การจำแนกประเภทข้อมูลแบบเพิ่มขยายโดยใช้กฎความสัมพันธ์

INCREMENTAL CLASSIFICATION BASED ON ASSOCIATION RULES

สรารักษ์ ธนารัตน์

SARARAK TANARAT

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2553

KMITL-2011-IT-M-001-002

INCREMENTAL CLASSIFICATION BASED ON ASSOCIATION RULES

SARARAK TANARAT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2010

KMITL-2011-IT-M-001-002

หัวข้อวิทยานิพนธ์	การจำแนกประเภทข้อมูลแบบเพิ่มขยายโดยใช้กฎความสัมพันธ์
นักศึกษา	นางสาวสรารักษ์ ธนารัตน์
รหัสนักศึกษา	49066738
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
พ.ศ.	2553
อาจารย์ที่ปรึกษา	รศ.ดร.วรพจน์ กรีสุระเดช

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้นำเสนอการจำแนกประเภทข้อมูลแบบเพิ่มขยายโดยใช้กฎความสัมพันธ์ โดยเป็นการปรับปรุงเทคนิคการจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์ซึ่งเป็นการนำเอาสองเทคนิคที่สำคัญในดาต้าไมนิ่งคือการจำแนกประเภทข้อมูลและการค้นหากฎความสัมพันธ์มาประยุกต์ใช้งานร่วมกัน มีอัลกอริทึมชื่อว่า Classification Based on Association Rules (CBA)

เมื่อมีการเพิ่มขยายของข้อมูล ความสัมพันธ์ระหว่างข้อมูลอาจจะเปลี่ยนไป ซึ่งเป็นเหตุผลที่ทำให้ training data เปลี่ยนไปด้วยเช่นกัน และเพื่อปรับปรุงกฎความสัมพันธ์ให้ถูกต้องอยู่เสมอ จึงมีความจำเป็นที่จะต้องค้นหาความสัมพันธ์ใหม่ งานวิจัยนี้ได้นำเสนออัลกอริทึมใหม่ใช้สำหรับจำแนกประเภทข้อมูลแบบเพิ่มขยายโดยใช้กฎความสัมพันธ์ มีชื่อเรียกว่า Incremental Classification Based on Association Rules (ICBA) โดยเป็นการนำหลักการของอัลกอริทึม FUP เข้ามาประยุกต์ร่วมกับอัลกอริทึม CBA

จากการทดลองแสดงให้เห็นว่า อัลกอริทึม ICBA สามารถทำงานได้เร็วกว่าอัลกอริทึม CBA โดยที่กฎความสัมพันธ์และแบบจำลองของทั้งสองอัลกอริทึมมีค่าเท่ากัน เมื่อมีการเพิ่มขยายของ training data

Thesis	Incremental Classification Based on Association Rules
Student	Ms.Sararak Tanarat
Student ID.	49066738
Degree	Master of Science
Program	Information Technology
Major	Information System
Year	2010
Thesis Advisor	Assoc. Prof. Dr. Worapoj Kreesuradej

ABSTRACT

This thesis proposes a new classification approach called Associative Classification which integrates both association rule mining and classification task. In this study, an incremental updating technique is applied to associative classification for constructing classification system when a new training dataset is appended to an old training dataset. The proposed algorithm, called Incremental Classification Based on Association Rules (ICBA), is based on the concept of Fast Update algorithm (FUP) algorithm. The experiment results show that the proposed algorithm has execution time better than CBA algorithm.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงเป็นอย่างดี ด้วยความกรุณาจากอาจารย์ผู้ควบคุมวิทยานิพนธ์ รศ.ดร. วรพจน์ กรีสุระเดช ที่ให้คำปรึกษา ชี้แนวทางในการในการวิจัย และการแก้ปัญหาของงานวิจัยนี้จนสำเร็จลุล่วง ตลอดจนประสิทธ์ประสาทวิชาความรู้อื่นๆ ที่ไม่สามารถหาได้ในห้องเรียน

ขอกราบพระคุณคณาจารย์คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ ตลอดระยะเวลาที่ศึกษาอยู่ในสถานศึกษาแห่งนี้

ขอขอบคุณห้องวิจัยและปฏิบัติการ Data Mining and Data Exploration Lab (DME Lab) คณะเทคโนโลยีสารสนเทศ ที่สนับสนุนอุปกรณ์ในการทำวิจัยจนสำเร็จ

ขอขอบคุณสำนักงานพัฒนาเทคโนโลยีอวกาศและภูมิสารสนเทศ (องค์การมหาชน) หน่วยงานต้นสังกัด ที่ให้เวลาในการทำวิจัยอย่างเต็มที่จนสำเร็จลุล่วง

ขอขอบคุณ เพื่อนๆ พี่น้องนักศึกษาทุกคนที่ได้ให้ความช่วยเหลือและแบ่งปันความรู้ รวมทั้งเป็นที่ปรึกษา เมื่อเกิดปัญหาในการวิจัย

สุดท้ายนี้ขอกราบขอบพระคุณบิดา มารดา พี่น้องร่วมอุทรที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่อง จนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์ที่ได้รับจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่บิดา มารดา อันเป็นที่รักและเคารพยิ่ง ซึ่งเป็นผู้ให้กำเนิดและทำให้ข้าพเจ้าได้มีวันนี้

สรารักษ์ ธนารัตน์

สารบัญ

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญตาราง (ต่อ)

สารบัญรูป

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

สารบัญรูป (ต่อ)

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การทำเหมืองข้อมูลเป็นกระบวนการที่สำคัญในการสกัดแยกข้อมูล (Extract Data) ทำการสำรวจวิเคราะห์ ค้นหาลักษณะรูปแบบความสัมพันธ์ของข้อมูลที่น่าสนใจ หรือข้อมูลที่มีรูปแบบ (Pattern) ซึ่งมีความหมาย มีอยู่จริงในฐานข้อมูลและเชื่อถือได้ รวมถึงสามารถนำข้อมูลที่มีอยู่มาใช้ประโยชน์ได้ตรงตามความต้องการสูงที่สุด เนื่องจากไม่มีเทคนิคใดของการทำเหมืองข้อมูลจะสามารถตอบคำถามได้ทุกปัญหา จึงทำให้มีเทคนิคสำหรับการทำเหมืองข้อมูลเป็นจำนวนมาก การคัดเลือกเทคนิคให้เหมาะสมกับปัญหาจึงเป็นเรื่องสำคัญอย่างยิ่ง

การจำแนกประเภทข้อมูล (Data Classification) [8] เป็นเทคนิคหนึ่งที่สำคัญของการทำเหมืองข้อมูล โดยเทคนิคนี้จะใช้ในการค้นหาความรู้เพื่อสรุปหาแบบจำลองของฐานข้อมูลสำหรับนำไปใช้ทำนายข้อมูลใหม่ๆ (Unseen data) การจำแนกประเภทข้อมูลเป็นเทคนิคแบบ Supervise Learning นั่นคือ ก่อนที่จะสร้างแบบจำลองของข้อมูลขึ้นมาได้นั้น จะต้องทำการสอนระบบเสียก่อน หลักการทำงานของเทคนิคนี้คือจะทำการแบ่งข้อมูลออกเป็นสองส่วนคือ ส่วนหนึ่งใช้สอนระบบ (Training data) ซึ่งจะเป็นการสร้างแบบจำลองจากข้อมูลในอดีตที่มีข้อมูลเป็นจำนวนมาก และอีกส่วนหนึ่งใช้ทดสอบแบบจำลอง(Testing Data) ซึ่งจะทดสอบความน่าเชื่อถือประสิทธิภาพของแบบจำลองที่ถูกสร้างขึ้นมาว่ามีความแม่นยำเหมาะสมที่จะนำไปใช้งานกับข้อมูลใหม่ (Unseen data) หรือไม่ โดยทั่วไปสัดส่วนระหว่างข้อมูลที่ใช้สอนระบบกับข้อมูลที่ใช้ทดสอบแบบจำลองจะอยู่ที่ประมาณ 80 ต่อ 20

อีกเทคนิคหนึ่งที่สำคัญในการทำดาต้าไมนิ่ง คือการค้นหากฎความสัมพันธ์ (Association Rule Discovery) [8]หลักการทำงานโดยสรุปคือ ค้นหารูปแบบความสัมพันธ์ของข้อมูลจากข้อมูลจำนวนมากในฐานข้อมูลเพื่อนำไปใช้ในการวิเคราะห์ และทำนายคาดการณ์ (Prediction) จัดข้อมูลเหล่านั้นให้อยู่ในรูปแบบของกฎความสัมพันธ์ โดยกฎความสัมพันธ์จะอยู่ในรูปของ IF X THEN Y เช่น IF milk THEN bread ความหมายของกฎความสัมพันธ์นี้คือ ถ้าลูกค้าซื้อนมแล้วจะซื้อขนมปังไปด้วย

การจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์ (Associative Classification) [4] เป็นอีกเทคนิคหนึ่งของการทำเหมืองข้อมูลซึ่งมีประสิทธิภาพในการทำนายกลุ่มของข้อมูลแม่นยำกว่าเทคนิคการจำแนกประเภทข้อมูลทั่วไป การจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์นั้น เป็นการนำสองเทคนิคที่สำคัญในการทำเหมืองข้อมูล คือเทคนิคการจำแนกประเภทข้อมูลและการค้นหาความสัมพันธ์ มาประยุกต์ใช้งานร่วมกัน เป้าหมายของเทคนิคการจำแนกประเภทข้อมูลโดย

ใช้กฎความสัมพันธ์คือ เพื่อทำนายกลุ่มให้กับข้อมูลใหม่ที่ไม่ทราบกลุ่ม (Unknown class data) การจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์มีอัลกอริทึมชื่อว่า CBA (Classification Based on Association rules) ซึ่งจะแบ่งการทำงานออกเป็น 2 ส่วนหลักๆคือ ส่วนของการสร้างกฎความสัมพันธ์ (Rule Generator) โดยที่กฎความสัมพันธ์จะอยู่ในรูปของกฎความสัมพันธ์แบบมีคลาส (Class Association Rule: CAR) และส่วนที่สร้างแบบจำลองเพื่อใช้ในการทำนาย (Classifier Builder)

การจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์ เป็นการนำเอากฎความสัมพันธ์แบบมีคลาสมาสร้างเป็นแบบจำลองเพื่อใช้ในการทำนายกลุ่มให้กับข้อมูล ดังนั้นเมื่อมีการเพิ่มขยายข้อมูลในฐานข้อมูล อาจทำให้มีผลต่อการเปลี่ยนแปลงของกฎความสัมพันธ์ เนื่องจากข้อมูลที่ใช้สอนระบบ (Training data) ได้เปลี่ยนแปลงไป จากการเปลี่ยนแปลงดังกล่าว ทำให้จะต้องทำการสแกนทั้งฐานข้อมูลเก่าที่มีอยู่และฐานข้อมูลใหม่ที่มีการเพิ่มขึ้นของ transaction เพื่อทำการสร้างกฎความสัมพันธ์ให้สอดคล้องกับฐานข้อมูลที่เปลี่ยนแปลงไป ซึ่งจะทำให้เสียเวลาในการทำงานมากขึ้น จึงได้นำหลักการ Incremental association rule เข้ามาประยุกต์ใช้ เพื่อนำความรู้ที่มีอยู่เดิมมาใช้งานให้เกิดประโยชน์สูงสุด

1.2 วัตถุประสงค์ของงานวิจัย

วิทยานิพนธ์ฉบับนี้มีวัตถุประสงค์ที่จะปรับปรุงอัลกอริทึม CBA ซึ่งเป็นอัลกอริทึมสำหรับการจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์ เมื่อมีการเพิ่มขยายของ Training data โดยทำการเสนออัลกอริทึม ICBA ซึ่งเป็นอัลกอริทึมที่ช่วยเพิ่มประสิทธิภาพในด้านเวลาการทำงานของอัลกอริทึม CBA โดยที่อัลกอริทึม ICBA จะต้องสร้างแบบจำลองสำหรับจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์ในกรณีที่มีการเพิ่มขยายของ Training data ด้วยเวลาที่เร็วกว่าอัลกอริทึม CBA

1.3 สมมติฐานของการศึกษา

ในงานวิจัยนี้ได้มุ่งเน้นไปที่การลดเวลาการทำงานของอัลกอริทึม CBA เมื่อมีการเพิ่มขยายของ Training data โดยการนำเสนออัลกอริทึม ICBA ซึ่งอัลกอริทึม ICBA จะทำงานได้เร็วกว่าอัลกอริทึม CBA ก็ต่อเมื่อ Training data ที่เป็นส่วนของการเพิ่มขยายนั้น มีค่าทางสถิติที่ใกล้เคียงกับ Training data เดิม เนื่องจากถ้าค่าทางสถิติของข้อมูลไม่ใกล้เคียงกัน อัลกอริทึมอาจจะไม่สามารถนำความรู้เดิมที่เคยมีอยู่มาใช้งานให้เกิดประโยชน์ได้เต็มที่ และเซตกฎความสัมพันธ์ใหม่ที่เกิดจาก Increment training data จะต้องมีกฎความสัมพันธ์บางส่วนเหมือนกับสมาชิกในเซตของกฎความความสัมพันธ์ที่เกิดจาก Original training data

1.4 ขอบเขตการวิจัย

วิทยานิพนธ์ฉบับนี้ได้ศึกษาเกี่ยวกับวิธีการจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์เมื่อมีการเพิ่มขยายของข้อมูล โดยอัลกอริทึม FUP [5] เป็นอัลกอริทึมสำหรับใช้ค้นหากฎความสัมพันธ์ของข้อมูลในกรณีเพิ่มข้อมูลเข้าสู่ฐานข้อมูล เพื่อให้สามารถค้นหากฎความสัมพันธ์เมื่อมีการเพิ่มขึ้นของข้อมูล จึงได้ปรับปรุงอัลกอริทึม CBA [4]ให้สามารถค้นหาจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์ในกรณีที่มีการเพิ่มของข้อมูลในฐานข้อมูลได้

1.5 ขั้นตอนของการศึกษา

ขั้นตอนในการศึกษาวิธีการวิจัย จากเริ่มจนถึงสิ้นสุดการทำงานวิจัยดังนี้

1. ศึกษาทฤษฎีและงานวิจัยจากเอกสาร บทความต่างๆ ในส่วนที่เกี่ยวข้องกับการทำงานวิจัยในฉบับนี้
2. กำหนดหัวข้อ วัตถุประสงค์ ขอบเขตการทำงานวิจัย
3. วิเคราะห์และปรับปรุงอัลกอริทึม
4. เตรียมข้อมูลเพื่อใช้ทดลอง
5. พัฒนาโปรแกรม ทดสอบ และแก้ไขข้อผิดพลาด
6. รวบรวมผลการทดลองจากการทำงานของโปรแกรม
7. วิเคราะห์และสรุปผลการทดลอง
8. ดำเนินการจัดทำเอกสารงานวิจัย

วิทยานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาทั้งหมดออกเป็น 5บทคือ

บทที่ 1 ความเป็นมาของงานวิจัย ความมุ่งหมายและวัตถุประสงค์ สมมติฐาน ขอบเขตของการวิจัย และขั้นตอนการศึกษา

บทที่ 2 ทฤษฎีพื้นฐานที่ใช้ในการวิจัย

บทที่ 3 การจำแนกประเภทข้อมูลแบบเพิ่มขยายโดยใช้กฎความสัมพันธ์

บทที่ 4 การทดลองและวิเคราะห์ผลการทดลอง

บทที่ 5 สรุปผลการวิจัย

บทที่ 2
ทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้อง

2.1 การจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์ (Associative Classification)

การจำแนกประเภทข้อมูล (Classification) [8] เป็นเทคนิคหนึ่งในการทำเหมืองข้อมูล ซึ่งเป็นเทคนิคที่ใช้สำหรับการสืบค้นความรู้ที่ซ่อนอยู่ในข้อมูลเพื่อทำการสรุปหาแบบจำลองหรือโมเดลของฐานข้อมูลนั้น โดยผลลัพธ์ที่ได้จะนำไปใช้ในการทำนายข้อมูลใหม่ที่ไม่เคยพบมาก่อน (Unseen data) โดยที่การจำแนกประเภทข้อมูลเป็นเทคนิคแบบ Supervise Learning คือการที่จะสร้างแบบจำลองได้นั้นจะต้องทำการสอนระบบเสียก่อน ข้อมูลจะถูกแบ่งออกเป็นสองส่วนในสัดส่วน 80:20 โดยที่ข้อมูล 80 ส่วนจะใช้สำหรับสอนระบบ (Training data) และอีก 20 ส่วนจะนำไปใช้สำหรับทดสอบความแม่นยำของแบบจำลอง (Testing) โดยจะนำข้อมูลที่ถูกแบ่งไว้สำหรับสอนระบบเข้าสู่กระบวนการสร้างแบบจำลอง เมื่อได้แบบจำลองแล้วจะนำแบบจำลองนั้นไปทดสอบความแม่นยำกับข้อมูลที่แบ่งไว้อีกส่วนหนึ่ง หากแบบจำลองมีความแม่นยำไม่ผ่านเกณฑ์ที่กำหนดไว้ก็จะต้องกลับไปปรับปรุงกระบวนการสร้างแบบจำลองใหม่จนกว่าจะได้ค่าความแม่นยำที่พอใจ เมื่อแบบจำลองผ่านเกณฑ์ความแม่นยำที่กำหนดไว้แล้ว ก็จะสามารถนำแบบจำลองนั้นไปใช้จำแนกประเภทข้อมูลที่ไม่เคยพบมาก่อนต่อไป

การค้นหากฎความสัมพันธ์ (Association Rule Discovery) [8] เป็นอีกหนึ่งเทคนิคในการทำเหมืองข้อมูลที่สำคัญ ซึ่งมีหลักการทำงานคือ จะทำการค้นหาความสัมพันธ์ของข้อมูลจากฐานขนาดใหญ่ที่มีอยู่แล้วนำไปสร้างเป็นกฎความสัมพันธ์ เพื่อนำกฎความสัมพันธ์นั้นไปใช้ช่วยในการวิเคราะห์ตัดสินใจ อัลกอริทึมในการค้นหากฎความสัมพันธ์ที่รู้จักกันดีคือ อัลกอริทึมอะพริออริ (Apriori) [2] ซึ่งเป็นอัลกอริทึมที่นิยมใช้ในการค้นหากฎความสัมพันธ์

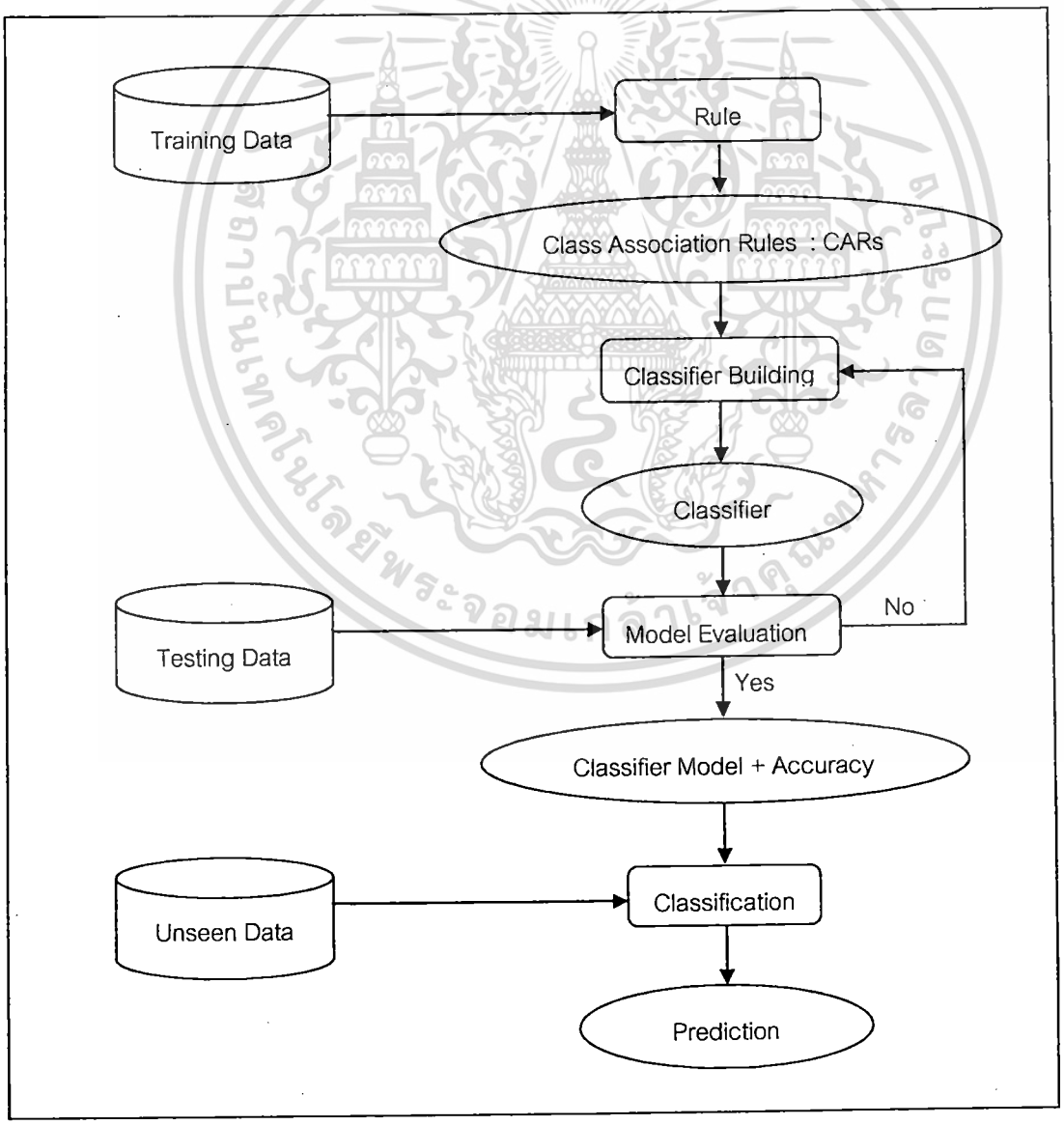
เทคนิคการจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์(Associative Classification) [4] เป็นเทคนิคหนึ่งในการจำแนกประเภทข้อมูลที่น่าสนใจ ให้ความแม่นยำในการทำนายสูง เนื่องจากมีการนำเทคนิคการค้นหากฎความสัมพันธ์เข้ามารวมกับเทคนิคการจำแนกประเภทข้อมูล

เป้าหมายของเทคนิคการจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์ คือใช้ทำนายกลุ่มให้กับข้อมูลในอนาคตที่ไม่ทราบกลุ่ม ซึ่งการทำงานจะถูกแบ่งออกเป็น 2 ส่วนคือ ส่วนแรกเป็นส่วนของการสร้างกฎความสัมพันธ์ (Rule Generator : RG) และส่วนที่สองเป็นส่วนของการสร้างแบบจำลองเพื่อนำไปใช้ในการทำนาย (Classifier Builder : CB)

ในการสร้างกฎความสัมพันธ์สำหรับเทคนิคการจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์นั้น จะใช้หลักการเดียวกันกับ เทคนิคการค้นหากฎความสัมพันธ์เกือบทั้งหมด คือ

หลักการของอัลกอริทึมอะพริออริ แต่สิ่งที่แตกต่างกันคือจะมีข้อจำกัดในการสร้างกฎเพิ่มขึ้น เนื่องจากเทคนิคการจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์นั้น จะสนใจเฉพาะกฎความสัมพันธ์ที่ทางด้านขวามือเป็น Class เท่านั้น ซึ่งเรียกว่า กฎความสัมพันธ์แบบมีคลาส (class association rules : CARs) [4] คือกฎความสัมพันธ์ที่สับเซตของกฎทางด้านขวามือจะต้องเป็นแอตทริบิวท์ class เท่านั้นรูปแบบของกฎความสัมพันธ์แบบมีคลาส

การทำงานในส่วนที่สองได้แก่ ส่วนของการสร้างแบบจำลองเพื่อทำนายกลุ่มให้กับข้อมูล (Classifier Builder : CB) หลังจากได้เซตของกฎความสัมพันธ์แบบมีคลาสทั้งหมดแล้วจากขั้นตอนแรก ให้นำเซตของกฎความสัมพันธ์ที่ได้นั้นมาทำการเรียงศักย์ของกฎ และสร้างแบบจำลองเพื่อใช้ในการทำนายต่อไป ภาพรวมการทำงานของเทคนิคการจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์แสดงดังรูปที่ 2.1



รูปที่ 2.1 ภาพรวมการทำงานของเทคนิคการจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์

2.1.1 อัลกอริทึม CBA (Classification Based on Association rules) [4]

อัลกอริทึม CBA เป็นอัลกอริทึมต้นแบบของเทคนิคการจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์ซึ่งจะประกอบไปด้วยสองส่วนคือ ส่วนของการสร้างกฎความสัมพันธ์ เรียกว่า CBA-RG (Classification Based on Association Rules – Rule Generator) ดังแสดงในรูปที่ 2.2 และส่วนของการสร้างแบบจำลองในการทำนายเรียกว่า CBA-CB (Classification Based on Association Rules – Classifier Building) ดังแสดงในรูปที่ 2.3

```
1. F1 = {large 1-ruleitems};
2. CAR1 = genRules(F1);
3. prCAR1 = pruneRules(CAR1);
4. for (k = 2; Fk-1 ≠ Ø; k++) do
5.     Ck  = candidateGen (Fk-1);
6.     for each data case d∈D do
7.       Cd = ruleSubset(Ck, d);
8.       for each candidate c∈Cd do;
9.         c.condsupCount++;
10.              if d.class = c.class then c.rulesupCount++
11.        end
12.    end
13.    Fk = {c∈Ck | c.rulesupCount ≥ minsup};
14.    CARk = genRules(Fk);
15.    prCARk = pruneRules(CARk);
16. end
17. CARs = Uk CARk ;
18. prCARs =  UkprCARk ;
```

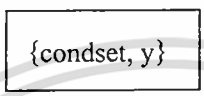
รูปที่ 2.2 อัลกอริทึม CBA-RG

```
1.  R = sort (R);
2.  for each rule r ϵ R in sequence do
3.     temp =  Ø;
4.     for each case d∈D do
5.        if d satisfies the conditions of r then
6.           store d.id in temp and mark r if it correctly
             Classifies d;
7.     if r is marked then
8.        insert r at the end of C;
9.        delete all the cases with the ids in temp from D;
10        selecting a default class for the current C;
11.       compute the total number of errors of C;
12.    end
13. end
14. Find the first rule p in C with the lowest total
    number of errors and drop all the rules after p in C;
15. Add the default class associated with p to end of C,
    And return C (our classifier).
```

รูปที่ 2.3 อัลกอริทึม CBA-CB

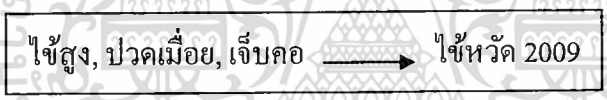
2.1.2 อัลกอริทึม CBA-RG

แนวคิดพื้นฐานที่ใช้ในอัลกอริทึมส่วนของ CBA-RG มีจุดมุ่งหมายในการค้นหา ruleitem ทั้งหมดที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำซึ่งจะเรียกว่า large-ruleitem หลังจากนั้นจะตรวจสอบค่าความเชื่อมั่นของ ruleitem ทั้งหมด ruleitem ใดไม่ผ่านค่าความเชื่อมั่นขั้นต่ำจะถูกพรุนทิ้งไป ผลลัพธ์ที่ได้จากอัลกอริทึม CBA-RG คือเซตของกฎความสัมพันธ์แบบมีคลาสที่มีค่าความเชื่อมั่นมากกว่าหรือเท่ากับค่าความเชื่อมั่นขั้นต่ำที่ได้กำหนดไว้ รูปแบบของ ruleitem ของกฎความแบบมีคลาสแสดงดังรูปที่ 2.4



รูปที่ 2.4 รูปแบบ ruleitem ของกฎความสัมพันธ์แบบมีคลาส

เมื่อ condset เป็นเซตของ ruleitem และ y คือคลาสปลายทาง (class label) ตัวอย่างของกฎความสัมพันธ์แบบมีคลาสเป็นแสดงดังรูปที่ 2.5



รูปที่ 2.5 ตัวอย่างกฎความสัมพันธ์แบบมีคลาส

จากตัวอย่าง ความหมายของกฎความสัมพันธ์คือ ถ้ามีอาการไข้สูง ร่วมกับการปวดเมื่อยและเจ็บคอ จะจัดว่าอยู่ในกลุ่มของผู้ป่วยไข้หวัด 2009

แต่ละ ruleitem จะมีค่า threshold สองค่าหลักๆคือ ค่าสนับสนุน (support) และค่าความเชื่อมั่น (confidence) โดยที่ค่าสนับสนุนสามารถหาได้จากสมการที่ 2.1

$$(\text{rulesupCount} / |D|) * 100\% \qquad (2.1)$$

และ ค่าความเชื่อมั่น(confidence) หาได้จากสมการที่ 2.2

$$(\text{rulesupCount} / \text{condsupCount}) * 100\% \qquad (2.2)$$

โดยที่

- support count of the ruleitem (rulesupCount) คือ จำนวนครั้งของทรานแซคชั่นที่เกิด condset และคลาสปลายทางเป็น y พร้อมกัน
- |D| คือ ขนาดของฐานข้อมูลหรือจำนวนแถวของข้อมูล (Transactions)

- support count of the condset (condsupCount) คือ จำนวนครั้งของทรานแซคชั่นที่เกิด condset ขึ้นในฐานข้อมูล

ตารางที่ 2.1 ตัวอย่างข้อมูลคนไข้

Female	High	True	3	Well
Female	High	False	1	Well
Male	Low	True	2	Well
Female	High	True	2	Sick
Male	High	True	3	Sick

กำหนดให้ ค่า Minimum support = 25% ค่า Minimum confidence = 50%

ขั้นตอนการทำงานของอัลกอริทึม CBA-RG มีดังต่อไปนี้

- ทำการค้นหา 1-ruleitem ทั้งหมด จากตัวอย่างข้อมูลคนไข้ในตารางที่ 2.1 สามารถหา 1-ruleitem ได้ดังแสดงในตารางที่ 2.2

ตารางที่ 2.2 1- ruleitem ทั้งหมด

1-ruleitem		rulesupCount	support
Male	Well	1	20%
Male	Sick	1	20%
Female	Well	2	40%
Female	Sick	1	20%
High	Well	2	40%
High	Sick	2	40%
Low	Well	1	20%
True	Well	2	40%
True	Sick	2	40%
False	Well	1	20%
1	Well	1	20%
2	Well	1	20%
2	Sick	1	20%
3	Well	1	20%
3	Sick	1	20%

จากตารางที่ 2.2 จะเห็นว่า ruleitem ที่ถูกแรเงาคือ large 1-ruleitem เนื่องจากมีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ สำหรับ 1-ruleitem ที่ไม่ถูกแรเงาจะไม่นำมาพิจารณาต่อไป large 1-ruleitem ทั้งหมดแสดงดังตารางที่ 2.3

ตารางที่ 2.3 large 1-ruleitem ทั้งหมด

large 1-ruleitem		rulesupCount	support
Female	Well	2	40%
High	Well	2	40%
High	Sick	2	40%
True	Well	2	40%
True	Sick	2	40%

สำหรับกรณีที่ ruleitem มี condset เหมือนกัน จะทำการเลือกกฎความสัมพันธ์ที่มีค่าความเชื่อมั่นสูงที่สุด (highest confidence) เรียกว่า Possible rule(PR) เป็นตัวแทนกลุ่มของ ruleitem ที่เหลือ เช่น

จากรูปที่ 2.6 $ruleitem_1$ ประกอบด้วย แอตทริบิวท์ Sex มีค่า ruleitem เป็น Male และแอตทริบิวท์ Cholesterol มีค่า ruleitem เป็น High และ $ruleitem_2$ ประกอบด้วย แอตทริบิวท์ Sex มีค่าruleitem เป็น Male และแอตทริบิวท์ Cholesterol มีค่า ruleitem เป็น Highซึ่ง ruleitem ทั้งสอง ruleitem มีค่า condset เหมือนกัน แต่มีคลาสปลายทางต่างกันคือ มีคลาสปลายทางเป็น Well และ Sick

$ruleitem_1$= {(Sex, Male) , (Cholesterol, High)}, {class : Well}

$ruleitem_2$= {(Sex, Male) , (Cholesterol, High)}, {class : Sick}

รูปที่ 2.6 ruleitem ที่มี condset เหมือนกัน แต่มีคลาสปลายทางต่างกัน

กำหนดให้

ค่า condsupCount เท่ากับ 3

ค่า rulesupCount ของ $ruleitem_1$ เท่ากับ 2

ค่า rulesupCount ของ $ruleitem_2$ เท่ากับ 1

ขนาดของฐานข้อมูล (D) เท่ากับ 10

เมื่อคำนวณค่าความเชื่อมั่นตามสมการที่ 2.2 จะได้ค่าความเชื่อมั่นดังนี้

สำหรับ ruleitem$_1$ จะได้ค่าความเชื่อมั่นเท่ากับ (2/3) *100% = 66.7%

สำหรับ ruleitem$_2$ จะได้ค่าความเชื่อมั่นเท่ากับ (1/3) * 100% = 33.3%

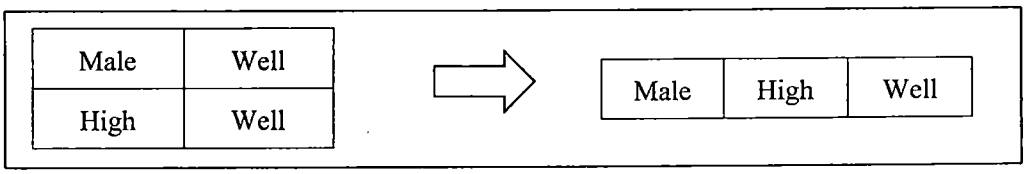
จากผลลัพธ์ที่ได้แสดงให้เห็นว่า ruleitem$_1$มีค่าความเชื่อมั่นสูงกว่า ruleitem$_2$ ดังนั้น ruleitem$_1$ คือ {(Sex, Male) , (Cholesterol, High)}, {class : Well} [support = 20%, confidence = 66.7%] จะถูกเลือกไปสร้างกฎความสัมพันธ์ต่อไป โดยที่กฎความสัมพันธ์นั้นจะต้องมีค่าความเชื่อมั่นมากกว่าค่าความเชื่อมั่นขั้นต่ำ (Minimum confidence)

หลังจากนั้นนำ large 1-ruleitem ในระดับที่หนึ่งไปสร้างเป็นกฎความสัมพันธ์แบบมีคลาส (CARs) และ prune กฎความสัมพันธ์ที่มีค่าความเชื่อมั่นไม่ผ่านค่าความเชื่อมั่นขั้นต่ำทิ้งดังแสดงในตารางที่ 2.4

ตาราง 2.4 กฎความสัมพันธ์แบบมีคลาสระดับที่หนึ่ง

CARs (1-ruleitem)	support	confidence
Female → Well	40%	67%
High → Well	40%	50%
High → Sick	40%	50%
True → Well	40%	50%
True → Sick	40%	50%

- ขั้นตอนต่อไปทำการค้นหาตั้งแต่ large 2-ruleitem ขึ้นไปจนถึง large k-ruleitem สำหรับการตั้งแต่ large 2-ruleitem ขึ้นไปจะต้องทำการสร้าง candidate itemset ก่อนโดยการนำเอา ruleitem ระดับก่อนหน้ามา join กัน ซึ่งตามหลักการ join ของอัลกอริทึม CBA นั้น ruleitem ที่สามารถ join กันได้จะต้องเป็น ruleitem ที่อยู่ในคลาสเดียวกันและไม่อยู่ใน แอตทริบิวท์เดียวกัน พิจารณาจากตารางที่ 2.7 จะเห็นว่า ruleitem (Male , Well) ไม่สามารถทำการ join กับ ruleitem (Female , Well) ได้ แม้ว่าจะอยู่ในคลาสเดียวกันก็ตาม เนื่องจาก Male และ Female เป็น ruleitem ที่มาจากแอตทริบิวท์เดียวกัน และ ruleitem (Male , Well) ก็ไม่สามารถทำการ join กับ ruleitem (High , Sick) ได้เช่นกัน เนื่องจากเป็น ruleitem ที่ไม่ได้อยู่ในคลาสเดียวกัน แต่ ruleitem (Male , Well) จะสามารถทำการ join กับ ruleitem (High , Well) ได้ โดยเมื่อทำการ join แล้ว ผลลัพธ์ที่ได้คือ (Male, High, Well) ดังแสดงในรูปที่ 2.7



รูปที่ 2.7 วิธีการ join ruleitem

นำ large ruleitem จากตารางที่ 2.4 มา join กันจะได้ผลลัพธ์ดังตารางที่ 2.5 ซึ่งเรียกว่า candidate 2-ruleitem (C_2)

ตารางที่ 2.5 candidate 2- ruleitem (C_2)

2- ruleitem			rulesupCount	support
Famale	High	Well	2	40%
Female	True	Well	1	20%
High	True	Well	1	20%
High	True	Sick	2	40%

หลังจากนั้นทำการตรวจสอบ ruleitemใน candidate itemset ว่าเป็นสมาชิกใน large ruleitem ในระดับก่อนหน้าหรือไม่ หากไม่เป็นสมาชิกใน large ruleitem ในระดับก่อนหน้าแสดงว่า large ruleitem นั้นก็จะไม่สามารถเป็น large ruleitem ในระดับต่อไปได้ จะไม่นำ ruleitem นั้นมาพิจารณา

เมื่อได้ candidate itemset แล้ว อัลกอริทึมจะทำการคำนวณค่าสนับสนุนของแต่ละ ruleitem เพื่อนำไปใช้ในการสร้าง large-itemset ต่อไป โดยจะพิจารณาค่าสนับสนุนของ ruleitem ทุกตัวใน candidate itemset ว่ามี ruleitem ตัวใดบ้างที่มีค่าสนับสนุนมากกว่าค่าสนับสนุนขั้นต่ำ จากตารางที่ 2.5 จะเห็นว่า ruleitem ที่ถูกแรเงา คือ ruleitem ที่มีค่าสนับสนุนมากกว่าค่าสนับสนุนขั้นต่ำ ดังนั้น large 2- ruleitem จะแสดงได้ดังตารางที่ 2.6

ตารางที่ 2.6 large 2- ruleitem ทั้งหมด

large 2-ruleitem			rulesupCount	support
Female	High	Well	2	40%
High	True	Sick	2	40%

จากตารางที่ 2.6 จะเห็นว่าไม่มี ruleitemใด join กันตามหลักการทำงานของอัลกอริทึม CBA-RG ได้เลย ดังนั้นอัลกอริทึมจะไม่วนลูปต่อไปอีกเนื่องจากไม่สามารถทำการหา candidate itemset ได้ หลังจากนั้นนำ large 2- ruleitem ไปสร้างเป็นกฎความสัมพันธ์แบบมีคลาส (CARs) และ prune กฎความสัมพันธ์ที่มีค่าความเชื่อมั่นไม่ผ่านค่าความเชื่อมั่นขั้นต่ำทิ้งดังแสดงในตารางที่ 2.7 จะเห็นว่าทุกๆกฎความสัมพันธ์ผ่านค่าความเชื่อมั่นทั้งหมด ดั้งนั้นทุกกฎความสัมพันธ์จะถูกเลือก

ตาราง 2.7 กฎความสัมพันธ์แบบมีคลาสระดับที่สอง

CARs (2-ruleitem)	support	confidence
Female , High → Well	40%	67%
High , True → Sick	40%	67%

- นำกฎความสัมพันธ์แบบมีคลาสตั้งแต่ระดับที่สองขึ้นไปมารวมกันทุกระดับแล้ว พิจารณา condset ของทุก ruleitem สำหรับกรณีที่ ruleitem มี condset เหมือนกัน จะทำการเลือกกฎความสัมพันธ์ที่มีค่าความเชื่อมั่นสูงที่สุด (highest confidence) เรียกว่า Possible rule(PR) เป็นตัวแทนกลุ่มของ ruleitem ที่เหลือ เช่นเดียวกับการทำในรอบที่หนึ่งหลังจากนั้นทำการ prune กฎความสัมพันธ์ที่มีค่าความเชื่อมั่นน้อยกว่าค่าความเชื่อมั่นขั้นต่ำทิ้ง เมื่อพิจาณาตารางที่ 2.7 จะเห็นได้ว่าทุกกฎความสัมพันธ์มีค่าความเชื่อมั่นมากกว่าค่าความเชื่อมั่นขั้นต่ำทั้งหมด

ผลลัพธ์ของอัลกอริทึม CBA-RG คือเซตของกฎความสัมพันธ์แบบมีคลาสจากตัวอย่างข้อมูลคนไข้แสดงดังตารางที่ 2.8

ตารางที่ 2.8 กฎความสัมพันธ์แบบมีคลาสทุกระดับ

	CARs	support	confidence
r_1	Female → Well	25%	67%
r_2	High → Well	25%	50%
r_3	High → Sick	25%	50%
r_4	True → Well	25%	50%
r_5	True → Sick	25%	50%
r_6	Female , High → Well	25%	67%
r_7	High , True → Sick	25%	67%

2.1.3 อัลกอริทึม CBA-CB

เป็นอัลกอริทึมที่ใช้สำหรับการสร้างแบบจำลองจากกฎความสัมพันธ์แบบมีคลาสซึ่งเป็นผลลัพธ์ที่ได้จากอัลกอริทึม CBA-RG โดยอัลกอริทึม CBA-CB จะมีขั้นตอนการทำงาน ดังนี้

- ทำการเรียงศักย์ของกฎความสัมพันธ์ทั้งหมดโดยมีหลักการเรียงศักย์ดังนี้

$r_i \succ r_j$ (r_i มีศักย์สูงกว่า r_j) ได้ก็ต่อเมื่อ

- ค่าความเชื่อมั่น (confidence) ของ r_i มากกว่า r_j หรือ
- ค่าความเชื่อมั่น (confidence) เท่ากัน แต่ค่าสนับสนุน (support) r_i มากกว่า r_j หรือ
- ทั้งค่าความเชื่อมั่น(confidence)และค่าสนับสนุน(support)เท่ากันแต่ r_i ถูกสร้างมาก่อน r_j

ตารางที่ 2.9 ผลลัพธ์ของการเรียงศักย์ของกฎความสัมพันธ์

	CARs	support	confidence
r_1	Female → Well	25%	67%
r_6	Female , High → Well	25%	67%
r_7	High , True → Sick	25%	67%
r_2	High → Well	25%	50%
r_3	High → Sick	25%	50%
r_4	True → Well	25%	50%
r_5	True → Sick	25%	50%

- ขั้นตอนต่อไปทำการวนลูปพิจารณาข้อมูลในฐานข้อมูลทีละตัวร่วมกับกฎความสัมพันธ์ทีละกฎ ว่ากฎความสัมพันธ์สามารถรองรับ (Satisfy) ตัวข้อมูลทางด้านซ้าย (condset) และทางด้านขวา (class) หรือไม่ ได้ โดยกฎความสัมพันธ์ที่มีศักย์สูงที่สุดจะถูกเลือกมาพิจารณาก่อน ถ้าตัวข้อมูลนั้นถูกรองรับโดยกฎความสัมพันธ์ จะทำการเพิ่มข้อมูลตัวนั้นเข้าไปที่ temp พร้อมทั้ง mark กฎความสัมพันธ์ที่พิจารณาร่วมกัน หลังจากนั้นให้เพิ่มกฎความสัมพันธ์เข้าไปในเซต C และทำการลบข้อมูลที่ถูกเพิ่มเข้าไปใน temp โดยจะลบออกจากฐานข้อมูล D เมื่อลบเสร็จเรียบร้อยแล้ว ให้หา default class เพื่อใช้ทำนายให้กับข้อมูลที่ไม่มีกฎใดๆรองรับได้เลย พร้อมทั้งทำการหาค่าเปอร์เซ็นต์ความผิดพลาดด้วย และสำหรับข้อมูลที่กฎความสัมพันธ์สามารถรองรับได้ทั้งซ้ายและขวาแล้ว จะไม่ถูกนำมาพิจารณาในรอบต่อไป วนลูปจนครบทุกกฎความสัมพันธ์หรือจนกว่าจะไม่มีข้อมูลให้พิจารณา

ตารางที่ 2.10 วิธีพิจารณาการรองรับข้อมูลของกฎความสัมพันธ์

Female	High	True	3	Well
Female	High	False	1	Well
Male	Low	True	2	Well
Female	High	True	2	Sick
Male	High	True	3	Sick

จากตารางที่ 2.10 พิจารณาที่ Female , Well และกฎความสัมพันธ์ r_1 จะพบว่ากฎความสัมพันธ์สามารถรองรับตัวข้อมูลได้ทั้งทางซ้ายและทางขวา (พิจารณาส่วนที่แรเงา) ดังนั้นกฎความสัมพันธ์ r_1 จะถูกเพิ่มเข้าไปในเซต C และ ฐานข้อมูลจะเป็นดังนี้

ตารางที่ 2.11 training data หลังจากพิจารณาการรองรับข้อมูลของกฎความสัมพันธ์แล้ว

Male	Low	True	2	Well
Female	High	True	2	Sick
Male	High	True	3	Sick

จากตารางที่ 2.11พบว่า ข้อมูลถูกลบไปสองทรานแซคชั่น และเซต C มีค่าดังนี้ ทำการหา default class และคำนวณค่าความผิดพลาดจากฐานข้อมูลที่เหลือ ในที่นี้จะพบว่า default class เป็น Sick ดังนั้นเซต C ซึ่งเป็นเซตของแบบจำลอง จะมีค่าเป็นดังนี้ $C = \{r_1$, default class = Sick , error = 33%} หลังจากนั้นวนรอบทำซ้ำจนครบทุกกฎความสัมพันธ์ เซตของแบบจำลองจะเปลี่ยนแปลงไปเรื่อยๆจนกว่าจะทำการวนรอบครบทุกกฎความสัมพันธ์

อัลกอริทึมจะคำนวณค่าความผิดพลาดของแบบจำลองที่ได้ทุกๆรอบ เมื่อพบว่ากฎความสัมพันธ์ใดทำให้ค่าความผิดพลาดของแบบจำลองเพิ่มขึ้น กฎความสัมพันธ์นั้นจะไม่ถูกเพิ่มเข้าไปในเซตของแบบจำลองและจะหยุดทำการวนรอบทันที ผลลัพธ์ของแบบจำลองที่ได้แสดงดังรูปที่ 2.8

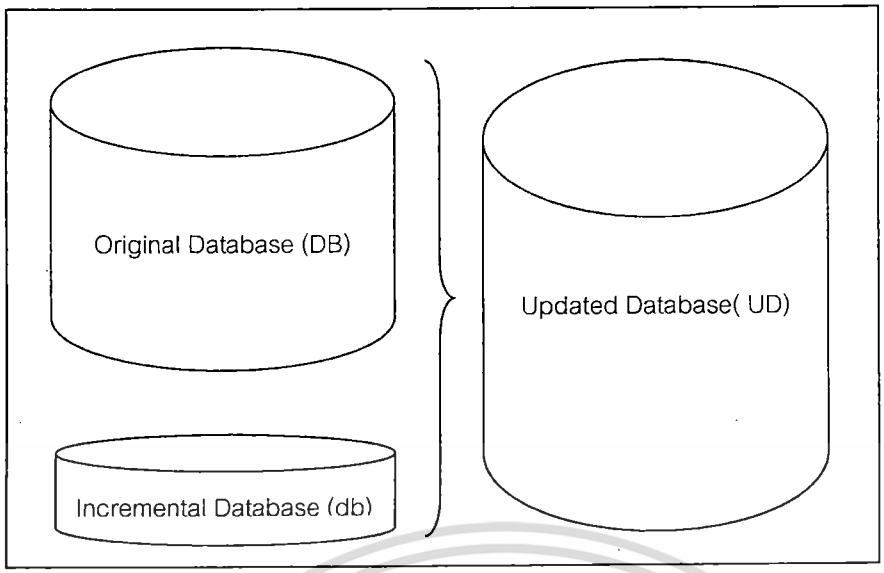
$R = \{r_1, r_3, r_4$, default class = Sick$\}$ error= 15%

รูปที่ 2.8 ตัวอย่างแบบจำลองที่ใช้สำหรับทำนาย

2.2 การค้นหากฎความสัมพันธ์ของการเพิ่มขยายข้อมูล (Incremental Association Rule Discovery) [5]

การค้นหากฎความสัมพันธ์ของการเพิ่มข้อมูล เป็นการค้นหากฎความสัมพันธ์เนื่องจากมีการเพิ่มขึ้นข้อมูลใหม่เข้าสู่ฐานข้อมูลทำให้มีผลต่อการปรับปรุงค่าสนับสนุนของ itemset ในฐานข้อมูล อาจมีผลทำให้กฎความสัมพันธ์เดิมที่เคยเป็นกฎความสัมพันธ์ที่แข็งแกร่งเปลี่ยนเป็นกฎความสัมพันธ์ที่อ่อนแอหรืออาจทำให้กฎความสัมพันธ์ที่อ่อนแอเปลี่ยนเป็นกฎความสัมพันธ์ที่แข็งแกร่ง ทำให้ต้องมีการค้นหากฎความสัมพันธ์ใหม่เพื่อให้กฎความสัมพันธ์ถูกต้องอยู่เสมอ และในการค้นหาค่าสนับสนุนของ itemsetในฐานข้อมูลเก่า จะใช้เวลาในการค้นหามากเพราะโดยทั่วไปแล้วฐานข้อมูลเก่ามีขนาดใหญ่ และฐานข้อมูลที่เพิ่มใหม่จะมีขนาดเล็กกว่า

สำหรับฐานข้อมูลเพิ่มใหม่ หรือ Incremental Database (db) เมื่อเพิ่มรวมเข้ากับฐานข้อมูลเดิม หรือ Original Database (DB) จะเรียกว่าฐานข้อมูลปรับปรุง หรือ Update Database (UD' : DB+db) ดังแสดงในรูปที่ 2.9 ภายหลังการเพิ่มข้อมูลเข้าสู่ฐานข้อมูลอาจทำให้ large itemset เกิดการเปลี่ยนแปลงไปจากเดิมที่เคยค้นหาความสัมพันธ์ไว้



รูปที่ 2.9 ฐานข้อมูลที่มีการเปลี่ยนแปลง

การค้นหากฎความสัมพันธ์ด้วย FUP Algorithm

อัลกอริทึม FUP เป็นอัลกอลิทึมสำหรับการค้นหากฎความสัมพันธ์เมื่อมีการเพิ่มข้อมูลใหม่เข้าสู่ฐานข้อมูล และเป็นอัลกอริทึมแรกที่นำเสนอการเพิ่มข้อมูลเข้าสู่ฐานข้อมูล โดยอัลกอลิทึมนี้มีจุดมุ่งหมายคือการลดการค้นหาในฐานข้อมูลเดิมที่มีขนาดใหญ่ ซึ่งจะเน้นให้ความสนใจในข้อมูลใหม่ที่เพิ่มเข้าสู่ฐานข้อมูล โดยนำความรู้ที่เคยได้จากการทำไมนิ่งค้นหาค่า large itemset ก่อนหน้าการเพิ่มข้อมูลเข้าสู่ฐานข้อมูลมาใช้ประโยชน์ เพื่อลดจำนวนการค้นหา itemset ในทุกทรานแซคชั่นที่มีในฐานข้อมูลทั้งหมดเพื่อนับค่าสนับสนุนของแต่ละ itemset จะแตกต่างจากอัลกอริทึมอะพิโอริที่ต้องทำการค้นหาจำนวนของแต่ละ itemset โดยการเข้าไปค้นหาในฐานข้อมูลทั้งหมด แม้ว่าจะมีการเพิ่มข้อมูลเข้าสู่ฐานข้อมูลเข้าไปน้อยมากก็ตามโดยไม่สนใจว่าจะมีความรู้เดิมที่เคยได้จากการไมนิ่งค่า large itemset จากฐานข้อมูลเดิมอยู่ ที่สามารถนำมาใช้ให้เกิดประโยชน์ต่อได้อีก เพราะในบางกรณีอัลกอริทึม FUP อาจไม่ต้องทำการสแกนหาในฐานข้อมูลเดิมซ้ำ

ในส่วนของสมาชิก candidate itemsets การทำงานของอัลกอริทึม FUP จะมีการเก็บเฉพาะ itemset ของฐานข้อมูลเพิ่มใหม่ โดยจัดเก็บ itemset ที่ไม่เป็นสมาชิกของ large itemsets ของฐานข้อมูลเดิมที่เคยได้ทำการไมนิ่งค้นหากฎความสัมพันธ์มาแล้ว ซึ่งในอัลกอริทึมอะพิโอริจะเป็นการสร้าง candidate itemsets จากฐานข้อมูลเดิมและข้อมูลเพิ่มใหม่ทำให้ได้ candidate itemsets เป็นจำนวนมาก เพราะจะรวมเอาทั้ง itemset ที่มีอยู่แล้วและ itemset ที่มีใหม่มารวมกันทำให้มี itemset ที่ต้องทำการสแกนหาหลายครั้ง ซึ่งในส่วนนี้อัลกอริทึม FUP สามารถลดการสแกนค้นหาในฐานข้อมูลได้ เพราะโดยส่วนใหญ่ฐานข้อมูลเพิ่มใหม่นั้นมีจำนวนข้อมูลที่น้อยกว่าฐานข้อมูลเดิม

อัลกอริทึม FUP มีการพิจารณาในส่วนของ itemset ที่ไม่สามารถเป็น large itemsets โดยนำความรู้จาก large itemsets ของฐานข้อมูลเดิมนำมาพิจารณาร่วมกับ large itemsets ใหม่ เพื่อเป็นการลดการค้นหาในฐานข้อมูลเพิ่มใหม่ในขั้นตอนถัดไป

การทำงานของอัลกอริทึม FUP เพื่อการค้นหากฎความสัมพันธ์ยังอยู่บนพื้นฐานของอัลกอริทึมอะพริออริ โดยจะมีการวนรอบการทำงานซ้ำเพื่อหาความสัมพันธ์ของข้อมูลจาก 1-itemse ไปจนถึง k-itemset อัลกอริทึม FUP นี้จะใช้ค่าสนับสนุนขั้นต่ำเดียวกันทั้งหมดเหมือนกับอัลกอริทึมอะพริออริ และอัลกอริทึม FUP ยังได้นำเอาขั้นตอนการ join ของอัลกอริทึมอะพริออริมาใช้เพื่อสร้างความสัมพันธ์ของitemset และได้นำการทำงานบางส่วนที่เป็นจุดด้อยของอัลกอริทึมอะพริออริมาปรับปรุง เพื่อทำให้การค้นหากฎความสัมพันธ์ของข้อมูลมีประสิทธิภาพมากกว่าอัลกอริทึมอะพริออริ

```
Input: DB: the original database (with its size, i.e., the total number of transaction,
           equal to D);
       L_k: the set of all large k- itemsets in DB, where k= 1, …, r;
       db: an increment database (with its size equal to d);
Output: L': The set of all large itemsets in DB ∪ db.
Method: The 1st iteration:/* find L'_1, the set of all large 1-itemsets in DB ∪ db */
    W = L_1; C = φ ; L'_1 = φ ; P = φ ;/* W: winners, C: candidate sets,
    L'_1: initialized, P: for optimization */
    for all T ∈ db do /* scan db */
      for all 1-itemset X ⊆ T do {
        if X ∈ W then X.support_d++;
        else {
          if X ∉ C
           then { C = C ∪ {X}; X.support_d = 0;}
          X.supprt_d++;}/*init the support cont and add X into C */
        }
    for all X ∈ W do /* put winners into L'_1 */
      if X.support_UD ≥ s ×(D + d)
       then L'_1 = L'_1 ∪{X};
    for all X ∈ C do /*prune candidate sets in C*/
      if X.support_d < s ×d
       then { C = C – {X}; P = P ∪{X};}/* P will be used for optimization */
    for all T ∈ DB do /* scan DB */
      for all 1-itemset X ⊆ T do {
        if X ∈ C then X.support_D++;
        if X ∈ P then removes X from T; /* Transaction T is reduced */
      };
    for all X ∈ C do /* put winners into L_1' */
      if X.support_UD ≥ s × (D + d)
       then L'_1 = L'_1 ∪{X};
    return L'_1. /* end of the 1st iteration */
```

รูปที่ 2.10 อัลกอริทึม FUP สำหรับหา large 1-itemset

จากอัลกอริทึม FUP สำหรับการค้นหา large 1- itemset (L'_1) ดังแสดงในรูปที่ 2.10 สามารถอธิบายรายละเอียดขั้นตอนการทำงานได้ดังนี้

- ขั้นตอนแรกจะทำการสแกนค้นหา itemsetในฐานข้อมูลเพิ่มใหม่ สำหรับทุก itemset ที่เป็นสมาชิกอยู่ภายใน L_1 (large 1-itemsets) ของฐานข้อมูลเดิม เพื่อปรับปรุงค่าสนับสนุนใน itemset ที่กำลังพิจารณา โดยหากว่า itemset นั้นมีค่าสนับสนุนที่น้อยกว่าค่าสนับสนุนขั้นต่ำของฐานข้อมูลปรับปรุง ($X.support_{UD} < s \times (D+d)$) ก็ไม่สามารถผ่านเกณฑ์ไปเป็น itemset ใน L'_1ของฐานข้อมูลปรับปรุง ดังนั้นจะเรียก itemset นั้นว่า loser และถ้า itemset ที่พิจารณามีค่าสนับสนุนที่มากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำของฐานข้อมูลปรับปรุง ($X.support_{UD} \geq s(D+d)$) ดังนั้นจะเรียก itemset ที่ผ่านเกณฑ์ดังกล่าวว่า Winner

- หลังจากทำการสแกนค้นหาในฐานข้อมูลเพิ่มใหม่จากขั้นตอนที่กล่าวมาแล้ว อัลกอริทึมจะทำการสร้าง C_1 (candidate 1-itemset) เพื่อเก็บ itemset ที่ไม่ได้เป็นสมาชิกของ L_1 ในฐานข้อมูลเดิม ($X \notin L_1$) โดย itemset ใดที่เป็นสมาชิกใน C_1 ($X \in C_1$) หากค่าสนับสนุนของ itemsetใดมีค่าสนับสนุนต่ำกว่าค่าสนับสนุนขั้นต่ำของฐานข้อมูลเพิ่มใหม่ ($X.support_d < s \times d$) จะถูกตัดออกไปจาก C_1 เรียก itemset เหล่านั้นว่า loser เพราะว่า itemset นั้นไม่สามารถเป็น itemsetใน L'_1 ได้ จะทำการเก็บ itemset ที่ถูกตัดออกจาก C_1 ไว้ใน P เพื่อนำมาใช้ในการพิจารณาเมื่อมีการสแกนในฐานข้อมูลเดิม จะทำการตัด itemset X ที่เป็นสมาชิกของ P ($X \in P$) ออกจากทรานแซคชั่น T ที่เป็นสมาชิกในฐานข้อมูลเดิม ($T \in DB$) เพราะ itemset นี้ไม่สามารถเป็น large Itemset ได้ในรอบต่อไป

- เมื่อได้ itemset ทั้งหมดใน C_1 นำ itemset ที่มีมาพิจารณาว่าแต่ละ itemset มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำของฐานข้อมูลเพิ่มใหม่หรือไม่($X.support_d < s \times d$) สำหรับ itemset ที่ผ่านเกณฑ์จะนำไปทำการสแกนค้นหาในฐานข้อมูลเดิม เพื่อปรับปรุงค่าสนับสนุน โดยพิจารณาว่ามี itemset ใดที่เหมือนกันกับ itemsetใน C_1 จะนับจำนวนค่าสนับสนุนของแต่ละ itemset นั้นว่ามีค่าเป็นเท่าใด เมื่อได้ค่าสนับสนุนของ itemset ที่ค้นหาในฐานข้อมูลเดิมมารวมกับค่าสนับสนุนของ itemset ใน C_1 แล้วนำค่ามาพิจารณาว่า ถ้าค่าสนับสนุน itemsetใดมีค่ามากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำของฐานข้อมูลปรับปรุง ก็จะปรากฏอยู่ใน L'_1 ถ้าปรากฏว่าค่าสนับสนุนของ itemsetไม่ผ่านค่าสนับสนุนขั้นต่ำของฐานข้อมูลปรับปรุง ($X.support_{UD} \geq s(D+d)$) จะเรียก itemset นั้นว่า Loser

สำหรับตัวอย่างของขั้นตอนการค้นหา large 1-itemset นั้นสามารถแสดงได้ดังรูปที่ 2.11 ถึง 2.12

ตารางที่ 2.12 ตัวอย่างข้อมูลและ large itemset ของ original dataset

DB	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
db	T_{10}	B D
	T_{11}	D F
	T_{12}	A B C D

itemset	support
{A}	6/9
{B}	6/9
{C}	6/9
{E}	4/9
{A,B}	5/9
{A,C}	4/9
{B,C}	4/9
{A,B,C}	4/9

1-itemset (DB)	$support_{DB}$
{A}	6/9
{B}	6/9
{C}	6/9
{E}	4/9

(ก.)

1-itemset (db)	$support_{db}$
{A}	1/3
{B}	2/3
{C}	1/3
{D}	3/3
~~{F}~~	~~1/3~~

(ข.)

รูปที่ 2.11 1-itemset ของ original dataset (ก.) และ 1-itemset ของ increment dataset (ข.)

large 1-itemset	support
{A}	7/12
{B}	8/12
{C}	7/12
{D}	6/12

รูปที่ 2.12 large 1-itemset ทั้งหมด และค่าสนับสนุน

จากรูปที่ 2.11 จะเห็นว่า 1-itemset ที่ถูกแรเงา คือ 1-itemset ที่ไม่ได้เป็นสมาชิกใน large 1-itemset ของฐานข้อมูลเดิม ซึ่งจะต้องทำการพิจารณาค่าสนับสนุนว่ามีค่ามากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำของฐานข้อมูลใหม่หรือ จากรูปแสดงให้เห็นว่า itemset {F} ไม่ผ่านเงื่อนไขดังกล่าว อัลกอริทึมจะไม่นำ itemset {F} มาพิจารณาปรับค่าสนับสนุนต่อไป และจากรูปที่ 2.12 แสดงให้เห็น 1-itemset ทั้งหมดสำหรับฐานข้อมูลที่มีการปรับปรุงซึ่งทำการปรับค่าสนับสนุนเรียบร้อยแล้วโดยจะนำค่าสนับสนุนจากฐานข้อมูลเริ่มต้นและฐานข้อมูลใหม่มารวมกัน

```
The k-th iteration:
 /*for k = 2 or larger, repeat this program fragment to find L'k.
 the set of all large k-itemsets in the  updated database, until either L'k
returned is empty or db = φ */
W = Lk; Lk' = φ ;  /*W: winners; Lk': initialized */
C = apriori_gen1(L'k-1) - Lk;
/* the size-k candidate sets */
for all k-itemset X∈ W do /* prune off losers in W */
  for all (k-1)-itemset Y ∈ Lk-1 – L'k-1 do
    if Y ⊆ X then { W = W – {X}; break;}
for all T ∈ db do {  /* scan db */
  for all X ∈Subset(W,T) do X.supportd++;
 /* Subset(W,T) returns all the sets in W contained in T*/
for all X ∈ Subset(C,T) do X.supportd++; /* find support of all X ∈ C */
Reduce db (T);
/* Some items in transactions in db can be removed, discussed in next
section */
}
for all X ∈ W do  /* put the winners from W into L'k */
  if  X.supportUD ≥ s × ( D+d )
   then L'k = L'k ∪{X};
for all X ∈ C do /* prune candidate sets in C*/
  if X.supportd < s × d then C = C – {X};
for all T ∈ DB do { /* scan DB */
  for all X ∈ Subset(C,T) do X.supportD++
Reduce_DB(T); }
/* Some items in transactions in DB can be removed, discussed in next
section */
for all X ∈ C do
  if X.supportUD ≥ s × (D + d)
   then L'k = L'k∪{X};
return Lk'. /* the end of the k-th iteration */
```

รูปที่ 2.13 อัลกอริทึม FUP สำหรับหาตั้งแต่ large 2-itemset ขึ้นไป

จากรูปที่ 2.13 แสดงอัลกอริทึม FUP สำหรับการค้นหา large itemset ตั้งแต่ 2-itemset ขึ้นไป สามารถอธิบายขั้นตอนการทำงานได้ดังต่อไปนี้

- เมื่อทำในรอบแรกเสร็จแล้วจะได้ L'_1 ของฐานข้อมูลปรับปรุงและจะทำการหา C_2 (candidate 2-itemset) ด้วยการ join L'_1 กับ L'_1 เมื่อเสร็จการ join จะตัด itemset ที่มี itemset ที่เหมือนกับ L_2 ออกจาก C_2 จากนั้นจะมาพิจารณาว่ามี itemset ที่ไม่จำเป็นต้องสแกนบ้าง โดยจะพิจารณาจาก $L_{k-1} - L'_{k-1}$

ตัวอย่าง $L_1 = \{I_1, I_2, I_3\}$ $L_2 = \{I_1I_2, I_2I_3\}$ และ $L'_1 = \{I_1, I_2, I_4\}$

จากตัวอย่างพบว่า item I_1 และ I_2 ของ L_1 ก็อยู่ใน L'_1 มีเพียง I_3 ใน L_1 เพียง item เดียวเท่านั้นที่ไม่เหมือน item ใน L'_1 ดังนั้นจะตัด itemset ที่มี I_3 เป็นสมาชิกที่ itemset นั้นอยู่ใน L_2 ออกไปคือ I_2I_3 เพราะไม่สามารถเป็น large Itemset ได้ ทำให้เหลือเพียง W ก็คือ itemset $\{I_1I_2\}$

- ทำการสแกนฐานข้อมูลเพิ่มใหม่ เพื่อนับค่าสนับสนุนของ itemset (X) ที่เป็นสมาชิกในซับเซตใน W กับ T โดยที่ T เป็นสมาชิกในฐานข้อมูลเพิ่มใหม่ ในขั้นตอนนี้จะทำการค้นหาค่าสนับสนุนitemset X ที่มีอยู่ใน C ภายในฐานข้อมูลเพิ่มใหม่ด้วย

สำหรับทุก X ที่เป็นสมาชิกอยู่ภายใน W จะถูกนำมาพิจารณา ถ้า itemsetใดที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำของฐานข้อมูลปรับปรุง itemset นั้นก็จะปรากฏอยู่ใน L'_k

สำหรับ X ที่เป็นสมาชิกอยู่ภายใน C จะถูกนำมาพิจารณาว่าถ้า itemset นั้นมีค่าสนับสนุนน้อยกว่าค่าสนับสนุนขั้นต่ำของฐานข้อมูลเพิ่มใหม่ ก็จะทำการตัด itemset ดังกล่าวออกจาก C

- ทำการสแกนทุกทรานแซคชั่น T ที่เป็นสมาชิกในฐานข้อมูลเดิม และสำหรับ itemset ที่เป็นสมาชิกอยู่ในสับเซต C กับ T จะนับค่าสนับสนุนของ itemset ดังกล่าวที่มีอยู่ในฐานข้อมูลเดิม สำหรับทุกๆ X ที่เป็นสมาชิกของ C ถ้า itemset ที่พิจารณาใดมีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำของฐานข้อมูลปรับปรุง หาก itemset ที่กำลังพิจารณานั้นผ่านเกณฑ์ค่าสนับสนุนขั้นต่ำดังกล่าวแล้วก็จะปรากฏอยู่ใน L'_k

$L'_1 \bowtie L'_1$	$support_{db}$
{A,B}	1/3
{A,C}	1/3
{A,D}	1/3
{B,C}	1/3
{B,D}	2/3
{C,D}	1/3

L_2	$support_{DB}$
{A,B}	5/9
{A,C}	4/9
{B,C}	4/9

รูปที่ 2.14 $L'_1 \bowtie L'_1$ เปรียบเทียบกับ large 2-itemset (L_2)

จากรูปที่ 2.14 จะเห็นว่า itemset ที่ถูกแรเงาใน $L'_1 \bowtie L'_1$ คือ itemset ที่ซ้ำกับ L_2 อัลกอริทึมจะทำการปรับค่าสนับสนุนโดยนำค่าสนับสนุนที่หาได้จากฐานข้อมูลใหม่ ไปรวมกับค่าสนับสนุนของฐานข้อมูลเดิม ซึ่งจะได้ผลลัพธ์แสดงดังรูปที่ 2.15

2-itemset	$support_{UD}$
{A,B}	6/12
{A,C}	5/12
{B,C}	5/12

รูปที่ 2.15 2-itemset ที่ทำการปรับค่าสนับสนุนแล้ว

และสำหรับ 2-itemset ที่ไม่เป็นสมาชิกใน L_2 จะต้องทำการพิจารณาก่อนว่า ค่าสนับสนุนของ 2-itemset นั้นผ่านค่าสนับสนุนขั้นต่ำในฐานข้อมูลใหม่หรือไม่ อัลกอริทึมจะทำการปรับค่าสนับสนุนเฉพาะ 2-itemset ที่ผ่านค่าสนับสนุนขั้นต่ำในฐานข้อมูลใหม่เท่านั้น

จากรูปที่ 2.16 จะเห็นว่า มี 2-itemset {B,D}เพียงตัวเดียวที่ผ่านค่าสนับสนุนขั้นต่ำ ดังนั้นจะทำการปรับค่าสนับสนุนใหม่ แต่หลังจากปรับค่าสนับสนุนแล้ว 2-itemset {B,D} มีค่าสนับสนุนในฐานข้อมูลปรับปรุงไม่ผ่านค่าสนับสนุนขั้นต่ำที่ได้กำหนดไว้ large 2-itemset แสดงดังรูปที่ 2.17 และอัลกอริทึมจะทำการวนรอบซ้ำจนกว่าจะไม่สามารถหา large-itemset ได้อีก

$(L'_1 \bowtie L'_1) - L_2$	$support_{db}$
{A,D}	1/3
{B,D}	2/3
{C,D}	1/3

$(L'_1 \bowtie L'_1) - L_2$	$support_{UD}$
{B,D}	4/12

รูปที่ 2.16 $L'_1 \bowtie L'_1$ ที่ไม่เป็นสมาชิก ใน L_2

large 2-itemset (L'_2)	$support_{UD}$
{A,B}	6/12
{A,C}	5/12
{B,C}	5/12

รูปที่ 2.17 large 2-itemset ของ updated training data

สำหรับ large itemset ทั้งหมด ของฐานข้อมูลปรับปรุงแสดงได้ดังรูปที่ 2.8

large itemset (L')	$support_{UD}$
{A}	7/12
{B}	8/12
{C}	7/12
{D}	6/12
{A,B}	6/12
{A,C}	5/12
{B,C}	5/12
{A,B,C}	5/12

รูปที่ 2.18 large itemset ทั้งหมดของ updated training data (L')

บทที่ 3
การจำแนกประเภทข้อมูลแบบเพิ่มขยาย โดยใช้กฎความสัมพันธ์

ในปัจจุบันองค์กรต่างๆมีการทำเทคโนโลยีสารสนเทศเข้ามาประยุกต์ร่วมกับการดำเนินงานในแต่ละองค์กร ซึ่งจะช่วยให้การดำเนินงานเป็นไปอย่างสะดวกและรวดเร็ว ฐานข้อมูลเป็นส่วนหนึ่งในการประยุกต์เทคโนโลยีสารสนเทศในองค์กรต่างๆ ข้อมูลของแต่ละองค์กรเป็นจำนวนมากจะถูกจัดเก็บในรูปของฐานข้อมูลขนาดใหญ่ และข้อมูลดังกล่าวนั้นก็จะมีจำนวนเพิ่มมากขึ้นเรื่อยๆ

เนื่องจากการทำเหมืองข้อมูลเป็นการค้นหาความรู้ที่ซ่อนอยู่ในข้อมูล ทำให้เมื่อมีข้อมูลเพิ่มเข้ามาใหม่ความรู้ดังกล่าวนั้นอาจจะเปลี่ยนไป ผู้วิจัยจึงได้เล็งเห็นความสำคัญของการนำความรู้เก่าที่มีอยู่เดิมมาใช้กับฐานข้อมูลที่มีการเปลี่ยนแปลงให้เกิดประโยชน์สูงที่สุด สำหรับงานวิจัยนี้จะเป็นการนำเอาหลักการของการค้นหากฎความสัมพันธ์แบบเพิ่มขยายข้อมูลมาประยุกต์เข้ากับการจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์

กำหนดให้

3.1 การจำแนกประเภทข้อมูลแบบเพิ่มขยายโดยใช้กฎความสัมพันธ์

เนื่องจากมีข้อมูลเพิ่มเข้ามาในฐานข้อมูล (db) ทำให้ความสัมพันธ์ระหว่างข้อมูลมีการเปลี่ยนแปลงเกิดขึ้น จากที่ได้กล่าวไว้แล้วในบทที่2 ว่า การจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์นั้น เป็นการสร้างแบบจำลองจากกฎความสัมพันธ์แบบมีคลาส ซึ่งจะต้องมีการแบ่งข้อมูลออกเป็นสองส่วน ส่วนแรกสำหรับสอนระบบ เรียกว่า training data และส่วนที่ 2 ใช้สำหรับทดสอบแบบจำลองที่สร้างขึ้นมาเรียกว่า testing data เมื่อมีการเพิ่มขยายของข้อมูล จะทำให้ training data เปลี่ยนไป แบบจำลองที่ได้อาจจะไม่ถูกต้อง ดังนั้นจึงจำเป็นที่จะต้องทำการค้นหากฎความสัมพันธ์แบบมีคลาสขึ้นมาใหม่ เพื่อให้กฎความสัมพันธ์และแบบจำลองถูกต้องอยู่เสมอ สำหรับงานวิจัยนี้ได้นำหลักการทำงานของอัลกอริทึม FUP [5] เข้ามาประยุกต์ใช้ทั้งในขั้นตอนการค้นหากฎความสัมพันธ์และขั้นตอนการสร้างแบบจำลอง ซึ่งหลักการของอัลกอริทึม FUP มีจุดมุ่งหมายลดการค้นหาในฐานข้อมูลขนาดใหญ่และนำความรู้เดิมที่มีอยู่มาใช้ประโยชน์สูงสุด

ผู้วิจัยได้นำเสนออัลกอริทึมใหม่สำหรับการจำแนกประเภทข้อมูลแบบเพิ่มขยายโดยใช้กฎความสัมพันธ์ เรียกว่า อัลกอริทึม ICBA (Incremental Classification Based on Association Rules) ซึ่งอัลกอริทึมจะแบ่งการทำงานออกเป็นสองส่วนดังนี้

- ส่วนของการสร้างกฎความสัมพันธ์สำหรับข้อมูลแบบเพิ่มขยาย เรียกว่า Incremental Classification Based on Association Rules – Rule Generator (ICBA-RG)
- ส่วนของการสร้างแบบจำลองสำหรับข้อมูลแบบเพิ่มขาย เรียกว่า Incremental Classification Based on Association Rule – Classifier Building (ICBA-CB)

3.1.1 อัลกอริทึม ICBA-RG

แนวคิดของอัลกอริทึม ICBA-RG คล้ายกับอัลกอริทึม CBA-RG ที่ได้กล่าวไว้แล้วในบทที่ 2 ส่วนที่แตกต่างกันคือ อัลกอริทึม ICBA-RG จะนำหลักการของอัลกอริทึม FUP เข้ามาประยุกต์ใช้งานร่วมกัน

การสแกน training data เพื่อทำการหาค่าสนับสนุนของแต่ละ ruleitem ของอัลกอริทึม ICBA-RG จะแตกต่างกับอัลกอริทึม CBA-RG เนื่องจากอัลกอริทึม CBA-RG จะทำการสแกนทุกๆทรานแซคชั่นใน training data เพื่อทำการหาค่าสนับสนุนโดยไม่สนใจความรู้เดิมที่มีอยู่ ไม่ว่าจะมีการเพิ่มขึ้นของข้อมูลเป็นจำนวนมากหรือน้อยก็ตามซึ่งจะทำให้ใช้เวลามากเกินความจำเป็น แต่สำหรับอัลกอริทึม ICBA-RG จะลดจำนวนครั้งในการสแกนลงด้วยการนำความรู้เดิมที่ได้จาก training data เก่ามาใช้เพื่อให้เกิดประโยชน์สูงสุด อัลกอริทึม ICBA-RG แสดงดังรูปที่ 3.1

```
1   C'1 = large 1-ruleitem in db
2   for each X ∈ C'1 do
3       if X ∈ C'1 and X ∈ F1 do
4           scan db to update X.supportUD then
5               if X.supportUD ≥ s × (D+d) do
6                   insert X at the end of L'1
7       else
8           if X.supportd ≥ (s × d ) do
9               scan UD to update X.supportUD then
10                  if X.supportUD ≥ s × (D+d) do
11                      insert X at the end of L'1
12  end
13  for (k=2; Lk-1 ≠ ∅; k++) do
14      C'k = (L'k-1 ⊗ L'k-1 ) – Lk-1
15      for each X ∈ C'k do
16          if X ∈ C'k and X ∈ Ck do
17          scan db to update X.supportUD then
18              if X.supportUD ≥ s × (D+d) do
19                  insert X at the end of L'k
20      else
21          if X.supportd ≥ (s × d ) do
22              scan UD to update X.supportUD then
23                  if X.supportUD ≥ s × (D+d) do
24                      insert X at the end of L'k
25  end
26  CARs' = pruneRule(L')
```

รูปที่ 3.1 อัลกอริทึม ICBA-RG

จากรูปที่ 3.1 สามารถอธิบายขั้นตอนการทำงานของอัลกอริทึม ICBA-RG ได้ดังนี้

- ทำการค้นหา 1- ruleitem ทั้งหมดในฐานข้อมูลใหม่ แล้วนำไปตรวจสอบว่าเป็นสมาชิกอยู่ใน large 1-ruleitem (L_1) ของ training data เดิมหรือไม่ ถ้า ruleitem นั้นๆ (X) เป็นสมาชิกของ large 1-ruleitem ใน training data เดิมจะทำการปรับค่าสนับสนุนใหม่โดยนำค่าสนับสนุนของ ruleitem นั้นจาก training data เดิมและค่าสนับสนุนที่คำนวณจาก training data ที่เพิ่มใหม่มารวมกัน โดย ถ้า $X.support_{UD} \geq s \times (D+d)$ แล้ว ruleitem นั้นๆจะถูกเรียกว่า "winner" แต่ถ้า $X.support_{UD} < s \times (D+d)$ ruleitem นั้นจะถูกเรียกว่า "loser" โดย ruleitem ที่เป็น winner จะถูกนำไปเก็บไว้ที่ L'_1 และ ruleitem ที่เป็น loser จะถูก prune ทิ้งไป แต่ถ้า ruleitem นั้นไม่เป็นสมาชิกใน large 1-ruleitem ของ training data เดิม ให้ตรวจสอบค่าสนับสนุนของ ruleitem นั้นว่ามีค่ามากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำของ training data ที่เพิ่มใหม่หรือไม่ ($X.support_{UD} \geq s \times d$) ถ้าค่าสนับสนุนของ ruleitem นั้นมีค่าน้อยกว่าค่าสนับสนุนขั้นต่ำของ training data ที่เพิ่มใหม่จะไม่นำ ruleitem นั้นไปพิจารณาต่อ แต่ถ้าค่าสนับสนุนของ ruleitem มากกว่าค่าสนับสนุนขั้นต่ำของ training data ที่เพิ่มใหม่ อัลกอริทึมจะทำการปรับค่าสนับสนุนของ ruleite, ด้วยการนำค่าสนับสนุนของ ruleitem นั้นจาก training data เดิมและค่าสนับสนุนที่คำนวณจาก training data ที่เพิ่มใหม่มารวมกัน เมื่อทำการปรับค่าสนับสนุนเรียบร้อยแล้วจะพิจารณาค่าสนับสนุนของ training data ที่ปรับปรุงว่ามีค่ามากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำหรือไม่ หากค่าสนับสนุนของ ruleitem นั้นมีค่ามากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ $X.support_{UD} \geq s \times (D+d)$ ruleitem นั้นจะถูกเรียกว่า "winner" แต่ถ้า $X.support_{UD} < s \times (D+d)$ ruleitem นั้นจะถูกเรียกว่า "loser" โดย ruleitem ที่เป็น winner จะถูกนำไปเก็บไว้ที่ L'_1 และ ruleitem ที่เป็น loser จะถูก prune ทิ้งไป สามารถแสดงตัวอย่างได้ดังนี้

กำหนดให้

DB = training data เดิม

db = training data ที่เพิ่มใหม่

UD = training data ที่ปรับปรุงแล้ว (DB+db)

d = จำนวนทรานแซคชั่นใน training data ที่เพิ่มใหม่

D = จำนวนทรานแซคชั่นใน training data เดิม

s = ค่าสนับสนุนขั้นต่ำ (20%)

c = ค่าความเชื่อมั่นขั้นต่ำ (50%)

CARs = กฎความสัมพันธ์แบบมีคลาสของ training data เดิม

CARs' = กฎความสัมพันธ์แบบมีคลาสของ training data ที่ปรับปรุง

X = ruleitem

$X.support_d$ = ค่าสนับสนุนของไอเทมเซต X ใน training data ที่เพิ่มใหม่

$X.support_{UD}$ = ค่าสนับสนุนของไอเทมเซต X ใน training data ที่ปรับปรุง

ตารางที่ 3.1 ตัวอย่างข้อมูลคนไข้

	Sex	Cholesterol (250)	Blood sugar < 120	Vessel color number	Status
DB	Male	High	False	1	Sick
	Male	Low	True	3	Well
	Female	Low	False	3	Sick
	Female	High	True	2	Well
	Female	High	False	3	Well
	Male	Low	True	1	Sick
	Female	Low	False	3	Sick
	Female	High	True	3	Well
	Male	Low	False	1	Few
	Female	Low	True	1	Well
	Male	High	True	3	Sick
	Male	Low	True	1	Well
	Male	Low	False	2	Well
	Female	Low	False	2	Few
	Male	Low	False	3	Sick
db	Female	High	True	3	Well
	Female	High	False	1	Well
	Male	Low	True	2	Well
	Female	High	True	2	Sick
	Male	High	True	3	Sick

1 – itemset (DB)		support
Male	Well	3
Male	Sick	4
Female	Well	4
High	Well	3
Low	Well	4
Low	Sick	4
True	Well	5
False	Sick	4
3	Well	3
3	Sick	4

(ก.)

1 – itemset (db)		support
Male	Well	1
Male	Sick	1
Female	Well	2
Female	Sick	1
High	Well	2
High	Sick	2
Low	Well	1
True	Well	2
True	Sick	2
False	Well	1
1	Well	1
2	Well	1
2	Sick	1
3	Well	1
3	Sick	1

(ข.)

รูปที่ 3.2 large 1-ruleitem (L_1) ของ training data เดิม (ก.) และ 1-ruleitem ของ training data ที่เพิ่มใหม่ (ข.)

จากรูปที่ 3.2 ทำการเปรียบเทียบให้เห็นว่า มี 1-ruleitem ตัวใดบ้างที่เป็นสมาชิกของ large 1-ruleitem ใน training data เดิม โดยนำ large 1-ruleitem (L_1) มาลบกับ 1-ruleitem ของ training data ที่เพิ่มใหม่ ได้ผลลัพธ์ดังรูปที่ 3.3 พิจารณารูปที่ 3.3 อัลกอริทึมจะทำการปรับค่าสนับสนุนของ 1-ruleitem ให้เป็นค่าสนับสนุนของ training data ปรับปรุง โดยจะปรับปรุงทั้ง 1-ruleitem ที่มาจาก training data เดิม และ 1-ruleitem ของ training data ที่เพิ่มเข้ามาใหม่ หลังจากนั้นจะทำการพิจารณาค่าสนับสนุนของแต่ละ ruleitem ว่าผ่านค่าสนับสนุนขั้นต่ำที่ได้กำหนดไว้หรือไม่ และสำหรับ 1-ruleitem ที่ไม่ได้เป็นสมาชิกของ large 1-ruleitem ใน training data เดิม จะต้องตรวจสอบค่าสนับสนุนใน training data ใหม่ที่เพิ่มเข้ามาของ 1-ruleitem นั้นก่อนจึงปรับค่าสนับสนุน โดยจะปรับค่าสนับสนุนเฉพาะ 1-ruleitem ที่มีค่าสนับสนุนใน training data ที่เพิ่มใหม่มากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำเท่านั้น

1 – ruleitem		$support_{DB}$	$support_{UD}$
Male	Well	3	4
Male	Sick	4	5
Female	Well	4	6
High	Well	3	5
Low	Well	4	5
Low	Sick	4	4
True	Well	5	7
False	Sick	4	4
3	Well	3	4
3	Sick	4	5

(ก.)

1 – ruleitem		$support_{db}$	$support_{UD}$
Female	Sick	1	3
High	Sick	2	4
True	Sick	2	4
False	Well	1	3
1	Well	1	3
2	Well	1	3
2	Sick	1	1

(ข.)

รูปที่ 3.3 1-ruleitem ที่เป็นสมาชิกของ large 1-ruleitem ใน training data เดิม (ก.) และ 1- ruleitem ที่ ไม่เป็นสมาชิกของ large 1-ruleitem ใน training data เดิม (ข.)

จากรูปที่ 3.3 พบว่าทุก 1-ruleitem สำหรับ training data ที่เพิ่มใหม่ที่มีค่าสนับสนุนผ่านค่าสนับสนุนขั้นต่ำที่กำหนดไว้ อัลกอริทึมจะทำการปรับค่าสนับสนุนให้ 1-ruleitem ทุกตัว

1 – ruleitem		$support_{UD}$
Male	Well	4
Male	Sick	5
Female	Well	6
High	Well	5
High	Sick	4
Low	Well	5
Low	Sick	4
True	Well	7
True	Sick	4
False	Sick	4
3	Well	4
3	Sick	5

รูปที่ 3.4 1-ruleitem ที่ปรับค่าสนับสนุนแล้วทั้งหมด

กำหนดให้ $s \times (D+d) = 0.2 \times (15+5) = 4$

จากรูปที่ 3.4 จะเห็นว่า ทุกๆ 1-ruleitem มีค่า support $\geq s \times (D+d)$ ดังนั้นทุกตัวเป็น winner จึงนำไปเก็บไว้ที่ L'_1 และกฎความสัมพันธ์แบบมีคลาสระดับที่ 1 สำหรับ training data ที่เพิ่มขยาย แสดงดังรูปที่ 3.5

id	(CARs')	confidence
r'_1	Male $\longrightarrow$ Sick	50%
r'_2	Female $\longrightarrow$ Well	60%
r'_3	High $\longrightarrow$ Well	55%
r'_4	True $\longrightarrow$ Well	63%
r'_5	3 $\longrightarrow$ Sick	55%

รูปที่ 3.5 กฎความสัมพันธ์แบบมีคลาสระดับที่ 1 สำหรับ training data เพิ่มขยาย

ในขั้นตอนการสร้างกฎความสัมพันธ์ สำหรับทุกๆ ruleitem ที่มี condset เหมือนกัน จะเลือกเพียงกฎเดียว โดยจะเลือกกฎความสัมพันธ์ที่มีค่าความเชื่อมั่นสูงสุด เช่น

กำหนดให้

1. Male High $\longrightarrow$ Well ค่าความเชื่อมั่นเท่ากับ 30%
2. Male High $\longrightarrow$ Sick ค่าความเชื่อมั่นเท่ากับ 70%

อัลกอริทึมจะทำการเลือกกฎที่ 2 เนื่องจากมีค่าความเชื่อมั่นสูงกว่า และกฎความสัมพันธ์ใดที่มีค่าความเชื่อมั่นน้อยกว่าค่าความเชื่อมั่นขั้นต่ำจะไม่ถูกนำไปใช้ในการสร้างแบบจำลอง

จากรูปที่ 3.6 จะเห็นว่ากฎความสัมพันธ์ใดที่ผ่านค่าความเชื่อมั่นขั้นต่ำทุกกฎ ดังนั้นจะถูกนำไปใช้สำหรับการสร้างแบบจำลองต่อไป

- ขั้นตอนต่อไปทำการหา candidate 2- ruleitem โดยทำเหมือนกับอัลกอริทึม CBA นั่นคือ นำ ruleitem ใน L'_1 มา join กัน ($L' \bowtie L'_1$) สำหรับการ join จะใช้วิธีการเดียวกันกับอัลกอริทึม CBA –RG ได้ผลลัพธ์ดังรูปที่ 3.7 (ก.) จากนั้นนำผลลัพธ์ที่ได้ไปลบ L_2 ออก จะได้ 2-ruleitem ที่ไม่เป็นสมาชิกใน 2-ruleitem ของ training data เดิม แสดงดังรูปที่ 3.7 (ข.) จากนั้นทำการพิจารณาค่าสนับสนุนของ 2-ruleitem ใน training data ที่เพิ่มใหม่ ว่ามีค่ามากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำหรือไม่ หากผ่านค่าสนับสนุนขั้นต่อ อัลกอริทึมจะทำการปรับค่าสนับสนุนให้ 2-ruleitem นั้นๆ เพื่อเป็นค่าสนับสนุนของ training data ที่ปรับปรุงแล้ว เมื่อปรับค่าสนับสนุนแล้ว ทำการค้นหา 2-ruleitem ที่เป็น winner คือ 2-ruleitem ที่มีค่าสนับสนุนมีค่ามากกว่าหรือเท่ากับ $s \times (D+d)$ แล้วนำ 2-ruleitem ที่เป็น winner ไปเก็บไว้ที่เซต L'_2

($L'_1 \otimes L'_1$)	
Male , High	Well
Male , Low	Well
Male , True	Well
Male , 3	Well
Male , High	Sick
Male , Low	Sick
Male , True	Sick
Male , False	Sick
Male , 3	Sick
Female , High	Well
Female , Low	Well
Female , True	Well
Female , 3	Well
High , True	Well
High , 3	Well
High , True	Sick
High , False	Sick
High , 3	Sick
Low , True	Well
Low , 3	Well
Low , True	Sick
Low , False	Sick
Low , 3	Sick
True , 3	Well
True , 3	Sick

(ก.)

Large 2-ruleitem (L_2)	
Female , Low	Well
Female , High	Well
Female , True	Well
Low , True	Well
Low , False	Sick
Low , 3	Sick
False , 3	Sick

(ข.)

รูปที่ 3.6 ผลลัพธ์ของ $L'_1 \otimes L'_1$ (ก.) และ large 2-ruleitem (L_2) ของ training data เดิม (ข.)

จากนั้นพิจารณากรณี $L_{k-1} - L'_{k-1}$ เพื่อตัด ruleitem ที่ไม่สามารถเป็น large ruleitem ได้

กำหนดให้ L_1 = { (High ,Well), (False ,Well), (3 , Well) }

L_2 = {(High, False, Well) , (False, 3, Well)}

L'_1 = {(High ,Well) , (False , Well) , (True,Sick)}

จะเห็นว่า (High ,Well) และ (False, Well) ต่างก็เป็นสมาชิกใน L'_1 มีเพียง (3 , Well) ที่ไม่เป็นสมาชิกใน L'_1 ดังนั้นจะตัด ruleitem ใน L_2 ที่มีสับเซต (3 , Well) ออก เพราะไม่สามารถเป็น large-2-ruleitem ได้ ทำให้ L_2 ที่จะนำมาพิจารณาเหลือเพียง L_2 = (High, False, Well) เมื่อทำการตัด ruleitem ที่มีสับเซตออกทั้งหมดแล้ว อัลกอริทึมจะนำ L_2 ที่เหลือไปปรับค่าสนับสนุน โดยจะทำการสแกน training data เฉพาะส่วนที่เพิ่มใหม่เท่านั้น จากตัวอย่างจะพบว่า $L_{k-1} - L'_{k-1}$ มีค่าเท่ากับเซตว่าง ดังนั้น 2-ruleitem ทุกตัวใน L_2 จะถูกนำไปปรับค่าสนับสนุนทั้งหมด ผลลัพธ์ที่ได้แสดงในรูปที่ 3.7 เมื่อปรับค่าสนับสนุนแล้ว ทำการค้นหา 2-ruleitem ที่เป็น winner คือ 2-ruleitem ที่มีค่าสนับสนุนมีค่ามากกว่าหรือเท่ากับ $s \times (D+d)$ แล้วนำ 2-ruleitem ที่เป็น winner ไปเก็บไว้ที่เซต L'_2 ต่อไป

2 – ruleitem		$support_{DB}$	$support_{UD}$
Male , Low	Well	3	4
Female , High	Well	3	5
Female ,True	Well	3	4
Low , True	Well	3	4
Low , False	Sick	3	3
Low , 3	Sick	3	3
False , 3	Sick	3	3

รูปที่ 3.7 large 2-ruleitem (L_2) ของ training data เดิม และค่าสนับสนุน

large 2 – ruleitem (L'_2)		$support_{UD}$
Male	Well	4
Male	Sick	5
Female	Well	6
High	Well	5

รูปที่ 3.8 large 2-ruleitem (L_2) ของ training data เดิม ที่ทำการปรับค่าสนับสนุนแล้ว

จากรูปที่ 3.7 จะเห็นว่า 2-ruleitem ที่ถูกแรเงา มีค่าสนับสนุนมากกว่าค่าสนับสนุนขั้นต่ำ จะถูกเรียกว่า "winner" และนำไปเก็บไว้ที่ L'_2 ต่อไป L'_2 ทั้งหมดแสดงดังรูปที่ 3.8

id	(CARs')	confidence
r'_6	Male , Low ⟶ Well	50%
r'_7	Female , High ⟶ Well	60%
r'_8	Female , True ⟶ Well	55%
r'_9	Low , True ⟶ Well	63%

รูปที่ 3.9 กฎความสัมพันธ์แบบมีคลาสระดับที่ 2 สำหรับ training data เพิ่มขยาย

ในขั้นตอนการสร้างกฎความสัมพันธ์จะทำเช่นเดียวกับกฎความสัมพันธ์ระดับที่ 1 คือ สำหรับทุกๆ ruleitem ที่มี condset เหมือนกัน จะเลือกเพียงกฎเดียว โดยจะเลือกกฎความสัมพันธ์ที่มีค่าความเชื่อมั่นสูงสุดเท่านั้น และกฎความสัมพันธ์ใดที่มีค่าความเชื่อมั่นน้อยกว่าค่าความเชื่อมั่นขั้นต่ำจะไม่ถูกนำไปใช้ในการสร้างแบบจำลอง กฎความสัมพันธ์แบบมีคลาสทั้งหมด แสดงดังรูปที่ 3.10

id	(CARs')	confidence
r'_1	Male ⟶ Sick	50%
r'_2	Female ⟶ Well	60%
r'_3	High ⟶ Well	55%
r'_4	True ⟶ Well	63%
r'_5	3 ⟶ Sick	55%
r'_6	Male , Low ⟶ Well	57%
r'_7	Female , High ⟶ Well	83%
r'_8	Female , True ⟶ Well	80%
r'_9	Low , True ⟶ Well	100%

รูปที่ 3.10 กฎความสัมพันธ์แบบมีคลาสทุกระดับ สำหรับ training data เพิ่มขยาย

อัลกอริทึม ICBA-RG จะทำการวนรอบซ้ำ เพื่อทำการค้นหา large ruleitem และกฎความสัมพันธ์แบบมีคลาสในทุกระดับ จนกว่าจะไม่สามารถหาได้ และนำกฎความสัมพันธ์แบบมีคลาสในทุกระดับมารวมกันแล้วนำไปสร้างแบบจำลองโดยใช้อัลกอริทึม ICBA-CB

3.1.2 อัลกอริทึม ICBA-CB

เป็นการนำผลลัพธ์ที่ได้จากขั้นตอนของการค้นหากฎความสัมพันธ์ คือเซตของกฎความสัมพันธ์แบบมีคลาสของข้อมูลเพิ่มขยาย (CARs') มาสร้างเป็นแบบจำลองในการทำนาย ตามอัลกอริทึม ICBA-CB แสดงดังรูปที่ 3.11

```
1  CARs' = sort (CARs')
2  for each rule r' ∈ CARs' in sequence do
3      if r' ∈ CARs then
4          for each case x in db do
5              if x satisfies the condition of r' do
6                  store x.id in temp and mark r' if it correctly classifies x;
7          if r' is marked then
8              insert r' at the end of C'
9              delete all the cases with the ids in temp from db;
10             selecting a default class for the current C'
11             compute the total number of errors of C'
12         end
13     else
14     if r' ∉ CARs then
15         for each case x in UD do
16             if x satisfies the condition of r' do
17                 store x.id in temp and mark r' if it correctly classifies x;
18         if r' is marked then
19             insert r' at the end of C'
20             delete all the cases with the ids in temp from UD;
21             selecting a default class for the current C'
22             compute the total number of errors of C'
23         end
24 end
25 Find the first rule p' in C' with the lowest total number of errors and drop
   all the rule after p' in C'
26 Add the default class associated with p' to end of C' and return C'
```

รูปที่ 3.11 อัลกอริทึม ICBA-CB สำหรับสร้างแบบจำลอง

การทำงานของอัลกอริทึม ICBA-CB สามารถอธิบายเป็นขั้นตอนได้ดังต่อไปนี้

- ทำการเรียงศักย์ของกฎความสัมพันธ์แบบมีคลาสใน CARs' โดยมีวิธีในการพิจารณาศักย์ของกฎความสัมพันธ์ดังนี้

$r_i \succ r_j$ (r_i มีศักย์สูงกว่า r_j) ได้ก็ต่อเมื่อ

- ค่าความเชื่อมั่น (confidence) ของ r_i มากกว่า r_j หรือ
- ค่าความเชื่อมั่น (confidence) เท่ากัน แต่ค่าสนับสนุน (support) r_i มากกว่า r_j หรือ
- ทั้งค่าความเชื่อมั่น(confidence)และค่าสนับสนุน(support)เท่ากันแต่ r_i ถูกสร้างมาก่อน r_j

ผลลัพธ์ของการเรียงกฎความสัมพันธ์ตามศักย์ของกฎ แสดงดังรูปที่ 3.12

id	(CARs')	confidence
r'_9	Low , True → Well	100%
r'_7	Female , High → Well	83%
r'_8	Female , True → Well	80%
r'_4	True → Well	63%
r'_2	Female → Well	60%
r'_6	Male , Low → Well	57%
r'_3	High → Well	55%
r'_5	3 → Sick	55%
r'_1	Male → Sick	50%

รูปที่ 3.12 กฎความสัมพันธ์แบบมีคลาสของข้อมูลเพิ่มขยาย (CARs') ที่ทำการเรียงศักย์แล้ว

- เลือกกฎความสัมพันธ์เพื่อสร้างแบบจำลอง ซึ่งจะเลือกกฎความสัมพันธ์ที่มีศักย์สูงที่สุดก่อน จากนั้นจะทำการวนลูปกฎความสัมพันธ์ โดยจะนำกฎความสัมพันธ์ r'_k ไปทดสอบกับกฎความสัมพันธ์แบบมีคลาสของ training data เดิม (CARs) ก่อน ถ้า $r'_k \in$ CARs ให้นำกฎความสัมพันธ์ไปทดสอบกับตัวข้อมูลในส่วนของ training data ที่เพิ่มใหม่เท่านั้น แต่ถ้า $r'_k \notin$ CARs ให้นำกฎความสัมพันธ์ไปทดสอบกับตัวข้อมูลในส่วนของ training data ที่ปรับปรุงแล้ว ซึ่งจะมีวิธีการทดสอบคือ พิจารณาว่า กฎความสัมพันธ์นั้นสามารถรองรับ (Satisfy) ตัวข้อมูลทางด้านซ้าย (condset) และทางด้านขวา (class) หรือไม่ โดยจะต้องวนลูปกฎความสัมพันธ์กับข้อมูลทุกๆทรานแซคชั่น เมื่อทำการวนลูปครบทุกทรานแซคชั่นแล้ว จะนำกฎความสัมพันธ์ไปเก็บไว้ที่ C ' หลังจากนั้นทำการหา default class เพื่อใช้ทำนายให้กับข้อมูลที่ไม่มีกฎใดๆรองรับได้เลย พร้อมทั้งทำการหาค่าเปอร์เซนต์ความผิดพลาดด้วย และสำหรับข้อมูลที่กฎความสัมพันธ์สามารถรองรับได้ทั้งซ้ายและขวาแล้ว จะไม่ถูกนำมาพิจารณาในรอบต่อไป

ตัวอย่าง ในการทำสอบกฎความสัมพันธ์ r'_7 กับข้อมูล

ตารางที่ 3.2 กฎความสัมพันธ์แบบมีคลาสสำหรับ training data เดิม (CARs)

id	Class association rule (CARs)	confidence
r_1	Male ⟶ Sick	50%
r_2	Female ⟶ Well	57%
r_3	High ⟶ Well	60%
r_4	True ⟶ Well	71%
r_5	False ⟶ Sick	50%
r_6	Male , Low ⟶ Well	50%
r_7	Female , High ⟶ Well	100%
r_8	Female , True ⟶ Well	100%
r_9	Low , True ⟶ Well	75%
r_{10}	Low , False ⟶ Sick	50%
r_{11}	Low , 3 ⟶ Sick	75%
r_{12}	False , 3 ⟶ Sick	75%
r_{13}	Low , False , 3 ⟶ Sick	100%

พิจารณาที่ r'_7 จะเห็นว่า r'_7 เป็นสมาชิกใน CARs ด้วย ดังนั้นจะทำการทดสอบกฎความสัมพันธ์ r'_7 กับฐาน training data ที่เพิ่มใหม่เท่านั้น

ตารางที่ 3.3 training data ที่เพิ่มใหม่

Female	High	True	3	Well
Female	High	False	1	Well
Male	Low	True	2	Well
Female	High	True	2	Sick
Male	High	True	3	Sick

จากตารางจะเห็นว่าส่วนที่แรเงาเป็นส่วนที่กฎความสัมพันธ์สามารถรองรับข้อมูลได้ทั้งซ้ายและขวา เมื่อนำข้อมูลไปทดสอบกับกฎความสัมพันธ์ r'_7 ซึ่งข้อมูลทั้ง 2 ทรานแซคชั่นนี้จะไม่ถูกนำไปพิจารณาอีก อัลกอริทึมจะทำการวนลูปจนกว่าจะพิจารณากฎความสัมพันธ์ครบทุกกฎหรือ ไม่มีข้อมูลเหลือให้พิจารณาแล้ว

การวนลูปในแต่ละรอบจะต้องพิจารณาค่าเปอร์เซ็นต์ความผิดพลาดด้วย ทุกครั้งที่มีการเพิ่มกฎความสัมพันธ์เข้าไปใน C ' ต้องระวังไม่ให้ค่าเปอร์เซ็นต์ความผิดพลาดเพิ่มขึ้น หากเพิ่มเข้าไปแล้วทำให้ค่าเปอร์เซ็นต์ความผิดพลาดเพิ่มขึ้น ก็จะไม่เพิ่มกฎความสัมพันธ์นั้นๆเข้าไปใน C ' และจะหยุดทำการวนรอบทันที ซึ่ง C ' เป็นผลลัพธ์ของแบบจำลอง (classifier) ที่จะใช้ในการทำนายข้อมูลให้กับข้อมูลที่ไม่ทราบกลุ่มต่อไป สำหรับตัวอย่างแบบจำลองที่ใช้ในการทำนายแสดงดังรูปที่

$R' = \{r'_1, r'_3, r'_4$, default class = Sick$\}$ error= 15%

รูปที่ 3.13 ตัวอย่างแบบจำลองสำหรับข้อมูลแบบเพิ่มขยาย

บทที่ 4
การทดลองและวิเคราะห์ผลการทดลอง

บทนี้จะกล่าวถึงวัตถุประสงค์ของการทดลอง, ชุดข้อมูลที่ใช้ในการทดลอง, การทดลอง, ผลการทดลองและสรุปผลการทดลอง ในการทดสอบประสิทธิภาพของอัลกอริทึมสำหรับการจำแนกประเภทข้อมูลแบบเพิ่มขยายโดยใช้กฎความสัมพันธ์ใช้ โปรแกรม MATLAB R2007b ในการพัฒนาโปรแกรมสำหรับการทดลอง และทำการทดสอบบนเครื่องคอมพิวเตอร์พีซีที่มีคุณสมบัติของเครื่องดังนี้

- CPU : Intel(R) Core(TM) i7 2.93 GHz
- RAM : 3 GB
- Harddisk : 500GB
- Operation System :Microsoft WindowXP Professtional Version 2002 Service Pack 3

4.1 วัตถุประสงค์การทดลอง

งานวิจัยนี้นำเสนออัลกอริทึม ICBA ซึ่งเป็นอัลกอริทึมที่ทำการเพิ่มประสิทธิภาพด้านเวลาการทำงานของอัลกอริทึม CBA เมื่อมีการเพิ่มขยายของ training data และเพื่อทดสอบประสิทธิภาพของอัลกอริทึม ICBA จึงได้ทำการทดลองตามวัตถุประสงค์ของการทดลองดังต่อไปนี้

- เพื่อวัดความถูกต้องของกฎความสัมพันธ์แบบมีคลาสและแบบจำลองโดยเปรียบเทียบระหว่าง 2 อัลกอริทึมคือ CBA และ ICBA
- เพื่อวัดประสิทธิภาพโดยเปรียบเทียบระหว่าง2 อัลกอริทึมคือ CBA และ ICBA

4.2 ข้อมูลที่ใช้ในการทดลอง

ชุดข้อมูลที่นำมาใช้ทำการทดลองในงานวิจัย Incremental Classification Based On Association Rules Algorithm มี3 ชุดดังต่อไปนี้

4.2.1 ชุดข้อมูล Adult Dataset เป็นข้อมูลจาก UCI Machine Learning Repository [9]โดยมีรายละเอียดดังนี้ มีจำนวนข้อมูลทั้งหมด 45,222 ทรานแซคชั่น, 15 แอตทริบิวท์ (Att 1. – Att 15.) โดยกำหนดให้ แอตทริบิวท์ที่ 15 เป็นแอตทริบิวท์คลาส รายละเอียดของแอตทริบิวท์มีดังนี้

- Att 1.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 8ค่า
- Att 2.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 7ค่า
- Att 3.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 8ค่า
- Att 4.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 16ค่า
- Att 5.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 4ค่า
- Att 6.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 7ค่า

- Att 7.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 14ค่า
- Att 8.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 6ค่า
- Att 9.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 5ค่า
- Att10.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 2ค่า
- Att11.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 7ค่า
- Att12.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 3ค่า
- Att13.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 7ค่า
- Att14.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 41ค่า
- Att 15.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 2ค่า

4.2.2 ชุดข้อมูล Nursery Dataset เป็นข้อมูลจากUCI Machine Learning Repository [9]โดยมีรายละเอียดดังนี้ มีจำนวนข้อมูลทั้งหมด 12,960 ทรานแซคชั่น, 9 แอตทริบิวท์ (Att 1. – Att 9.) โดยกำหนดให้ แอตทริบิวท์ที่ 9 เป็นแอตทริบิวท์คลาส รายละเอียดของแอตทริบิวท์มีดังนี้

- Att 1.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 3 ค่า
- Att 2.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 5 ค่า
- Att 3.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 4 ค่า
- Att 4.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 4 ค่า
- Att 5.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 3 ค่า
- Att 6.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 2 ค่า
- Att 7.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 3 ค่า
- Att 8.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 3 ค่า
- Att 9.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 5ค่า

4.2.3 ชุดข้อมูล Mushroom Datasetเป็นข้อมูลจาก UCI Machine Learning Repository [9]โดยมีรายละเอียดดังนี้มีจำนวนข้อมูลทั้งหมด5,644ทรานแซคชั่น, 23แอตทริบิวท์ (Att 1. – Att23.) โดยกำหนดให้ แอตทริบิวท์ที่ 1เป็นแอตทริบิวท์คลาส รายละเอียดของแอตทริบิวท์มีดังนี้

- Att 1.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 2ค่า
- Att 2.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 6ค่า
- Att 3.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 4 ค่า
- Att 4.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 8ค่า
- Att 5.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 2ค่า
- Att 6.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 7ค่า
- Att 7.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 2ค่า
- Att 8.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 2ค่า

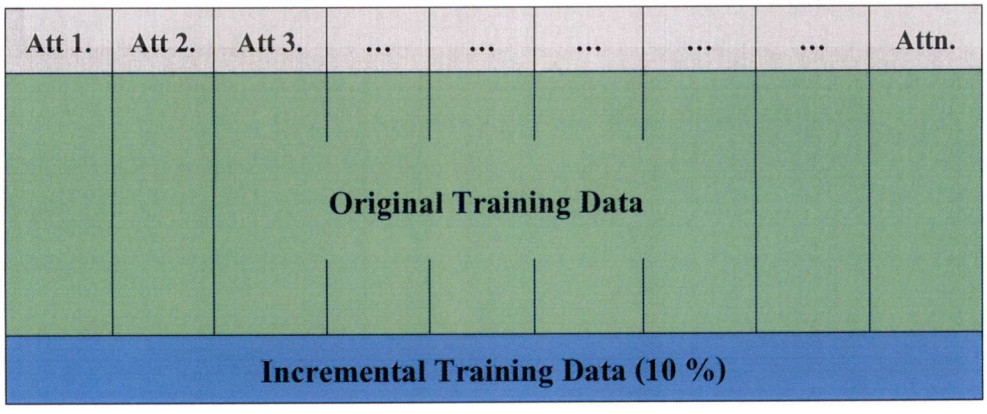
- Att 9.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 2ค่า
- Att10.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 9ค่า
- Att11.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 2ค่า
- Att12.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 4ค่า
- Att13.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 4ค่า
- Att14.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 4ค่า
- Att15.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 7ค่า
- Att16.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 7ค่า
- Att17.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 1ค่า
- Att18.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 2ค่า
- Att19.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 3ค่า
- Att20.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 4ค่า
- Att21.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 6ค่า
- Att22.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 6ค่า
- Att 23.มีค่าที่เป็นไปได้ในแอตทริบิวท์ทั้งหมด 6ค่า

4.3 แนวทางการทดลอง

ในการทดลองเพื่อวัดความถูกต้องและประสิทธิภาพของอัลกอริทึมสำหรับการจำแนกประเภทข้อมูลแบบเพิ่มขยายโดยใช้กฎความสัมพันธ์จะแบ่งออกเป็น 3ชุดการทดลองดังต่อไปนี้

4.3.1 ทำการทดลองกับการเพิ่ม training data 10%

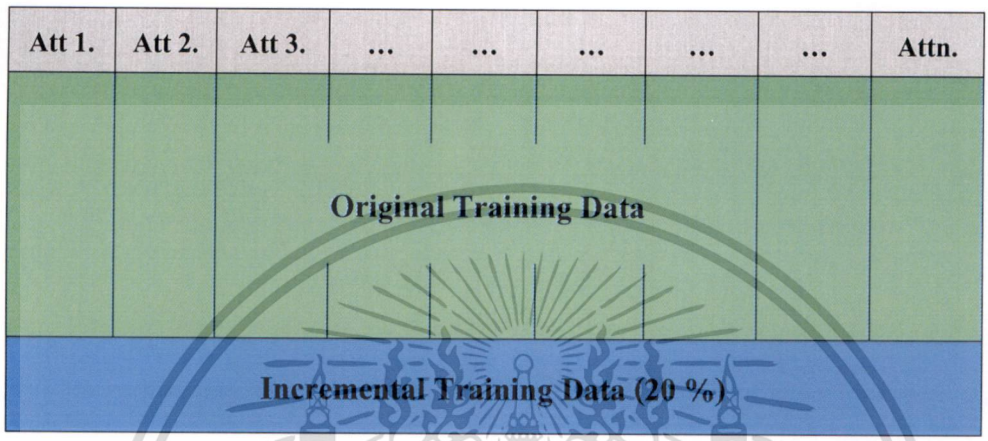
ทำการทดลองกับชุดข้อมูลทั้งสามชุดโดยจะพิจารณาเวลาการทำงานและความถูกต้องของเซตของกฎความสัมพันธ์แบบมีคลาสและแบบจำลองที่ใช้ในการทำนาย เมื่อมีการเพิ่มขึ้นของ training data10% จาก training data เดิม โดยทดลองกับ ค่า minimum support 3ค่า minimum confidence 3 ค่าโดยจะมีชุดของ training data ที่เพิ่มขึ้นจำนวน 3 ชุดแล้วนำมาหาค่าเฉลี่ย



รูปที่ 4.1 ลักษณะการเพิ่มขยาย training data ของการทดลองที่ 1

4.3.2 ทำการทดลองกับการเพิ่ม training data 20%

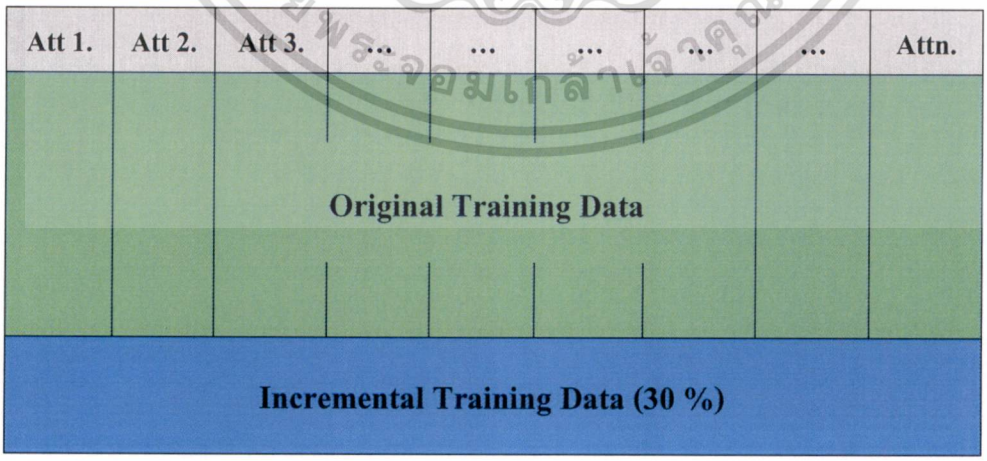
ทำการทดลองกับชุดข้อมูลทั้งสามชุดโดยจะพิจารณาเวลาการทำงานและความถูกต้องของเซตของกฎความสัมพันธ์แบบมีคลาสและแบบจำลองที่ใช้ในการทำนาย เมื่อมีการเพิ่มขึ้นของ training data20% จาก training data เดิม โดยทดลองกับ ค่า minimum support 3 ค่า minimum confidence 3 ค่าโดยจะมีชุดของ training data ที่เพิ่มขึ้นจำนวน 3 ชุดแล้วนำมาหาค่าเฉลี่ย



รูปที่ 4.2 ลักษณะการเพิ่มขยาย training data ของการทดลองที่ 2

4.3.3 ทำการทดลองกับการเพิ่ม training data 30%

ทำการทดลองกับชุดข้อมูลทั้งสามชุดโดยจะพิจารณาเวลาการทำงานและความถูกต้องของเซตของกฎความสัมพันธ์แบบมีคลาสและแบบจำลองที่ใช้ในการทำนาย เมื่อมีการเพิ่มขึ้นของ training data30% จาก training data เดิม โดยทดลองกับ ค่า minimum support 3 ค่า minimum confidence 3 ค่าโดยจะมีชุดของ training data ที่เพิ่มขึ้นจำนวน 3 ชุดแล้วนำมาหาค่าเฉลี่ย



รูปที่ 4.3 ลักษณะการเพิ่มขยาย training data ของการทดลองที่ 3

4.4 ผลการทดลอง

4.4.1 ผลการทดลองที่ 1 (เพิ่ม training data 10%)

การทดลองที่ 1 ทำการทดลองกับข้อมูล Adult Dataset โดยจะทำการเพิ่ม training data 10% ของ training data เดิม และมีสมมติฐานของการทดลองคือ จำนวนของกฎความสัมพันธ์แบบมีคลาสที่ถูกสร้างด้วยอัลกอริทึม CBA และ ICBA จะต้องเท่ากัน รวมทั้งจำนวนของกฎความสัมพันธ์ในแบบจำลองสำหรับการจำแนกประเภทข้อมูลของอัลกอริทึม CBA และ ICBA ต้องเท่ากันด้วย แต่อัลกอริทึม ICBA จะต้องใช้เวลาในการทำงานน้อยกว่าอัลกอริทึม CBA และอัลกอริทึม ICBA มีจำนวนครั้งที่ทรานแซคชั่นที่ถูกตรวจสอบในขั้นตอนการสร้างแบบจำลองน้อยกว่าอัลกอริทึม CBA สำหรับการทดลองที่ 1 กำหนดค่าตัวแปรต่างๆดังนี้ minimum support = 15% , 20% และ 25%, minimum confidence = 70%, 80% และ 90% และทำการทดลองกับtraining data จำนวน 3 ชุดที่แตกต่างกันคือ ชุด A , ชุด B และชุด Cจากนั้นนำผลการทดลองของทั้ง 3 ชุดมาหาค่าเฉลี่ย ผลการทดลองที่ 1 แสดงดังตารางที่ 4.1 , 4.2และกราฟเปรียบเทียบเวลาในการทำงานแสดงดังรูปที่ 4.4,4.5 และ 4.6

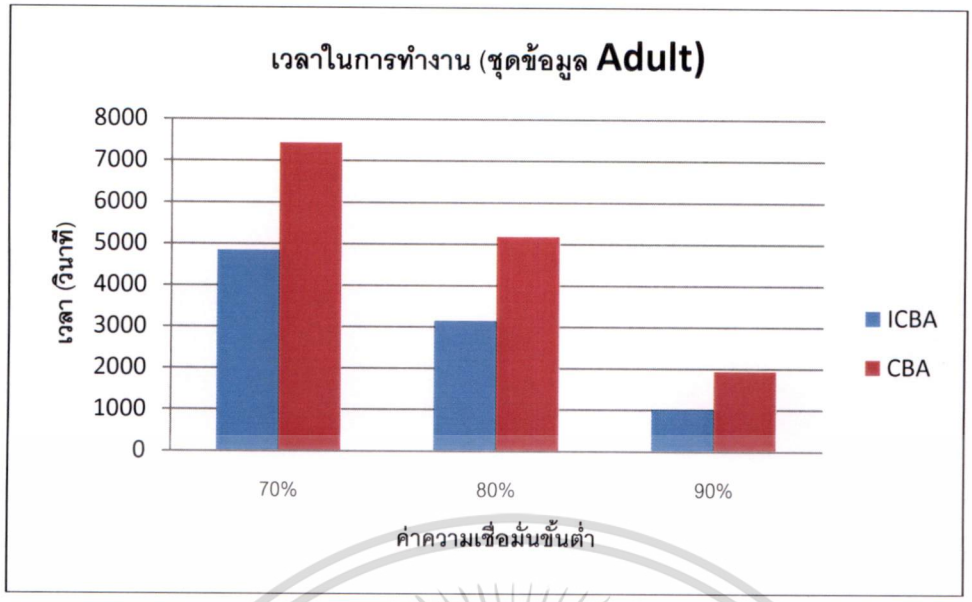
จากตารางที่ 4.1 จะเห็นได้ว่าการทดลองได้ผลตามสมมติฐานที่กำหนดไว้คือ จำนวนกฎความสัมพันธ์แบบมีคลาสที่ถูกสร้างมาจากทั้งสองอัลกอริทึมมีจำนวนเท่ากัน , จำนวนกฎความสัมพันธ์ในแบบจำลองสำหรับการจำแนกประเภทข้อมูลของทั้งสองอัลกอริทึมก็มีจำนวนเท่ากัน และจำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบในขั้นตอนการสร้างแบบจำลองของอัลกอริทึม ICBA น้อยกว่าอัลกอริทึม CBA เนื่องจากในส่วนของการสร้างแบบจำลองของอัลกอริทึม ICBA, มีการนำเอาความรู้เดิมมาใช่นั่นคืออัลกอริทึมจะทำการตรวจสอบว่ากฎความสัมพันธ์ที่พิจารณาอยู่เป็นกฎความสัมพันธ์ที่เคยเกิดขึ้นมาก่อนจาก training dataset เดิมหรือไม่ หากทำการตรวจสอบแล้วว่ากฎความสัมพันธ์นั้นเคยเกิดขึ้นมาจาก training dataset เดิม อัลกอริทึม ICBA จะทำการสแกนทรานแซคชั่นเพื่อตรวจสอบการรองรับข้อมูลสอนระบบของกฎความสัมพันธ์นั้นๆเฉพาะ training dataset ที่เพิ่มขยาย โดยไม่จำเป็นต้องสแกนทรานแซคชั่น training dataset เดิม ซึ่งในขั้นตอนนี้เป็นการประยุกต์เอาหลักการของอัลกอริทึม FUP มาใช้ จากตารางที่ 4.1 แสดงให้เห็นว่าจำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบของอัลกอริทึม ICBA น้อยกว่าอัลกอริทึม CBAมาก และเมื่อพิจารณาตารางที่ 4.2 ซึ่งเป็นตารางสำหรับการแสดงเวลาในการทำงานเปรียบเทียบกันระหว่าง 2 อัลกอริทึม และแสดงให้อยู่ในรูปของกราฟดังรูปที่ 4.4, 4.5 และ 4.6 ซึ่งเป็นกราฟเปรียบเทียบเวลาการทำงานของทั้งสองอัลกอริทึมพบว่า อัลกอริทึม ICBA ทำงานได้เร็วกว่าอัลกอริทึม CBA และเวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์มีค่าน้อยมากซึ่งแสดงให้เห็นว่าอัลกอริทึมไม่ได้เสียเวลาในการเปรียบเทียบกฎความสัมพันธ์เลย

ตารางที่ 4.1 ผลการทดลองที่ 1

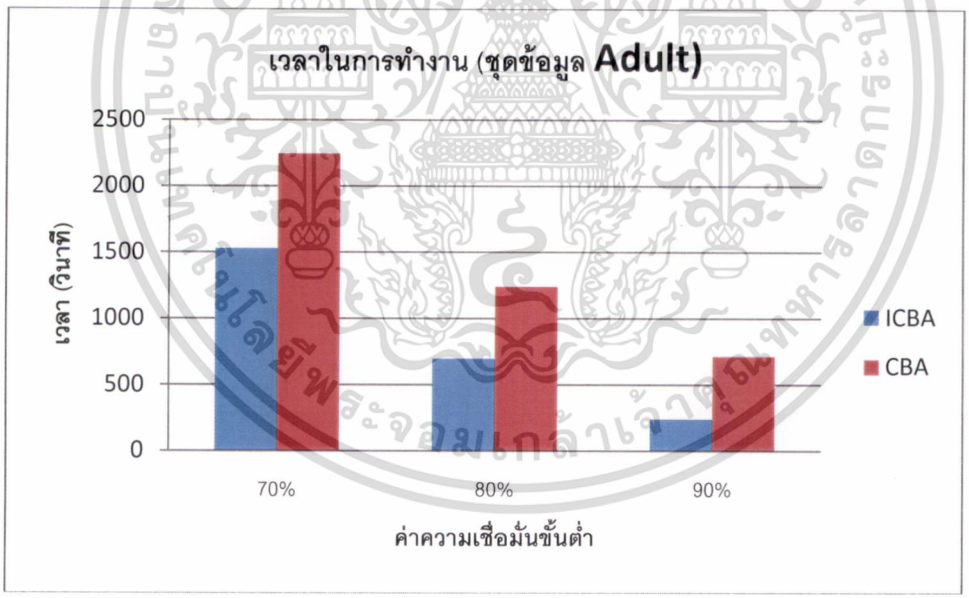
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	จำนวนกฎความสัมพันธ์แบบมีคลาส	จำนวนกฎความสัมพันธ์ในแบบจำลองเพื่อการจำแนกประเภทข้อมูล	จำนวนครั้งที่การแขวนขั้นถูกตรวจสอบ (ครั้ง)	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลสอนระบบ	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลทดสอบระบบ
15%	70%	ICBA	1,391	1,359	935,306	75%	73%
		CBA	1,391	1,359	6,489,082	75%	73%
	80%	ICBA	1,328	1,105	814,595	73%	72%
		CBA	1,328	1,105	5,323,196	73%	72%
	90%	ICBA	656	185	505,445	74%	73%
		CBA	656	185	2,262,832	74%	73%
20%	70%	ICBA	847	815	425,732	74%	71%
		CBA	847	815	3,061,782	74%	71%
	80%	ICBA	784	577	390,058	72%	70%
		CBA	784	577	2,370,215	72%	70%
	90%	ICBA	224	61	133,409	75%	74%
		CBA	224	61	678,231	75%	74%
25%	70%	ICBA	268	192	269,445	70%	69%
		CBA	268	192	1,801,960	70%	69%
	80%	ICBA	168	148	228,362	72%	72%
		CBA	168	148	1,278,060	72%	72%
	90%	ICBA	48	33	51,674	76%	74%
		CBA	48	33	183,806	76%	74%

ตารางที่ 4.2 เวลาในการทำงานของการทดลองที่ 1

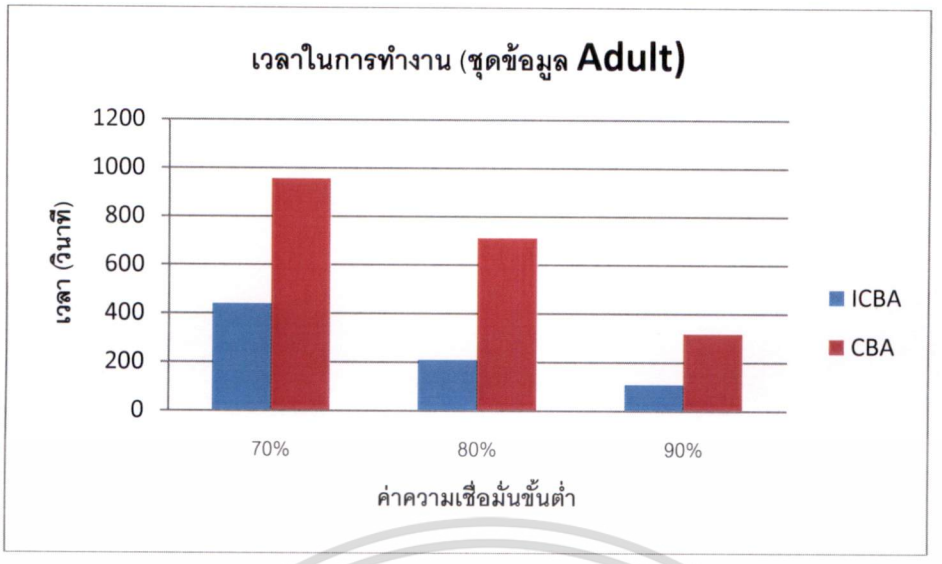
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	เวลาที่ใช้ในการสร้างเซตของกฎความสัมพันธ์แบบมีคลาส (วินาที)	เวลาที่ใช้ในการสร้างแบบจำลองเพื่อการจำแนกประเภทข้อมูล (วินาที)	เวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์(วินาที)	เวลารวม (วินาที)
15%	70%	ICBA	1,161.8257	3,681.5471	0.4828	4,843.3728
		CBA	2,536.5330	4,879.2333	0	7,415.766
	80%	ICBA	1,151.4955	1,988.1667	0.3483	3139.6622
		CBA	2,030.9667	3,141.6333	0	5,172.6000
	90%	ICBA	861.8278	142.4089	0.0871	1,004.2367
		CBA	1,531.7667	402.2935	0	1,934.0602
20%	70%	ICBA	569.5577	957.719	0.1992	1,527.2767
		CBA	1,002.5544	1,244.6667	0	2,247.2211
	80%	ICBA	423.6687	269.9991	0.1270	693.6678
		CBA	1,004.6876	238.8498	0	1,243.5374
	90%	ICBA	217.7716	22.4185	0.0757	240.1901
		CBA	604.4466	112.5684	0	717.015
25%	70%	ICBA	37.2448	402.5313	0.1026	439.7761
		CBA	500.4909	454.054	0	954.5449
	80%	ICBA	37.1647	170.8875	0.0583	208.0522
		CBA	501.2533	207.7213	0	708.9746
	90%	ICBA	97.2677	10.1133	0.0037	107.3810
		CBA	300.5856	16.7388	0	317.3244



รูปที่ 4.4 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 10% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 15% (Adult dataset)



รูปที่ 4.5 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 10% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 20% (Adult dataset)



รูปที่ 4.6 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 10% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 25% (Adult dataset)

4.4.2 ผลการทดลองที่ 2 (เพิ่ม training data 10%)

การทดลองที่ 2 ทำการทดลองกับข้อมูล Mushroom Dataset โดยจะทำการเพิ่ม training data 10% ของ training data เดิม และมีสมมติฐานของการทดลองคือ จำนวนของกฎความสัมพันธ์แบบมีคลาสที่ถูกสร้างด้วยอัลกอริทึม CBA และ ICBA จะต้องเท่ากัน รวมทั้งจำนวนของกฎความสัมพันธ์ในแบบจำลองสำหรับการจำแนกประเภทข้อมูลของอัลกอริทึม CBA และ ICBA ต้องเท่ากันด้วย แต่อัลกอริทึม ICBA จะต้องใช้เวลาในการทำงานน้อยกว่าอัลกอริทึม CBA และอัลกอริทึม ICBA มีจำนวนครั้งที่ทรานแซคชั่นที่ถูกตรวจสอบในขั้นตอนการสร้างแบบจำลองน้อยกว่าอัลกอริทึม CBA สำหรับการทดลองที่ 2 กำหนดค่าตัวแปรต่างๆดังนี้ minimum support = 35% , 40% และ 45%, minimum confidence = 50%, 70% และ 90% และทำการทดลองกับtraining data จำนวน 3 ชุดที่แตกต่างกันคือ ชุด A , ชุด B และชุด C จากนั้นนำผลการทดลองของทั้ง 3 ชุดมาหาค่าเฉลี่ย ผลการทดลองที่ 2 แสดงดังตารางที่ 4.3 , 4.4 และกราฟเปรียบเทียบเวลาในการทำงานแสดงดังรูปที่ 4.7, 4.8 และ 4.9

จากตารางที่ 4.3 จะเห็นได้ว่าการทดลองได้ผลตามสมมติฐานที่กำหนดไว้คือ จำนวนกฎความสัมพันธ์แบบมีคลาสที่ถูกสร้างมาจากทั้งสองอัลกอริทึมมีจำนวนเท่ากัน , จำนวนกฎความสัมพันธ์ในแบบจำลองสำหรับการจำแนกประเภทข้อมูลของทั้งสองอัลกอริทึมก็มีจำนวนเท่ากัน และจำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบในขั้นตอนการสร้างแบบจำลองของอัลกอริทึม ICBA น้อยกว่าอัลกอริทึม CBA เนื่องจากในส่วนของการสร้างแบบจำลองของอัลกอริทึม ICBA, มีการนำเอาความรู้เดิมมาใช้นั่นคืออัลกอริทึมจะทำการตรวจสอบว่ากฎความสัมพันธ์ที่พิจารณาอยู่เป็นกฎความสัมพันธ์ที่เคยเกิดขึ้นมาก่อนจาก training dataset เดิมหรือไม่ หากทำการตรวจสอบแล้ว

ว่ากฎความสัมพันธ์นั้นเคยเกิดขึ้นมาจาก training dataset เดิม อัลกอริทึม ICBA จะทำการสแกนทรานแซคชั่นเพื่อตรวจสอบการรองรับข้อมูลสอนระบบของกฎความสัมพันธ์นั้นๆเฉพาะ training dataset ที่เพิ่มขยาย โดยไม่จำเป็นต้องสแกนทรานแซคชั่นtraining dataset เดิม ซึ่งในขั้นตอนนี้เป็นการประยุกต์เอาหลักการของอัลกอริทึม FUP มาใช้ จากตารางที่ 4.3 แสดงให้เห็นว่าจำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบของอัลกอริทึม ICBA น้อยกว่าอัลกอริทึม CBA มาก และเมื่อพิจารณาตารางที่ 4.4 ซึ่งเป็นตารางสำหรับการแสดงเวลาในการทำงานเปรียบเทียบกันระหว่าง 2 อัลกอริทึม และแสดงให้อยู่ในรูปของกราฟดังรูปที่ 4.7, 4.8 และ 4.9 ซึ่งเป็นกราฟเปรียบเทียบเวลาการทำงานของทั้งสองอัลกอริทึมพบว่า อัลกอริทึม ICBA ทำงานได้เร็วกว่าอัลกอริทึม CBA และเวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์มีค่าน้อยมากซึ่งแสดงให้เห็นว่าอัลกอริทึมไม่ได้เสียเวลาในการเปรียบ เทียบกฎความสัมพันธ์เลย

ตารางที่ 4.3 ผลการทดลองที่ 2

ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	จำนวนกฎความสัมพันธ์แบบดิสครีต	จำนวนกฎความสัมพันธ์ในแบบจำลองเพื่อการจำแนกประเภทข้อมูล	จำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบ (ครั้ง)	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลสอนระบบ	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลทดสอบระบบ
35%	50%	ICBA	1455	1431	176964	85%	84%
		CBA	1455	1431	1701959	85%	84%
	70%	ICBA	1389	1166	164732	81%	78%
		CBA	1389	1166	1630188	81%	78%
	90%	ICBA	744	318	93073	83%	83%
		CBA	744	318	924812	83%	83%
40%	50%	ICBA	804	772	121778	81%	79%
		CBA	804	772	970452	81%	79%
	70%	ICBA	741	540	115205	83%	80%
		CBA	741	540	897603	83%	80%
	90%	ICBA	224	121	27311	84%	79%
		CBA	224	121	299435	84%	79%

ตารางที่ 4.3 (ต่อ) ผลการทดลองที่ 2

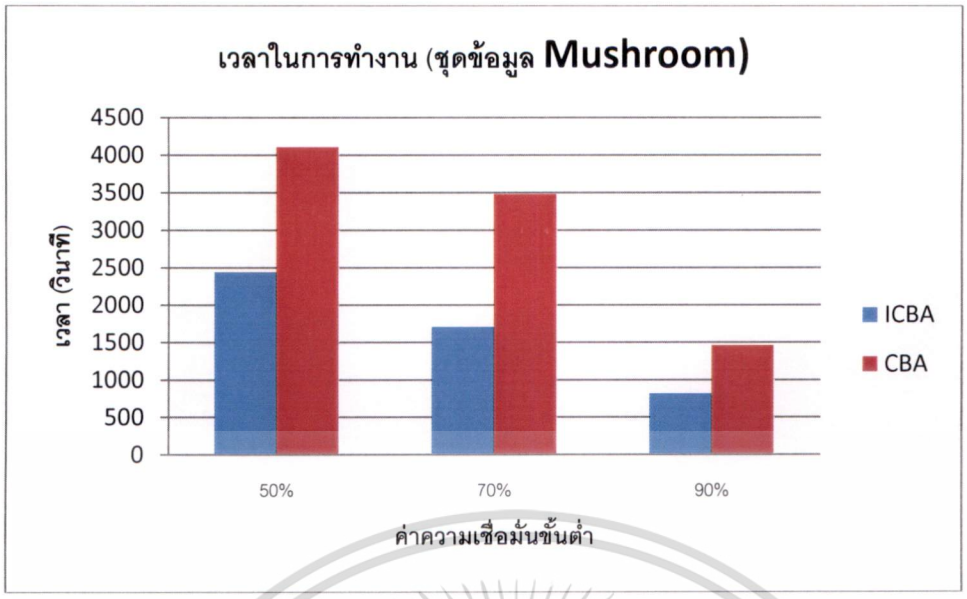
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	จำนวนกฎความสัมพันธ์แบบมีคลาส	จำนวนกฎความสัมพันธ์ในแบบจำลองเพื่อการจำแนกประเภทข้อมูล	จำนวนครั้งที่ทราเนเซคชั่นถูกตรวจสอบ (ครั้ง)	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลสอนระบบ	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลทดสอบระบบ
45%	50%	ICBA	196	188	37636	77%	71%
		CBA	196	188	236926	77%	71%
	70%	ICBA	157	84	33567	80%	78%
		CBA	157	84	191907	80%	78%
	90%	ICBA	48	33	6018	78%	77%
		CBA	48	33	65681	78%	77%

ตารางที่ 4.4 เวลาในการทำงานของการทดลองที่ 2

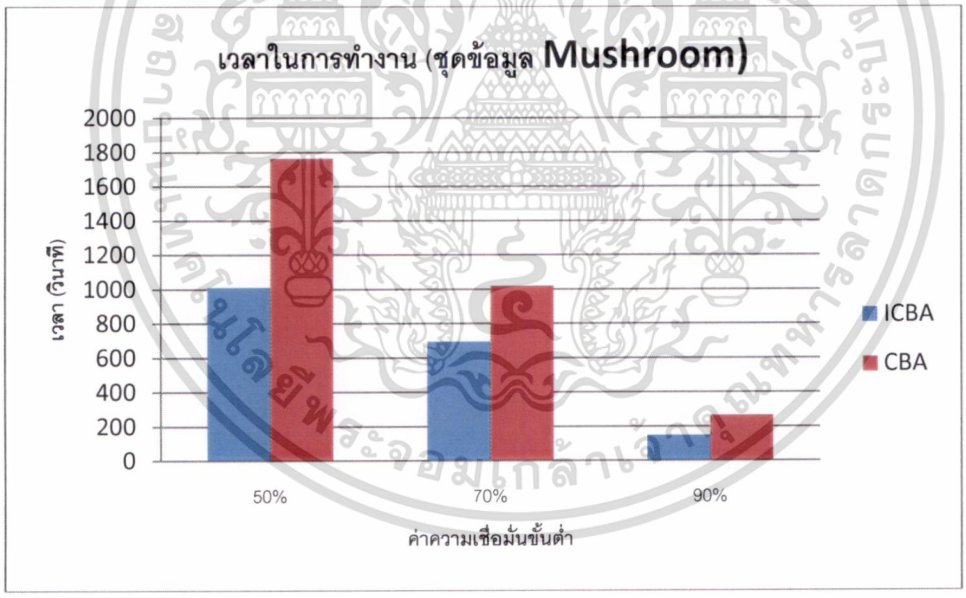
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	เวลาที่ใช้ในการสร้างเซตของกฎความสัมพันธ์แบบมีคลาส (วินาที)	เวลาที่ใช้ในการสร้างแบบจำลองเพื่อการจำแนกประเภทข้อมูล (วินาที)	เวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์(วินาที)	เวลารวม (วินาที)
35%	50%	ICBA	540.5747	1896.2000	1.0157	2436.775
		CBA	1443.2008	2656.8667	0	4100.0680
	70%	ICBA	387.6365	1316.6333	0.9398	1704.2700
		CBA	1028.2102	2444.8000	0	3473.0100
	90%	ICBA	170.7263	638.3629	0.4736	809.0892
		CBA	742.6894	712.1451	0	1454.835

ตารางที่ 4.4 (ต่อ) เวลาในการทำงานของการทดลองที่ 2

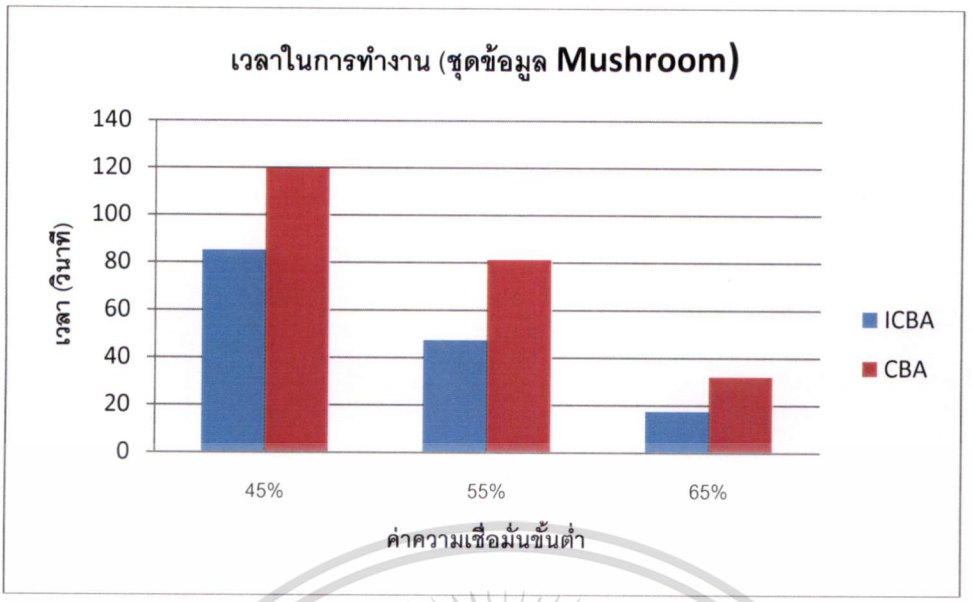
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	เวลาที่ใช้ในการสร้างเซตของกฎความสัมพันธ์แบบมีคลาส (วินาที)	เวลาที่ใช้ในการสร้างแบบจำลองเพื่อการจำแนกประเภทข้อมูล (วินาที)	เวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์(วินาที)	เวลารวม (วินาที)
40%	50%	ICBA	418.6598	592.1326	0.3518	1010.7920
		CBA	997.1169	768.8197	0	1765.9370
	70%	ICBA	118.6599	573.1943	0.3171	691.8542
		CBA	316.6642	704.9997	0	1021.664
	90%	ICBA	83.6508	61.9497	0.0863	145.6005
		CBA	196.6497	71.4899	0	268.1396
45%	50%	ICBA	54.4950	30.5691	0.0570	85.0641
		CBA	75.1065	44.5115	0	119.6180
	70%	ICBA	20.4729	26.8490	0.0422	47.3219
		CBA	48.0074	33.1656	0	81.1730
	90%	ICBA	14.7270	2.6169	0.0126	17.3439
		CBA	27.3100	4.7621	0	32.0721



รูปที่ 4.7 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 10% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 35% (Mushroom dataset)



รูปที่ 4.8 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 10% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 40% (Mushroom dataset)



ปที่ 4.9 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 10%โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 45% (Mushroom dataset)

4.4.3 ผลการทดลองที่ 3 (เพิ่ม training data 10%)

การทดลองที่ 3 ทำการทดลองกับข้อมูล Nursery Dataset โดยจะทำการเพิ่ม training data 10% ของ training data เดิม และมีสมมติฐานของการทดลองคือ จำนวนของกฎความสัมพันธ์แบบมีคลาสที่ถูกสร้างด้วยอัลกอริทึม CBA และ ICBA จะต้องเท่ากัน รวมทั้งจำนวนของกฎความสัมพันธ์ในแบบจำลองสำหรับการจำแนกประเภทข้อมูลของอัลกอริทึม CBA และ ICBA ต้องเท่ากันด้วย แต่อัลกอริทึม ICBA จะต้องใช้เวลาในการทำงานน้อยกว่าอัลกอริทึม CBA และอัลกอริทึม ICBA มีจำนวนครั้งที่ทรานแซคชั่นที่ถูกตรวจสอบในขั้นตอนการสร้างแบบจำลองน้อยกว่าอัลกอริทึม CBA สำหรับการทดลองที่ 3กำหนดค่าตัวแปรต่างๆดังนี้ minimum support = 1% , 3% และ 5%, minimum confidence = 45%, 55% และ 65% และทำการทดลองกับtraining data จำนวน 3 ชุดที่แตกต่างกันคือ ชุด A , ชุด B และชุด C จากนั้นนำผลการทดลองของทั้ง 3 ชุดมาหาค่าเฉลี่ย ผลการทดลองที่ 3 แสดงดังตารางที่ 4.5 , 4.6 และกราฟเปรียบเทียบเวลาในการทำงานแสดงดังรูปที่ 4.10, 4.11 และ 4.12

จากตารางที่ 4.5 จะเห็นได้ว่าการทดลองได้ผลตามสมมติฐานที่กำหนดไว้คือ จำนวนกฎความสัมพันธ์แบบมีคลาสที่ถูกสร้างมาจากทั้งสองอัลกอริทึมมีจำนวนเท่ากัน , จำนวนกฎความสัมพันธ์ในแบบจำลองสำหรับการจำแนกประเภทข้อมูลของทั้งสองอัลกอริทึมก็มีจำนวนเท่ากัน และจำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบในขั้นตอนการสร้างแบบจำลองของอัลกอริทึม ICBA น้อยกว่าอัลกอริทึม CBA เนื่องจากในส่วนของการสร้างแบบจำลองของอัลกอริทึม ICBA, มีการนำเอาความรู้เดิมมาใช้นั่นคืออัลกอริทึมจะทำการตรวจสอบว่ากฎความสัมพันธ์ที่พิจารณาอยู่

เป็นกฎความสัมพันธ์ที่เคยเกิดขึ้นมาก่อนจาก training dataset เดิมหรือไม่ หากทำการตรวจสอบแล้วว่ากฎความสัมพันธ์นั้นเคยเกิดขึ้นมาจาก training dataset เดิม อัลกอริทึม ICBA จะทำการสแกนทรานแซคชั่นเพื่อตรวจสอบการรองรับข้อมูลสอนระบบของกฎความสัมพันธ์นั้นๆเฉพาะ training dataset ที่เพิ่มขยาย โดยไม่จำเป็นต้องสแกนทรานแซคชั่นtraining dataset เดิม ซึ่งในขั้นตอนนี้เป็นการประยุกต์เอาหลักการของอัลกอริทึม FUP มาใช้ จากตารางที่ 4.5 แสดงให้เห็นว่าจำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบของอัลกอริทึม ICBA น้อยกว่าอัลกอริทึม CBA มาก และเมื่อพิจารณาตารางที่ 4.6 ซึ่งเป็นตารางสำหรับการแสดงเวลาในการทำงานเปรียบเทียบกันระหว่าง 2 อัลกอริทึม และแสดงให้อยู่ในรูปของกราฟดังรูปที่ 4.10, 4.11 และ 4.12 ซึ่งเป็นกราฟเปรียบเทียบเวลาการทำงานของทั้งสองอัลกอริทึมพบว่า อัลกอริทึม ICBA ทำงานได้เร็วกว่าอัลกอริทึม CBA และเวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์มีค่าน้อยมากซึ่งแสดงให้เห็นว่าอัลกอริทึมไม่ได้เสียเวลาในการเปรียบ เทียบกฎความสัมพันธ์เลย

ตารางที่ 4.5 ผลการทดลองที่ 3

ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	จำนวนกฎความสัมพันธ์แบบคลาส	จำนวนกฎความสัมพันธ์ในแบบจำลองเพื่อการจำแนกประเภทข้อมูล	จำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบ (ครั้ง)	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลสอนระบบ	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลทดสอบระบบ
1%	45%	ICBA	225	147	149280	69%	65%
		CBA	225	147	440156	69%	65%
	55%	ICBA	201	159	126417	68%	63%
		CBA	201	159	319047	68%	63%
	65%	ICBA	102	45	82046	67%	61%
		CBA	102	45	306482	67%	61%
3%	45%	ICBA	75	38	36816	66%	65%
		CBA	75	38	140292	66%	65%
	55%	ICBA	57	37	32672	67%	65%
		CBA	57	37	131261	67%	65%
	65%	ICBA	23	22	17725	68%	66%
		CBA	23	22	85825	68%	66%

ตารางที่ 4.5 (ต่อ) ผลการทดลองที่ 3

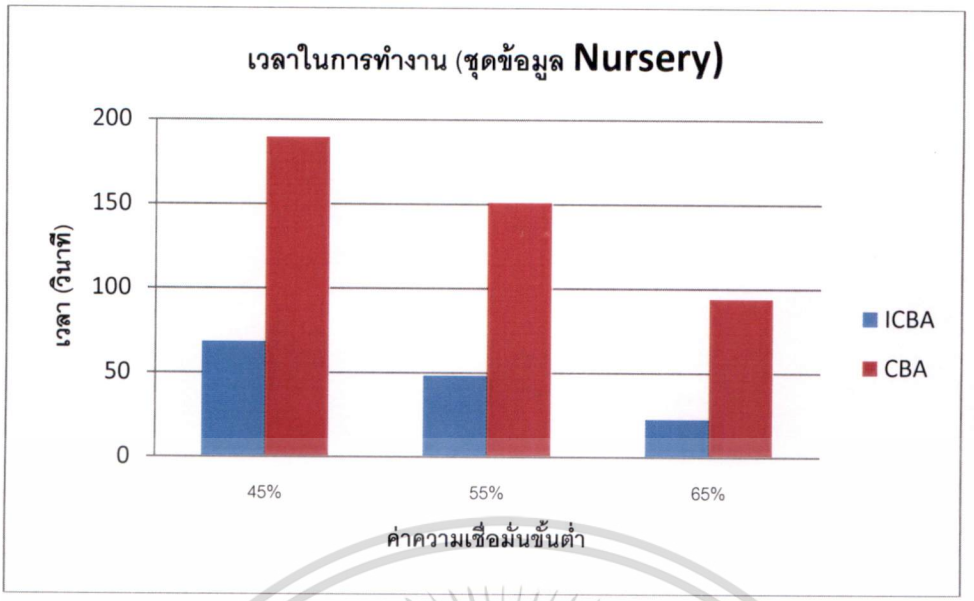
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	จำนวนกฎความสัมพันธ์แบบมีคลาส	จำนวนกฎความสัมพันธ์ในแบบจำลองเพื่อการจำแนกประเภทข้อมูล	จำนวนครั้งที่ทราเวอร์สต้นไม้ถูกตรวจสอบ (ครั้ง)	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลสอนระบบ	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลทดสอบระบบ
5%	45%	ICBA	35	27	29915	68%	63%
		CBA	35	27	87832	68%	63%
	55%	ICBA	27	24	22126	67%	61%
		CBA	27	24	80319	67%	61%
	65%	ICBA	14	14	13800	66%	65%
		CBA	14	14	52491	66%	65%

ตารางที่ 4.6 เวลาในการทำงานของการทดลองที่ 3

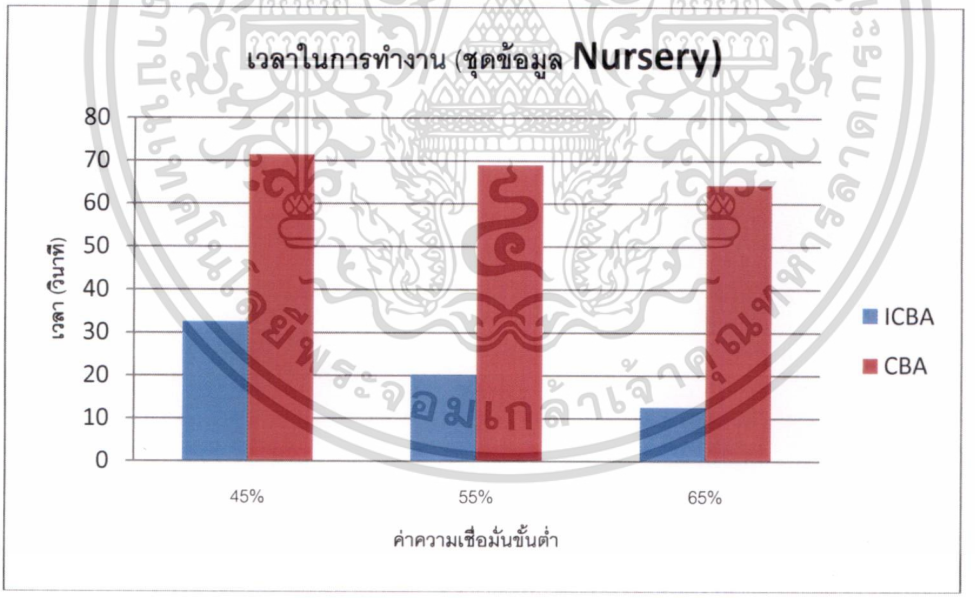
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	เวลาที่ใช้ในการสร้างชุดของกฎความสัมพันธ์แบบมีคลาส (วินาที)	เวลาที่ใช้ในการสร้างแบบจำลองเพื่อการจำแนกประเภทข้อมูล (วินาที)	เวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์(วินาที)	เวลารวม (วินาที)
1%	45%	ICBA	12.1837	56.0008	0.1127	68.1845
		CBA	108.1651	81.1266	0	183.2917
	55%	ICBA	12.1704	35.9085	0.0900	48.0789
		CBA	97.1765	53.4893	0	150.6658
	65%	ICBA	12.1751	10.1973	0.0457	22.3724
		CBA	73.2172	20.7306	0	93.9478

ตารางที่ 4.6 (ต่อ) เวลาในการทำงานของการทดลองที่ 3

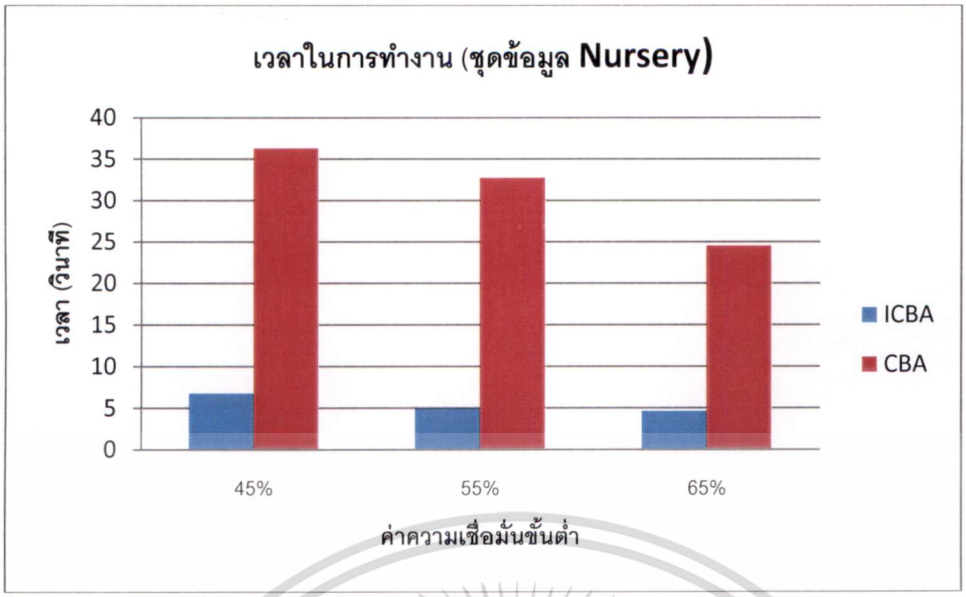
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	เวลาที่ใช้ในการสร้างเซตของกฎความสัมพันธ์แบบมีคลาส (วินาที)	เวลาที่ใช้ในการสร้างแบบจำลองเพื่อการจำแนกประเภทข้อมูล (วินาที)	เวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์(วินาที)	เวลารวม (วินาที)
3%	45%	ICBA	26.4532	5.9726	0.0280	32.4258
		CBA	59.9731	11.5700	0	71.5431
	55%	ICBA	16.4330	3.7299	0.0170	20.1629
		CBA	60.3107	8.8042	0	69.1149
	65%	ICBA	10.4537	2.1533	0.0068	12.6070
		CBA	60.1452	4.2504	0	64.3956
5%	45%	ICBA	3.7918	2.9650	0.0145	6.7568
		CBA	31.2352	5.0373	0	36.2725
	55%	ICBA	3.5103	1.4000	0.0073	4.9103
		CBA	28.3814	4.2974	0	32.6788
	65%	ICBA	3.5318	1.0664	0.0035	4.5982
		CBA	21.6780	2.7848	0	24.4628



รูปที่ 4.10 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 10% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 1% (Nursery dataset)



รูปที่ 4.11 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 10% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 3% (Nursery dataset)



รูปที่ 4.12 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 10%โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 5% (Nursery dataset)

4.4.4 ผลการทดลองที่ 4 (เพิ่ม training data 20%)

การทดลองที่ 4 ทำการทดลองกับข้อมูล Adult Dataset โดยจะทำการเพิ่ม training data 20% ของ training data เดิม และมีสมมติฐานของการทดลองคือ จำนวนของกฎความสัมพันธ์แบบมีคลาสที่ถูกสร้างด้วยอัลกอริทึม CBA และ ICBA จะต้องเท่ากัน รวมทั้งจำนวนของกฎความสัมพันธ์ในแบบจำลองสำหรับการจำแนกประเภทข้อมูลของอัลกอริทึม CBA และ ICBA ต้องเท่ากันด้วย แต่อัลกอริทึม ICBA จะต้องใช้เวลาในการทำงานน้อยกว่าอัลกอริทึม CBA และอัลกอริทึม ICBA มีจำนวนครั้งที่ทรานแซคชั่นที่ถูกตรวจสอบในขั้นตอนการสร้างแบบจำลองน้อยกว่าอัลกอริทึม CBA สำหรับการทดลองที่ 4กำหนดค่าตัวแปรต่างๆดังนี้ minimum support = 15% , 20% และ 25%, minimum confidence = 70%, 80% และ 90% และทำการทดลองกับtraining data จำนวน 3 ชุดที่แตกต่างกันคือ ชุด A , ชุด B และชุด C จากนั้นนำผลการทดลองของทั้ง 3 ชุดมาหาค่าเฉลี่ย ผลการทดลองที่ 4 แสดงดังตารางที่ 4.7, 4.8 และกราฟเปรียบเทียบเวลาในการทำงานแสดงดังรูปที่ 4.13, 4.14และ 4.15

จากตารางที่ 4.7จะเห็นได้ว่าการทดลองได้ผลตามสมมติฐานที่กำหนดไว้คือ จำนวนกฎความสัมพันธ์แบบมีคลาสที่ถูกสร้างมาจากทั้งสองอัลกอริทึมมีจำนวนเท่ากัน , จำนวนกฎความสัมพันธ์ในแบบจำลองสำหรับการจำแนกประเภทข้อมูลของทั้งสองอัลกอริทึมก็มีจำนวนเท่ากัน และจำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบในขั้นตอนการสร้างแบบจำลองของอัลกอริทึม ICBA น้อยกว่าอัลกอริทึม CBA เนื่องจากในส่วนของการสร้างแบบจำลองของอัลกอริทึม ICBA, มีการนำเอาความรู้เดิมมาใช้นั่นคืออัลกอริทึมจะทำการตรวจสอบว่ากฎความสัมพันธ์ที่พิจารณาอยู่

เป็นกฎความสัมพันธ์ที่เคยเกิดขึ้นมาก่อนจาก training dataset เดิมหรือไม่ หากทำการตรวจสอบแล้วว่ากฎความสัมพันธ์นั้นเคยเกิดขึ้นมาจาก training dataset เดิม อัลกอริทึม ICBA จะทำการสแกนทรานแซคชั่นเพื่อตรวจสอบการรองรับข้อมูลสอนระบบของกฎความสัมพันธ์นั้นๆเฉพาะ training dataset ที่เพิ่มขยาย โดยไม่จำเป็นต้องสแกนทรานแซคชั่นtraining dataset เดิม ซึ่งในขั้นตอนนี้เป็นการประยุกต์เอาหลักการของอัลกอริทึม FUP มาใช้ จากตารางที่ 4.7แสดงให้เห็นว่าจำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบของอัลกอริทึม ICBA น้อยกว่าอัลกอริทึม CBA มาก และเมื่อพิจารณาตารางที่ 4.8 ซึ่งเป็นตารางสำหรับการแสดงเวลาในการทำงานเปรียบเทียบกันระหว่าง 2 อัลกอริทึม และแสดงให้อยู่ในรูปของกราฟดังรูปที่ 4.13, 4.14และ 4.15 ซึ่งเป็นกราฟเปรียบเทียบเวลาการทำงานของทั้งสองอัลกอริทึมพบว่า อัลกอริทึม ICBA ทำงานได้เร็วกว่าอัลกอริทึม CBA และเวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์มีค่าน้อยมากซึ่งแสดงให้เห็นว่าอัลกอริทึมไม่ได้เสียเวลาในการเปรียบ เทียบกฎความสัมพันธ์เลย

ตารางที่ 4.7 ผลการทดลองที่ 4

ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	จำนวนกฎความสัมพันธ์แบบคลาส	จำนวนกฎความสัมพันธ์ในแบบจำลองเพื่อการจำแนกประเภทข้อมูล	จำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบ (ครั้ง)	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลสอนระบบ	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลทดสอบระบบ
15%	70%	ICBA	1684	1572	1616921	74%	71%
		CBA	1684	1572	7406461	74%	71%
	80%	ICBA	1347	1015	1408702	72%	70%
		CBA	1347	1015	6348316	72%	70%
	90%	ICBA	893	584	934459	75%	74%
		CBA	893	584	2930723	75%	74%
20%	70%	ICBA	1039	934	776936	73%	72%
		CBA	1039	934	3395658	73%	72%
	80%	ICBA	792	516	731624	74%	73%
		CBA	792	516	2902440	74%	73%
	90%	ICBA	431	379	332718	74%	71%
		CBA	431	379	927103	74%	71%

ตารางที่ 4.7 (ต่อ) ผลการทดลองที่ 4

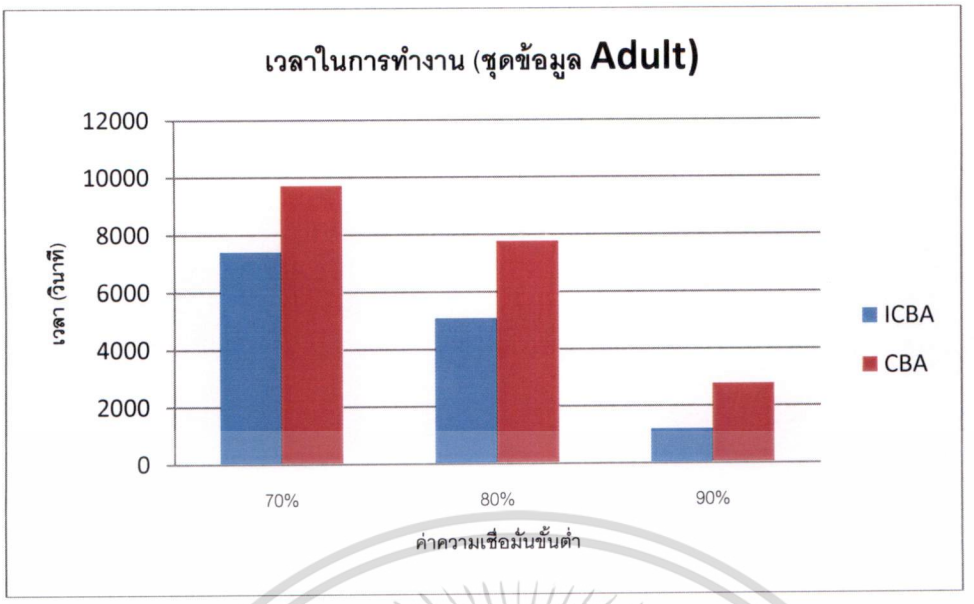
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	จำนวนกฎความสัมพันธ์แบบมีคลาส	จำนวนกฎความสัมพันธ์ในแบบจำลองเพื่อการจำแนกประเภทข้อมูล	จำนวนครั้งที่ทราเวอร์ซฌชั้นถูกตรวจสอบ (ครั้ง)	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลสอนระบบ	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลทดสอบระบบ
25%	70%	ICBA	493	175	448732	76%	74%
		CBA	493	175	1973955	76%	74%
	80%	ICBA	214	180	397963	73%	715
		CBA	214	180	1622406	73%	71%
	90%	ICBA	103	84	134927	75%	72%
		CBA	103	84	318046	75%	72%

ตารางที่ 4.8 เวลาในการทำงานของการทดลองที่ 4

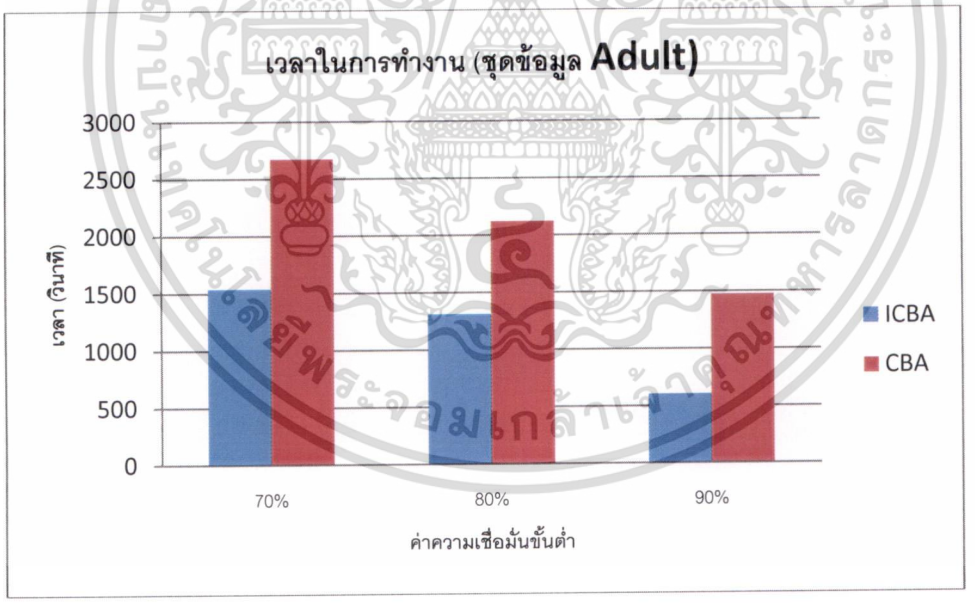
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	เวลาที่ใช้ในการสร้างเซตของกฎความสัมพันธ์แบบมีคลาส (วินาที)	เวลาที่ใช้ในการสร้างแบบจำลองเพื่อการจำแนกประเภทข้อมูล (วินาที)	เวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์(วินาที)	เวลารวม (วินาที)
15%	70%	ICBA	2389.7719	5022.6000	0.5188	7412.3720
		CBA	3499.2333	6203.2667	0	9702.5000
	80%	ICBA	1389.0128	3684.8667	0.4061	5073.8800
		CBA	2909.7667	4852.8333	0	7762.6000
	90%	ICBA	589.8824	596.6316	0.1136	1186.5140
		CBA	1802.3000	967.534	0	2769.8340

ตารางที่ 4.8 (ต่อ) เวลาในการทำงานของการทดลองที่ 4

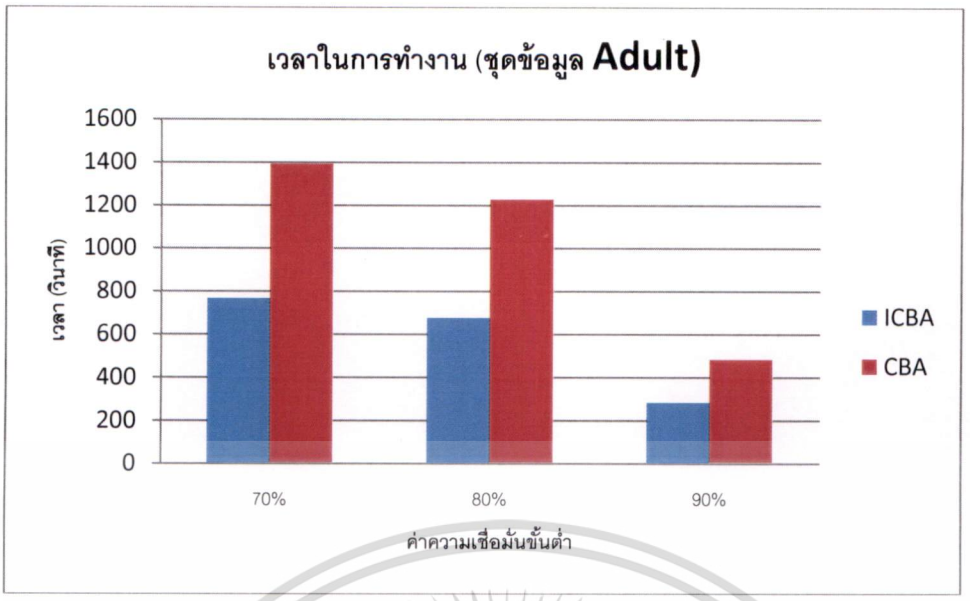
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	เวลาที่ใช้ในการสร้างเซตของกฎความสัมพันธ์แบบมีคลาส (วินาที)	เวลาที่ใช้ในการสร้างแบบจำลองเพื่อการจำแนกประเภทข้อมูล (วินาที)	เวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์(วินาที)	เวลารวม (วินาที)
20%	70%	ICBA	459.2888	1073.7333	0.2202	1533.0220
		CBA	1107.0667	1570.3667	0	2677.4330
	80%	ICBA	362.1463	942.6555	0.1699	1304.8020
		CBA	903.0667	1220.7752	0	2123.8420
	90%	ICBA	159.3857	442.8396	0.0295	602.2253
		CBA	708.3333	767.2277	0	1475.5610
25%	70%	ICBA	385.5777	380.8994	0.1020	766.4771
		CBA	656.3993	736.4826	0	1392.882
	80%	ICBA	325.4156	349.9239	0.0750	675.3395
		CBA	597.3674	629.5032	0	1226.8710
	90%	ICBA	185.3779	96.8018	0.0067	282.1797
		CBA	355.2080	128.0023	0	483.2103



รูปที่ 4.13 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 20% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 15% (Adult dataset)



รูปที่ 4.14 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 20% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 20% (Adult dataset)



รูปที่ 4.15 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 20% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 25% (Adult dataset)

4.4.5 ผลการทดลองที่ 5 (เพิ่ม training data 20%)

การทดลองที่ 5 ทำการทดลองกับข้อมูล Mushroom Dataset โดยจะทำการเพิ่ม training data 20% ของ training data เดิม และมีสมมติฐานของการทดลองคือ จำนวนของกฎความสัมพันธ์แบบมีคลาสที่ถูกสร้างด้วยอัลกอริทึม CBA และ ICBA จะต้องเท่ากัน รวมทั้งจำนวนของกฎความสัมพันธ์ในแบบจำลองสำหรับการจำแนกประเภทข้อมูลของอัลกอริทึม CBA และ ICBA ต้องเท่ากันด้วย แต่อัลกอริทึม ICBA จะต้องใช้เวลาในการทำงานน้อยกว่าอัลกอริทึม CBA และอัลกอริทึม ICBA มีจำนวนครั้งที่ทรานแซคชั่นที่ถูกตรวจสอบในขั้นตอนการสร้างแบบจำลองน้อยกว่าอัลกอริทึม CBA สำหรับการทดลองที่ 5กำหนดค่าตัวแปรต่างๆดังนี้ minimum support = 35% , 40% และ 45%, minimum confidence = 50%, 70% และ 90% และทำการทดลองกับtraining data จำนวน 3 ชุดที่แตกต่างกันคือ ชุด A , ชุด B และชุด C จากนั้นนำผลการทดลองของทั้ง 3 ชุดมาหาค่าเฉลี่ย ผลการทดลองที่ 5 แสดงดังตารางที่ 4.9 , 4.10 และกราฟเปรียบเทียบเวลาในการทำงานแสดงดังรูปที่ 4.16, 4.17 และ 4.18

จากตารางที่ 4.9 จะเห็นได้ว่าการทดลองได้ผลตามสมมติฐานที่กำหนดไว้คือ จำนวนกฎความสัมพันธ์แบบมีคลาสที่ถูกสร้างมาจากทั้งสองอัลกอริทึมมีจำนวนเท่ากัน , จำนวนกฎความสัมพันธ์ในแบบจำลองสำหรับการจำแนกประเภทข้อมูลของทั้งสองอัลกอริทึมก็มีจำนวนเท่ากัน และจำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบในขั้นตอนการสร้างแบบจำลองของอัลกอริทึม ICBA น้อยกว่าอัลกอริทึม CBA เนื่องจากในส่วนของการสร้างแบบจำลองของอัลกอริทึม ICBA, มีการนำเอาความรู้เดิมมาใช้นั่นคืออัลกอริทึมจะทำการตรวจสอบว่ากฎความสัมพันธ์ที่พิจารณาอยู่

เป็นกฎความสัมพันธ์ที่เคยเกิดขึ้นมาก่อนจาก training dataset เดิมหรือไม่ หากทำการตรวจสอบแล้วว่ากฎความสัมพันธ์นั้นเคยเกิดขึ้นมาจาก training dataset เดิม อัลกอริทึม ICBA จะทำการสแกนทรานแซคชั่นเพื่อตรวจสอบการรองรับข้อมูลสอนระบบของกฎความสัมพันธ์นั้นๆเฉพาะ training dataset ที่เพิ่มขยาย โดยไม่จำเป็นต้องสแกนทรานแซคชั่นtraining dataset เดิม ซึ่งในขั้นตอนนี้เป็นการประยุกต์เอาหลักการของอัลกอริทึม FUP มาใช้ จากตารางที่ 4.9 แสดงให้เห็นว่าจำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบของอัลกอริทึม ICBA น้อยกว่าอัลกอริทึม CBA มาก และเมื่อพิจารณาตารางที่ 4.10 ซึ่งเป็นตารางสำหรับการแสดงเวลาในการทำงานเปรียบเทียบกันระหว่าง 2 อัลกอริทึม และแสดงให้อยู่ในรูปของกราฟดังรูปที่ 4.16, 4.17 และ 4.18 ซึ่งเป็นกราฟเปรียบเทียบเวลาการทำงานของทั้งสองอัลกอริทึมพบว่า อัลกอริทึม ICBA ทำงานได้เร็วกว่าอัลกอริทึม CBA และเวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์มีค่าน้อยมากซึ่งแสดงให้เห็นว่าอัลกอริทึมไม่ได้เสียเวลาในการเปรียบ เทียบกฎความสัมพันธ์เลย

ตารางที่ 4.9 ผลการทดลองที่ 5

ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	จำนวนกฎความสัมพันธ์แบบคลาส	จำนวนกฎความสัมพันธ์ในแบบจำลองเพื่อการจำแนกประเภทข้อมูล	จำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบ (ครั้ง)	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลสอนระบบ	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลทดสอบระบบ
35%	50%	ICBA	1434	1402	332453	83%	80%
		CBA	1434	1402	1836320	83%	80%
	70%	ICBA	1371	1145	318219	86%	81%
		CBA	1371	1145	1760720	86%	81%
	90%	ICBA	741	302	181125	84%	84%
		CBA	741	302	1005120	84%	84%
40%	50%	ICBA	815	783	239692	82%	82%
		CBA	815	783	1065080	82%	82%
	70%	ICBA	752	545	227050	81%	78%
		CBA	752	545	986162	81%	78%
	90%	ICBA	224	97	53000	85%	86%
		CBA	224	97	325125	85%	86%

ตารางที่ 4.9 (ต่อ) ผลการทดลองที่ 5

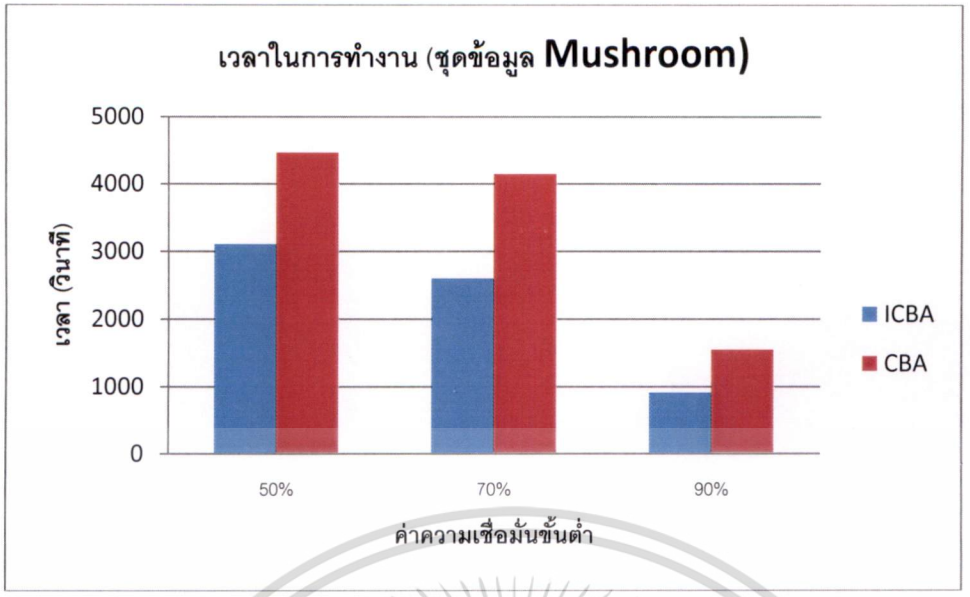
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	จำนวนกฎความสัมพันธ์แบบมีคลาส	จำนวนกฎความสัมพันธ์ในแบบจำลองเพื่อการจำแนกประเภทข้อมูล	จำนวนครั้งที่ทำนายแชดขึ้นถูกตรวจสอบ (ครั้ง)	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลสอนระบบ	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลทดสอบระบบ
45%	50%	ICBA	202	194	60096	84%	80%
		CBA	202	194	264985	84%	80%
	70%	ICBA	163	84	51958	79%	78%
		CBA	163	84	215897	79%	78%
	90%	ICBA	48	33	11875	83%	83%
		CBA	48	33	71538	83%	83%

ตารางที่ 4.10 เวลาในการทำงานของการทดลองที่ 5

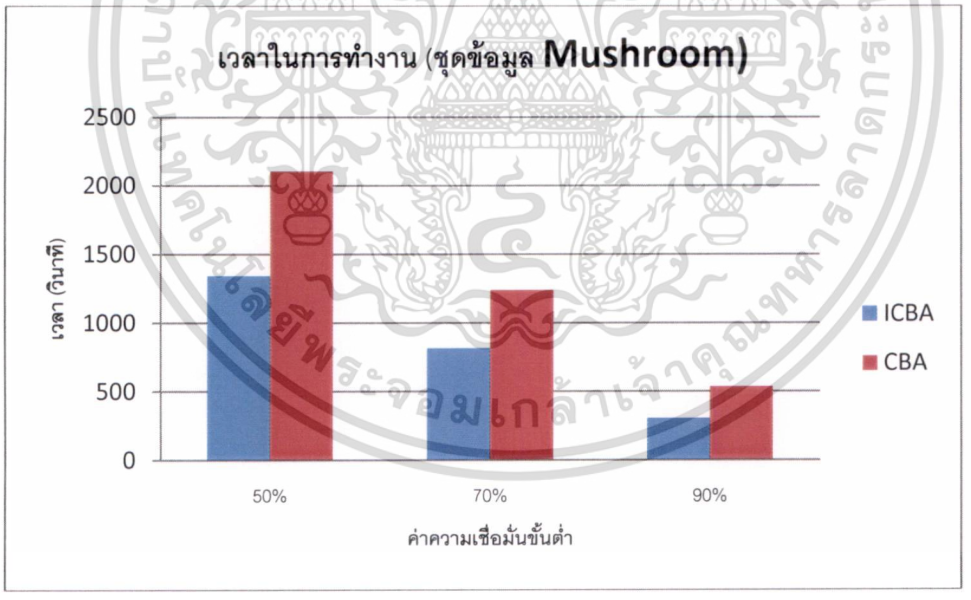
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	เวลาที่ใช้ในการสร้างแชดของกฎความสัมพันธ์แบบมีคลาส (วินาที)	เวลาที่ใช้ในการสร้างแบบจำลองเพื่อการจำแนกประเภทข้อมูล (วินาที)	เวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์(วินาที)	เวลารวม (วินาที)
35%	50%	ICBA	691.364	2143.0333	0.9281	3104.3970
		CBA	1549.8373	2915.2000	0	4465.0370
	70%	ICBA	491.4499	2108.3667	0.8620	2599.8170
		CBA	1556.6523	2588.7000	0	4145.3520
	90%	ICBA	191.3669	715.5238	0.4866	906.8907
		CBA	650.6309	891.0877	0	1541.7190
		CBA	47.5658	25.2781	0	72.8439

ตารางที่ 4.10 (ต่อ) เวลาในการทำงานของการทดลองที่ 5

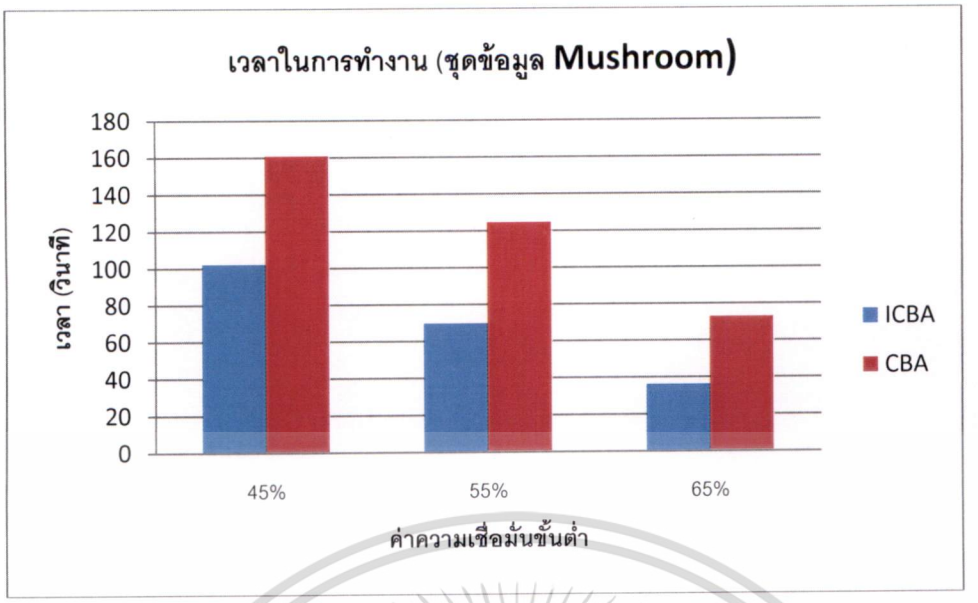
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	เวลาที่ใช้ในการสร้างเซตของกฎความสัมพันธ์แบบมีคลาส (วินาที)	เวลาที่ใช้ในการสร้างแบบจำลองเพื่อการจำแนกประเภทข้อมูล (วินาที)	เวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์(วินาที)	เฉลี่ย (วินาที)
40%	50%	ICBA	543.6879	794.4523	0.3553	1338.1400
		CBA	1136.7236	973.6375	0	2100.361
	70%	ICBA	247.5282	564.8687	0.3162	812.3969
		CBA	410.3103	831.6359	0	1241.9460
	90%	ICBA	143.3982	162.1471	0.0845	305.5453
		CBA	261.7637	274.2056	0	535.9693
45%	50%	ICBA	58.0488	44.0716	0.0565	102.1204
		CBA	97.5031	63.1965	0	160.6996
	70%	ICBA	38.0361	31.5798	0.0420	69.6159
		CBA	77.7128	46.9580	0	124.6708
	90%	ICBA	23.0035	12.9331	0.0125	35.9366
		CBA	47.5658	25.2781	0	72.8439



รูปที่ 4.16 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 20%โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 35% (Mushroom dataset)



รูปที่ 4.17 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 20%โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 40% (Mushroom dataset)



รูปที่ 4.18 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 20% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 45% (Mushroom dataset)

4.4.6 ผลการทดลองที่ 6 (เพิ่ม training data 20%)

การทดลองที่ 5 ทำการทดลองกับข้อมูล Nursery Dataset โดยจะทำการเพิ่ม training data 20% ของ training data เดิม และมีสมมติฐานของการทดลองคือ จำนวนของกฎความสัมพันธ์แบบมีคลาสที่ถูกสร้างด้วยอัลกอริทึม CBA และ ICBA จะต้องเท่ากัน รวมทั้งจำนวนของกฎความสัมพันธ์ในแบบจำลองสำหรับการจำแนกประเภทข้อมูลของอัลกอริทึม CBA และ ICBA ต้องเท่ากันด้วย แต่อัลกอริทึม ICBA จะต้องใช้เวลาในการทำงานน้อยกว่าอัลกอริทึม CBA และอัลกอริทึม ICBA มีจำนวนครั้งที่ทรานแซคชั่นที่ถูกตรวจสอบในขั้นตอนการสร้างแบบจำลองน้อยกว่าอัลกอริทึม CBA สำหรับการทดลองที่ 6กำหนดค่าตัวแปรต่างๆดังนี้ minimum support = 1% , 3% และ 5%, minimum confidence = 45%, 55% และ 65% และทำการทดลองกับtraining data จำนวน 3 ชุดที่แตกต่างกันคือ ชุด A , ชุด B และชุด C จากนั้นนำผลการทดลองของทั้ง 3 ชุดมาหาค่าเฉลี่ย ผลการทดลองที่ 6 แสดงดังตารางที่ 4.11, 4.12 และกราฟเปรียบเทียบเวลาในการทำงานแสดงดังรูปที่ 4.19, 4.20 และ 4.21

จากตารางที่ 4.11จะเห็นได้ว่าการทดลองได้ผลตามสมมติฐานที่กำหนดไว้คือ จำนวนกฎความสัมพันธ์แบบมีคลาสที่ถูกสร้างมาจากทั้งสองอัลกอริทึมมีจำนวนเท่ากัน , จำนวนกฎความสัมพันธ์ในแบบจำลองสำหรับการจำแนกประเภทข้อมูลของทั้งสองอัลกอริทึมก็มีจำนวนเท่ากัน และจำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบในขั้นตอนการสร้างแบบจำลองของอัลกอริทึม ICBA น้อยกว่าอัลกอริทึม CBA เนื่องจากในส่วนของการสร้างแบบจำลองของอัลกอริทึม ICBA, มีการนำเอาความรู้เดิมมาใช้นั่นคืออัลกอริทึมจะทำการตรวจสอบว่ากฎความสัมพันธ์ที่พิจารณาอยู่

เป็นกฎความสัมพันธ์ที่เคยเกิดขึ้นมาก่อนจาก training dataset เดิมหรือไม่ หากทำการตรวจสอบแล้วว่ากฎความสัมพันธ์นั้นเคยเกิดขึ้นมาจาก training dataset เดิม อัลกอริทึม ICBA จะทำการสแกนทรานแซคชั่นเพื่อตรวจสอบการรองรับข้อมูลสอนระบบของกฎความสัมพันธ์นั้นๆเฉพาะ training dataset ที่เพิ่มขยาย โดยไม่จำเป็นต้องสแกนทรานแซคชั่นtraining dataset เดิม ซึ่งในขั้นตอนนี้เป็นการประยุกต์เอาหลักการของอัลกอริทึม FUP มาใช้ จากตารางที่ 4.11แสดงให้เห็นว่าจำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบของอัลกอริทึม ICBA น้อยกว่าอัลกอริทึม CBA มาก และเมื่อพิจารณาตารางที่ 4.12 ซึ่งเป็นตารางสำหรับการแสดงเวลาในการทำงานเปรียบเทียบกันระหว่าง 2 อัลกอริทึม และแสดงให้อยู่ในรูปของกราฟดังรูปที่ 4.19, 4.20และ 4.21 ซึ่งเป็นกราฟเปรียบเทียบเวลาการทำงานของทั้งสองอัลกอริทึมพบว่า อัลกอริทึม ICBA ทำงานได้เร็วกว่าอัลกอริทึม CBA และเวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์มีค่าน้อยมากซึ่งแสดงให้เห็นว่าอัลกอริทึมไม่ได้เสียเวลาในการเปรียบ เทียบกฎความสัมพันธ์เลย

ตารางที่ 4.11 ผลการทดลองที่ 6

ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	จำนวนกฎความสัมพันธ์แบบมีคลาส	จำนวนกฎความสัมพันธ์ที่ผ่านเกณฑ์ในแบบจำลองเพื่อการจำแนกประเภทข้อมูล	จำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบ (ครั้ง)	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลสอนระบบ	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลทดสอบระบบ
1%	45%	ICBA	270	164	244732	61%	59%
		CBA	270	164	470799	61%	59%
	55%	ICBA	210	152	199608	65%	64%
		CBA	210	152	451201	65%	64%
	65%	ICBA	103	61	115016	66%	62%
		CBA	103	61	325182	66%	62%
3%	45%	ICBA	78	43	73527	67%	63%
		CBA	78	43	150847	67%	63%
	55%	ICBA	59	35	65461	65%	60%
		CBA	59	35	140290	65%	60%
	65%	ICBA	21	20	33277	67%	63%
		CBA	21	20	89470	67%	63%

ตารางที่ 4.11 (ต่อ) ผลการทดลองที่ 6

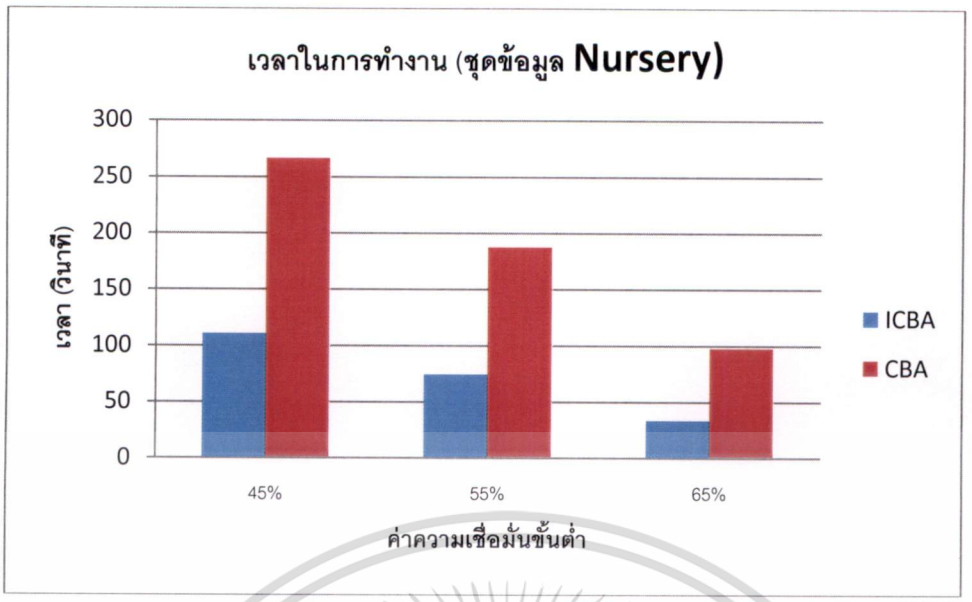
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	จำนวนกฎความสัมพันธ์แบบมีคลาส	จำนวนกฎความสัมพันธ์ในแบบจำลองเพื่อการจำแนกประเภทข้อมูล	จำนวนครั้งที่ทำนายชนิดสัตว์ถูกตรวจสอบ (ครั้ง)	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลสอนระบบ	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลทดสอบระบบ
5%	45%	ICBA	38	25	38417	62%	59%
		CBA	38	25	92707	62%	59%
	55%	ICBA	27	23	28195	60%	60%
		CBA	27	23	84543	60%	60%
	65%	ICBA	12	12	17655	64%	65%
		CBA	12	12	55366	64%	65%

ตารางที่ 4.12 เวลาในการทำงานของการทดลองที่ 6

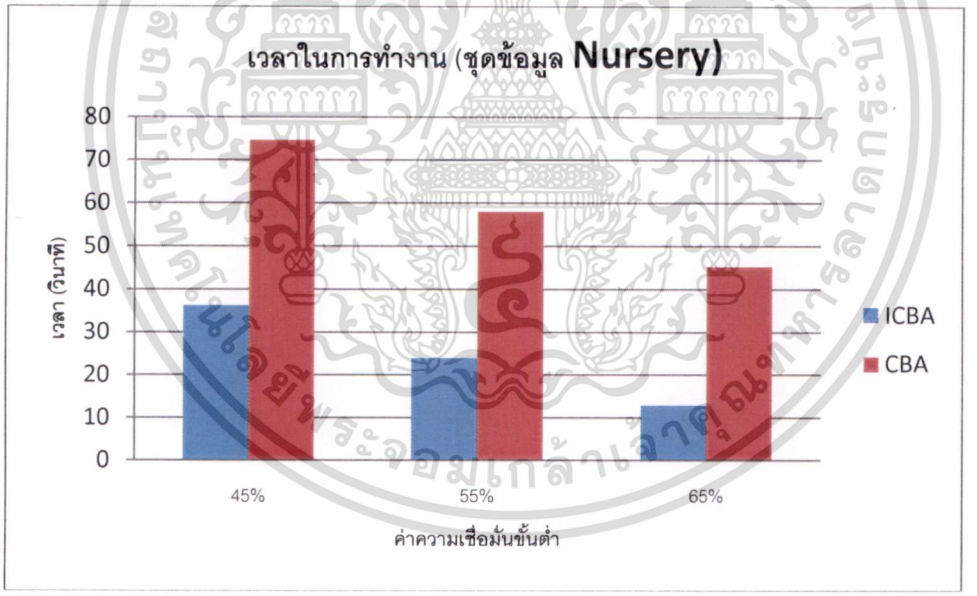
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	เวลาที่ใช้ในการสร้างชุดของกฎความสัมพันธ์แบบมีคลาส (วินาที)	เวลาที่ใช้ในการสร้างแบบจำลองเพื่อการจำแนกประเภทข้อมูล (วินาที)	เวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์(วินาที)	เวลารวม (วินาที)
1%	45%	ICBA	39.0509	71.3554	0.1070	110.4063
		CBA	173.6066	92.6482	0	266.2548
	55%	ICBA	30.2012	43.8074	0.0775	74.0086
		CBA	124.1000	63.2761	0	187.3761
	65%	ICBA	21.1927	11.9087	0.0376	33.1014
		CBA	73.8834	23.4232	0	97.3066
		CBA	25.2990	2.9267	0	28.2257

ตารางที่ 4.12 (ต่อ) เวลาในการทำงานของการทดลองที่ 6

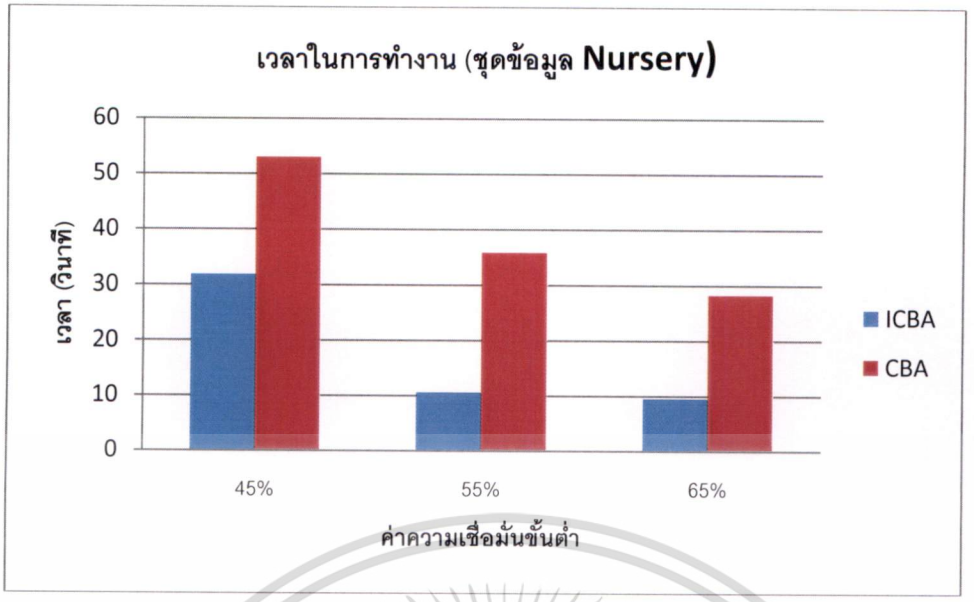
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	เวลาที่ใช้ในการสร้างเซตของกฎความสัมพันธ์แบบมีคลาส (วินาที)	เวลาที่ใช้ในการสร้างแบบจำลองเพื่อการจำแนกประเภทข้อมูล (วินาที)	เวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์(วินาที)	เวลารวม (วินาที)
3%	45%	ICBA	29.1203	7.0101	0.0273	36.1304
		CBA	62.7461	11.9536	0	74.6997
	55%	ICBA	19.0803	4.7086	0.0163	23.7889
		CBA	48.8245	9.2770	0	58.1015
	65%	ICBA	10.0778	2.7984	0.0060	12.8762
		CBA	40.4943	4.5893	0	45.3836
55%	45%	ICBA	28.2520	3.5429	0.0152	31.7949
		CBA	47.3904	5.5861	0	52.9765
	55%	ICBA	7.8276	2.6780	0.0071	10.5056
		CBA	31.2229	4.5896	0	35.8125
	65%	ICBA	7.8315	1.6225	0.0034	9.4540
		CBA	25.2990	2.9267	0	28.2257



รูปที่ 4.19 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 20% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 1% (Nursery dataset)



รูปที่ 4.20 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 20% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 3% (Nursery dataset)



รูปที่ 4.21 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 20%โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 5% (Nursery dataset)

4.4.7 ผลการทดลองที่ 7 (เพิ่ม training data 30%)

การทดลองที่ 7 ทำการทดลองกับข้อมูล Adult Dataset โดยจะทำการเพิ่ม training data 30% ของ training data เดิม และมีสมมติฐานของการทดลองคือ จำนวนของกฎความสัมพันธ์แบบมีคลาสที่ถูกสร้างด้วยอัลกอริทึม CBA และ ICBA จะต้องเท่ากัน รวมทั้งจำนวนของกฎความสัมพันธ์ในแบบจำลองสำหรับการจำแนกประเภทข้อมูลของอัลกอริทึม CBA และ ICBA ต้องเท่ากันด้วย แต่อัลกอริทึม ICBA จะต้องใช้เวลาในการทำงานน้อยกว่าอัลกอริทึม CBA และอัลกอริทึม ICBA มีจำนวนครั้งที่ทรานแซคชั่นที่ถูกตรวจสอบในขั้นตอนการสร้างแบบจำลองน้อยกว่าอัลกอริทึม CBA สำหรับการทดลองที่ 7 กำหนดค่าตัวแปรต่างๆดังนี้ minimum support = 15% , 20% และ 25%, minimum confidence = 70%, 80% และ 90% และทำการทดลองกับtraining data จำนวน 3 ชุดที่แตกต่างกันคือ ชุด A , ชุด B และชุด C จากนั้นนำผลการทดลองของทั้ง 3 ชุดมาหาค่าเฉลี่ย ผลการทดลองที่ 7 แสดงดังตารางที่ 4.13 , 4.14 และกราฟเปรียบเทียบเวลาในการทำงานแสดงดังรูปที่ 4.22, 4.23 และ 4.24

จากตารางที่ 4.13 จะเห็นได้ว่าการทดลองได้ผลตามสมมติฐานที่กำหนดไว้คือ จำนวนกฎความสัมพันธ์แบบมีคลาสที่ถูกสร้างมาจากทั้งสองอัลกอริทึมมีจำนวนเท่ากัน , จำนวนกฎความสัมพันธ์ในแบบจำลองสำหรับการจำแนกประเภทข้อมูลของทั้งสองอัลกอริทึมก็มีจำนวนเท่ากัน และจำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบในขั้นตอนการสร้างแบบจำลองของอัลกอริทึม ICBA น้อยกว่าอัลกอริทึม CBA เนื่องจากในส่วนของการสร้างแบบจำลองของอัลกอริทึม ICBA, มีการนำเอาความรู้เดิมมาใช้นั่นคืออัลกอริทึมจะทำการตรวจสอบว่ากฎความสัมพันธ์ที่พิจารณาอยู่

เป็นกฎความสัมพันธ์ที่เคยเกิดขึ้นมาก่อนจาก training dataset เดิมหรือไม่ หากทำการตรวจสอบแล้วว่ากฎความสัมพันธ์นั้นเคยเกิดขึ้นมาจาก training dataset เดิม อัลกอริทึม ICBA จะทำการสแกนทรานแซคชั่นเพื่อตรวจสอบการรองรับข้อมูลสอนระบบของกฎความสัมพันธ์นั้นๆเฉพาะ training dataset ที่เพิ่มขยาย โดยไม่จำเป็นต้องสแกนทรานแซคชั่นtraining dataset เดิม ซึ่งในขั้นตอนนี้เป็นการประยุกต์เอาหลักการของอัลกอริทึม FUP มาใช้ จากตารางที่ 4.13 แสดงให้เห็นว่าจำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบของอัลกอริทึม ICBA น้อยกว่าอัลกอริทึม CBA มาก และเมื่อพิจารณาตารางที่ 4.14 ซึ่งเป็นตารางสำหรับการแสดงเวลาในการทำงานเปรียบเทียบกันระหว่าง 2 อัลกอริทึม และแสดงให้อยู่ในรูปของกราฟดังรูปที่ 4.22, 4.23 และ 4.24 ซึ่งเป็นกราฟเปรียบเทียบเวลาการทำงานของทั้งสองอัลกอริทึมพบว่า อัลกอริทึม ICBA ทำงานได้เร็วกว่าอัลกอริทึม CBA และเวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์มีค่าน้อยมากซึ่งแสดงให้เห็นว่าอัลกอริทึมไม่ได้เสียเวลาในการเปรียบ เทียบกฎความสัมพันธ์เลย

ตารางที่ 4.13 ผลการทดลองที่ 7

ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	จำนวนกฎความสัมพันธ์แบบมีคลาส	จำนวนกฎความสัมพันธ์ในแบบจำลองเพื่อการจำแนกประเภทข้อมูล	จำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบ (ครั้ง)	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลสอนระบบ	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลทดสอบระบบ
15%	70%	ICBA	1831	1534	2389365	72%	70%
		CBA	1831	1534	8439617	72%	70%
	80%	ICBA	1493	1192	1925247	75%	74%
		CBA	1493	1192	7078895	75%	74%
	90%	ICBA	924	607	1301506	70%	69%
		CBA	924	607	3540174	70%	69%
20%	70%	ICBA	1276	1097	1025412	73%	69%
		CBA	1276	1097	3739872	73%	69%
	80%	ICBA	968	739	982073	77%	78%
		CBA	968	739	3323610	77%	78%
	90%	ICBA	524	362	487382	74%	74%
		CBA	524	362	1169908	74%	74%

ตารางที่ 4.13 (ต่อ) ผลการทดลองที่ 7

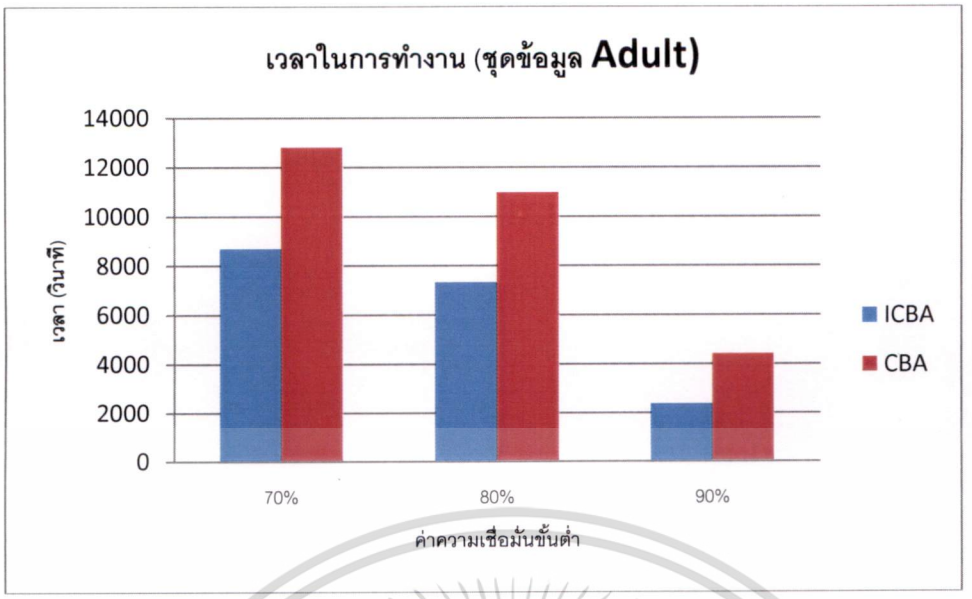
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	จำนวนกฎความสัมพันธ์แบบมีคลาส	จำนวนกฎความสัมพันธ์ในแบบจำลองเพื่อการจำแนกประเภทข้อมูล	จำนวนครั้งที่การแทนเซตนั้นถูกตรวจสอบ (ครั้ง)	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลสอนระบบ	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลทดสอบระบบ
25%	70%	ICBA	501	335	634530	75%	71%
		CBA	501	335	2140215	75%	71%
	80%	ICBA	386	274	570083	76%	76%
		CBA	386	274	1872618	76%	76%
	90%	ICBA	232	91	214195	79%	75%
		CBA	232	91	399826	79%	75%

ตารางที่ 4.14 เวลาในการทำงานของการทดลองที่ 7

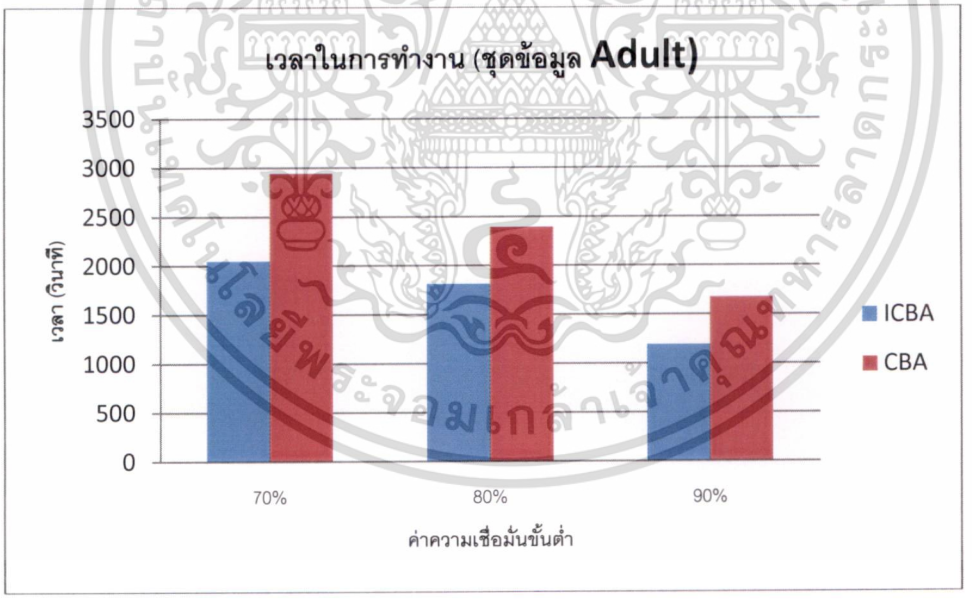
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	เวลาที่ใช้ในการสร้างเซตของกฎความสัมพันธ์แบบมีคลาส (วินาที)	เวลาที่ใช้ในการสร้างแบบจำลองเพื่อการจำแนกประเภทข้อมูล (วินาที)	เวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์(วินาที)	เวลารวม (วินาที)
15%	70%	ICBA	1558.8973	7115.5667	0.5840	8674.4640
		CBA	3522.6667	9288.2000	0	12810.8700
	80%	ICBA	1258.8732	6053.4667	0.4499	7312.3400
		CBA	2717.1333	8243.8333	0	10960.9700
	90%	ICBA	560.1918	1794.3542	0.1361	2354.5460
		CBA	912.1333	3477.7604	0	4389.8940

ตารางที่ 4.14 (ต่อ) เวลาในการทำงานของการทดลองที่ 7

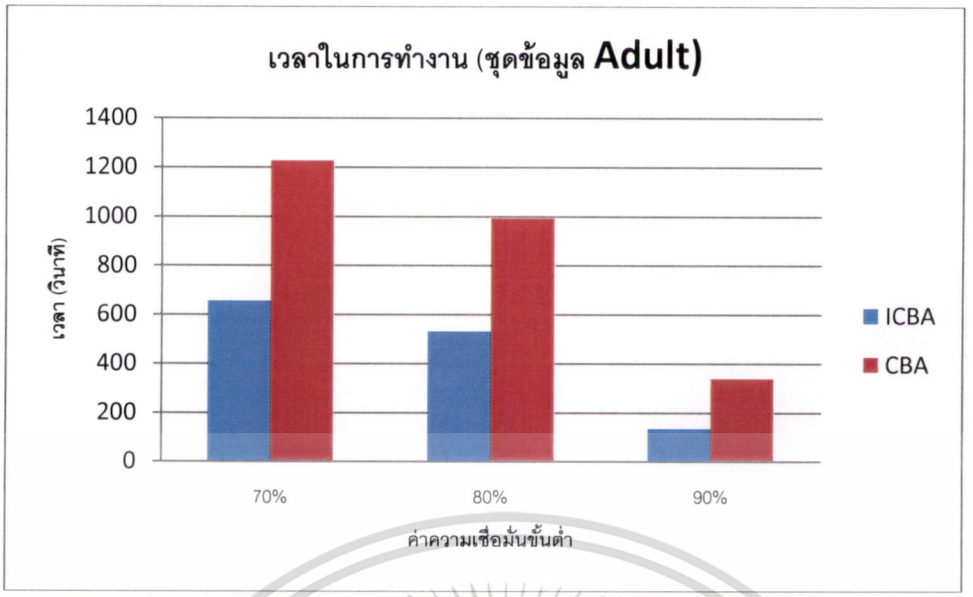
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	เวลาที่ใช้ในการสร้างเซตของกฎความสัมพันธ์แบบมีคลาส (วินาที)	เวลาที่ใช้ในการสร้างแบบจำลองเพื่อการจำแนกประเภทข้อมูล (วินาที)	เวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์(วินาที)	เวลารวม (วินาที)
20%	70%	ICBA	931.5521	1113.7000	0.2321	2045.2520
		CBA	1334.4000	1611.5000	0	2945.9000
	80%	ICBA	731.2720	1076.4107	0.1914	1807.6830
		CBA	1135.4667	1260.2000	0	2395.6670
	90%	ICBA	431.4537	758.8400	0.0398	1190.2940
		CBA	585.3333	1090.2170	0	1675.5500
25%	70%	ICBA	183.4778	470.8235	0.1075	654.3013
		CBA	598.8401	627.3110	0	1226.1510
	80%	ICBA	128.5436	400.7893	0.0871	529.3329
		CBA	400.0615	592.0283	0	992.0898
	90%	ICBA	72.5765	60.3615	0.0081	132.9380
		CBA	201.2471	134.4748	0	335.7219



รูปที่ 4.22 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 30% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 15% (Adult dataset)



รูปที่ 4.23 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 30% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 20% (Adult dataset)



รูปที่ 4.24 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 30% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 25% (Adult dataset)

4.4.8 ผลการทดลองที่ 8 (เพิ่ม training data 30%)

การทดลองที่ 8 ทำการทดลองกับข้อมูล Mushroom Dataset โดยจะทำการเพิ่ม training data 30% ของ training data เดิม และมีสมมติฐานของการทดลองคือ จำนวนของกฎความสัมพันธ์แบบมีคลาสที่ถูกสร้างด้วยอัลกอริทึม CBA และ ICBA จะต้องเท่ากัน รวมทั้งจำนวนของกฎความสัมพันธ์ในแบบจำลองสำหรับการจำแนกประเภทข้อมูลของอัลกอริทึม CBA และ ICBA ต้องเท่ากันด้วย แต่อัลกอริทึม ICBA จะต้องใช้เวลาในการทำงานน้อยกว่าอัลกอริทึม CBA และอัลกอริทึม ICBA มีจำนวนครั้งที่ทรานแซคชั่นที่ถูกตรวจสอบในขั้นตอนการสร้างแบบจำลองน้อยกว่าอัลกอริทึม CBA สำหรับการทดลองที่ 8 กำหนดค่าตัวแปรต่างๆดังนี้ minimum support = 35% , 40% และ 45%, minimum confidence = 50%, 70% และ 90% และทำการทดลองกับtraining data จำนวน 3 ชุดที่แตกต่างกันคือ ชุด A , ชุด B และชุด C จากนั้นนำผลการทดลองของทั้ง 3 ชุดมาหาค่าเฉลี่ย ผลการทดลองที่ 8 แสดงดังตารางที่ 4.15 , 4.16 และกราฟเปรียบเทียบเวลาในการทำงานแสดงดังรูปที่ 4.25, 4.26 และ 4.27

จากตารางที่ 4.15 จะเห็นได้ว่าการทดลองได้ผลตามสมมติฐานที่กำหนดไว้คือ จำนวนกฎความสัมพันธ์แบบมีคลาสที่ถูกสร้างมาจากทั้งสองอัลกอริทึมมีจำนวนเท่ากัน , จำนวนกฎความสัมพันธ์ในแบบจำลองสำหรับการจำแนกประเภทข้อมูลของทั้งสองอัลกอริทึมก็มีจำนวนเท่ากัน และจำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบในขั้นตอนการสร้างแบบจำลองของอัลกอริทึม ICBA น้อยกว่าอัลกอริทึม CBA เนื่องจากในส่วนของการสร้างแบบจำลองของอัลกอริทึม ICBA, มีการนำเอาความรู้เดิมมาใช้นั่นคืออัลกอริทึมจะทำการตรวจสอบว่ากฎความสัมพันธ์ที่พิจารณาอยู่

เป็นกฎความสัมพันธ์ที่เคยเกิดขึ้นมาก่อนจาก training dataset เดิมหรือไม่ หากทำการตรวจสอบแล้วว่ากฎความสัมพันธ์นั้นเคยเกิดขึ้นมาจาก training dataset เดิม อัลกอริทึม ICBA จะทำการสแกนทรานแซคชั่นเพื่อตรวจสอบการรองรับข้อมูลสอนระบบของกฎความสัมพันธ์นั้นๆเฉพาะ training dataset ที่เพิ่มขยาย โดยไม่จำเป็นต้องสแกนทรานแซคชั่นtraining dataset เดิม ซึ่งในขั้นตอนนี้เป็นการประยุกต์เอาหลักการของอัลกอริทึม FUP มาใช้ จากตารางที่ 4.15 แสดงให้เห็นว่าจำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบของอัลกอริทึม ICBA น้อยกว่าอัลกอริทึม CBA มาก และเมื่อพิจารณาตารางที่ 4.16 ซึ่งเป็นตารางสำหรับการแสดงเวลาในการทำงานเปรียบเทียบกันระหว่าง 2 อัลกอริทึม และแสดงให้อยู่ในรูปของกราฟดังรูปที่ 4.25, 4.26 และ 4.27 ซึ่งเป็นกราฟเปรียบเทียบเวลาการทำงานของทั้งสองอัลกอริทึมพบว่า อัลกอริทึม ICBA ทำงานได้เร็วกว่าอัลกอริทึม CBA และเวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์มีค่าน้อยมากซึ่งแสดงให้เห็นว่าอัลกอริทึมไม่ได้เสียเวลาในการเปรียบ เทียบกฎความสัมพันธ์เลย

ตารางที่ 4.15 ผลการทดลองที่ 8

ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	จำนวนกฎความสัมพันธ์แบบดิสเคลาส	จำนวนกฎความสัมพันธ์ในแบบจำลองเพื่อการจำแนกประเภทข้อมูล	จำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบ (ครั้ง)	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลสอนระบบ	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลทดสอบระบบ
35%	50%	ICBA	1436	1413	489756	82%	82%
		CBA	1436	1413	1989527	82%	82%
	70%	ICBA	1371	1153	464089	86%	86%
		CBA	1371	1153	1904321	86%	86%
	90%	ICBA	741	300	264942	85%	84%
		CBA	741	300	1087169	85%	84%
40%	50%	ICBA	839	807	844357	81%	79%
		CBA	839	807	4137906	81%	79%
	70%	ICBA	776	572	332598	83%	84%
		CBA	776	572	718544	83%	84%
	90%	ICBA	227	97	634489	84%	81%
		CBA	227	97	348861	84%	81%

ตารางที่ 4.15 (ต่อ) ผลการทดลองที่ 8

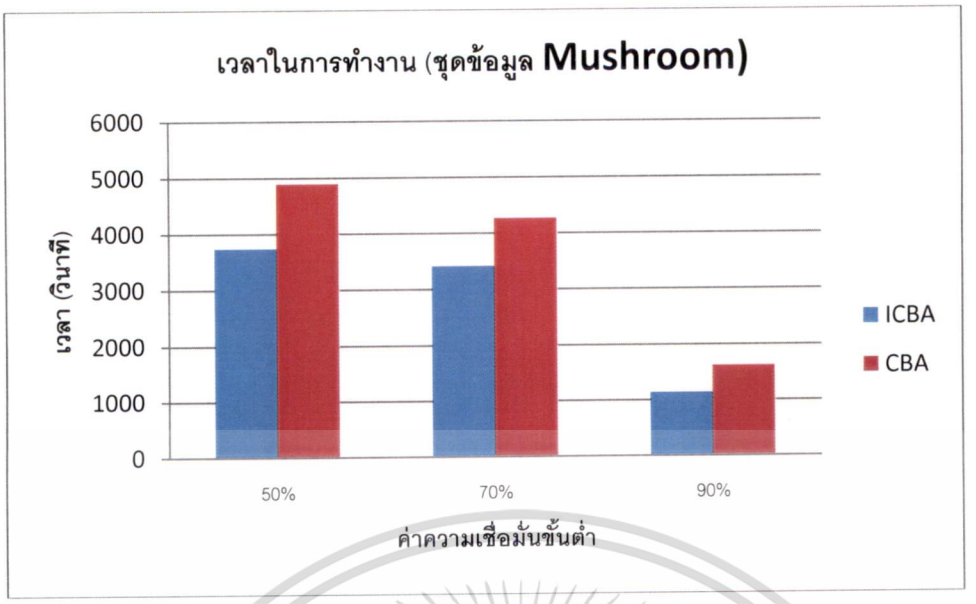
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	จำนวนกฎความสัมพันธ์แบบมีคลาส	จำนวนกฎความสัมพันธ์ในแบบจำลองเพื่อการจำแนกประเภทข้อมูล	จำนวนครั้งที่ทราเวอร์สต้นไม้ถูกตรวจสอบ (ครั้ง)	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลสอนระบบ	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลทดสอบระบบ
45%	50%	ICBA	231	220	122261	82%	80%
		CBA	231	220	323654	82%	80%
	70%	ICBA	189	102	107126	84%	79%
		CBA	189	102	267630	84%	79%
	90%	ICBA	51	36	22382	83%	83%
		CBA	51	36	81459	83%	83%

ตารางที่ 4.16 เวลาในการทำงานของการทดลองที่ 8

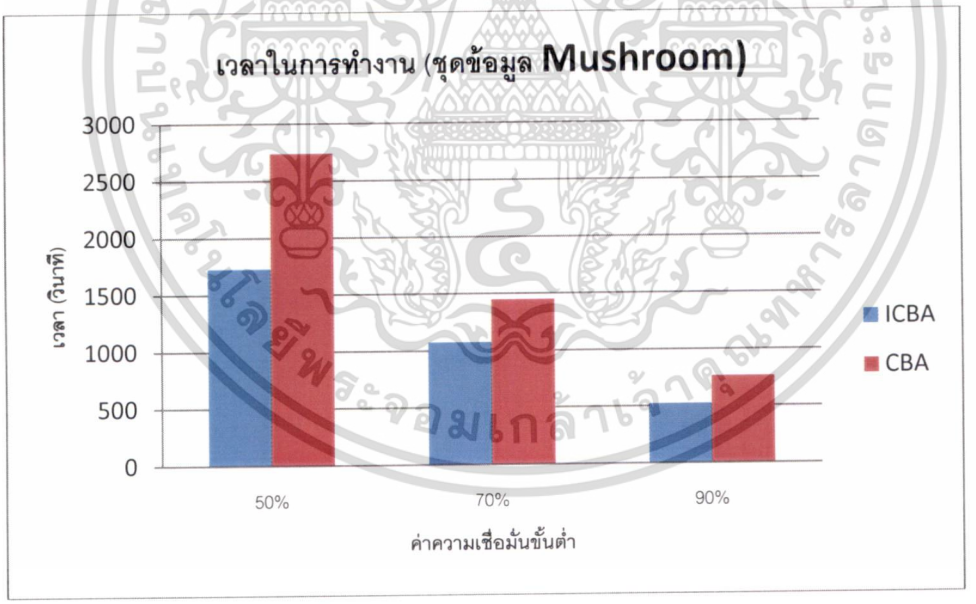
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	เวลาที่ใช้ในการสร้างเซตของกฎความสัมพันธ์แบบมีคลาส (วินาที)	เวลาที่ใช้ในการสร้างแบบจำลองเพื่อการจำแนกประเภทข้อมูล (วินาที)	เวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์ (วินาที)	เวลารวม (วินาที)
35%	50%	ICBA	856.6218	2874.5667	0.9528	3731.1890
		CBA	1916.3113	2972.0667	0	4888.3780
	70%	ICBA	668.5967	2736.3000	0.9483	3404.8970
		CBA	1443.8495	2818.5333	0	4262.3830
	90%	ICBA	326.8998	798.8038	0.5129	1128.7040
		CBA	793.7584	814.8149	0	1608.5730

ตารางที่ 4.16 (ต่อ) เวลาในการทำงานของการทดลองที่ 8

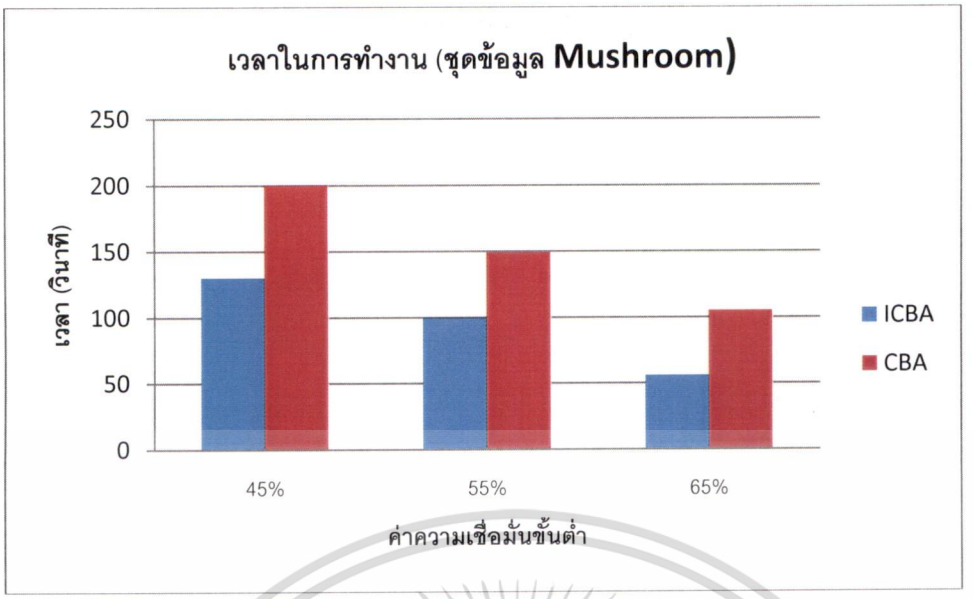
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	เวลาที่ใช้ในการสร้างเซตของกฎความสัมพันธ์แบบมีคลาส (วินาที)	เวลาที่ใช้ในการสร้างแบบจำลองเพื่อการจำแนกประเภทข้อมูล (วินาที)	เวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์(วินาที)	เวลารวม (วินาที)
40%	50%	ICBA	759.9829	962.6256	0.3710	1722.6090
		CBA	1651.9539	1090.3858	0	2742.3400
	70%	ICBA	359.9119	712.4781	0.3402	1072.3900
		CBA	551.6107	895.8150	0	1447.4260
	90%	ICBA	254.9293	268.3799	0.0899	523.3092
		CBA	385.1556	378.8930	0	764.0486
45%	50%	ICBA	73.0778	56.9948	0.2270	130.0726
		CBA	105.4438	94.7484	0	200.1922
	70%	ICBA	52.9195	46.5958	0.2474	99.5153
		CBA	85.4043	63.5540	0	148.9583
	90%	ICBA	31.8771	23.8501	0.0138	55.7272
		CBA	65.3578	39.5362	0	104.8940



รูปที่ 4.25 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 30% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 35% (Mushroom dataset)



รูปที่ 4.26 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 30% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 40% (Mushroom dataset)



รูปที่ 4.27 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 30% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 45% (Mushroom dataset)

4.4.9 ผลการทดลองที่ 9 (เพิ่ม training data 30%)

การทดลองที่ 9 ทำการทดลองกับข้อมูล Nursery Dataset โดยจะทำการเพิ่ม training data 30% ของ training data เดิม และมีสมมติฐานของการทดลองคือ จำนวนของกฎความสัมพันธ์แบบมีคลาสที่ถูกสร้างด้วยอัลกอริทึม CBA และ ICBA จะต้องเท่ากัน รวมทั้งจำนวนของกฎความสัมพันธ์ในแบบจำลองสำหรับการจำแนกประเภทข้อมูลของอัลกอริทึม CBA และ ICBA ต้องเท่ากันด้วย แต่อัลกอริทึม ICBA จะต้องใช้เวลาในการทำงานน้อยกว่าอัลกอริทึม CBA และอัลกอริทึม ICBA มีจำนวนครั้งที่ทรานแซคชั่นที่ถูกตรวจสอบในขั้นตอนการสร้างแบบจำลองน้อยกว่าอัลกอริทึม CBA สำหรับการทดลองที่ 9กำหนดค่าตัวแปรต่างๆดังนี้ minimum support = 1% , 3% และ 5%, minimum confidence = 45%, 55% และ 65% และทำการทดลองกับtraining data จำนวน 3 ชุดที่แตกต่างกันคือ ชุด A , ชุด B และชุด C จากนั้นนำผลการทดลองของทั้ง 3 ชุดมาหาค่าเฉลี่ย ผลการทดลองที่ 9 แสดงดังตารางที่ 4.17 , 4.18 และกราฟเปรียบเทียบเวลาในการทำงานแสดงดังรูปที่ 4.28, 4.29 และ 4.30

จากตารางที่ 4.17 จะเห็นได้ว่าการทดลองได้ผลตามสมมติฐานที่กำหนดไว้คือ จำนวนกฎความสัมพันธ์แบบมีคลาสที่ถูกสร้างมาจากทั้งสองอัลกอริทึมมีจำนวนเท่ากัน , จำนวนกฎความสัมพันธ์ในแบบจำลองสำหรับการจำแนกประเภทข้อมูลของทั้งสองอัลกอริทึมก็มีจำนวนเท่ากัน และจำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบในขั้นตอนการสร้างแบบจำลองของอัลกอริทึม ICBA น้อยกว่าอัลกอริทึม CBA เนื่องจากในส่วนของการสร้างแบบจำลองของอัลกอริทึม ICBA, มีการนำเอาความรู้เดิมมาใช้นั่นคืออัลกอริทึมจะทำการตรวจสอบว่ากฎความสัมพันธ์ที่พิจารณาอยู่

เป็นกฎความสัมพันธ์ที่เคยเกิดขึ้นมาก่อนจาก training dataset เดิมหรือไม่ หากทำการตรวจสอบแล้วว่ากฎความสัมพันธ์นั้นเคยเกิดขึ้นมาจาก training dataset เดิม อัลกอริทึม ICBA จะทำการสแกนทรานแซคชั่นเพื่อตรวจสอบการรองรับข้อมูลสอนระบบของกฎความสัมพันธ์นั้นๆเฉพาะ training dataset ที่เพิ่มขยาย โดยไม่จำเป็นต้องสแกนทรานแซคชั่นtraining dataset เดิม ซึ่งในขั้นตอนนี้เป็นการประยุกต์เอาหลักการของอัลกอริทึม FUP มาใช้ จากตารางที่ 4.17 แสดงให้เห็นว่าจำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบของอัลกอริทึม ICBA น้อยกว่าอัลกอริทึม CBA มาก และเมื่อพิจารณาตารางที่ 4.18 ซึ่งเป็นตารางสำหรับการแสดงเวลาในการทำงานเปรียบเทียบกันระหว่าง 2 อัลกอริทึม และแสดงให้อยู่ในรูปของกราฟดังรูปที่ 4.28, 4.29 และ 4.30 ซึ่งเป็นกราฟเปรียบเทียบเวลาการทำงานของทั้งสองอัลกอริทึมพบว่า อัลกอริทึม ICBA ทำงานได้เร็วกว่าอัลกอริทึม CBA และเวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์มีค่าน้อยมากซึ่งแสดงให้เห็นว่าอัลกอริทึมไม่ได้เสียเวลาในการเปรียบ เทียบกฎความสัมพันธ์เลย

ตารางที่ 4.17 ผลการทดลองที่ 9

ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	จำนวนกฎความสัมพันธ์แบบมีคลาส	จำนวนกฎความสัมพันธ์ในแบบจำลองเพื่อการจำแนกประเภทข้อมูล	จำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบ (ครั้ง)	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลสอนระบบ	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลทดสอบระบบ
1%	45%	ICBA	267	173	282245	67%	63%
		CBA	267	173	513676	67%	63%
	55%	ICBA	209	127	266899	63%	63%
		CBA	209	127	490475	63%	63%
	65%	ICBA	99	63	149748	64%	64%
		CBA	99	63	345956	64%	64%
3%	45%	ICBA	77	50	98755	65%	59%
		CBA	77	50	153948	65%	59%
	55%	ICBA	58	38	87687	62%	60%
		CBA	58	38	143312	62%	60%
	65%	ICBA	20	19	40579	66%	66%
		CBA	20	19	87293	66%	66%

ตารางที่ 4.17 (ต่อ) ผลการทดลองที่ 9

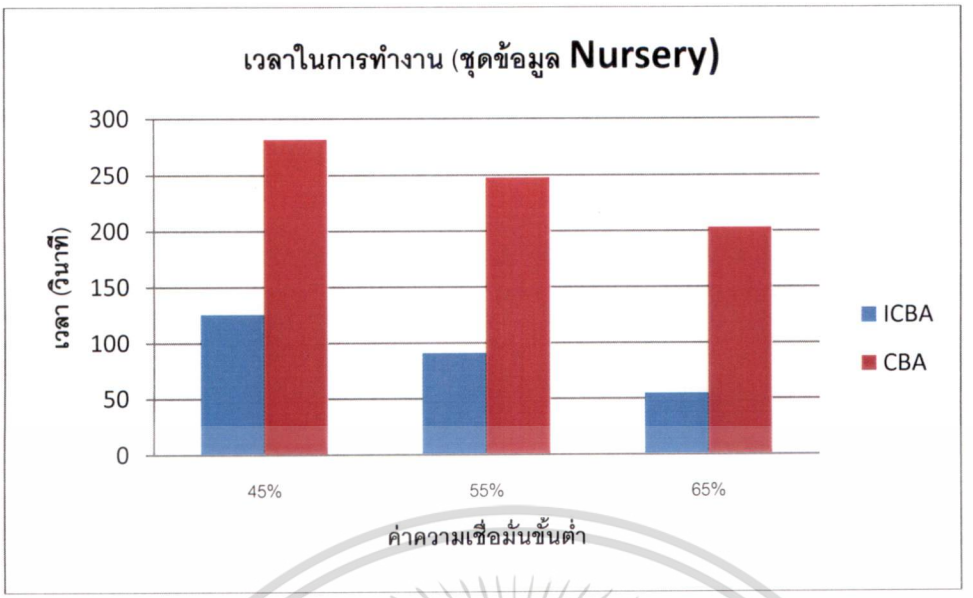
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	จำนวนกฎความสัมพันธ์แบบคลาส	จำนวนกฎความสัมพันธ์ในแบบจำลองเพื่อการจำแนกประเภทข้อมูล	จำนวนครั้งที่ทรานแซคชั่นถูกตรวจสอบ (ครั้ง)	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลสอนระบบ	ค่าความแม่นยำเมื่อตรวจสอบในข้อมูลทดสอบระบบ
5%	45%	ICBA	38	24	37035	61%	59%
		CBA	38	24	100489	61%	59%
	55%	ICBA	28	21	29320	63%	61%
		CBA	28	21	92167	63%	61%
	65%	ICBA	12	12	18554	60%	62%
		CBA	12	12	62989	60%	62%

ตารางที่ 4.18 เวลาในการทำงานของการทดลองที่ 9

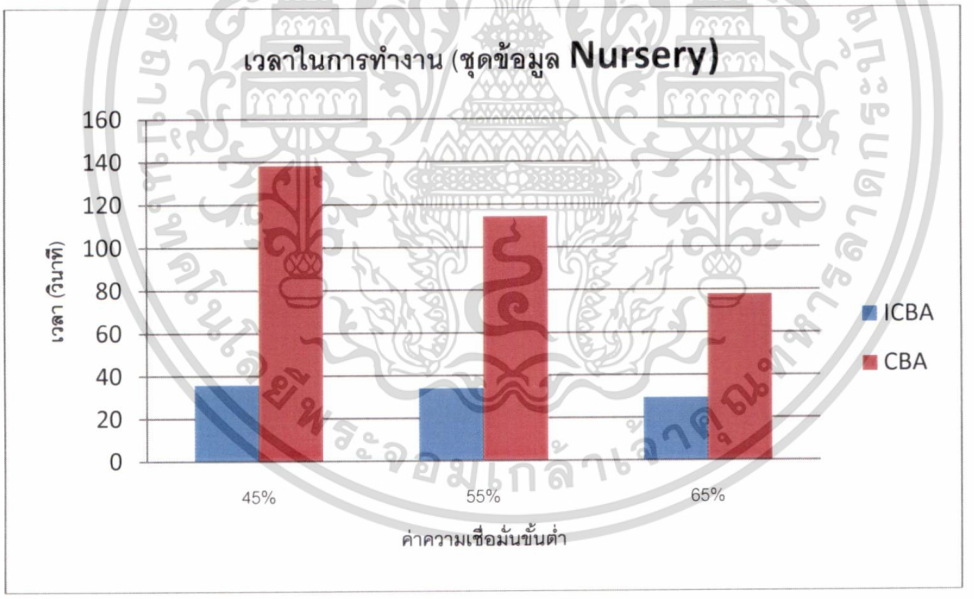
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	เวลาที่ใช้ในการสร้างเซตของกฎความสัมพันธ์แบบคลาส (วินาที)	เวลาที่ใช้ในการสร้างแบบจำลองเพื่อการจำแนกประเภทข้อมูล (วินาที)	เวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์(วินาที)	เวลารวม (วินาที)
1%	45%	ICBA	41.3678	84.0303	0.1205	125.3981
		CBA	178.4096	103.1022	0	281.5118
	55%	ICBA	41.3844	49.263	0.0854	90.6474
		CBA	177.9109	69.3069	0	247.2178
	65%	ICBA	41.674	12.7966	0.0398	54.4706
		CBA	177.6511	24.819	0	202.4701

ตารางที่ 4.18 (ต่อ) เวลาในการทำงานของการทดลองที่ 9

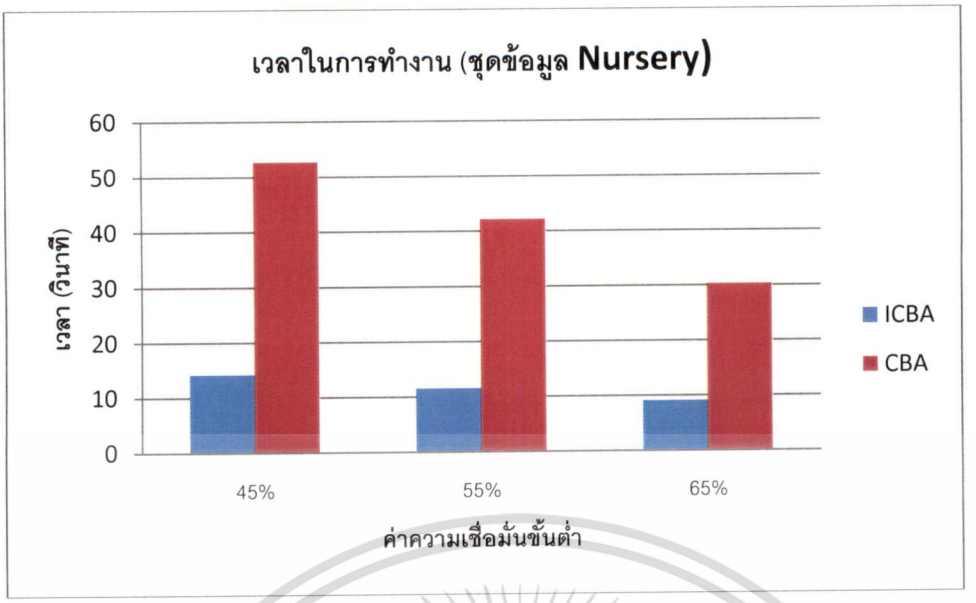
ค่าสนับสนุนขั้นต่ำ	ค่าความเชื่อมั่นขั้นต่ำ	อัลกอริทึม	เวลาที่ใช้ในการสร้างเซตของกฎความสัมพันธ์แบบดิสครีต (วินาที)	เวลาที่ใช้ในการสร้างแบบจำลองเพื่อการจำแนกประเภทข้อมูล (วินาที)	เวลาที่ใช้ในการตรวจสอบกฎความสัมพันธ์(วินาที)	เวลารวม (วินาที)
3%	45%	ICBA	22.6110	13.2178	0.0274	35.8288
		CBA	115.9887	22.3054	0	138.2941
	55%	ICBA	22.7394	11.3435	0.0171	34.0829
		CBA	97.2340	17.1372	0	114.3712
	65%	ICBA	22.6684	6.8353	0.0056	29.5037
		CBA	67.4555	10.3060	0	77.7615
55%	45%	ICBA	11.8721	2.2403	0.0150	14.1124
		CBA	46.9187	5.6947	0	52.6134
	55%	ICBA	9.9694	1.5255	0.0073	11.4949
		CBA	37.1758	4.9016	0	42.0774
	65%	ICBA	7.1537	1.8315	0.0036	8.9852
		CBA	26.9023	3.2439	0	30.1462



รูปที่ 4.28 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 30% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 1% (Nursery dataset)



รูปที่ 4.29 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 30% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 3% (Nursery dataset)



รูปที่ 4.30 กราฟเปรียบเทียบเวลาในการทำงานระหว่าง 2 อัลกอริทึม เมื่อมีการเพิ่มขยาย training data 30% โดยกำหนดค่าสนับสนุนขั้นต่ำที่ 5% (Nursery dataset)

4.5 สรุปผลการทดลอง

จากผลการทดลองทั้งหมด 9 กรณีคือทดลองกับชุดข้อมูล, จำนวนข้อมูลที่เพิ่มขึ้น, ค่าสนับสนุนขั้นต่ำและค่าความเชื่อมั่นที่แตกต่างกัน3 ค่า ผลการทดลองแสดงให้เห็นว่า จำนวนของกฎความสัมพันธ์ ที่สร้างจากอัลกอริทึม ICBA เท่ากับอัลกอริทึม CBA แบบจำลองที่สร้างได้จากทั้งสองอัลกอริทึมออกมาเหมือนกัน ทำให้ทราบว่า อัลกอริทึม ICBA มีผลลัพธ์ที่ถูกต้องเป็นไปตามสมมติฐานที่ได้วางไว้ และเมื่อพิจารณาจากกราฟแสดงการเปรียบเทียบเวลาที่ใช้ในการทำงานของสองอัลกอริทึม พบว่าทุกๆการทดลองอัลกอริทึม ICBA สามารถทำงานได้เร็วกว่าอัลกอริทึม CBA เมื่อมีการเพิ่มขยายของ training data เนื่องจากอัลกอริทึม ICBA มีการนำความรู้เดิมมาใช้ประโยชน์ในทั้งสองส่วนของอัลกอริทึม สำหรับส่วนแรกของอัลกอริทึม อัลกอริทึม ICBA สามารถลดเวลาที่ใช้ในการสแกน training data เพื่อหาค่าสนับสนุน และลดจำนวนของการเชื่อมความสัมพันธ์ระหว่าง ruleitem ซึ่งในการเชื่อมความสัมพันธ์เป็นขั้นตอนที่ใช้เวลาค่อนข้างนานและในส่วนที่สองของอัลกอริทึมพบว่า มีการลดจำนวนครั้งในการสแกน training data เพื่อตรวจสอบการรองรับข้อมูลของกฎความสัมพันธ์ด้วยเช่นกัน

บทที่ 5
สรุปผลการวิจัย

การจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์ (Associative Classification) เป็นเทคนิคของการจำแนกประเภทข้อมูลเทคนิคหนึ่งที่ได้รับความนิยม เนื่องจากเป็นเทคนิคที่มีค่าความแม่นยำสูงกว่าเทคนิคการจำแนกประเภทข้อมูลแบบทั่วไป เทคนิคการจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์ เป็นเทคนิคที่มีการนำเอาสองเทคนิคสำคัญในการทำเหมืองข้อมูลมาประยุกต์ใช้งานร่วมกัน คือ การจำแนกประเภทข้อมูล (Data Classification) และการค้นหากฎความสัมพันธ์ (Association Rule Discovery)

การจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์จะมีขั้นตอนการทำงานแบ่งออกเป็นสองส่วนหลักๆด้วยกันคือ ส่วนแรกเป็นส่วนของการสร้างเซตของกฎความสัมพันธ์จากข้อมูลที่ถูกแบ่งไว้เพื่อเป็นข้อมูลสอนระบบ (training data) และการสร้างกฎความสัมพันธ์มีพื้นฐานอยู่บนอัลกอริทึม อะพริออริ โดยจะสนใจเฉพาะกฎความสัมพันธ์ที่ทางขวามือเป็นค่าของคลาสเท่านั้น หรือที่เรียกว่า กฎความสัมพันธ์แบบมีคลาส (Class Association Rule : CAR) เมื่อได้เซตของกฎความสัมพันธ์แบบมีคลาสแล้ว ในส่วนที่สองจะเป็นการนำเซตของกฎความสัมพันธ์มาสร้างเป็นแบบจำลองสำหรับใช้ทำนายข้อมูลที่ไม่ทราบค่าต่อไป แต่ก่อนที่จะนำแบบจำลองไปใช้งาน จะต้องมีการตรวจสอบประสิทธิภาพของแบบจำลองก่อน โดยนำแบบจำลองไปทดสอบทำนายค่าให้กับข้อมูลที่ถูกแบ่งไว้สำหรับทดสอบแบบจำลอง (testing data) จากงานวิจัยในอดีตพบว่า เทคนิคการจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์มีความแม่นยำสูง แม้ว่าข้อมูลจะมีความหลากหลายของคลาสก็ตาม

เมื่อมีการเพิ่มขึ้นของฐานข้อมูลทำให้ข้อมูลสอนระบบเปลี่ยนแปลงไป ซึ่งอาจจะส่งผลต่อกฎความสัมพันธ์ที่ได้ เนื่องจากความสัมพันธ์ระหว่างข้อมูลได้เปลี่ยนแปลงไป ดังนั้นเพื่อเป็นการปรับแบบจำลองให้ถูกต้องและใหม่ล่าสุดสำหรับฐานข้อมูลนั้นๆ จึงจำเป็นที่จะต้องทำการค้นหากฎความสัมพันธ์ขึ้นมาใหม่ โดยการทำวิธีการเดิมใหม่ตั้งแต่เริ่มต้น แต่เนื่องจากการทำเหมืองข้อมูลเป็นการทำงานกับฐานข้อมูลขนาดใหญ่ ดังนั้นเมื่อต้องเริ่มการทำงานใหม่ตั้งแต่แรกโดยไม่ได้สนใจว่าจะมีการเพิ่มขยายมากหรือน้อยก็ตาม ด้วยเหตุผลนี้เองทำให้เสียเวลาค่อนข้างมาก จึงได้มีการพัฒนาอัลกอริทึม FUP ขึ้นมา เพื่อนำมาใช้สำหรับการค้นหาความสัมพันธ์ในกรณีที่มีการเพิ่มขยายของข้อมูล

งานวิจัยการจำแนกประเภทข้อมูลแบบเพิ่มขยายโดยใช้กฎความสัมพันธ์ได้ทำการเสนออัลกอริทึม ICBA (Incremental Classification Based on Association Rule Algorithm)ซึ่งเป็นการนำเอาแนวคิดของอัลกอริทึม FUP มาประยุกต์เพื่อลดเวลาในการทำงานของอัลกอริทึม CBA ซึ่ง

เป็นอัลกอริทึมต้นแบบของการจำแนกประเภทข้อมูลโดยใช้กฎความสัมพันธ์เมื่อมีการเพิ่มขยายของ training dataset

อัลกอริทึม ICBA แบ่งการทำงานออกเป็น 2 ส่วนหลักๆคือ ส่วนของการสร้างกฎความสัมพันธ์เรียกว่า Incremental Classification Based on Association Rule Algorithm : Rule generator (ICBA-RG) และส่วนของการสร้างแบบจำลองเพื่อจำแนกประเภทข้อมูลเรียกว่า Incremental Classification Based on Association Rule Algorithm : Classifier Building (ICBA-CB) ในส่วนของการสร้างกฎความสัมพันธ์ เป็นส่วนของการสร้างกฎความสัมพันธ์แบบมีคลาส อัลกอริทึมจะทำการค้นหา large ruleitem ทั้งหมดที่เกิดจาก increment training dataset จากนั้นพิจารณา large ruleitem ที่ได้ว่าเคยเป็น large ruleitem ของ original training dataset หรือไม่ หาก large ruleitem ที่กำลังพิจารณาอยู่เคยเป็น large ruleitem ของ original training dataset มาก่อน อัลกอริทึมจะทำการสแกนเฉพาะในส่วนของ increment training dataset เพื่อปรับค่าสนับสนุนของ large ruleitem นั้น แต่หาก large ruleitem ที่พิจารณาเป็น large ruleitem ที่เกิดขึ้นมาใหม่ อัลกอริทึมจะทำการสแกนทั้ง updated training dataset เพื่อปรับค่าสนับสนุน การเลือกสแกนเฉพาะ training dataset ที่จำเป็นมีส่วนช่วยลดเวลาในการทำงานของอัลกอริทึมได้ และในส่วนของการสร้างแบบจำลองเพื่อการจำแนกประเภทข้อมูลอัลกอริทึมจะทำการตรวจสอบกฎความสัมพันธ์หลังที่ทำการเรียงศักย์เรียบร้อยแล้ว ว่ากฎความสัมพันธ์ที่พิจารณาอยู่นั้น เป็นกฎความสัมพันธ์ที่เคยเกิดขึ้นใน original training dataset หรือไม่ หากเป็นกฎความสัมพันธ์ที่เคยเกิดขึ้นมาก่อนแล้ว อัลกอริทึมจะเลือกสแกนเฉพาะในส่วน increment training dataset เพื่อตรวจสอบการรองรับข้อมูล แต่หากเป็นกฎความสัมพันธ์ที่ไม่เคยเกิดขึ้นมาก่อน อัลกอริทึมก็จะสแกน updated training dataset ทั้งหมดเพื่อตรวจสอบการรองรับข้อมูลของกฎความสัมพันธ์ จากในขั้นตอนนี้จะเห็นได้ว่า การเลือกสแกน training dataset ในส่วนที่จำเป็นสามารถช่วยลดเวลาในการทำงานได้

จากผลการทดลองทั้งหมด 3 ชุดซึ่งเป็นข้อมูลจาก UCI Machining Learning โดยทดลองทั้งหมด 9 กรณีคือ แต่ละชุดข้อมูลจะมีค่าสนับสนุนขั้นต่ำ และค่าความเชื่อมั่นขั้นต่ำที่แตกต่างกัน 3 ค่า พบว่าอัลกอริทึม ICBA สามารถลดเวลาในการทำงานเมื่อมีการเพิ่มขยายของ training data ได้จริงตามสมมติฐาน เมื่อเปรียบเทียบกับอัลกอริทึม CBA เนื่องจากอัลกอริทึม ICBA ได้นำความรู้เดิมที่มีอยู่มาใช้งานให้เกิดประโยชน์ ด้วยการนำแนวคิดของอัลกอริทึม FUP มาใช้ทำให้อัลกอริทึม ICAB สามารถลดจำนวนครั้งในการสแกน training datasetได้เป็นอย่างดี

บรรณานุกรม

[1] Agrawal, R., Imielinski, T., and Swami, A. "Mining association rules between sets of items in large database." **Proceeding of the 1993 ACM SIGMOD Conference on Management of data discovery and data mining.** Washington DC, May 1993. pp 207-216.

[2] Agrawal, R., and Srikant, R., "Fast algorithm for mining association rules." Proceedings of **20th VLDB Conference Santiago**. Chile, September 1994. pp 487-499.

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

[4] Bing Liu., Wynne Hsu., Yimming Ma. "Integrating Classification and Association Rule Mining." **Proceeding of the Fourth International Conference on Knowledge Discovery and Data Mining**, NY, August 1998. pp 80-86.

[5] Cheung, D.W., Han, J., Ng, V.T. and Wong, C.Y., "Maintenance of Discovered Association Rule in Large Database: An incremental updating technique" **In 12 th IEEE International Conference on Data Engineering,** New Orleans, February 1996. pp 106-114

[6] Cheung, D.W., Lee, S.D. and Kao, B. "A General Incremental Technique for Maintaining Discovered Association Rules" **Proceeding of the 5th International Conference on Database System for Advanced Applications**, Melbourne, April 1997. pp 185-194.

[7] Thabtah, F. "Challenges and Interesting Research Directions in Associative Classification." **Proceeding of the 6th IEEE International Conference on Data Mining Workshops,** Hong Kong, December 2006. pp 785-792.

[8] Han, J. and Kamber, M. 2006 **Data Mining: Concepts and Techniques.** 2nd ed. San Francisco : Morgan Kaufmann Publishers.

[9] **UCI Machining Learning Respository**. [Online]. Available : http://archive.ics.uci.edu/ml/.

ภาคผนวก

บทความวิจัยที่ได้รับการตีพิมพ์และเผยแพร่

Administered by: UCMSS
13223-1 Black Mountain Road,
Suite 135
San Diego, CA 92129-2658

WORLDCOMP'11
July 18-21, 2011
Monte Carlo Resort
Las Vegas Nevada, USA

Official Contact: Prof. H. R. Arabnia
Phone: (706) 542-3480
Fax: (706) 542-2966
Email: hra@cs.uga.edu
www.world-academy-of-science.org

April 22, 2011

Ms. Sararak Tanarat
Kamol Apartment Room 422
556/1 soi Rimsuan Village , Ladkrabang,
Ladkrabang, Bangkok 10520
Thailand

Dear Ms. Sararak Tanarat,

This notification letter is to inform you that the paper entitled "***Incremental Classification Based on Association Rules Algorithm (ICBA)***" which was submitted to The 2011 International Conference on Data Mining (DMIN'11) has been accepted for publication.

You have been invited by the members of the Organizing Committee of The 2011 World Congress in Computer Science, Computer Engineering, and Applied Computing (WORLDCOMP'11) to attend the annual summer conference series to be held in Las Vegas, Nevada, USA (July 18-21, 2011). Universal Conference Management Systems & Support (UCMSS), who is managing the operation of these conferences, wishes to congratulate you on behalf of the Conference Committee and extend this invitation to you for presentation of the above paper at this event. Enclosed, please find a letter addressed to the United States Consulate General for US VISA purposes.

We hope that you will take advantage of this exceptional opportunity to join us. We are confident that you will enjoy the scientific program that is being offered at this year's event.

Congratulations once again, and thank you for your contribution to the conference. We look forward to welcoming you at the conference in Las Vegas.

Sincerely,

Universal Conference Management Systems & Support
Official Seal

Kaveh D. Arbtan
UCMSS
San Diego, California
U.S.A.

Incremental Classification Based on Association Rules Algorithm (ICBA)

S. Tanarat[1], and W. Kreesuradej[2]
[1]Information Technology, King Mongkut's Institiute of Technology Ladkrabang, Bangkok,Thailand
[2]Information Technology, King Mongkut's Institiute of Technology Ladkrabang, Bangkok,Thailand

Abstract - *In this study, an incremental updating technique is applied to associative classification for constructing classification system when a new training dataset is appended to an old training dataset. The proposed algorithm, called Incremental Classification Based on Association Rules (ICBA). ICBA has 2 phases which are rule generator phase (ICBA-RG) and classifier building phase (ICBA-CB). In order to reduce the execution time, we applied the concept of Fast Update algorithm (FUP) algorithm to both phases of our algorithm. The experiment results show that the proposed algorithm has execution time better than CBA algorithm.*

Keywords: Incremental Associative Classification, Associative Classification, Class Association Rule.

1 Introduction

Associative Classification [3] is a framework that integrates classification and association rule mining [1,2]. The goal of associative classification is to build a model that uses association rules for classification to predict future data objects for which the class label is unknown.

Model Construction generally consists of two major phases: rule generation and classifier building. Firstly, the rule generation is discovering the set of class association rules (CARs) which satisfy the user specified constraints denoted respectively by minimum support and minimum confidence thresholds. Secondly, a classifier is built by choosing a subset of the generated class association rules (CARs). Many studies have shown that Associative Classification is often more accurate than do traditional classification techniques.

When a new training dataset is appended to an old training dataset, the classifier that uses association rules may need to be changed in order to reflect any change in the new training dataset. As a brute force technique to deal with this situation, both old training dataset and a new training dataset are merged into an updated training dataset. Then, the model construction process starts building a classifier based on the updated training dataset. This brute force technique is time consuming and inefficiency.

Therefore, a new algorithm, called Incremental Classification Based on Association Rules (ICBA) algorithm, is proposed. The objective of this algorithm is to solve these problems more efficiently. As a result, the proposed algorithm has faster execution time faster than that of the previous algorithm.

2 Related Work

2.1 Associative Classification (AC)

Associative Classification is considered as a new approach for classification. The framework of associative classification is integration of classification and association rule mining. The first associative classification algorithm is called Classification Based on Association Rules (CBA) [3].The algorithm has two major phases:

- CBA – Rule Generator (CBA-RG)
- CBA – Classifier Building (CBA-CB).

CBA-RG algorithm generates a complete set of class association rules (CARs) that satisfy the minimum support and minimum confidence thresholds. To generate the set of class association, CBA-RG algorithm finds all large ruleitem by making multiple pass over data similar to Apiori algorithm. Ruleitems are large ruleitem if their supports are greater than or equal to minimum support. For all ruleitems with the same condset, the ruleitem that have the highest confidence is chosen as possible rule. The result of this step is the set of CARs.

CBA-CB algorithm sorts the set of CARs according to the precedence relation ($\succ$). The rule ranking is defined as follows:

Given two rule r_i and r_j; $r_i \succ r_j$ (r_i has higher precedence over r_j), if one of the following holds good:

1. The confidence of r_i is greater than that of r_j
2. Their confidence are the same but support of r_i is greater than that of r_j
3. Both confidences and supports of r_i and r_j are the same, but r_i is generated before r_j

After rule ranking, each training instance is covered by a rule having the highest precedence among the rules that can

over the case. The rule that do not cover any training instances are removed. Then, training instances that do not fall into any of the observed classed are added to a default class. Finally, rules that do not improve the accuracy of the classifier are discarded. The remaining rules and the default class of the last rule are formed as associative classifier.

2.2 An Incremental Updating Technique

When new transactions are added to the database shown in figure 1, association rules may be changed. For dynamic databases, several incremental updating techniques have been developed for mining association rule. An Incremental Updating Technique [5,6] is proposed for dynamic database which new transactions are appended.

The concept of incremental updating technique is to reuse large itemset of previous mining to obtain the update large itemset of an incremental database. Fast Update algorithm (FUP) was first introduced in [4]. The algorithm handles database with transactions insertion only. An efficient algorithm FUP is presented for computing the large itemset in the updated database. It is shown that the information from the old large itemset can be reused.

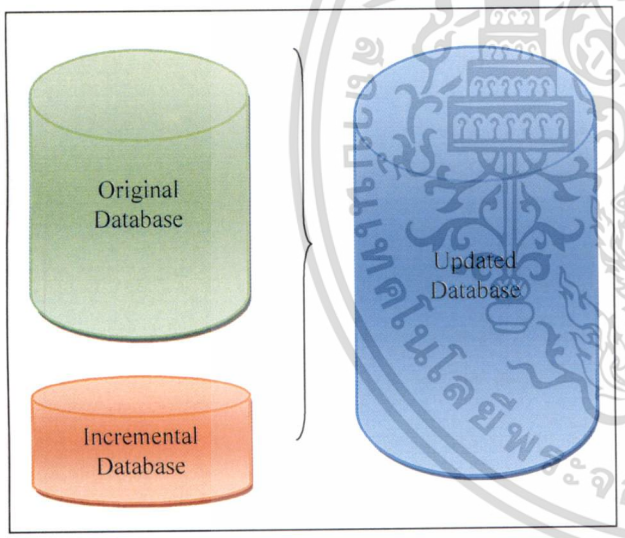


Fig. 1. Incremental Database

3 Incremental Classification Based on Association Rule Algorithm

When a new training dataset is appended to an old training dataset, an associative classifier may need to be changed in order to reflect any change in the new training dataset. However, when the training dataset changed, the existed Associative Classification algorithm always scans the changed training dataset in order to reflect the changes done. So far, there are rarely researches on the incremental learning of associative classification but there had been some studies on incremental association rule discovery algorithms that we can use their ideas for reference to solve the incremental associative classification problem. We propose an efficient incremental associative classification algorithm, a new algorithm is proposed to update an associative classifier when a new training dataset is appended to an old training dataset. The algorithm called Incremental Classification Based on Association Rules (ICBA) algorithm [8], is based on the concept of FUP algorithm to solve this problem.

The algorithm is divided into two parts. As the first part, ICBA-RG algorithm shown in figure 2 discovers the set of class association rule (CARs') in updated training datasets. Then, the second part is building a classifier for an updated training datasets. The algorithm for the second part shown in figure 3 is called ICBA-CB algorithm.

According to Figure 2, the steps of rule generation part are outlined as follows. An incremental training dataset (t) is scanned to determine large 1-ruleitem of an updated training dataset (UT) shown at line 1-16. If a candidate 1-ruleitem is a member of previous large 1-ruleitem, its support is updated. On the other hand, if a candidate 1-ruleitem is not a member of old large 1-ruleitemset, our algorithm checks the support of the ruleitem in an incremental training datasets (t). If the support of the ruleitem in an incremental training datasets is equal or above minimum support of incremental training datasets i.e., support $\geq (s \times d)$ when s is minimum support and d is size of incremental training dataset, then the algorithm scans original training datasets (T) to update support count of ruleitem ($X.support_{UT}$). In this paper, we called ruleitem which its support is greater than minimum support "winner" and ruleitem which its support is lower than minimum support "loser". As shown at line 17-33, the large k-ruleitems of updated training datasets are determined when k is greater than or equal to 2. Candidate k-ruleitems are generated by applying candidateGen function shown in line 18, this function is joining step similar to Apriori algorithm (see this function details in [2]). At k-th iteration, loser in L_k will be filtered out in a scan of t. The filtering is done by two steps. Firstly , a large k- ruleitems in L_k containing the ruleitem that cannot be the winner in the k-th iteration will be filtered out by ruleSubset function shown in line 19.

And the second , ICBA-RG filtered out loser in L_k without checking it against t. The set of losers $Y = L_k - L_{k-1}$ have been identified in line 21. Therefore, any sets of $X \in L_k$, which have subset Y such that $Y \in L_k - L_{k-1}$, cannot be large ruleitem and are filtered out from L_k. Then, if a ruleitem is a member of L_k and its support is equal or above $s \times (D+d)$ it becomes large k-itemset of update training datasets (L'_k). On the other hand, if a ruleitem is a member of C'_k and its support less than $s \times d$, the item will be removed from candidate k-ruleitem. If the support of ruleitem is equal or above $(s \times (D+d)$, the ruleitem is inserted into L'_k which will be generated to class association rules (CARs) by genRule function[3.] at line 14 and 32. Finally, pruneRule function shown at line 15 and 33 is prune CARs by minimum confidence same as Apriori

algorithm. All rules in CARs' which have their confidence less than minimum confidence ICBA-RG will be filter out.

```
Input : UT = The updated training datasets
        T  = The original training datasets with the total number of
             transactions, equal to D.
        t  = The incremental training datasets with the total of
             number transaction equal to d.
        L'k = The set of large-ruleitems in UT
        W  = L1 (large 1-ruleitem of T)
        s  = minimum support

Output : CARs' = The set of class association rules of updated
                 training datasets.

1.  for all X ∈ t do
2.    if X ∈ W do
3.      scan t to update X.support_UT then
4.        if X.support_UT ≥ s × (D + d) do
5.          if X.support_UT ≥ s × (D + d) do
6.            insert X at the end of L'1
7.    else
8.      if X ∉ W do
9.        for all X ∉ W do
10.         if X.support_UT ≥ (s × d) do
11.           scan UT to update X.support_UT then
12.             if X.support_UT ≥ s × (D + d) do
13.               insert X at the end of L'1
14. CARs1 = genRules(L'1)
15. prCARs1' = pruneRules(CARs1)
16. end
17. for (k=2; Lk-1 ≠∅; k++) do
18.   C'k = candidateGen (L'k-1) - Lk
19.   C's = ruleSubset (C'k)
20.       for all k-1 ruleitem in C's do
21.         Y = Lk-1 - L'k-1 do
22.           if Y ⊆ X then W = W-{X}
23.   for all X ∈ C's do
24.     if X.support_UT ≥ s × (T + t) do
25.       scan UT to update X.support_UT then
26.         insert X at the end of L'k
27.   for X ∈ W do
28.     scan UT to update X.support_UT then
29.       if X.support_UD ≥ s × (T + t) do
30.         insert X at the end of L'k
31.   end
32. CARs'k = genRules(L'k)
33. prCARs'k = pruneRules(CARs'k)
34. end
```

Fig. 2 ICBA-RG algorithm

For the last phase, ICBA-CB algorithm shown in figure 3 builds a classifier using CARs'. In this phase, in order to reduce the execution time we try to scan transactions of training datasets as less as we can.

ICBA-CB has three steps to build the classifier. The first step which is at line 1 is sorting the set of CARs' according to the relation "≻".

Then, the second step at line 2-24 is selecting rules for classifier following the sorted sequence. For each rule which is member of CARs, we go through t to find those cases covered by the rule. If selected rules are not member of CARs, our algorithm goes through UT to find those covered case instead. We mark selected rules if they correctly classify a case (line 6 and line 17). If selected rules can correctly classify at least one case, they may be our potential rule in our classifier. The rules that do not cover any case are removed and the cases that do not fall into any of the observed classes are added to a default class (in case we stop selecting more rule for our classifier (C')). The algorithm computes and records the total number of error made by classifier and default class. When there is no rule of training case left, the rule selection process is completed.

```
Input : CARs = Set of class association rules of original training
               datasets
        R'   = Class association rule is CARs'
        UT   = Updated training datasets
        T    = Original training datasets
        t    = Incremental training datasets
Output : C'  = Classifier

1.  R' = sort (R')
2.  for each rule r' ∈ R' in sequence do
3.    if r' ∈ CARs then
4.      for each case X in t do
5.        if X satisfies the condition of r' do
6.          store X.id in temp and mark r' if it
            correctly classifies t ;
7.      if r' is marked then
8.      insert r' at the end of C' (our classifier)
9.        delete all the ruleitem with the id in temp from db
10.       selecting a default class for the current C'
11.       compute the total number of errors of C'
12.   end
13.   else
14.     if r' ∉ CARs then
15.       for each case X in UT do
16.         if X satisfies the condition of r' do
17.           store X.id in temp and mark r' if it
              correctly classifies X;
18.     if r' is marked then
19.     insert r' at the end of C' (our classifier)
20.     delete all the ruleitem with the id in temp from UT
21.     selecting a default class for the current C'
22.     compute the total number of errors of C'
23.   end
24. end
25. Find the first rule p' in C' with the lowest total
    number of errors and drop all the rule after p' in C'
26. Add the default class associated with p' to end of C'
    and return C'
```

Fig. 3 ICBA-CB algorithm

The third step at line 25-26 is discarding those rules in the classifier that do not improve the accuracy. The cutoff rule is the first rule at which there is the least number of errors recorded on UT. The remaining rules and the default class of the last rule in the classifier form our classifier.

4 Experiments

The comprehensive experiment is conducted to evaluate the efficiency of the proposed algorithm. We compare ICBA performance with CBA algorithm and we use 3 datasets (2

different minimum support thresholds and 2 different minimum confidence thresholds) from UCI Machine Learning Repository *[http://archive.ics.uci.edu/ml]*. The execution time show in figure 4, 5 and 6.

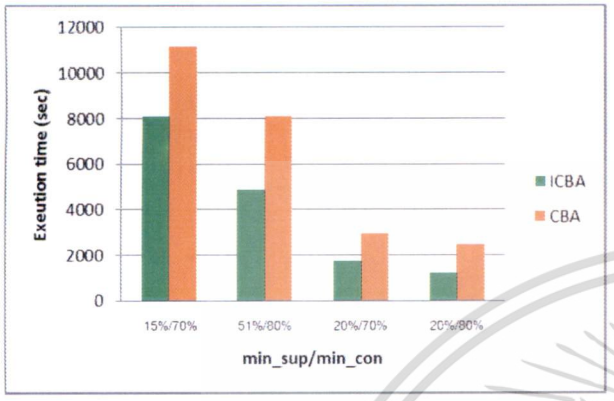


Fig. 4 Execution time of Adult dataset

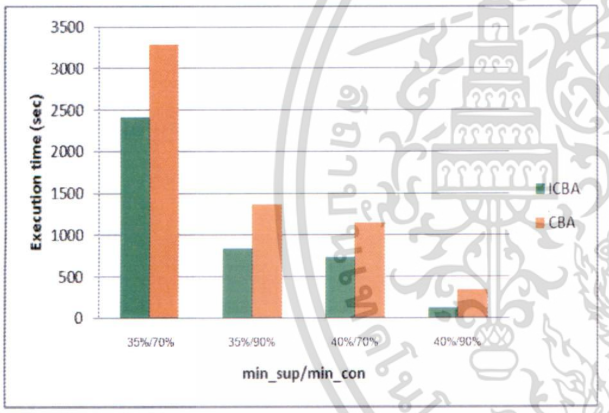


Fig. 5 Execution time of Mushroom dataset

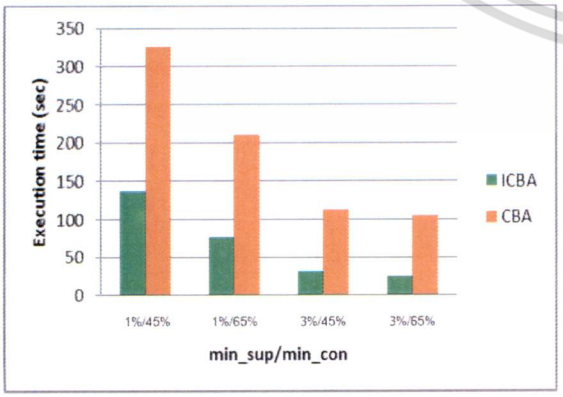


Fig. 6 Execution time of Nursery dataset

5 Conclusion

In this study, we propose an improved new classification based on association rules algorithm called Incremental classification based on association rules (ICBA). The experiment results show that our algorithm is more efficient than CBA algorithm. In the future, further researches and experiments on the proposed algorithm will be presented.

6 Reference

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

[2] R. Agrawal, and R. Srikant, "Fast Algorithm for Mining Association Rules," Proceedings of the International Conference on Very Large Database, Santiago, Chile, 1994, pp. 487-499.

[3] Bing Liu., Wynne Hsu., Yimming Ma. "Integrating Classification and Association Rule Mining", In Proc. Of the Fourth International Conference on Knowledge Discovery and Data Mining, New York, NY, pp.80-86, 1998

[4] D. Cheung, J. Han, V. Ng, and C. Y. Wong. "Maintenance of Discovered Association Rules in Lange Database: An Incremental Updating Technique." Proceedings of the 12th IEEE International Conference on Data Engineering, 1996, pp. 106-114.

[5] D. Cheung, S.D. Lee, and B. Kao. "A General Incremental Technique for Maintaining Discovered Association Rules," Proceedings of the 5th International Conference on Database System for Advanced Applications, Melbourne, Australia, 1997, pp. 185-194.

[6] F. Thabtah, "Challenges and Interesting Research Directions in Associative Classification," Proceeding of the Sixth IEEE International Conference on Data Mining Workshops, 2006, pp.785-792.

Notification of Acceptance of 3rd IEEE ICIME 2011

第三届 IEEE 信息管理与工程国际会议

May 21 - 22, 2011, Zhengzhou, China

www.icime.org

Dear Sararak Tanarat and Worapoj Kreesuradej.

Paper ID: I460

Paper Title: Incremental Updated Association Rules Based Classifier For Changing Training Dataset

Congratulations! The review processes for 2011 3rd IEEE International Conference on Information management and engineering (IEEE ICIME 2011) has been completed. The conference received submissions from 35 different countries and regions, which were reviewed by international experts, and 500 papers have been selected for presentation and publication. Based on the recommendations of the reviewers and the Technical Program Committees, we are pleased to inform you that your paper identified above has been accepted for publication and oral presentation. You are cordially invited to present the paper orally at ICIME 2011 to be held on 21-22, May 2011, Zhengzhou, China.

ICIME 2011 is co-sponsored by IEEE and Zhengzhou Institute of Aeronautical Industry Management, co-sponsored by Henan University of Technology, University of Electronics Science and Technology of China, Sichuan Institute of Electronics

(Important) So in order to register the conference and have your paper included in the proceeding successfully, you must finish following SIX steps.

1. Revise your paper according to the Review Comments in the attachment carefully.

2. Format your paper according to the Template carefully.
http://www.icime.org/IEEE.doc (DOC Format)

3. Download and complete the Registration Form.
http://www.icime.org/ICIME.Registration.doc (English)
http://www.icime.org/ICIME.CNRegistration.doc (中国大陆作者注册表)

4. Finish the payment of Registration fee at the Bank. (The bank transfer information can be found in the Registration form)

http://www.icime.org/ICIME.Registration.doc (English)
http://www.icime.org/ICIME.CNRegistration.doc (中国大陆作者注册表)

5. Finish the IEEE Copyright Form
http://www.icime.org/IEEECopyrightForm.doc

6. Send your final papers (both .doc and .pdf format), filled registration form (.doc format), copyright form (.jpg format) and the scanned payment (in jpg format) to us at icime@vip.163.com (Before April 1, 2011)

ICIME 2011 will check the format of all the registered papers first, so the authors don't need to upload the paper to the IEEE. After the registration, we will send all qualified papers to the IEEE for publishing directly.

The ICIME 2011 conference proceeding will proudly published by IEEE Press, which will be included in IEEE Xplore, and indexed by Ei Compendex, INSPEC and Thomson ISI (ISTP).

Maybe some unforeseeable events could prevent a few authors not to attend the event to present their papers, so if you and your co-author(s) could not attend ICIME 2011 to present your paper for some reasons, please inform us. And we will send you, the official receipt of registration fee, proceedings, and/or other materials after ICIME 2011 free of charge.

Please strictly adhere to the format specified in the conference template while preparing your final paper. If you have any problem in preparing the final paper, please feel free to contact us via icime@vip.163.com. For the most updated information on the conference, please check the conference website www.icime.org. The Conference Program will be available at the website in early May, 2011.

Finally, we would like to further extend our congratulations to you and we are looking forward to meeting you in Zhengzhou, China!

Yours sincerely,

INTERNATIONAL ASSOCIATION OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY
IACSIT
Regn. No
53122E55K

ICIME 2011 Organizing Committees
www.icime.org
Zhengzhou, China

Incremental Updated Association Rules Based Classifier For Changing Training Dataset

Sararak Tanarat
Faculty of Information Technology
King Mongkut's Institute of technology Ladkrabang
Bangkok, 10520 Thailand
som_x@hotmail.com

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

stract— Recent studies in data mining have proposed a new ssification approach called Associative Classification which egrates both association rule mining and classification task. this study, an incremental updating technique is applied to ociative classification for constructing classification system order to reduce the execution time, when a new training aset is appended to an old training dataset. The proposed orithm, called Incremental Classification Based on sociation Rules (ICBA), is based on the concept of Fast date algorithm (FUP) algorithm. The experiment results ow that the proposed algorithm has execution time faster n CBA algorithm.

Keywords- Associative Classification; class association rules; remental association rules.

I. INTRODUCTION

Associative Classification [3] is a framework that egrates classification [7] and association rule mining [1, The goal of associative classification is to build a model it uses association rules for classification to predict future ta objects for which the class label is unknown.

Model Construction generally consists of two major ases: rule generation and classifier building. Firstly, the le generation is discovering the set of class association les (CARs) which satisfy the user specified constraints noted respectively by minimum support and minimum nfidence thresholds. Secondly, a classifier is built by oosing a subset of the generated class association rules 'ARs). Many studies have shown that Associative assification is often more accurate than traditional ssification techniques.

When a new training dataset is appended to an old ining dataset, the classifier that uses association rules may ed to be changed in order to reflect any change in the new ining dataset. As a brute force technique to deal with this uation, both old training dataset and a new training dataset merged into an updated training dataset. Then, the model nstruction process starts building a classifier based on the dated training dataset. This brute force technique is time nsuming and inefficiency.

Therefore, a new algorithm, called Incremental assification Based on Association Rules (ICBA) gorithm, is proposed. The objective of this algorithm is to lve these problems more efficiently. As a result, the proposed algorithm has execution time faster than that of the previous algorithm.

II. RELATED WORK

A. Associative Classification (AC)

Associative Classification is considered as a new approach for classification. The framework of associative classification is integration of classification and association rule mining. The first associative classification algorithm is called Classification Based on Association Rules (CBA) [3].The algorithm has two major phases:

- CBA – Rule Generator (CBA-RG)
- CBA – Classifier Building (CBA-CB).

CBA-RG algorithm shown in figure 1 generates a complete set of class association rules (CARs) that satisfy the minimum support and minimum confidence thresholds. To generate the set of class association, CBA-RG algorithm finds all frequent ruleitems by making multiple passes over data similar to Apriori algorithm. Ruleitems are a frequent ruleitem if their supports are greater than or equal to minimum support. For all ruleitems with the same condset, the ruleitem that has the highest confidence is chosen as possible rule. The result of this step is the set of CARs.

CBA-CB algorithm is shown in figure 2. The algorithm sorts the set of CARs according to the precedence relation ($\succ$). The rule ranking is defined as following:

Given two rule r_i and r_j, $r_i \succ r_j$ (r_i has higher precedence over r_j), if one of the following holds good:

1. The confidence of r_i is greater than that of r_j
2. Their confidence are the same but support of r_i is greater than that of r_j
3. Both the confidences and supports of r_i and r_j are the same, but r_i is generated before r_j

After rule ranking, each training instance is covered by a rule having the highest precedence among the rules that can cover the case. The rules that do not cover any training instances are removed. Then, training instances that do not fall into any of the observed classed are added to a default class. Finally, rules that do not improve the accuracy of the classifier are discarded. The remaining rules and the default class of the last rule are formed as associative classifier.

```
. F1 = {large 1-ruleitems};
2. CAR1 = genRules(F1);
3. prCAR1 = pruneRules(CAR1);
4. for (k = 2; Fk-1 ≠ Ø; k++) do
5.     Ck  = candidateGen (Fk-1);
6.     for each data case d∈D do
7.         Cd = ruleSubset(Ck, d);
8.         for each candidate c∈Cd do;
9.             c.condsupCount++;
10.                if d.class = c.class then c.rulesupCount++
11.            end
12.    end
13.    Fk = {c∈Ck | c.rulesupCount ≥ minsup};
14.    CARk = genRules(Fk);
15.    prCARk = pruneRules(CARk);
16. end
17. CARs = ∪k CARk ;
18. prCARs = ∪kprCARk ;
```

Figure 1. CBA-RG algorithm

```
1.  R = sort (R);
2.  for each rule r ϵ R in sequence do
3.     temp = Ø;
4.     for each case d∈D do
5.        if d satisfies the conditions of r then
6.           store d.id in temp and mark r if it correctly
             Classifies d;
7.     if r is marked then
8.        insert r at the end of C;
9.        delete all the cases with the ids in temp from D;
10        selecting a default class for the current C;
11.       compute the total number of errors of C;
12.    end
13. end
14. Find the first rule p in C with the lowest total
    number of errors and drop all the rules after p in C;
15. Add the default class associated with p to end of C,
    And return C (our classifier).
```

Figure 2. CBA-CB algorithm

An Incremental Updating Technique

An Incremental Updating Technique [5, 6] is proposed r dynamic database which new transactions are appended. hen new transactions are added to the database shown in ure 3, association rules maybe change. For this problem, st Update algorithm (FUP) [4] shown in figure 4 and ure 5 is the first algorithm that proposed to solve it. The orithm handles database with transactions insertion only. efficient algorithm FUP is presented for computing the ge itemsets in the updated database. It shows that the ormation from the old large itemsets can be reused. The st iteration of FUP algorithm shown in figure 4 is filters the losers and obtains the first set of winners from the ginal large1-itemsets. It removes size-one loser then erates size-one candidate sets and finds size one winner. r the second iteration and beyond, FUP algorithm moves other losers, prunes candidate sets and finds maining winners. The same algorithm is applied to the later rations until no large itemsets is found.

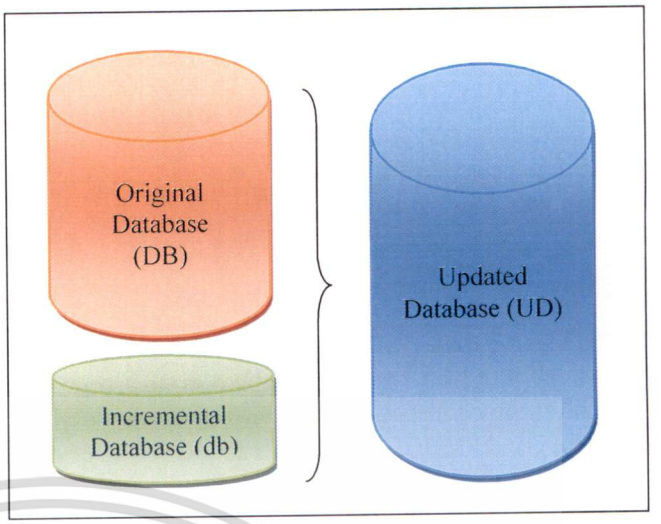


Figure 3. Updated Database

```
Input : DB = the original database
        Lk = the set of all large k-itemsets in DB, where
             k = 1,..., r;
        db = an incremental database
        s  = minimum support threshold
Output: L' = the set of all large itemsets in DB ∪ db.
The 1st iteration :

1. W = L1 ; C = Ø ; L'1 = Ø ; P = Ø;
   /* W: winners, C: candidate sets, L1': initialized,
   P: for optimization*/
2. for all T∈db do /*scan db*/
3.    for all 1-itemset X ⊆ T do{
4.       if X∈W then X.supportd++;
5.       else {
6.          if X ∉ C
7.          then {C = C ∪ {X}; X.supportd = 0;}
            /*init the support count and add X into C */
8.          X.supportd++;}
9. };
10. for all X∈W do /*put winners in to L1' */
11.    if X.supportUD ≥ s × (D + d)
12.       then L'1 = L'1 ∪ {X};
13. for all X∈C do /*prune candidate sets in C*/
14.    if X.supportd < s × d
15.       then {C = C−{X}; P = P ∪ {X};}
          /*P will be used for optimization*/
16. for all T∈DB do /* scan DB*/
17. for all 1-itemset X ⊆ T do {
18.    if X∈C then X.supportD ++;
19.    if X∈P then removes X from T;
       /* Transacsion T is reduced*/
20. };
21. for all X∈C do /*put the winners into L'1 */
22.    if X.supportUD ≥ s × (D + d)
23.       then L'1 = L'1 ∪ {X}
24. return L'1 /* end of the 1st iteration*/
```

Figure 4. FUP algorithm (1st iteration)

```
W = Lk ; L'k = Ø;
/* W: winners; L'k : initialized */
C = apriori-gen (L'k-1) - Lk ;
/* the size-k candidate sets*/
for all k-itemset X∈W do
  /*prune off loser in W*/
   for all (k-1) – itemset Y∈Lk-1 - L'k-1 do
      if Y ⊆ X then { W = W – {X}; break; }
for all T ∈db do {/* scan db*/}
  for all X ∈Subset (W,T) do X.supportd ++ ;
  /* Subset (W,T) returns all the sets in W contained in T*/
  for all X ∈Subset (C,T) do X.supportd ++ ;
  /* find support of all X∈C */
  Reduce_db (T);
   /* Some item in transactions in db can be removed*/
). }
1. for all X∈W do
  /* put the winners from W into L'k */
2.   if X.supportUD ≥ s × (D + d)
3.      then L'k = L'k ∪ {X};
4. for all X∈C do /* prune candidate sets in C */
5.   if X.supportd < s × d then C = C – {X};
6. for all T ∈DB do {/* scan DB*/}
7.   for all X ∈Subset (C,T) do X.supportD ++;
8.   Reduce DB(T); }
   /* Some transactions in DB can be removed*/
9. for all X ∈C do
  /* put the winners from C into L'k */
0.   if X.supportUD ≥ s × (D + d)
1.      then L'k = L'k ∪ {X};
2. return L'k . /* The end of the kth iteration*/
```

Figure 5. FUP algorithm (k^{th} iteration)

```
1.  C '1 = 1-ruleitem in db
2.  for each X∈C '1 do
3.      if X∈C1 do
4.        scan db to update X.supportUD then
5.            if X.supportUD ≥ s × (D + d) do
6.              insert X at the end of L'1
7.      else
8.          if X.supportd ≥ (s × d) do
9.            scan UD to update X.supportUD then
10.               if X.supportUD ≥ s × (D + d) do
11.                 insert X at the end of L'1
12. end
13. for (k=2; Lk-1 ≠Ø; k++) do
14.   C 'k = candidateGen (L'k-1) - Lk
15.   Lk = ruleSubset (Lk-1 , L'k-1)
16.     for each X ∈C 'k do
17.        if X ∈Ck do
18.          scan db to update X.supportUD then
19.              if X.supportUD ≥ s × (D + d) do
20.              insert X at the end of L'k
21.        else
22.              if X.supportd ≥ (s × d) do
23.                scan UD to update X.supportUD then
24.                    if X.supportUD ≥ s × (D + d) do
25.                      insert X at the end of L'k
26.     end
27. end
28. CARs' = pruneRule (L')
```

Figure 6. ICBA-RG algorithm

III. INCREMENTAL ASSOCIATIVE CLASSIFICATION (IAC)

When a new training dataset is appended to an old training dataset, an associative classifier may need to be changed in order to reflect any change in the new training dataset. Here, a new algorithm is proposed to update an associative classifier when a new training dataset is appended to an old training dataset. The algorithm, called Incremental Classification Based on Association Rules (ICBA) algorithm, is based on the concept of FUP algorithm to solve this problem. The algorithm is divided into two parts. As the first part, ICBA-RG algorithm shown in figure 6 discovers the set of class association rule (CAR's) in updated database. Then, the second part is builds a classifier for an updated database. The algorithm for the second part shown in figure 7 is called ICBA-CB algorithm.

```
1.  CARs' = sort (CARs')
2.  for each rule r' ∈ CARs' in sequence do
3.     if r '∈ CARs then
4.       for each case d in db do
5.          if d satisfies the condition of r' do
6.            store d.id in temp and mark r' if it
              correctly classifies d;
7.       if r' is marked then
8.       insert r' at the end of C ' (our classifier)
9.         delete all the cases with the id in temp from db
10.        selecting a default class for the current C '
11.        compute the total number of errors of C '
12.    end
13.  else
14.    if r' ∉ CARs then
15.       for each case d in UD do
16.         if d satisfies the condition of r' do
17.           store d.id in temp and mark r' if it
              correctly classifies d;
18.       if r' is marked then
19.       insert r' at the end of C ' (our classifier)
20.       delete all the cases with the id in temp from UD
21.       selecting a default class for the current C '
22.       compute the total number of errors of C '
23.   end
24. end
25. Find the first rule p ' in C ' with the lowest total
    number of errors and drop all the rule after p' in C '
26. Add the default class associated with p' to end of C '
    and return C '
```

Figure 7. ICBA-CB algorithm

According to Figure 6, the steps of rule generation part are outlined as follows. An incremental database (db) is scanned to determine large 1-ruleitem of an updated database (UD) shown at line 1-12. If a candidate 1-ruleitem is member of previous large 1-ruleitemsets, its support is updated. On the other hand, if a candidate 1-ruleitem is not member of old large 1-ruleitemsets, our algorithm checks the support of the ruleitem in an incremental database. If the support of the ruleitem in an incremental database is equal or above minimum support of incremental database, i.e., support ≥ (s×d), then the algorithm scans DB to update support count of ruleitem.

As shown at line 13-27, the large k-ruleitems of updated abase are determined when k is greater than or equal to 2. ıdidate k-ruleitemsets are generated by applying didateGen function shown in line 14. Similar to the ning step of Apriori algorithm, a ruleSubset function wn in line 15 removes joined ruleitemsets which is ıilar to previous large k-ruleitemsets, i.e. L_k. Then, if a eitem is member of L_k and its support is equal or above s× +d) it becomes large k-itemset of update database (L'_k). the other hand, if a ruleitem is member of C_2 and then its pport less than (s×d) then the item is removed from ıdidate k-ruleitem. Then, if the support of ruleitem is ıal or above (s× (D+d), the ruleitem is inserted into L'_2. ially, genRule function shown at line 28 generates CARs.

For the last phase, ICBA-CB algorithm shown in figure 7 lds a classifier using CARs. It has three steps to build the ssifier. The first step which is at line 1 is sorting the set of ıRs' according to the relation ">".

Then, the second step at line 2-24 is selecting rules for ssifier following the sorted sequence. For each rule which member of CARs, we go through db to find those cases vered by the rule. If selected rules are not member of ıRs, our algorithm goes through UD to find those covered se instead. We mark selected rules if they correctly classify case (line 6 and line 17). If selected rules can correctly ssify at least one case, they may be our potential rule in r classifier. The rules that do not cover any case are moved and the case that do not fall into any of the served classes are added to a default class which mean that we stop selecting more rule for our classifier (C) this class ll be the default class. The algorithm computes and cords the total number of error that made by classifier and fault class. When there is no rule of training case left, the e selection process is completed.

The third step which is at line 25-26 is discarding those es in the classifier that is do not improve the accuracy. e cutoff rule is that the first rule at which there is the least mber of errors recorded on UD. The remaining rules and e default class of the last rule in the classifier forms our ıssifier.

IV. EXPERIMENTS

The experiment is conducted an extensive performance ıdy to evaluate the accuracy an efficiency of the proposed gorithm. We compared ICBA performance with CBA gorithm and we used 2 datasets (with different minimum pport but same minimum confidence at 60%) from UCI achine Learning Repository TABLE I shows the average execution time for ICBA algorithm and CBA algorithm th different datasets.

TABLE I. AVERAGE OF EXECUTION TIME

Dataset	minimum support	Average of Execution time (sec.)	
		CBA algorithm	*ICBA algorithm*
Mushroon	25/60	346.633	81.583
Mushroom	30/60	203.664	48.600
Nursery	25/60	13.458	2.430
Nursery	30/60	10.431	1.401

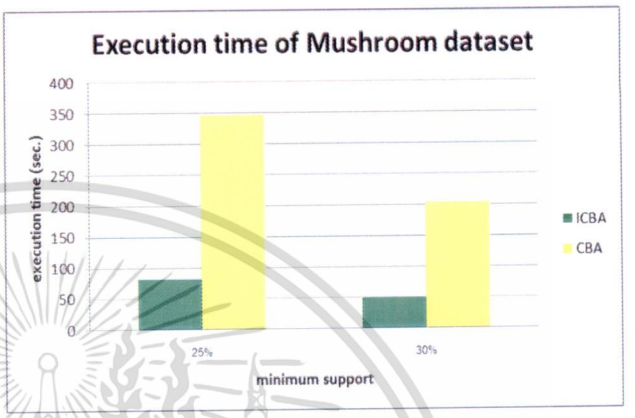


Figure 8. Execution time of Mushroom dataset

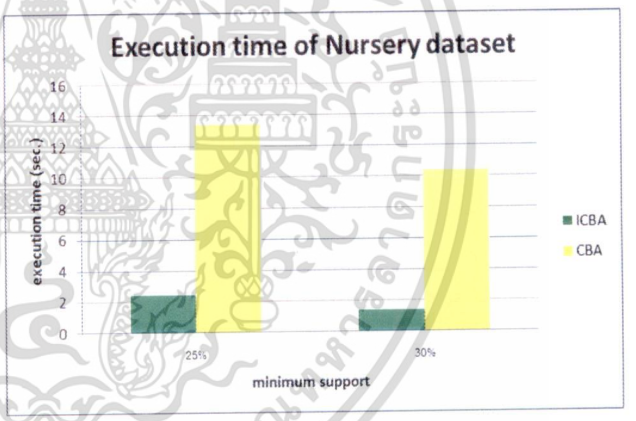


Figure 9. Execution time of Nursery dataset

V. CONCLUSION

In this study, we propose a new classification based on association rules algorithm called Incremental Classification Based on Association rules (ICBA). The experiment show that our algorithm is more efficient than CBA algorithm. In the future, further researches and experiments on the proposed algorithm will be presented.

VI. REFERENCES

R. Agrawal., T. Imielinski, and A. Swami, " Mining association rule between sets of items in large database", In Proceeding of the ACM SIGMOD Int'l Conf. on Management of Data, Washington, USA, May 1993, pp. 207-216.

R. Agrawal, and R. Srikant, "Fast Algorithm for Mining Association Rules," Proceedings of the International Conference on Very Large Database, Santiago, Chile, 1994, pp. 487-499.

Bing Liu., Wynne Hsu., Yimming Ma. "Integrating Classification and Association Rule Mining", In Proc. Of the Fourth International Conference on Knowledge Discovery and Data Mining, New York, NY, pp.80-86, 1998

D. Cheung, J. Han, V. Ng, and C. Y. Wong. "Maintenance of Discovered Association Rules in Lange Database: An Incremental Updating Technique," Proceedings of the 12th IEEE International Conference on Data Engineering, 1996, pp. 106-114.

D. Cheung, S.D. Lee, and B. Kao. "A General Incremental Technique for Maintaining Discovered Association Rules," Proceedings of the 5th International Conference on Database System for Advanced Applications, Melbourne, Australia, 1997, pp. 185-194.

F. Thabtah. "Challenges and Interesting Research Directions in Associative Classification," Proceeding of the Sixth IEEE International Conference on Data Mining Workshops, 2006, pp.785-792.

J. Han, and M. Kamber, "Data mining: Concepts and Techniques," Morgan Kaufmann Publishers. San Francisco, California, pp. 227-256, 2006.

ประวัติผู้เขียน

ชื่อ	นางสาวสรารักษ์ ธนารัตน์
วัน เดือน ปีเกิด	5 ตุลาคม 2525
ที่อยู่	หอพักธนารัตน์ 405 หมู่ที่ 5 ต.สุรนารี อ.เมือง จ.นครราชสีมา
ประวัติการศึกษา	2548 วิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า สาขาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีมหานคร
ประวัติการทำงาน	
พ.ศ. 2548-2549	ตำแหน่งผู้ช่วยอาจารย์สอนและวิจัย สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
พ.ศ. 2551- ปัจจุบัน	ตำแหน่งวิศวกรฝ่ายผลิตข้อมูลดาวเทียม ศูนย์ปฏิบัติการดาวเทียมภาคพื้นดิน สำนักงานพัฒนาเทคโนโลยีอวกาศและภูมิสารสนเทศ (องค์การมหาชน)