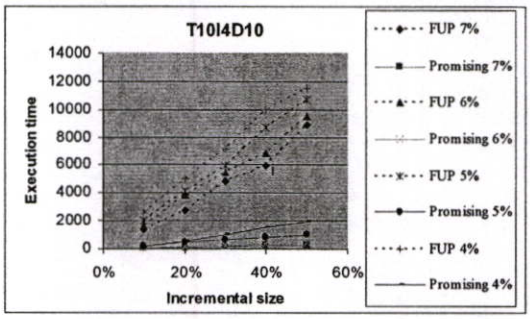


Figure 8. Execution Time comparison

V. CONCLUSIONS

We have proposed promising frequent algorithm for incremental association rule mining. Assuming that the two thresholds, minimum support and confidence, do not change, the promising frequent algorithm can guarantee to discover frequent itemsets. From the experiment, our algorithm has better running time than that of FUP algorithm. In the future, further researches and experiments on the proposed algorithm will be presented.

REFERENCES

[1] R. Agrawal., T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," In Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (A CM SIGMOD '93), Washington, USA,May 1993.

[2] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules." In Proceedings of 20 th Intl Conf. on Very Large Databases (VLDB'94), pages487-499,Santiago, Chile, 1994.

[3] D. W. Cheung, J. Han, V. T. Ng, and C. Y. Wong, "Maintenance of discovered association rules in large databases: An incremental updating technique,"In 12th IEEE International Conference on Data Engineering,1996.

[4] D. W. Cheung, S.D. Lee, B. Kao, "A General incremental technique for maintaining discovered association rules," In Proceedings of the 5 th Intl. Conf. on Database Systems for Advanced Applications (DASFAA'97), Melbourne, Australia, April 1997.

[5] N.F. Ayn, A.U. Tansel, and E. Arun, "An efficient algorithm to update large itemsets with early pruning." Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, August 1999.

[6] S. Thomas, S. Bodagala, K. Alsabti, and S. Ranka, "An efficient algorithm for the incremental updation of association rules in large databases," In Proceedings of the 3rd Intl. Conf. on Knowledge Discovery and Data Mining (KDD'97), New Port Beach, California,1997.

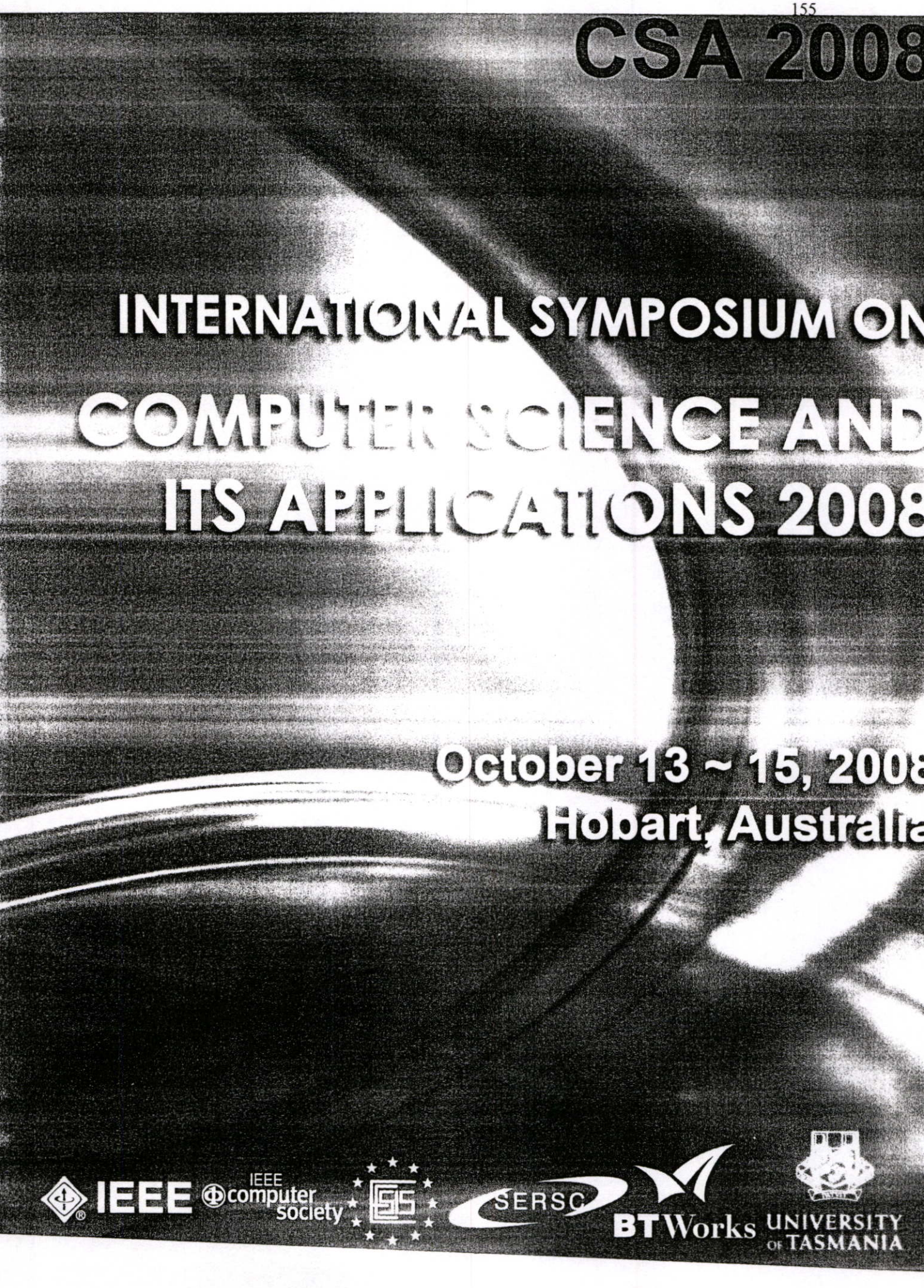
[7] C. H. Lee, C. R. Lin , and M. S. Chen, "Sliding-Window Filtering: An Efficient Algorithm for Incremental Mining ," ACM , 2000

[8] C.C. Chang, Y.C. Li and J.S. Lee, "An efficient algorithm for incremental mining of association rules," Proceedings of the 15th international workshop on research issues in data engineering: stream data mining and applications (RIDE-SDMA'05) ,IEEE, 2005

[9] A. A. Veloso et al., "Mining frequent itemsets in evolving databases," In Proc. 2nd SIAM Intl. Conf. on Data Mining, Arlington, VA, Apr. 2002

[10] K.L. Lee, G. Lee and A. L.P. Chen, "Efficient Graph-based algorithm for discovering and maintaining knowledge in large database," Proceedings of the third pacific-asia conference on methodologies for knowledge discovert and data mining, April 1999.

[11] N. L. Sarda and N. V. Srinivas, "An adaptive algorithm for incremental. mining of association rules," In Proc. 9th Intl.Workshop on Database and Expert System Applications,Vienna, Austria, pp. 240-245, Aug 1998.


CSA 2008
INTERNATIONAL SYMPOSIUM ON
COMPUTER SCIENCE AND
ITS APPLICATIONS 2008
October 13 ~ 15, 2008
Hobart, Australia
IEEE
IEEE computer society
SERSC
BT Works
UNIVERSITY OF TASMANIA

Probability-based incremental association rule discovery algorithm

Ratchadaporn Amornchewin
Faculty of Information Technology
King Mongkut's Institute of Technology
Ladkrabang
Bangkok, 10520 Thailand
ramornchewin@yahoo.com

Worapoj Kreesuradej
Faculty of Information Technology
King Mongkut's Institute of Technology
Ladkrabang
Bangkok, 10520 Thailand
worapoj@it.kmitl.ac.th

Abstract

In dynamic databases, new transactions are appended as time advances. This may introduce new association rules and some existing association rules would become invalid. Thus, the maintenance of association rules for dynamic databases is an important problem. In this paper, probability-based incremental association rule discovery algorithm is proposed to deal with this problem. The proposed algorithm uses the principle of Bernoulli trials to find expected frequent itemsets. This can reduce a number of times to scan an original database. This paper also proposes a new updating and pruning algorithm that guarantee to find all frequent itemsets of an updated database efficiently. The simulation results show that the proposed algorithm has a good performance.

1. Introduction

Data mining is one of the processes of Knowledge Discovery in Database (KDD) that is used for extracting information or pattern from large database. One major area of data mining is association rule mining [1] that discovers hidden knowledge in database. The association rule mining problem is to find out all the rules in the form of $X \Rightarrow Y$, where X and $Y \subset I$ are sets of items, called itemsets. The association rule discovery algorithm is usually decomposed into 2 major steps. The first step is find out all large itemsets that have support value exceed a minimum support threshold and the second steps is find out all the association rules that have value exceed a minimum confidence threshold.

However, a database is dynamic when new transactions are inserted into the database. This may introduce new association rules and some existing association rules would become invalid. As a brute force approach, apriori algorithm may be applied to mining a whole dynamic database when the database has been changed. However, this approach is very costly even if small amount of new transactions is inserted into a database. Thus, the association rule mining for a dynamic database is an important problem. Several research works [6, 7, 8, 9] have proposed several incremental algorithms to deal with this problem. Review of related works will be introduced in section 2.

In this paper, a new incremental algorithm, called probability-based incremental association rule discovery, is introduced. The goal of this work is to solve the updating problem of association rules after a number of new records have been added to a database. Based on probabilistic approach, our incremental algorithm predicts infrequent itemsets that have capable of being frequent itemsets after a number of new records have been added to a database. That infrequent itemsets is called expected frequent itemsets. Our algorithm can reduce a number of times to scan an original database. As a result, the algorithm has execution time faster than that of previous methods.

2. Previous work

An influential algorithm for association rule mining is Apriori [2]. Apriori computes frequent itemsets in a large database through several iterations based on a prior knowledge. Each iteration has 2 steps which are a joining step and a pruning step. For a frequent itemset, its support must be higher than a user-specified minimum support threshold. The association rule can be discoverd based on frequent itemsets.

For dynamic databases, several incremental updating techniques have been developed for mining association rules. One of the previous works for incremental association rule mining is FUP algorithm that was presented by Cheung et al [3]. The major idea of FUP is re-using frequent itemsets of previous

mining to update with frequent itemsets of an incremental database based on the concepts of Apriori.

Furthermore, negative border approach is presented by Toivonen [5], Thomas et al [6] and Feldman et al [8]. The negative border approach is an incremental mining algorithm based on FUP. The border itemset is not a frequent itemset but all its proper subsets are frequent itemsets. The approach need to keep a large number of border itemsets in order to reduce scanning times of an original database.

To reduce memory space, Hong et al [9] and Amornchewin et al[10] propose a new approach. The approach maintains both frequent itemsets and expected frequent itemsets. An expected frequent itemset is not a frequent itemset but is expected to become a frequent itemset when a new database is added to an original database. In order to guarantee that all frequent itemsets can be found when a new database is added to an original database, the approach can only allow very small size of an incremental database to insert into an original database.

Similarly, the proposed method in this paper also keeps not only frequent itemsets but also expected frequent itemsets from an original database. Unlike the previous works, this paper proposes a new technique for predicting expected frequent itemsets. Here, the principle of Bernoulli trials is used to predict the expected frequent itemsets. The expected frequent itemsets obtained from the proposed technique has lesser members than the border itemsets and the expected frequent itemsets obtained from the previous technique. This work also proposes a new updating algorithm that guarantee to find all frequent itemsets of a dynamic database efficiently.

3. Probability-based Incremental Association Rule Discovery Algorithm

When a dynamic database is inserted new transactions, not only some existing association rules may be invalidated but also some new association rules may be discovered. This is the case because frequent itemsets can be changed after inserting new transactions into a dynamic database. Therefore, an association rule discovery algorithm for a dynamic database has to maintain frequent itemsets when new transactions are inserted into the dynamic database.

The task of an association rule discovery algorithm for a dynamic database can be divided into three tasks. The first task is to update support count of existing frequent itemsets. The second task is to prune existing frequent itemsets that have support count below a minimum support threshold after updating the database. The third task is to discover new frequent itemsets that have support count equal or above a minimum support threshold after updating the database.

In this section, we describe our algorithm into 2 subsections. Firstly, probability-based expected frequent itemsets is presented. Secondly, updating frequent and expected frequent itemsets is introduced.

3.1 Probability-Based Expected Frequent Itemsets

For our algorithm, an original database, which is a database before being inserted new transactions, is firstly mined to find all frequent itemsets that satisfy a minimum support count, denoted $k_{original}$. Furthermore, the proposed algorithm also predicts and keeps expected frequent itemsets that may become frequent itemsets if new transactions are inserted into the original database.

Our assumption for the new algorithm is that the statistics of new transactions slightly change from original transactions and the maximum number of new transactions that be allowed to insert into an original database is available. According to the first assumption, the statistics of old transactions, obtained from previous mining, can be utilized for approximating that of new transactions. Therefore, the new algorithm uses support count of itemsets obtained from previous mining to approximate the probability of infrequent itemsets in an original database that may be capable of being frequent itemsets when new transactions are inserted into the original database.

Here, the process of inserting m transactions into an original database of n transaction can be considered as (m+n) Bernoulli trials, which are (m+n) sequence of identical trials. Each itemset has its probability of appearing in a transaction, denoted by p, i.e., the probability of success. According to the principle of Bernoulli trials, the probability of the number of an itemset to appearing in (n+m) transactions, denoted by P(x), can be found by the following equation:

$$P(x)=\binom{n+m}{x}\cdot p^{x}\cdot(1-p)^{n+m-x} \quad ,$$

where p is the probability of an itemset appearing in a transaction, m is a number of new transactions, and n is a number of transactions of an original database.

Thus, if k is a minimum support count after inserting new transactions into an original database, the probability of an itemset to be a frequent intemset in an updated database can be obtained as the following equations:

$$P\left(x \geq k\right)_{item} = 1 - P\left(x < k\right)_{item} \qquad (1)$$

Here, an expected frequent itemset is an itemset that is not a frequent itemset but has its probability to be a frequent intemset greater than $Prob_{pl}$. $Prob_{pl}$ is a threshold constant specified by users. $Prob_{pl}$ indicates the minimum confidence level that a promising

frequent itemset will be a frequent itemset after inserting new transaction into an original database.

3.2. Updating frequent and expected frequent itemsets

When new transactions are added to an original database, an old frequent k-itemset could become an infrequent k-itemset and an old expected frequent k-itemset could become a frequent k-itemset. This introduces new association rules and some existing association rules would become invalid. To deal with this problem, all k-itemsets must be updated when new transactions are added to an original database. The notation used in this section is given in Table1.

Table 1. The notation for Updating frequent and expected frequent itemsets algorithm

DB	Original database
db	Incremental Database
UP	Updated database
k	Number of itemset
σ	Minimum support
ρ	Minimum Expected Frequent
C_k	Candidate k-itemset
F_k	Frequent k-itemset
EF_k	Expected Frequent k-itemset

Here, a new updating frequent and expected frequent itemsets algorithm shown in Figure 1 is proposed in this paper. The algorithm consists of three phases. The first phase is updating 1- frequent and expected frequent itemsets, i.e. line1-3. The second phase is repeatedly updating the other frequent and expected frequent itemsets by using only an incremental database, i.e. line 6-11. The third phase is scanning an original database, i.e. line 13-22.

The First phase is updating 1- frequent and expected frequent itemsets. According to our propose, the 1-candidate itemsets of an updated database, i.e. C_1^{UP}, can be found by combining the 1-candidate itemsets of an original database, i.e. C_1^{DB}, with the 1-candidate itemsets of an incremental database, i.e. C_1^{db}. Then, the support count of C_1^{UP} can be updated by scanning only an incremental database. Then, the result of this phase is consist of 1-frequent and expected 1-itemsets of an updated database.

The second phase has 2 major steps which are a generating *k*- incremental candidate itemsets step and an updating support count of k- incremental frequent and k- incremental expected frequent itemsets step for *k* greater than or equal to 2. For *k*=2, the 2- incremental candidate itemsets are easily obtained by joining F_1^{UP} with F_1^{UP}. For *k*>2, the algorithm is firstly find *k*-candidate itemsets of an incremental database, i.e. C_k^{db}, by joining F_{k-1}^{db} with F_{k-1}^{db}. Similar to Apriori algorithm, the *k*- candidate itemsets of an incremental database can be the updated frequent itemsets, i.e. F_k^{UP}, only if the subsets of the *k*- candidate itemsets of an incremental database must be in the (*k*-1)- updated frequent. Thus, the *k*- incremental candidate itemsets, will keep only the k- candidate itemsets of an incremental database whose subsets of the *k*- candidate itemsets are in the (*k*-1) - updated frequent intemsets. This can prune the *k*- candidate itemsets of an incremental database that can't be the *k*- updated frequent intemsets.

Algorithm1 : Updating frequent and expected frequent itemsets Algorithm

Input : $DB, db, k, \sigma^{UP}, \rho^{UP}, \rho^{DB}, C_1^{DB}, F_1^{DB}, EF_1^{DB}$ *and their count*
Output : F_k^{UP}, EF_k^{UP}

1. $k = 1$
2. *if* $k = 1$
3. *Update* 1−*itemset*
4. $k = k + 1$
5. *else*
6. *for* $(k = 2; F_k^{UP} \neq \phi; k++)$ *do*
7. *Generate Candidate Itemset*
8. *Update* k − *itemset*(*return m*, *Temp_scanDB*)
9. *// m is the maximum itemset of Temp_scanDB*
10. $k = k + 1$
11. *end do*
12. *end if*
13. $k = 2$
14. *while* ($Temp\_scanDB_k \neq \phi$ *and* ($k \leq m$) *do*
15. *Scan Original Database for all* $X \in Temp\_scanDB_k$
16. *for all* $X \in Temp\_scanDB_k$ *do*
17. $c(X,UP) = c(X,DB) + c(X,db)$
18. *end do*
19. $F_k^{UP} = F_k^{UP} \cup \{X \mid X \in Temp\_scanDB_k \text{ and } c(X,UP) \geq \sigma^{UP}\}$
20. $EF_k^{UP} = EF_k^{UP} \cup \{X \mid X \in Temp\_scanDB_k \text{ and } \rho^{UP} \leq c(X,UP) < \sigma^{UP}\}$
21. $k = k + 1$
22. *end do*
23. *clear Temp_scanDB*

Figure 1. Updating frequent and expected frequent itmesets Algorithm

When any *k*-itemsets are not in the union set of the original *k*-frequent and the original *k*- expected frequent itemsets, but is in the k- incremental candidate itemsets, their support counts need to be specially updated. This is the case because their support counts obtained from an original database are not available. Here, their support counts in an original database are assumed to be equal to the sum of ρ^{DB} - 1 and their support counts from an incremental database. If any k-itemsets have support counts below updated min support count, i.e. σ^{UP}, the *k*-itemsets can't be the *k*-updated frequent itemsets. On the other hand, if any *k*-itemsets have support counts above or equal to an

updated min support count, the *k*-itemsets are likely to be the *k*-updated frequent itemsets. Thus, the *k*-itemsets, which have support counts above or equal to an updated min support count, are set aside for finding their true support counts from an original database.

At the third phase, an original database is scanned to find true support counts for the *k*-itemsets that are likely to be the *k*-updated frequent itemsets. The support counts of the likely *k*-updated frequent itemsets are found and updated by scanning an original database. Then, all *k*-updated frequent itemsets and *k*-updated expected frequent itemsets are found.

4. Experiments

To evaluate the performance of probability-based incremental association rules discovery algorithm, the algorithm is implemented and tested on a PC with a 2.8 GHz Pentium 4 processor, and 1 GB main memory. The experiments are conducted on a synthetic dataset, called T10I4D10K. The technique for generating the dataset is proposed by Agrawal and etc. [1]. The synthetic dataset comprises 110,000 transactions over 100 unique items, each transaction has 10 items on average and the maximal size itemset is 4.

Firstly, the proposed algorithm with $Prob_{pl}$ = 0.06 used to find association rules from an original database of 10,000 transactions. Then, the same sizes of incremental databases, i.e. 10% of the original database, are added to the original database for 100 trials. For comparison purpose, FUP and Borders algorithm are also used to find association rules from the same original database and the same incremental databases. Figure 2 and Table 2 show the average of execution time for FUP, Borders and our approach. The results also show that the proposed algorithm has much better running time than that of FUP.

Table 2. Average of Execution time

	Average of Execution time for 100 trials		
Min_sup (%)	FUP	Borders	Probability-Based
3%	3195.6939	2660.9297	2354.24533
4%	2274.4477	2087.8636	1931.19147

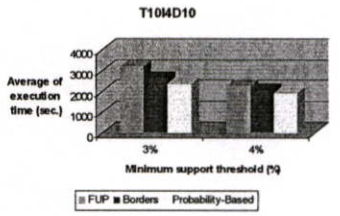


Figure 2. The execution time of FUP, Borders and the proposed algorithm

5. Conclusion

We have proposed probability-based incremental association rule discovery algorithm. Assuming that the two thresholds, minimum support and confidence, do not change, the algorithm can guarantee to discover all frequent itemsets. From the experiment, our algorithm has better running time than that of FUP and Borders algorithm. In the future, further researches and experiments on the proposed algorithm will be presented.

6. References

[1] R. Agrawal., T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases", In Proceeding of the ACM SIGMOD Int'l Conf. on Management of Data (A CM SIGMOD '93), Washington, USA,May 1993, pp. 207-216.

[2] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules", In Proceedings of 20 th Intl Conf. on Very Large Databases (VLDB'94), pages487-499, Santiago, Chile, September 1994, pp. 478 -499.

[3] D. Cheung, J. Han, V. Ng, and C. Y. Wong, "Maintenance of discovered association rules in large databases: An incremental updating technique", In 12th IEEE International Conference on Data Engineering, February 1996, pp. 106-114.

[4] D. W. Cheung, S.D. Lee, B. Kao, "A General incremental technique for maintaining discovered association rules" , In Proceedings of the 5 th Intl. Conf. on Database Systems for Advanced Applications (DASFAA'97), Melbourne, Australia, April 1997, pp. 185-194.

[5] H. Toivonen. "Sampling Large Databases for Association Rules", Proceeding of the 22th International conference on Very Large Data Bases, September 1996,pp. 134-145.

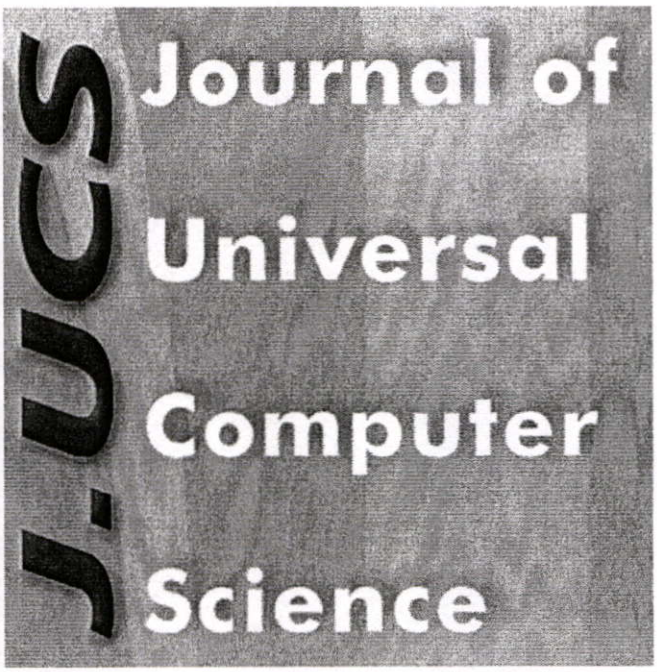
[6] S. Thomas, S. Bodagala, K. Alsabti, and S. Ranka, "An efficient algorithm for the incremental updation of association rules in large databases" , In Proceedings of the 3rd Intl. Conf. on Knowledge Discovery and Data Mining (KDD'97), New Port Beach, California, August 1997, pp. 263-266.

[7] C.C. Chang, Y.C. Li and J.S. Lee, "An efficient algorithm for incremental mining of association rules", Proceedings of the 15th international workshop on research issues in data engineering: stream data mining and applications (RIDE-SDMA'05), IEEE, 2005.

[8] R. Feldman, Y. Aumann, and O. Lipshtat. "Borders: An efficient algorithm for association generation in dynamic databases",Journal,Intelligent Information System, 1990, pp. 61-73.

[9] T.P. Hong C.Y. Wang and Y.H. Tao, "A new incremental data mining algorithm using pre-large itemsets", Journal, Intelligent Data Analysis, Vol. 5, No.2, pp. 111-129, 2001.

[10] R. Amornchewin and W. Kreesuradej, "Incremental association rule mining using promising frequent itemset algorithm", In Proceeding 6th International Conference on Information, Communications and Signal Processing, Dec. 10-13 2007, pp.1-5.



A publication of

Graz University of Technology and
Universiti Malaysia Sarawak

in cooperation with
Know-Center and **Campus02**

< enter

Unique Features of J.UCS

Imprint

J.UCS Journal of Universal Computer Science

Volume 15

Content of Issue 12

DOI: 10.3217/jucs-015-12

Journal of Universal Computer Science, vol. 15, no. 12 (2009), 2409-2428
submitted: 15/12/08, accepted: 25/6/09, appeared: 28/6/09 © *J.UCS*

Mining Dynamic Databases using Probability-Based Incremental Association Rule Discovery Algorithm

Ratchadaporn Amornchewin
(King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
ramornchewin@yahoo.com)

Worapoj Kreesuradej
(King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
worapoj@it.kmitl.ac.th)

Abstract: In dynamic databases, new transactions are appended as time advances. This paper is concerned with applying an incremental association rule mining to extract interesting information from a dynamic database. An incremental association rule discovery can create an intelligent environment such that new information or knowledge such as changing customer preferences or new seasonal trends can be discovered in a dynamic environment. In this paper, probability-based incremental association rule discovery algorithm is proposed to deal with this problem. The proposed algorithm uses the principle of Bernoulli trials to find expected frequent itemsets. This can reduce a number of times to scan an original database. This paper also proposes a new updating and pruning algorithm that guarantee to find all frequent itemsets of an updated database efficiently. The simulation results show that the proposed algorithm has better performance than that of previous work.

Keywords: Incremental Association Rule Discovery, Association Rule Discovery, Data mining
Categories: I.1.2, I.2.6

1 Introduction

Recent works in the field of databases have been used in business management, government administration, scientific, engineering data management, and many other applications. This explosive growth in data and databases has generated an urgent need for new techniques and tools that can intelligently and automatically transform the processed data into useful information and knowledge.

During the past few years, data mining has been considered as new techniques and tools for intelligently transforming the processed data into useful information and knowledge. Data mining provides the technique to analyze and convert mass volume of data and/or detect hidden patterns in data into valuable information. Data mining may be applied in an intelligence environment in a number of domains [Guo, Zhao 2008; Thuraisingham, Ceruti 2000; Du et al. 2008; Gong 2008; Juan et al. 2008 and Luhr et al. 2005].

An important technique of data mining is association rules mining [Argrawal 1993] which studies the buying behaviors of customers and thus improves the quality of business decisions. It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the databases. Association

rule mining is widely used in several business such as mobile data service environment in [Pawar, Aggarwal 2004], intelligent transportation system in [Juan et al. 2008], market basket analysis in [Brin 1997] and [Liu et al.1999], hospital information system in [Elfangary, Atteya 2008], etc.

The rules discovered from a database only reflect the current state of the database. However, in a dynamic database where new transactions are inserted frequently, association rules discovered in the previous database possibly no longer valid and interesting rules in the updated database. As a result, new business information such as changing customer preferences or new seasonal trends may not be discovered. To create an intelligent environment such that new business information can be discovered in a dynamic database, association rules algorithms should be capable of mining a dynamic database incrementally. Several research work [Ayan et al. 1999; Chang et al. 2005; Lee et al. 2001; Feldman et al. 1990; Ratchadaporn, Kreesuradej 2007; Sarda et al. 1998 and Veloso et al. 2002] have proposed several incremental algorithms to deal with this problem. The review of related work will be introduced in section 2.

In this paper, we propose a new incremental association rule discovery algorithm, called Probability-Based Incremental Association Rule Discovery Algorithm, for the environment of a dynamic database. The algorithm is capable of dynamically discovering new association rules when a number of new records have been added to a database. In our approach, we use the probability to predict infrequent itemsets that have capable of being frequent itemsets after a number of new records have been added to a database. That infrequent itemsets is called expected frequent itemsets. Our algorithm can reduce a number of times to scan an original database. As a result, the algorithm has execution time faster than that of previous methods.

2 Previous work

An influential algorithm for association rule mining is Apriori [Agrawal,Srikant 1993]. Apriori computes frequent itemsets in a large database through several iterations based on a prior knowledge. Each iteration has 2 steps which are a joining step and a pruning step. For a frequent itemset, its support must be higher than a user-specified minimum support threshold. The association rule can be discovered based on frequent itemsets. Based on Apriori algorithm, many new algorithms [Savesere et al. 1995] were designed with some modifications or improvements.

In dynamic databases, new transactions are appended as time advances. This may introduce new association rules and some existing association rules would become invalid. Thus, the maintenance of association rules for dynamic databases is an important problem. Maintaining association rules has been studied popularly in data mining. There are 2 main approaches to mining association rules for dynamic databases incrementally. The first approach assumes that association rules to be found are not stable over time. Recent patterns that represent new trends to be discovered are more interesting than old patterns. Thus, the first approach treats recent added transactions higher important than old transactions. The association rules obtained from the first approach are just the approximation of those obtained by re-running Apriori algorithm.

As the first approach, Zhang et al. [Zhang et al. 2003] proposed a weighting technique for discovering association rules in a dynamic database. The technique gives recent added transactions higher weight than old transactions. Then, frequent itemsets can be maintained incrementally by using a competitive model to promote infrequent itemsets to frequent itemset and degrade frequent itemset to infrequent itemset.

Similar to Zhang's work, Dudek et al. [Dudek et al. 2005] proposed a method of a time influence function in order to reflect that recent patterns are more interesting than old patterns. The time influence function gives recent added transactions higher weight than old transactions. In addition, this work also proposed using estimated support and confidence to deal with situation that new association rules are discovered only in new transactions but are not discovered only in old transactions. However, this work did not propose the method to estimate the estimated support and confidence.

Unlike the first approach, the second approach assumes that association rules to be found are stable over time. Thus, the second approach treats both old and new transactions as of equal importance. As a result, the association rules obtained from the second approach are the same as those obtained by re-running Apriori algorithm. According to Teng and Chen [Teng, Chen 2005], the second approach is categorized into Partition-based, Pattern growth and Apriori-based techniques.

Partition-based techniques partition a database into n partition and processes each partitions for an on going time variant database. For each frequent itemset must exist at least one of the n partitions. Sliding-window filtering (SWF) [Lee et al. 2001] is a partition algorithm for an incremental association mining. The concept of SWF is partitioning a transaction database into n partitions, P_1, P_2, ..., P_n and processing one by one. The filtering threshold is employed for selecting frequent itemset in each partition to deal with the candidate itemset generation. The key idea of SWF is to compute candidate 2-itemset as close to frequent 2-itemset as possible. SWF need one scan over the updated database for finding the frequent itemset from the candidate ones. Based on SWF, FI_SWF and CI_SWF [Chang, Yang 2003] are proposed to reduce the number of candidate itemsets of SWF.

Pattern-Growth techniques are basically based on FP-tree to discover association rules. Since FP-tree cannot be directly applied to the problem of incremental mining, two alternative forms of FP-tree, i.e., DB-tree [Ezeife, Su 2002] and Potential Frequent Pattern tree (PotFP-tree) [Ezeife, Su 2002], are proposed to solve the problem.

Apriori-based techniques, which are the most popular techniques, adopt the concepts of Apriori algorithm for incremental mining. Since the algorithm in this work can be classified as Apriori-based techniques, the more comprehensive reviews of Apriori-based techniques will be presented. The first incremental updating algorithm for maintaining association rules when new data are inserted into database is presented by Cheung et al. [Cheung et al. 1996]. FUP computes frequent itemsets using large itemsets found at previous iteration. The major idea of FUP is re-using frequent itemsets of previous mining to update with frequent itemsets of an increment database. At each iteration, the supports of the size-k frequent item sets of an original database are updated by scanning an increment database. As well as any k-frequent itemset of the increment database are updated by scanning the original database to

find the new frequent itemsets. As a result, FUP requires multiple-scan for maintaining frequent itemsets.

For enhance the efficiency of FUP algorithm, UWEP algorithm is presented by Ayan et al. [Ayan et al. 1999]. The major idea of UWEP algorithm is reducing the number of candidate itemset by pruning the supersets of an old frequent itemset in previous mining that no longer remain in an updated database with at most once scanning in original database.

To deal with the rescanning problem, negative border approach is presented by Toivonen [Toivonen 1996], Thomas et al. [Thomas et al. 1997] and Feldman et al. [Feldman et al. 1999].This approach maintains both frequent itemsets and border itemsets. The border itemset is not a frequent itemset but all its proper subsets are frequent itemsets. The approach need to keep a large number of border itemsets in order to reduce scanning times of an original database. Basically, the border-based algorithms start by scanning a new database. Then, the border-based algorithms update support counts of all frequent sets and border sets. Most updated frequent itemsets can be found not only from frequent itemsets but also from border itemsets. This can reduce scanning times of an original database. However, when new frequent itemsets are introduced as updated frequent itemsets, several database scanning is required to obtain support counts of the new frequent itemsets and their subsets. Adnan et al. [Adnan et al. 2005] shows that the execute time of the border-based algorithms can severely slower than that of Apriori when new frequent itemsets are introduced as updated frequent itemsets.

To reduce memory space, Tsai et al. [Tsai et al. 1999], Hong et al. [Hong et al. 2001] and Amornchewin and Kreesuradej [Amornchewin, Kreesuradej 2007] propose a new approach. The approach maintains both frequent itemsets and expected frequent itemsets. Based on FUP algorithm, Tsai et al. use not only support threshold but also the degree, called tolerance degree, to finding the expected frequent itemset. For awhile, Hong et al. use the 2 thresholds, upper support and lower support threshold, for mining frequent and expected frequent itmeset. Amornchewin and Kreesuradej proposes the idea for estimate expected frequent itemsets by using maximum support count of 1-itemsets obtained from prior mining. An expected frequent itemset is not a frequent itemset but is expected to become a frequent itemset when a new database is added to an original database. An expected frequent itemset is not a frequent itemset but is expected to become a frequent itemset when a new database is added to an original database. The expected frequent itemsets have lesser members than the border itemsets. As a result the approach uses lesser memory space than the border based approach. In order to guarantee that all frequent itemsets can be found when a new database is added to an original database. The approach of Hong et al. [Hong et al. 2001] and Amornchewin and Kreesuradej [Amornchewin, Kreesuradej 2007], can only allow very small size of an increment database to insert into an original database.

To deal with the problem that the previous approach can only allow very small size of an increment database to insert into an original database, a new algorithm, called Probability-based Incremental Association Rule Discovery Algorithm, is introduced by Amornchewin and Kreesuradej [Amornchewin, Kreesuradej 2008]. Similar to the previous approach, the new algorithm also keeps both frequent itemsets and expected frequent itemsets. However, the new algorithm introduces a new technique, which is based on the principle of Bernoulli trials, to estimate expected frequent

itemsets. The preliminary experiments show that the new technique can allow larger size of an increment database to insert into an original database than that of the previous approach. Moreover, the experiments also show that the new technique has execution time faster than that of previous methods.

This paper is the extension work of Probability-based Incremental Association Rule Discovery Algorithm, introduced by Amornchewin and Kreesuradej [Amornchewin, Kreesuradej 2008]. The paper provides more theoretical fundamentals of the algorithm and comprehensive experimental results than that of Amornchewin and Kreesuradej [Amornchewin, Kreesuradej 2008].

3 Probability-based Incremental Association Rule Discovery Algorithm

When a dynamic database is inserted new transactions, not only some existing association rules may be invalidated but also some new association rules may be discovered. This is the case because frequent itemsets can be changed after inserting new transactions into a dynamic database. The problem description is given as section 3.1. Then, we describe our algorithm into 2 subsections. Firstly, probability-based expected frequent itemsets is presented. Secondly, updating frequent and expected frequent itemsets is introduced.

3.1 Problem description

Let F be the set of all frequent itemsets in the original database, i.e. DB, σ be the minimum support threshold and |DB| be the number of transactions in DB. Assume that the support count of each frequent itemset in the original database is kept. After some updates, an increment database, i.e. db, is added into the original database DB, resulting in an updated database, i.e. UP. The numbers of transactions in db and UP are denoted |db| and |UP|, respectively, and the support counts of itemset X in DB, db and UP are c(X,DB), c(X,db), c(X,UP), respectively. With the same minimum support threshold σ, a frequent itemset X of DB remains a frequent itemset of UP if and only if its support in UP is greater than or equal to σ, i.e. $c(X,UP) \geq (|DB|+|db|)*\sigma$.

According to Tsai et al. [Tsai et al. 1999], an itemset, i.e. X, can be categorized into four cases:

Case 1: X is a frequent itemset in both DB and UP.

Case 2: X is a frequent itemset in DB and an infrequent itemset in UP.

Case 3: X is an infrequent itemset in DB and a frequent itemset UP.

Case 4: X is neither a frequent itemset in DB nor UP.

Obviously, the itemset of case 4 cannot change an association rule because the itemset is an infrequent itemset in both DB and UP. The itemset of case 1 and 2 can easily be discovered because updating support count of the frequent itemset, which is trivial tasks, requires only the frequency of the itemset from db. The task of discovering the itemset of case 3 is the hardest task because it requires to rescan an original database. Thus, how to efficiently discovery the itemset of case 3 is an

important problem. In the next subsection, a new algorithm to deal with this problem is proposed.

3.2 Predicting Expected Frequent Itemsets

The task of an association rule discovery algorithm for a dynamic database can be divided into three tasks. The first task is to update support count of existing frequent itemsets. The second task is to prune existing frequent itemsets that have support count below a minimum support threshold after updating the database. The third task is to discover new frequent itemsets that have support count equal or above a minimum support threshold after updating the database.

Updating support count of existing frequent itemsets and pruning existing frequent itemsets are trivial tasks. To perform updating and pruning tasks, the association rule discovery algorithm requires only frequency of itemsets from new transactions. Thus, an association rule discovery algorithm does not need to rescan an original database to perform the tasks.

Unlike the other tasks, the task for discovering new frequent itemsets requires not only frequency of itemsets from new transactions but also that from an original database. An association rule discovery algorithm need to rescan an original database to perform the tasks. Therefore, the discovering new frequent itemsets task is a nontrivial task for maintaining frequent itemsets

For our algorithm, an original database, which is a database before being inserted new transactions, is firstly mined to find all frequent itemsets that satisfy a minimum support count, denoted $k_{original}$. The proposed algorithm also predicts and keeps expected frequent itemsets that may become frequent itemsets if new transactions are inserted into the original database.

The process of inserting m transactions into an original database of n transaction can be considered as (m+n) Bernoulli trials, which are (m+n) sequence of identical trials. Each itemset has its probability of appearing in a transaction, denoted by $p_{itemset}$, i.e. the probability of success. According to the principle of Bernoulli trials, the probability of the number of an itemset to appearing in (n+m) transactions, denoted by $P(x)_{itemset}$, can be found by the following equation:

$$P(x)_{itemset} = \binom{n+m}{x} \cdot p_{itemset}{}^{x} \cdot (1 - p_{itemset})^{n+m-x} \quad ,$$

where $p_{itemset}$ is the probability of an itemset appearing in a transaction, m is a number of new transactions, and n is a number of transactions of an original database. Figure 1 shows the scheme of predicting expected frequent itemsets based on Bernoulli trials.

If k is a minimum support count after inserting new transactions into an original database, the probability of an itemset to be a frequent itemset in an updated database can be obtained as the following equation:

$$P(x \geq k)_{itemset} = 1 \quad P(x < k)_{itemset} \tag{1}$$

According to equation 1, $P(x < k)$ itemset can be found as the following equation:

$$P(x < k)_{itemset} = \sum_{x=0}^{k-1} \binom{n+m}{x} p_{itemset}^{\,x} (1 - p_{itemset})^{n+m-x} \qquad (2)$$

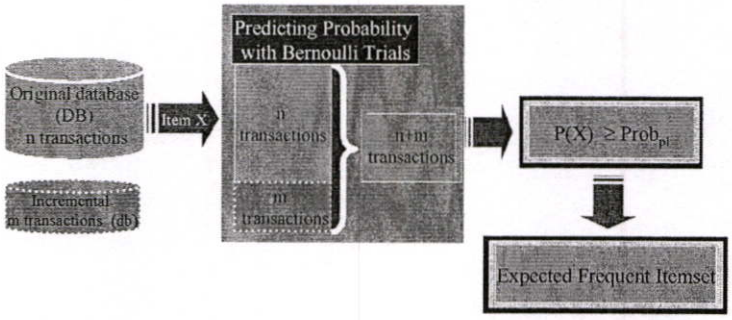


Figure1: The scheme of predicting expected frequent itemsets based on Bernoulli

Here, an expected frequent itemset is an itemset that is not a frequent itemset but has its probability to be a frequent itemset greater than $Prob_{pl}$. $Prob_{pl}$ is a threshold constant specified by users. $Prob_{pl}$ indicates the minimum confidence level that a promising frequent itemset will be a frequent itemset after inserting new transactions into an original database. The higher $Prob_{pl}$ is set, the lesser expected frequent itemsets are kept. As results, the algorithm may need more number of rescanning times in the original database when the algorithm performs the discovering new frequent item task.

From equation 2, the probability of success of an itemset, i.e., $p_{itemset}$, can be derived from the following equation:

$$p_{itemset} = \frac{c(itemset, DB) + c(itemset, db)}{|DB| + |db|}, \qquad (3)$$

where c(itemset, DB) is support counts obtained from an original database, c(itemset, db) is support counts obtained from an increment database, $|db|$ is the total number of transactions of an increment database, and |DB| is the total number of transactions of an original database.

According to the scheme of predicting expected frequent itemsets, an expected frequent itemset need to be obtained from an original database before an increment database is available. Therefore, the probability of success of an itemset has to approximate from an original database. Here, an original database, which has n transaction, is considered as a sample data of (n+m) transactions. The approximation of the probability of success of an itemset can be obtained as

$$\hat{p}_{itemset} = \frac{c(itemset, DB)}{|DB|}, \qquad (4)$$

where c(itemset, DB) is support counts obtained from an original database, i.e., DB, and $|DB|$ is the total number of transactions of an original database.

It is important to evaluate the accuracy of the approximation of the probability of success of an itemset. The following theorem gives a lower bound, i.e. ε, and maximum probability, i.e. δ, for an error of the approximation of the probability of success of an itemset.

Theorem 1[Mannila et al. 1994; Toivonen 1996] The probability that

$$error = \left| p_{itemset} - \hat{p}_{itemset} \right| > \varepsilon$$

at most δ. If the size of an original database satisfies the following equation:

$$|DB| \geq \frac{1}{2\varepsilon^2} ln \frac{2|UP|}{\delta},$$

where |DB| is the size of an original database and |UP| is the size of an updated database.

According to the theorem, if size of an original is sufficient large, $\hat{p}_{itemset}$ is a good approximation of $p_{itemset}$. Therefore, equation 2 can be rewritten as follows,

$$P(x < k)_{itemset} = \sum_{x=0}^{k-1} \binom{n+m}{x} \hat{p}_{itemset}^{\,x} (1 - \hat{p}_{itemset})^{n+m-x} \qquad (5)$$

As an example, an original database has 10 transactions, i.e. |DB| =10.Then, five new transactions are inserted into the original database, i.e. |db| =5. Here, minimum support count for mining association rules is set to 4 (40 percent). If we defined probability for a expected itemset is $Prob_{pl} = 0.10$, $P(x \geq k)$ is computed shown in figure 2.

TID	List of item
1	A, B, E
2	B, D
3	B, C
4	A, B, D
5	A, C
6	B, C
7	A, C
8	A, B, C, E
9	A, B, E
10	A, C

$$P(x \geq 6)_A = 1 - \sum_{x=0}^{5} \binom{15}{x} \frac{7}{10}^{x} \frac{3}{10}^{15-x} = 1$$

$$P(x \geq 6)_B = 1 - \sum_{x=0}^{5} \binom{15}{x} \frac{7}{10}^{x} \frac{3}{10}^{15-x} = 1$$

$$P(x \geq 6)_C = 1 - \sum_{x=0}^{5} \binom{15}{x} \frac{6}{10}^{x} \frac{4}{10}^{15-x} = 1$$

$$P(x \geq 6)_D = 1 - \sum_{x=0}^{5} \binom{15}{x} \frac{2}{10}^{x} \frac{8}{10}^{15-x} = 0.06$$

$$P(x \geq 6)_E = 1 - \sum_{x=0}^{5} \binom{15}{x} \frac{3}{10}^{x} \frac{7}{10}^{15-x} = 0.28$$

Figure 2: Transaction data and candidate 1-itemsets

From the example, the frequent 1- itemset is {A, B, C}. Item {E} is an expected frequent 1-itemset because its probability, which is equal to 0.28, is greater than $Prob_{pl}$.

The proposed algorithm generates candidate next k-itemsets by using both frequent k-itemsets and expected frequent k-itemsets. Firstly, new itemsets is obtained by union of frequent k-itemsets and expected frequent k-itemsets. Then, the candidate (k+1)-itemsets

is obtained by self joining the new itemset together. Both frequent (k+1)-itemsets and expected frequent (k+1)-itemsets can be found similar to that of k-itemsets. The example of generating candidate next k-itemsets is shown in figure 3.

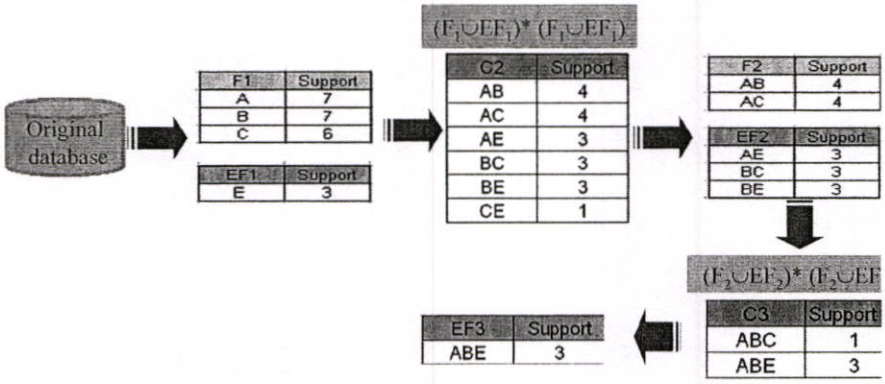


Figure 3: The example of generating candidate next k-itemsets

3.3 Updating frequent and expected frequent itemsets

When new transactions are added to an original database, an old frequent k-itemset could become an infrequent k-itemset and an old expected frequent k-itemset could become a frequent k-itemset. This introduces new association rules and some existing association rules would become invalid. To deal with this problem, all k-itemsets must be updated when new transactions are added to an original database. The notation used in this section is given in table1.

DB	Original database
db	Increment database
UP	Updated database
k	Number of itemset
σ	Minimum support
ρ	Minimum expected frequent
C_k	Candidate k-itemset
F_k	Frequent k-itemset
EF_k	Expected frequent k-itemset

Table 1: The notation for updating frequent and expected frequent itemsets algorithm

Here, a new updating algorithm shown in figure 4 is proposed in this paper. The algorithm consists of three phases. The first phase is updating 1- frequent and expected frequent itemsets, i.e. line1-3. The second phase is repeatedly updating the other frequent and expected frequent itemsets by using only an increment database, i.e. line 6-11. The third phase is scanning an original database, i.e. line 13-16.

Figure 5 shows the algorithm for updating 1- frequent and expected frequent itemsets. According to the algorithm, the 1-candidate itemsets of an updated database, i.e. C_1^{UP}, can be found by combining the 1-candidate itemsets of an original database, i.e. C_1^{DB}, with the 1-candidate itemsets of an increment database, i.e. C_1^{db}. Then, the support count of C_1^{UP} can be updated by scanning only an increment database, i.e. line 1-4. Then, the 1-frequent and expected itemsets of an updated database can be found as shown in line 5 and 6 respectively.

Algorithm1 : Main Algorithm

Input : $DB, db, k, \sigma^{UP}, \rho^{UP}, \rho^{DB}, C_1^{DB}, F_1^{DB}, EF_1^{DB}$ *and their count*

Output : F_k^{UP}, EF_k^{UP}

1. $k = 1$
2. *if* $k = 1$
3. *Update* $1-itemset$
4. $k = k + 1$
5. *else*
6. *for* $\left(k = 2; F_k^{UP} \neq \phi; k++\right)$ *do*
7. *Generate Candidate Itemset*
8. *Update* $k-itemset(return\ m, Temp\_scanDB)$
9. *// m is the* maximum *itemset of Temp_scanDB*
10. $k = k + 1$
11. *end do*
12. *end if*
13. $k = 2$
14. *while* $(Temp\_scanDBl_k \neq \phi$ *and* $(k \leq m)$ *do*
15. *Scan Original Database*$(Temp\_scanDB_k)$
16. $k = k + 1$
17. *end do*
18. *clear Temp_scanDB*

Figure 4: Main algorithm

The second phase has 2 major steps which are a generating k-increment candidate itemsets step and an updating support count of k-increment frequent and k-increment expected frequent itemsets step for k greater than or equal to 2. The algorithm for generating k- increment candidate itemsets for k greater than or equal to 2 is shown in figure 6. For k = 2, the 2-increment candidate itemsets are easily obtained by joining F_1^{UP} with F_1^{UP}, i.e. line 3. For k > 2, the algorithm is firstly find k-candidate itemsets of an increment database, i.e. C_k^{db}, by joining F_{k-1}^{db} with F_{k-1}^{db}, i.e. line 6. Similar to Apriori algorithm, the k-candidate itemsets of an increment database can be the

updated frequent itemsets, i.e. F_k^{UP}, only if the subsets of the k-candidate itemsets of an increment database must be in the (k-1)-updated frequent. Thus, the k- increment candidate itemsets, i.e. C_k^{new}, will keep only the k- candidate itemsets of an increment database whose subsets of the k- candidate itemsets are in the (k-1)-updated frequent itemsets, i.e. line 7-9. This can prune the k- candidate itemsets of an increment database that can't be the k- updated frequent itemsets.

Algorithm 2 : Updating 1-itemsets

Input : $DB, db, \sigma^{UP}, \rho^{UP}, C_1^{DB}, F_1^{DB}, EF_1^{DB}, C_1^{db}$ *and their count*

Output : $F_1^{UP}, EF_1^{UP}, C_1^{UP}$ *and their count*

1. *Scan db and find count* $c(X, db)$ *for all* $X \in C_1^{DB} \cup C_1^{db}$
2. *for all* $X \in C_1^{DB} \cup C_1^{db}$ *do*
3. $\quad c(X,UP) = c(X,DB) + c(X,db)$
4. *end do*
5. $F_1^{UP} = \left\{ X \in C_1^{UP} \middle| c(X,UP) \geq \sigma^{UP} \right\}$
6. $EF_1^{UP} = \left\{ X \in C_1^{UP} \middle| \rho^{UP} \leq c(X,UP) < \sigma^{UP} \right\}$

Figure 5: Update 1-itemsest algorithm

Algorithm 3: Generating Candidate k- itemsets

Input : $F_1^{UP}, F_{k-1}^{UP}, F_{k-1}^{db}, k$

Output : C_k^{new}

1. *if* $k = 2$ *then*
2. $\quad$ *if* $\left(length\left(F_1^{UP}\right) \geq 2\right)$ *then*
3. $\quad\quad C_2^{db} = F_1^{UP} * F_1^{UP}$
4. $\quad\quad$ *for all* $X \in C_2^{db}$ *do*
4. $\quad\quad\quad C_2^{new} = \left\{ X \in C_2^{db} \middle| X \notin \left(F_2^{DB} \cup EF_2^{DB}\right) \right\}$
5. $\quad\quad$ *end do*
6. $\quad$ *end if*
7. *else if* $k > 2$ *then*
8. $\quad\quad C_k^{db} = F_{k-1}^{db} * F_{k-1}^{db}$
9. $\quad\quad$ *for all* $X \in C_k^{db}$ *do*
10. $\quad\quad\quad C_k^{new} = \left\{ X \in C_k^{db} \middle| X \in F_{k-1}^{UP} \text{ and } X \notin \left(F_k^{DB} \cup EF_k^{DB}\right) \right\}$
11.. $\quad\quad$ *end do*
12. *end if*

Figure 6: Generating candidate k- itemsets algorithm

Figure 7 shows an algorithm for updating support count of k-updated frequent itemsets and k-updated expected frequent itemsets for k greater than or equal to 2. As shown in line 1, the algorithm scans an increment database to find and update support count of the k- updated candidate itemsets, i.e. C_k^{UP} . When any k-itemsets are not in the union set of the original *k*-frequent and the original *k*- expected frequent itemsets, i.e. k-itemsets $\notin$ $F_k^{DB} \cup EF_k^{DB}$, but is in the k- increment candidate itemsets, i.e. k-itemsets $\in$ C_k^{new} , their support counts need to be specially updated. This is the case because their support counts obtained from an original database are not available. Since these k-itemsets are not in $F_k^{DB} \cup EF_k^{DB}$, their support counts are at best equal to ρ^{DB} - 1 . Here, their support counts are assumed to be equal to the sum of ρ^{DB} - 1 and their support counts obtained from an increment database, i.e. $c(X, db)+(\rho^{DB} - 1)$. If any k- itemsets have support counts below updated min support count, i.e. σ^{UP}, the k- itemsets can't be the *k*-updated frequent itemsets. On the other hand, if any k- itemsets have support counts above or equal to an updated min support count, the k-itemsets are likely to be the k-updated frequent itemsets. Thus, the k- itemsets, which have support counts above or equal to an updated min support count, are set aside for finding their true support counts from an original database, i.e. line 9.

Algorithm 4 : Update ($k \geq 2$) itemset

Input : $DB, db, \sigma^{UP}, \rho^{UP}, \rho^{DB}, F_k^{DB}, EF_k^{DB}$ *and their count*

Ouiput: F_k^{UP} *and* EF_k^{UP}, F_k^{db}, *Temp_scanDB and their count, m*

1. *Scan db and find count* $c(X, db)$ *and* $c(Y, db)$
2. \\ $X \in (F_k^{DB} \cup EF_k^{DB})$ *and* $Y \in C_k^{new}$
3. *for all* $X \in (F_k^{DB} \cup EF_k^{DB} \cup C_k^{new})$ *do*
4. *if* $X \in (F_k^{DB} \cup EF_k^{DB})$ *and* $X \in C_k^{new}$ *then*
5. $c(X, UP) = c(X, DB) + c(X, db)$
6. *else if* $X \in (F_k^{DB} \cup EF_k^{DB})$ *and* $X \notin C_k^{new}$ *then*
7. $c(X, UP) = c(X, DB)$
8. *else if* $X \notin (F_k^{DB} \cup EF_k^{DB})$ *and* $X \in C_k^{new}$ *then*
9. $Temp\_scanDB_k = \{X \ (c(X, db) + (\rho^{DB} - 1)) \geq \sigma^{UP}\}$
10. *end if*
11. *end do*
12. $F_k^{UP} = \{X \ c(X, UP) \geq \sigma^{UP}\}$
13. $EF_k^{UP} = \{X \ \rho^{UP} \leq c(X, UP) < \sigma^{UP}\}$

Figure 7: Update ($k \geq 2$) itemset algorithm

At the third phase, an original database is scanned to find true support counts for the k-itemsets that are likely to be the *k*-updated frequent itemsets. The algorithm is shown in figure 8. The support counts of the likely k-updated frequent itemsets are found and updated by scanning an original database as shown in line 1-6. Then, all k-updated frequent itemsets and k- updated expected frequent itemsets are found as shown in line 7-8.

The figure 9 shows the increment database and the figure 10 shows the example of three phases of updating frequent and expected frequent itemsets. In the first phase, the 1-frequent and expected frequent itemset can be updated by scanning only an increment database. In the second phase, the 2-updated candidate itemset is generated by joining frequent 1-itemset of updated database together. The new candidate 2-itemset is pruned if it is member of frequent and expected frequent itemset of an original database. In this case, set of {AD, BD, CD} are new candidate 2-itemset in the updated database.

The increment database is scanned for 2-itemset of frequent, expected frequent and new updated candidate. Only new candidate itemset that has support count greater than or equal to minimum support of increment database, it require to scan in the original database.

Algorithm 5 : Scanning an original database

Input : $Temp\_scanDB_k, \sigma^{UP}, \rho^{UP}, F_k^{UP}, EF_k^{UP}$ *and their count*

Output : F_k^{UP}, EF_k^{UP} *and their count*

1. *Scan DB and obtain count* $c(X, DB)$ *for all* $Temp\_scanDB_k$
2. *for all* $X \in Temp\_scanDB_k$ *do*
3. $c(X,UP) = c(X, DB) + c(X, db)$
4. *end do*
5. $F_k^{new} = \{X \mid X \in Temp\_scanDB_k \text{ and } c(X,UP) \geq \sigma^{UP}\}$
6. $EF_k^{new} = \{X \mid X \in Temp\_scanDB_k \text{ and } \rho^{UP} \leq c(X,UP) < \sigma^{UP}\}$
7. $F_k^{UP} = F_k^{UP} \cup F_k^{new}$
8. $EF_k^{UP} = EF_k^{UP} \cup EF_k^{new}$

Figure 8: Algorithm for scanning an original database

Increment Database	
TID	List of item
11	A, B, D, E
12	B, C, D
13	B, D, E
14	A, B, C, D
15	A, C

Figure 9: Increment database

To reduce the number of itemset for scanning original database, if sum of any new candidate's support count and support of $prob_{pl}$ minus 1 is greater than minimum support of updated database, then it will be moved to Temp_scanDB. In this example, only set of BD is moved to Temp_scanDB. The last phase, original database is scanned for finding true support of BD. Finally, the frequent and expected frequent k-itemsets for updated database are found.

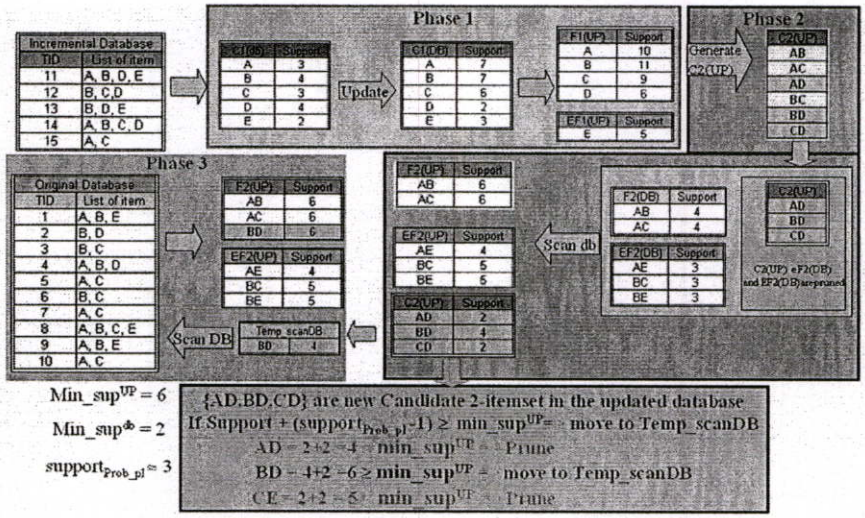


Figure 10: Example of three phases of updating frequent and expected frequent itemsets

4 Experiment

The purpose of this research is to explore an alternative way to improve the performance of the association rule mining while facing dynamic database environments. To evaluate the performance of probability-based incremental association rules discovery algorithm, the algorithm is implemented and tested on a PC with a 2.8 GHz Pentium 4 processor, and 1 GB main memory. The experiments

are conducted on a synthetic dataset, called T10I4D20K. The technique for generating the dataset is proposed by Agrawal [Agrawal 1994]. The synthetic dataset comprises 250,000 transactions over 70 unique items. Each transaction of the synthetic dataset has 10 items on average and the maximal size itemset is 4.

Firstly, we show the performance of our algorithm with minimum support 3% and 4%. The proposed algorithm with $Prob_{pl} = 0.06$ is used to find frequent itemsets, expected frequent itemsets and association rules from an original database of 20,000 transactions. Then, several sizes of increment databases, i.e. 25%, 50%, 75% and 100% of the original database, are added to the original database. Then, the proposed algorithm is used to maintain frequent itemsets, expected frequent itemsets and association rules of the updated database. For comparison purpose, FUP, Borders and Pre-large algorithm are also used to find frequent itemsets, expected frequent itemsets and association rules from the same original database and the same increment databases.

The experimental results with various minimum support thresholds are shown in figure 11, table 2 and table 3. Table 2 and table 3 show the number of frequent k-itemsets and k- expected itemsets in an original database for FUP, Borders, Pre-large algorithm and the proposed algorithm. Since these algorithms are in the second approach which is assumed that association rules to be found are stable over time, the association rules or frequent itemsets should be the same as those obtained from re-runing Apriori algorithm. Thus, each frequent itemset obtained from these algorithms is compared with that obtained from re-runing Apriori algorithm. The experiments show that all frequent itemset obtained from FUP, Borders, Pre-large algorithm and the proposed algorithm are the same as those obtained from re-runing Apriori algorithm.

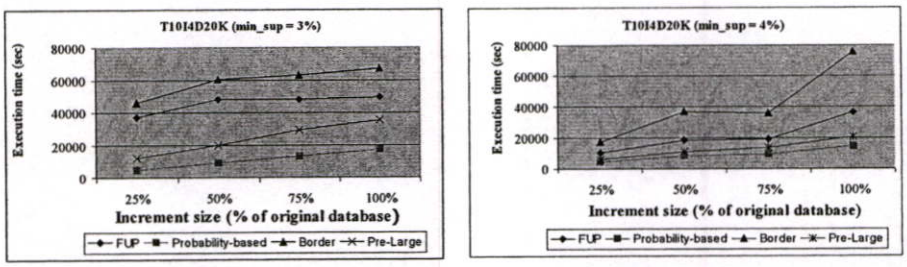


Figure 11: Execution time in FUP, Borders and Probability-based when (a)min_sup =3% and (b) min_sup =4%

Original Database : Min_sup = 3%										
k-item	Apriori		FUP		Borders		Pre-Large		Probability-based	
	Number of Frequent itemset	Number of kept Infrequent itemset	Number of Frequent itemset	Number of kept Infrequent itemset	Number of Frequent itemset	Number of kept Infrequent itemset	Number of Frequent itemset	Number of kept Infrequent itemset	Number of Frequent itemset	Number of kept Infrequent itemset
k=1	68	0	68	0	68	1	68	1	68	0
k=2	700	0	700	0	700	1578	700	418	700	14
k=3	158	0	158	0	158	5003	158	399	158	15
k=4	39	0	39	0	39	44	39	80	39	1
k=5	21	0	21	0	21	0	21	10	21	0
k=6	7	0	7	0	7	0	7	1	7	0
k=7	1	0	1	0	1	0	1	0	1	0

Table 2: Number of itemset of an original database for each algorithm with min_sup=3%

Original Database : Min_sup = 4%										
k-item	Apriori		FUP		Borders		Pre-Large		Probability-based	
	Number of Frequent itemset	Number of kept Infrequent itemset	Number of Frequent itemset	Number of kept Infrequent itemset	Number of Frequent itemset	Number of kept Infrequent itemset	Number of Frequent itemset	Number of kept Infrequent itemset	Number of Frequent itemset	Number of kept Infrequent itemset
k=1	65	0	65	0	65	4	65	3	65	1
k=2	442	0	442	0	442	1638	442	258	442	25
k=3	24	0	24	0	24	2160	24	134	24	3

Table 3: Number of itemset of an original database for each Algorithm with min_sup = 4%

From both table 2 and table 3, the results also show that the proposed technique requires slightly more space to keep the k- expected itemsets than that of FUP. This is the case because FUP does not need to keep some infrequent itemsets to maintain updated frequent itemsets but FUP requires multiple-scan an original database to maintain updated frequent itemsets. Thus, FUP requires more times for scanning an original database than that of the proposed algorithm. As a result, the proposed algorithm has better running time than that of FUP. As shown in figure 11, the proposed requires lesser memory space to keep the k- expected itemsets than that of Borders algorithm and Pre-large algorithm. However, the algorithm still has better running time than that of Borders algorithm and Pre-large algorithm. This is the case because the proposed algorithm, which is based on the principle of Bernoulli trials, has more efficient to predict expected frequent itemsets than Borders algorithm and Pre-large algorithm. Thus, the proposed algorithm requires less times for scanning an original database than Borders algorithm and Pre-large algorithm. As a result, the proposed algorithm has better running time than that of Borders algorithm and Pre-large algorithm.

	Average of Execution time for 100 trials (sec)			
min_sup (%)	**FUP**	**Borders**	**Pre-Large**	**Probability-Based**
4%	5900.722	7588.617	4207.582	2661.803

Table 4: The average execution time for FUP, Borders and Probability-based

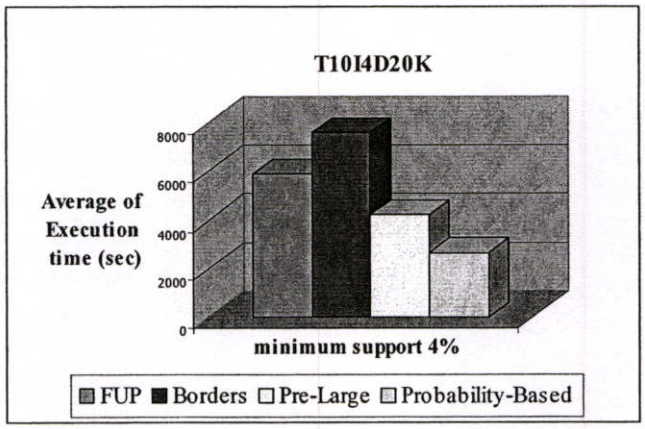


Figure 12: The Average of execution time in FUP, Borders and Probability-based when min_sup =4%

Secondly, the proposed algorithm with $Prob_{pl}$ = 0.06 used to find frequent itemsets, expected frequent itemsets and association rules from an original database of 20,000 transactions. Then, the same sizes of increment databases, i.e. 10% of the original database, are added to the original database for 100 trials. For comparison purpose, FUP, Borders and Pre-large algorithm are also used to find association rules from the same original database and the same increment databases. Table 4 and figure 12 show the average execution time of FUP, Borders, Pre-large and the proposed algorithm. The results show that the proposed algorithm has much better running time than that of the other algorithms.

5 Conclusion

To create an intelligent environment such that new information can be discovered in a dynamic database, this paper proposes mining a dynamic database using probability-based incremental association rule discovery algorithm. The proposed algorithm uses the principle of Bernoulli trials to find expected frequent itemsets. This can reduce a number of times to scan an original database. Assuming that the minimum support and confidence do not change, the algorithm can maintain association rules for a dynamic database. The experiments show that our algorithm has better running time than that of FUP, Borders and Prelarge algorithm.

References

[Adnan et al. 2005] Adnan, M., Alhajj, R., and Barker, K.: "Performance Analysis of Incremental Update of Association Rules Mining Approaches"; In Proceeding of 9th IEEE International Conference on Intellgent Engineering System 2005, September 16-19, 2005, 129-134.

[Agrawal et al. 1993] Agrawal, R., Imielinski, T., and Swami, A.: Mining association rules between sets of items in large databases , In Proceeding of the ACM SIGMOD Int'l Conf. on Management of Data (ACM SIGMOD '93), Washington, USA, May 1993, 207-216.

[Agrawal, Srikant 1994] Agrawal, R. and Srikant, R.: "Fast algorithms for mining association rules , In Proceedings of 20 th Intl Conf. on Very Large Databases (VLDB'94) "; Santiago, Chile (1994), 478 -499.

[Alon et al. 1992] Alon, N. and Spencer, J. H., 1992, "The probabilistic method"; Wiley Interscience, New York.

[Amornchewin, Kreesuradej 2007] Amornchewin, R., and Kreesuradej, W.: "Incremental association rule mining using promising frequent itemset algorithm "; In Proceeding 6th International Conference on Information, Communications and Signal Processing, Singapore, 2007.

[Amornchewin, Kreesuradej 2008] Amornchewin, R., and Kreesuradej, W.: "Probability-based incremental association rule discovery algorithm"; The 2008 International Symposium on Computer Science and its Applications (CSA-08), Australia, 2008.

[Ayan et al. 1999] Ayan, N.F., Tansel, A.U., and Arun, E.: "An efficient algorithm to update large itemsets with early pruning"; Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, August 1999, 287-291.

[Brin et al. 1997] Brin, S., Motwani, R., and Silverstein, C.: "Beyond Market Baskets: Generalizing Association Rules to Correlations"; Proceedings of the ACM SIGMOD Conference, 1997, 265-276.

[Chang, Yang 2003] Chang, C.-H. and Yang, S.-H.: "Enhancing SWF for Incremental Assocaition Mining by Itemset Maintenance"; Proceedings of 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining, April 2003.

[Chang et al. 2005] Chang, C.C., Li Y.C., and Lee, J.S.: "An efficient algorithm for incremental mining of association rules "; Proceedings of the 15th international workshop on research issues in data engineering: stream data mining and applications (RIDE-SDMA'05), IEEE, 2005.

[Cheung et al. 1996] Cheung, D., Han, J., Ng, V., and Wong, C. Y.: "Maintenance of discovered association rules in large databases: An incremental updating technique"; In 12th IEEE International Conference on Data Engineering, February 1996, 106-114.

[Cheung et al. 1997] Cheung, D. W., Lee, S.D., and Kao, B.: "A General incremental technique for maintaining discovered association rules";In Proceedings of the 5 th Intl. Conf. on Database Systems for Advanced Applications (DASFAA'97), Melbourne, Australia, April 1997, 185-194.

[Du et al. 2008] Du, Y. Zhao, M., Fan G.: "Research on Application of Improved Association rules Algorithm in intelligent QA system"; Second International Conference on Genetic and Evolutionary Computing, 2008.

[Dudek, Zgrzywa 2005] Dudek, D., Zgrzywa, A.:"The Incremental Method for Discovery of Association Rules"; Springer Berlin / Heidelberg, volume 30, 2005, 153-160.

[Elfangary, Atteya 2008] Elfangary, L.,Atteya, W.A.: "Mining Medical Database using Proposed Incremental Association rule Algorithm (PIA)"; Second International Conference on the Digital Society, IEEE (2008).

[Feldman et al. 1999] Feldman, R., Aumann,Y., and Lipshtat, O.: "Borders: An efficient algorithm for association generation in dynamic databases"; Journal of Intelligent Information System, 1999, 61-73.

[Ezeife, Su 2002] Ezeife, E. I. and Su, Y.: "Mining Incremental Association Rules with Generalized FP-Tree"; Proceedings of the 15th Conference of the Canadian Society for Computational Studies of Intelligence on Advances in Argificial Intelligence, May 2002, 147-160.

[Gong 2008] Gong M. :"Personalized E-learning System by Using Intelligent Algorithm"; 2008 Workshop on Knowledge Discovery and Data Mining, IEEE, 2008.

[Guo, Zhao 2008]Guo, M., Zhao Y.: "An Extensible Architecture for Personalized Information Services in An Ambient Intelligence Environment"; IEEE 2008.

[Han et al. 2000] Han, J.,Pei, J. and Yin, Y.: "Mining Frequent Pattern without Candidate Generation";Proceedings of the 2000 ACM-SIGMOD International Conference on Management of Data, May 2000, 355-359.

[Hong et al. 2001] Hong, T.P., Wang, C.Y., and Tao, Y.H.: "A new incremental data mining algorithm using pre-large itemsets"; Journal of Intelligent Data Analysis, Vol. 5, No.2 2001, 111-129.

[Juan et al. 2008] Juan X., Feng Y., Zhiyong Z.: "Association Rule Mining and Application in Intelligent Transportation System"; Proceedings of the 27th Chinese Control Conference, China, 2008, 538-540.

[Lee et al. 2001] Lee, C. H., Lin, C. R., and Chen,M. S.: "Sliding-Window Filtering: An Efficient Algorithm for Incremental Mining"; Proceeding of the ACM 10th International Conference on Information and Knowledge Management , November 2001.

[Liu et al. 1999] Liu, B., Hsu W., and Ma, Y.: "Mining Association Rules with Multiple Supports"; ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-99), SanDiego, CA, USA. August 1999.

[Luhr et al. 2005] Luhr, S., Venkatesh, S., and West, G.: "Emergent Intertransaction Association Rules for Abnormality Detection in Intelligent Environments"; Proceedings of the 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing Conference, December 2005, 343-347.

[Manku, Motwani 2002] Manku, G.S., and Motwani, R.: "Approximate Frequency Counts over Streaming Data"; Proceedings of the 28th International Conference on Very Large Database, August 2002, 346-357.

[Mannila et al. 1994]Mannila, H., Toivonen, H., and A. Verkamo I.: "Efficient algorithms for discovering association rules. In Knowledge Discovery in Databases"; 1994 AAAI Workshop (KDD'94), Seattle, Washington, July 1994, 181- 192.

[Pawar, Aggarwal 2004] Pawar P.A.and Aggrawal A. K.: "Associative Rule Mining of Mobile Data Services Usage for Preference Analysis, Personalization & Promotion"; Proceeding of WSEAS International Conference on Simulation, Modeling and Optimization, Izmir, Turkey, September 2004.

[Sarda, Srinivas 1998] Sarda, N. L., and Srinivas, N. V.: "An adaptive algorithm for incremental. mining of association rules"; In Proc. 9th Intl.Workshop on Database and Expert System Applications, Vienna, Austria, Aug 1998, 240-245.

[Savasere et al. 1995] Savasere, A., Omiecinski, E., and Navathe S.: "An Efficient Algorithm for Mining Association Rules in large database"; Proceedings of the 21st VLDB Conference, Zurich, Switzerland, 1995.

[Teng, Chen 2005] Teng W.-G. and Chen, M.-S.: " Incremental Mining on Association Rules"; Springer Berlin/ Heidelberg, vol. 180, 2005, 125-162.

[Thomas et al. 1997] Thomas, S., Bodagala, S., Alsabti, K., and Ranka, S.: "An efficient algorithm for the incremental updation of association rules in large databases"; In Proceedings of the 3rd Intl. Conf. on Knowledge Discovery and Data Mining (KDD'97), New Port Beach, California, August 1997, 263-266.

[Thuraisingham, Ceruti 2000] Thuraisingham, B.M., Ceruti, M.G.: "Understanding Data Mining and Applying it' to Command, Control, Communications and Intelligence Environments"; The Twenty-Fourth Annual International Computer Software and Applications Conference (COMPSAC), 2000, 171-174.

[Tsai et al 2005] Tsai, Paury S.M.,Lee, Chih-Chong, and Chen, Arbee L.P. : "An Efficient Approach for Incremental Association Rule Mining"; Proceedings of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining, Lecture Notes In Computer Science; Vol. 1574 archive, 1999.

[Toivonen 1996] Toivonen, H. 1996.: "Sampling large databases for association rules"; In Proceedings of the 22th International Conference on Very Large Databases (VLDB), September 1996, 134-145.

[Veloso et al. 2002] Veloso, A. Meira Jr., W., Carvalho, M., Parthasarathy, S., and Zaki ,M. J.: "Parallel, Mining frequent itemsets in evolving Databases"; In Proc. 2nd SIAM Intl. Conf. on Data Mining, Arlington, VA, April 2002.

[Zhang et al. 2003] Zhang, C., Zhang, S., and Webb, G. I.: "Identifying Approximate Itemsets of Interest in Large Databases"; Applied Intelligence 2003, 91–104.

[Zhang et al. 2003] Zhang, S., Zhang, C., and Yan X.: "Post-mining: maintenance of association rules by weighting"; Information System, 2003, 691-707.

ประวัติผู้เขียน

ชื่อ-นามสกุล	นางสาวรัชดาภรณ์ อมรชีวิน
วัน เดือน ปีเกิด	10 กุมภาพันธ์ พ.ศ. 2513
ที่อยู่	96/102 หมู่ 4 ถนน พหลโยธิน ตำบล เขาสามยอด อำเภอ เมืองลพบุรี จังหวัด ลพบุรี. 15000 โทร. 036-627606, 086-5115888.
ประวัติการศึกษา	2534 ระดับปริญญาตรีวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยหอการค้าไทย 2540 วิทยาศาสตรมหาบัณฑิต คณะบัณฑิตวิทยาลัย สาขาวิชาการจัดการเทคโนโลยีสารสนเทศ สถานศึกษาเดิม
ประวัติการทำงาน	2535- 2541 โปรแกรมเมอร์ บริษัท ไทยซอฟต์ จำกัด 2541- ปัจจุบัน อาจารย์คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยราชภัฏเทพสตรี
ผลงานวิจัย	- Amornchewin R. and Kreesuradej W., "Incremental association rule mining using promising frequent itemset algorithm", **In Proceeding 6th International Conference on Information, Communications and Signal Processing**, Dec. 10-13 2007, pp.1-5. - Amornchewin R. and Kreesuradej W. "Probability-Based Incremental Association Rule Discovery Algorithm" **International Symposium on Computer Science and Its Applications 2008,** 13-15 October 2008, pp.212-215. - Amornchewin R. and Kreesuradej W. "Mining Dynamic Databases using Probability-Based Incremental Association Rule Discovery Algorithm." **Journal of Universal Computer Science**, Vol. 15, no.12, 2009. pp. 2409-2428.