การพัฒนาขั้นตอนวิธีเพื่อจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือที่กะทัดรัด

นายไพชยนต์ คงไชย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2557

THE DEVELOPMENT OF DATA CLASSIFICATIONALGORITHM WITH COMPACT FUZZY ASSOCIATION RULES

PhaichayonKongchai

A Thesis Submitted in Partial Fulfillment of the Requirements for theDegree of Doctor of Philosophyin Computer Engineering

Suranaree University of Technology

Academic Year 2014

การพัฒนาขั้นตอนวิธีเพื่อจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือที่กะทัดรัด

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

_____

(รศ. ดร.นิตยา เกิดประสพ)
ประธานกรรมการ

_____

(รศ. ดร.กิตติศักดิ์ เกิดประสพ)
กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

_____

(ผศ. ดร.ศุภกฤษฎิ์นิวัฒนากูล)
กรรมการ

_____

(ผศ. ดร.สายสุนีย์จับโจร)
กรรมการ

_____

(ดร.ขุนเสก เสกขุนทด)
กรรมการ

_____

(ศ.ดร.ชูกิจ ลิมปิจำนงค์)(รศ.ร.อ.ดร.กนต์ธร ชำนิประศาสน์)
รองอธิการบดีฝ่ายวิชาการและนวัตกรรม คณบดีสำนักวิชาวิศวกรรมศาสตร์

ไพชยนต์ คงไชย : การพัฒนาขั้นตอนวิธีเพื่อจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือที่กะทัดรัด(THE DEVELOPMENT OF DATACLASSIFICATION ALGORITHM WITH COMPACT FUZZYASSOCIATION RULES)
อาจารย์ที่ปรึกษา: รองศาสตราจารย์ ดร.กิตติศักดิ์เกิดประสพ, 118หน้า

การทำเหมืองข้อมูลด้วยวิธีการจำแนกประเภทข้อมูล มีจุดประสงค์เพื่อนำโมเดลที่ได้จากกระบวนการเรียนรู้มาใช้ในการทำนายข้อมูลในอนาคต ซึ่งในปัจจุบันมีผู้วิจัยจำนวนมากให้ความสนใจที่จะพัฒนาประสิทธิภาพขั้นตอนวิธีการจำแนกประเภทข้อมูล เพื่อให้มีความแม่นยำในการจำแนกมากขึ้น และโมเดลที่ได้สามารถตีความหมายได้ง่าย แต่การที่จะเพิ่มประสิทธิภาพทั้งสองอย่างควบคู่กันไปนั้นยังไม่สามารถพัฒนาได้อย่างสมบูรณ์ ดังนั้นงานวิจัยนี้จึงได้เสนอการพัฒนาขั้นตอนวิธีเพื่อจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือที่กะทัดรัดเพื่อเพิ่มประสิทธิภาพการจำแนกประเภทข้อมูลในสองด้านคือ เพื่อให้โมเดลที่ได้รับมีความแม่นยำอยู่ในเกณฑ์ดีและสามารถตีความหมายได้ดีด้วย โดยได้นำเทคนิคการทำเหมืองข้อมูลเพื่อหากฎความสัมพันธ์มาผสมสานกับเทคนิคการจำแนกประเภทข้อมูลซึ่งเทคนิคการทำเหมืองข้อมูลด้วยการหากฎความสัมพันธ์นั้น มีจุดเด่นอยู่ที่จะค้นหาความสัมพันธ์จากข้อมูลทั้งหมดแล้วนำมาสร้างเป็นกฎความสัมพันธ์ จากนั้นงานวิจัยนี้จะใช้เทคนิคการจำแนกประเภทข้อมูลเพื่อมาคัดเลือกกฎความสัมพันธ์ เพื่อนำไปสร้างเป็นโมเดลที่ใช้ในการทำนายข้อมูลในอนาคตนอกจากนี้งานวิจัยนี้ยังใช้เทคนิคฟัซซีเซตมาควบคุมข้อมูลตัวเลขที่มีลักษณะเป็นค่าต่อเนื่อง เพื่อเพิ่มประสิทธิภาพในการจำแนกข้อมูลได้ดียิ่งขึ้น ในส่วนของผลการทดลองประสิทธิภาพได้ทำการทดลองกับอัลกอริทึม 3 ประเภท คือ อัลกอริทึมประเภทการจำแนกประเภทข้อมูล อัลกอริทึมประเภทการจำแนกข้อมูลด้วยกฎความสัมพันธ์ และอัลกอริทึมที่จำแนกข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือ เพื่อเปรียบเทียบค่าความถูกต้องในการจำแนกประเภทข้อมูลและจำนวนกฎที่ได้รับ

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ลายมือชื่อนักศึกษา_____

ปีการศึกษา 2557 ลายมือชื่ออาจารย์ที่ปรึกษา_____

PHAICHAYON KONGCHAI: THE DEVELOPMENT OF DATACLASSIFICATION ALGORITHMWITHCOMPACTFUZZYASSOCIATIONRULES. THESIS ADVISOR : ASSOC. PROF. KITTISAK KERDPRASOP, Ph.D., 118 PP.

DATA MINING/FUZZY ASSOCIATION RULE-BASEDCLASSIFIER

The objective of data classification is to find data model from the learning process to predict the future data. Currently, many researchers are interested inimproving the efficiency of data classification algorithm to obtain high accuracy and good interpretability of the model. But to obtain both criteria simultaneously is still unaccomplished bythe current methods.Therefore, this research proposes a method of data classification to obtain bothhighaccuracy and good interpretability of the model by the combination of association rule mining and data classification rule induction techniques.Association rule mining is good at finding relationshipsamong the whole data set and represents them as association rules. This research applies association rule mining for building a model to predict future data.This research also uses fuzzy set technique to control a continuous data to enhance efficiency of the data classification.To evaluate the performance of the proposed method, this researchcompares accuracy of the classification rules and the number of rules obtained from different kinds of data classification algorithms.These algorithms include the traditional data classification algorithms, the associative classification algorithms, and the fuzzy association rule-based classifier algorithms.

School ofComputer Engineering Student's Signature_____

Academic Year 2014 Advisor's Signature_____

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี ผู้วิจัยขอกราบขอบพระคุณ บุคคล และกลุ่มบุคคลต่างๆที่ได้กรุณาให้คำปรึกษา แนะนำ ช่วยเหลืออย่างดียิ่ง ทั้งในด้านวิชาการ และด้านการดำเนินงานวิจัย ดังต่อไปนี้

รองศาสตราจารย์ ดร.กิตติศักดิ์ เกิดประสพอาจารย์ที่ปรึกษาวิทยานิพนธ์ และรองศาสตราจารย์ ดร.นิตยา เกิดประสพ ที่ให้คำปรึกษาในการทำงานวิจัย การจัดรูปแบบ และช่วยตรวจทานความถูกต้องของวิทยานิพนธ์

คุณกัลญา พับโพธิ์ เลขานุการสาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ให้ความช่วยเหลือในการประสานงานด้านเอกสารระหว่างศึกษา

คุณกีระชาติ สุขสุทธิ์ที่ช่วยตรวจทานความถูกต้องของวิทยานิพนธ์และนักศึกษาบัณฑิตสาขาวิชาวิศวกรรมคอมพิวเตอร์ทุกท่านที่ให้คำปรึกษา

นอกจากนี้ขอขอบคุณมหาวิทยาลัยเทคโนโลยีสุรนารีที่ช่วยสนับสนุนทุนการศึกษาทุนวิจัยและค่าใช้จ่ายค่าเช่าที่พัก

ท้ายที่สุดที่จะลืมไม่ได้ ขอกราบขอบพระคุณ บิดา มารดา ที่ให้กำเนิด อบรม เลี้ยงดูด้วยความรัก และส่งเสริมการศึกษาเป็นอย่างดีโดยตลอด ทำให้ผู้วิจัยมีความรู้ ความสามารถ มีจิตใจที่เข้มแข็ง รวมทั้งเป็นกำลังใจที่ยิ่งใหญ่แก่ผู้วิจัย จนทำให้ผู้วิจัยประสบความสำเร็จในชีวิตเรื่อยมา

ไพชยนต์ คงไชย

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญตาราง(ต่อ)

ตารางที่ **หน้า**

สารบัญรูป

รูปที่ **หน้า**

บทที่ 1
บทนำ

1.1ความสำคัญและที่มาของปัญหาการวิจัย

การจำแนกประเภทข้อมูล(Data Classification) เป็นการทำเหมืองข้อมูลประเภทหนึ่งที่ได้รับความนิยมเป็นอย่างมากในการนำไปแก้ปัญหาทางวิทยาศาสตร์และอุตสาหกรรมโดยการใช้โมเดล (Model) หรือกฎการจำแนกประเภท(Classification Rule) ในการทำนายข้อมูลในอนาคต ซึ่งอัลกอริทึมส่วนใหญ่จะใช้หลักการค้นหาแบบฮิวริสติก (Heuristic Search)อย่างเช่น อัลกอริทึม C4.5 (Quinlan, 1993)อัลกอริทึม FOIL(Quinlan et al., 1993)และอัลกอริทึม RIPPER (Cohen, 1995)เพราะสามารถสร้างโมเดลที่ผู้ใช้เข้าใจได้ง่ายและมีประสิทธิภาพในการประมวลผลอย่างรวดเร็วแต่ความแม่นยำในการจำแนกมีไม่มากและยังมีอัลกอริทึมอีกประเภทหนึ่งที่ใช้หลักการของกล่องดำในการทำนายประเภทข้อมูลโดยที่ผู้ใช้เพียงแค่นำข้อมูลเข้า ซึ่งผลลัพธ์ที่ได้รับคือโมเดลที่มีค่าความถูกต้องที่สูง แต่ผู้ใช้อาจจะไม่สามารถตีความกฎที่เป็นผลลัพธ์นั้นได้

ต่อมาได้มีผู้คิดค้นอัลกอริทึมที่ผสมผสานระหว่างเทคนิคการทำเหมืองข้อมูลเพื่อหากฎความสัมพันธ์ (Association Rule Mining) และเทคนิคการสร้างกฎการจำแนกประเภท โดยการนำหลักการสร้างกฎความสัมพันธ์จากทุกข้อมูลมาสร้างเป็นกฎความสัมพันธ์แล้วใช้หลักการเลือกเฉพาะข้อมูลบางส่วนมาสร้างเป็นกฎความสัมพันธ์แบบจำแนกด้วยเทคนิคการจำแนกประเภท แล้วเรียกเทคนิคนี้ว่า การจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์ (Associative Classification)ซึ่งอัลกอริทึมแรกที่ใช้เทคนิคดังกล่าวมีชื่อว่า CBA (Ma, 1998)ที่มีความสามารถในการจำแนกประเภทข้อมูลได้ดีกว่าอัลกอริทึม C4.5 และต่อมามีผู้วิจัยสนใจพัฒนาเกี่ยวกับแนวคิดนี้มากขึ้น แต่เนื่องจากการสร้างกฎความสัมพันธ์มีข้อจำกัดที่ข้อมูลจะต้องเป็นค่าแบบกลุ่ม (Categorical Data) ทำให้มีการนำแนวคิดของฟัซซีมาใช้ในการควบคุมปัญหาการจัดการกับข้อมูลที่มีลักษณะเป็นตัวเลขค่าต่อเนื่อง (Continuous Data) ด้วยการแปลงข้อมูลดังกล่าวให้อยู่ในรูปแบบของฟัซซีเซต แต่ปัญหาในการใช้เทคนิคดังกล่าวคือได้รับกฎที่มีปริมาณมากเกินไปและตีความหมายได้ยากทำให้ในปัจจุบันยังคงเป็นปัญหาที่นักวิจัยให้ความสนใจ

ทำให้มีผู้วิจัยได้สรุปเป้าหมายของการวิจัยในการจำแนกประเภทข้อมูลมักจะเน้นที่ประสิทธิภาพ โดยประสิทธิภาพในการจำแนกประเภทข้อมูลจะขึ้นอยู่กับ 2 ตัวแปรคือ ค่าความถูกต้อง (Accuracy) และการตีความจากโมเดล (Interpretability of the Model) ซึ่งงานวิจัยส่วนใหญ่มุ่ง

เน้นเพื่อพัฒนาอัลกอริทึมเพื่อเพิ่มประสิทธิภาพดังกล่าว โดยสามารถแบ่งจุดประสงค์ของการพัฒนาอัลกอริทึมออกเป็น 4 รูปแบบ (Pach et al., 2008) คือ

- รูปแบบที่ 1 การพัฒนาอัลกอริทึมให้มีความสามารถในการจำแนกประเภทข้อมูลให้อยู่ในเกณฑ์ที่สามารถยอมรับได้และโมเดลสามารถตีความหมายได้ดีมาก
- รูปแบบที่ 2 การพัฒนาอัลกอริทึมให้มีความสามารถในการจำแนกประเภทข้อมูลอยู่ในเกณฑ์ที่ดีและโมเดลสามารถตีความหมายได้ดี
- รูปแบบที่ 3 การพัฒนาอัลกอริทึมให้มีความสามารถในการจำแนกประเภทข้อมูลอยู่ในเกณฑ์ที่ดีมากและโมเดลสามารถตีความหมายได้ และ
- รูปแบบที่ 4 การพัฒนาอัลกอริทึมให้มีความสามารถในการจำแนกประเภทข้อมูลอยู่ในเกณฑ์ที่ดีมากที่สุดแต่โมเดลไม่สามารถตีความหมายได้

ดังนั้นงานวิจัยนี้จึงได้เสนอการพัฒนาขั้นตอนวิธีเพื่อจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือที่กะทัดรัด(Classification with Compact Fuzzy Association Rules: CCFAR)เพื่อให้ได้เป้าหมายตามรูปแบบที่ 2 คือการพัฒนาอัลกอริทึมให้มีความสามารถในการจำแนกประเภทข้อมูลอยู่ในเกณฑ์ที่ดีและโมเดลสามารถตีความหมายได้ดีหรือมีขนาดและจำนวนที่กะทัดรัด(กฎที่ได้มีจำนวนน้อยและขนาดเล็ก) โดยการนำเทคนิคการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์ผสมผสานกับเทคนิคฟัซซีเซต และทำการเลือกกฎความสัมพันธ์เพื่อใช้ในการทำนายข้อมูลด้วยวิธีการใหม่ดังนี้

- วิธีการเลือกค่าเกณฑ์ที่เหมาะสมในการสร้างกฎความสัมพันธ์แบบคลุมเครือเพื่อใช้ในการจำแนกประเภทข้อมูล
- วิธีการเลือกกฎความสัมพันธ์แบบคลุมเครือเพื่อใช้ในการจำแนกประเภทข้อมูลให้มีขนาดเล็ก จำนวนไม่มาก และสามารถทำนายค่าที่เป็นไปได้ทั้งหมดของข้อมูล
- วิธีการเลือกกฎความสัมพันธ์แบบคลุมเครือเพื่อใช้ในการจำแนกประเภทข้อมูลโดยที่ผู้ใช้ไม่ต้องกำหนดพารามิเตอร์ของเกณฑ์ต่าง ๆ

นอกจากวิธีต่าง ๆ ดังกล่าวแล้ว งานวิทยานิพนธ์นี้ยังมีการทดสอบประสิทธิภาพของอัลกอริทึมที่พัฒนาขึ้นด้วยการทดสอบ 3 แบบ คือ การทดสอบหาเกณฑ์ที่เหมาะสมสำหรับการสร้างกฎ FCARs ของอัลกอริทึม CCFAR การทดสอบลักษณะการกระจายตัวของข้อมูลที่เหมาะสมสำหรับอัลกอริทึม CCFAR และการเปรียบเทียบประสิทธิภาพการจำแนกข้อมูลและจำนวนกฎที่ได้รับของอัลกอริทึม CCFAR กับอีก 9 อัลกอริทึม ซึ่งเกณฑ์ที่ใช้ในการทดสอบประสิทธิภาพ คือ ค่าความถูกต้องในการจำแนกข้อมูล จำนวนกฎที่ได้รับและค่าความเหมาะสมของกฎ

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อศึกษาและพัฒนาขั้นตอนวิธีการการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือให้มีความแม่นยำในการจำแนกอยู่ในเกณฑ์ดี
2. เพื่อศึกษาและพัฒนาอัลกอริทึมการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือ ทำให้กฎที่ได้รับนั้นสามารถตีความได้ง่ายและมีจำนวนน้อยแต่ยังครอบคลุมค่าที่เป็นไปได้ทั้งหมดของข้อมูล
3. เพื่อลดปัญหาการกำหนดค่าสนับสนุนขั้นต่ำและค่าความเชื่อมั่นขั้นต่ำ
4. เพื่อเปรียบเทียบหาเกณฑ์ที่เหมาะสมในการสร้างกฎการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือ

1.3 ข้อตกลงเบื้องต้น

1. ข้อมูลที่นำมาใช้จะต้องเป็นข้อมูลตัวเลขและไม่มีค่าที่สูญหาย
2. ข้อมูลที่นำมาประมวลผลจะต้องมีคลาสเป้าหมาย
3. กฎการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือที่ได้รับจะอยู่ในรูปแบบ ถ้า...แล้ว... (If… Then…)
4. ปัจจัยที่ได้ (เงื่อนไขที่อยู่ทางซ้ายมือของกฎ) จะอยู่ในรูปแบบของฟัซซีเซต
5. ในการกำหนดค่าของตัวแปรเชิงภาษา(หรือการกำหนดการแบ่งกลุ่มแบบฟัซซี) จะกำหนดให้มีค่าเป็น 3 ระดับ ได้แก่ระดับ Low ระดับ Medium และระดับ High

1.4 ขอบเขตของการวิจัย

1. การเปรียบเทียบประสิทธิภาพของอัลกอริทึมจะทำการเปรียบเทียบกับอัลกอริทึมที่มีความสามารถในการจำแนกประเภทข้อมูล 3 ประเภท คือ อัลกอริทึมประเภทการจำแนกประเภทข้อมูล(Data Classification) อัลกอริทึมประเภทการจำแนกข้อมูลด้วยกฎความสัมพันธ์(Associative Classification) และอัลกอริทึมที่จำแนกข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือ (Fuzzy Associative Classification)
2. การเปรียบเทียบประสิทธิภาพจะใช้เกณฑ์ค่าความถูกต้องและจำนวนของกฎที่ได้รับ
3. ข้อมูลที่นำมาใช้จะต้องมีขนาดไม่ใหญ่เกินความจุหน่วยความจำหลักของเครื่องคอมพิวเตอร์เพราะถ้าข้อมูลขนาดใหญ่เกินกว่านั้นอัลกอริทึมอาจจะไม่สามารถประมวลผลได้

4. การทดสอบใช้ข้อมูลสังเคราะห์และข้อมูลจริงจาก UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/)
5. ลักษณะของข้อมูลที่นำมาใช้จะต้องเป็นข้อมูลตัวเลข ไม่มีค่าสูญหาย และจะต้องเป็นข้อมูลที่มีคลาสเป้าหมาย

1.5 ประโยชน์ที่จะได้รับ

จากการศึกษาและพัฒนางานวิจัยนี้ผู้วิจัยคาดว่าอัลกอริทึมที่พัฒนาขึ้นจะเกิดประโยชน์ต่อผู้ใช้ในการสังเคราะห์กฎจำแนกประเภทข้อมูลในประเด็นต่าง ๆ ดังนี้

1. กฎที่ได้มีขนาดเล็กและมีจำนวนไม่มากทำให้ผู้ใช้สามารถเข้าใจได้ง่าย
2. ค่าความถูกต้องในการจำแนกข้อมูลสูงขึ้น
3. ลดปัญหาการใช้ข้อมูลตัวเลขต่อเนื่องในการประมวลผลแบบกฎความสัมพันธ์
4. สามารถใช้เกณฑ์ที่เหมาะสมในการสร้างกฎการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือ

บทที่ 2
ปริทัศน์วรรณกรรม

เนื้อหาในบทนี้ประกอบด้วยการทบทวนวรรณกรรมและงานวิจัยที่เกี่ยวข้อง โดยมีรายละเอียดทฤษฎีของฟัซซี (Fuzzy Theory)การประยุกต์ใช้ฟัซซีในการทำเหมืองข้อมูล (Applications of Fuzzy in Data Mining) การจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์ (Classification Based on Associations)การจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือ (Fuzzy Classification with Association Rules)และงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีของฟัซซี

ฟัซซีคือความคลุมเครือหรือความไม่แน่นอนที่ไม่สามารถบอกบางสิ่งได้อย่างชัดเจนว่าจริงหรือไม่จริง ใช่หรือไม่ อย่างเช่นประโยคที่ว่า คนที่มีอายุ 59 ปีเป็นผู้สูงอายุ แล้วคนที่มีอายุ 58 ปี 11 เดือน 30 วันจะเป็นผู้สูงอายุหรือไม่ ซึ่งเราไม่สามารถบอกได้ชัดเจนว่าอายุเท่าไรจึงจัดเป็นผู้สูงอายุ คำว่าสูงอายุมาจากความรู้สึกของคนเราซึ่งมีไม่เท่ากัน แต่ว่าคำว่าสูงอายุนี้ในระบบฟัซซีสามารถบ่งบอกได้ว่าเป็นคนสูงอายุอยู่ที่ระดับใด เช่น คนสูงอายุมาก คนสูงอายุน้อย เป็นต้น ระบบของฟัซซีนั้นจะประกอบไปด้วยเซตของสิ่งที่คลุมเครือซึ่งเราเรียกว่าฟัซซีเซต (Fuzzy Set)ทฤษฎีฟัซซีเซตนั้นถูกคิดค้นเมื่อปี ค.ศ. 1965 โดย L. A. Zadehซึ่งเป็นผลงานวิทยานิพนธ์ระดับปริญญาเอก ที่ได้ให้แนวคิดเกี่ยวกับฟัซซีเซตว่าเป็นเซตที่มีขอบเขตที่ไม่ชัดเจน ซึ่งจะแตกต่างจากเซตทวินัย(Crisp Set) หรือเซตปกติ ตรงที่จะมีระบุระดับที่แตกต่างกันในความเป็นไปได้ของการเป็นสมาชิกในแต่ละเซต ระดับความเป็นสมาชิกนี้สามารถหาได้จากฟังก์ชันสมาชิก (Membership Function) ซึ่งเป็นฟังก์ชันที่มีไว้เพื่อบอกระดับความเป็นสมาชิกของฟัซซีเซตนั้น ๆ เช่น คนที่อายุ 59 ปีจะเป็นคนสูงอายุที่ระดับ 0.4 และเป็นวัยกลางคนที่ระดับ 0.8 จะเห็นได้ว่าคนที่มีอายุ 59 ปีสามารถเป็นทั้งคนสูงอายุและวัยกลางคนได้โดยงานวิทยานิพนธ์นี้ได้นำการจัดกลุ่มแบบฟัซซีซีมีนมาช่วยในการบ่งบอกระดับความเป็นสมาชิกของเซตใด ๆ และได้อธิบายรายละเอียดในหัวข้อที่ 2.1.3

2.1.1ฟัซซีเซต

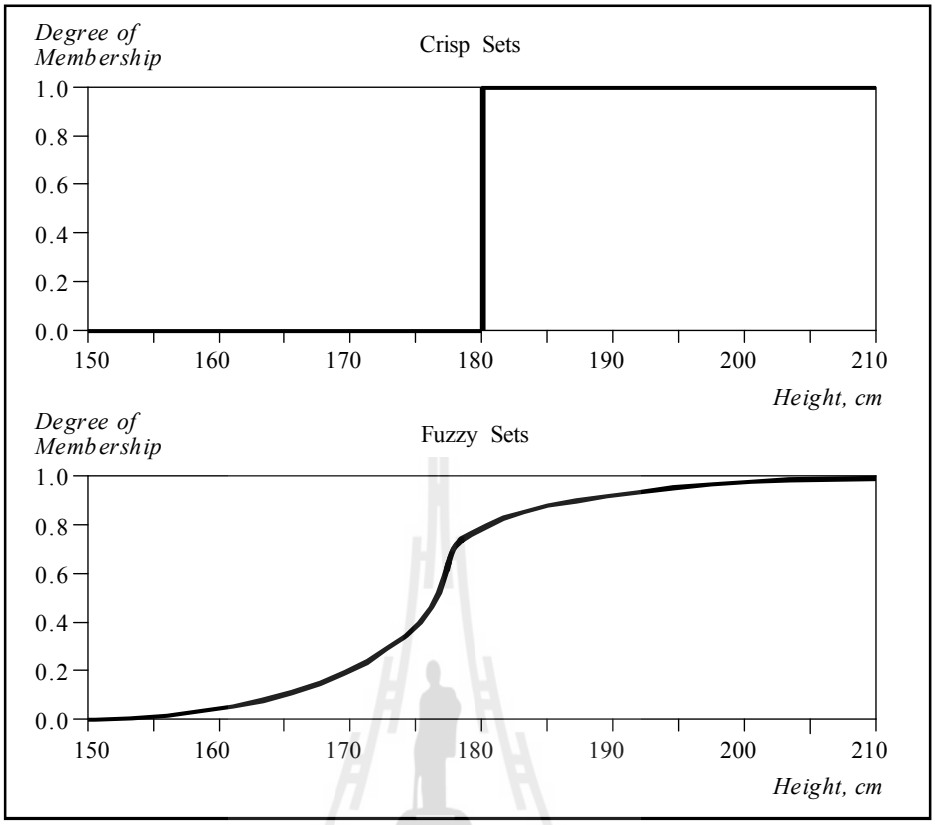
ฟัซซีเซตเป็นคำสองคำที่มาประกอบกันระหว่างคำว่า “ฟัซซี” ที่มีความหมายว่าคลุมเครือไม่แน่นอนอย่างเช่น ความรู้สึกของคนเราที่มีเกณฑ์การแยกแยะความรู้สึกที่ไม่เท่ากัน บางคนเปิดแอร์ที่อุณหภูมิ 15 องศาเซลเซียส บอกว่าเย็นสบายแต่อีกคนกลับบอกว่าหนาวมาก ความไม่

แน่นอนนี้เราบอกไม่ได้ว่ามันเป็นจริงหรือไม่จริงแต่เราสามารถบอกระดับความเป็นจริงนั้น ๆ ได้ ซึ่งจะแตกต่างจากความน่าจะเป็นที่สามารถบอกได้ว่าเป็นจริงหรือไม่จริงอย่างเช่น การโยนเหรียญ 1 ครั้ง สามารถบอกได้ว่าไม่เป็น หัว ก็เป็น ก้อย และอีกคำคือ “เซต” เซตในที่นี้หมายถึงเซตทางคณิตศาสตร์ที่จะประกอบไปด้วยสมาชิกที่มีความแตกต่างกัน ดังนั้นคำว่าฟัซซีเซต จึงหมายความว่าเซตที่ประกอบไปด้วยสมาชิกที่มีความคลุมเครือในระดับของความเป็นสมาชิกในเซต

ฟัซซีเซตอาจจะเรียกได้ว่าเป็นส่วนเติมเต็มของ เซตทวินัย (Crisp set) ซึ่งเป็นเซตที่มีค่าความเป็นสมาชิกของตัวแปรที่แน่นอน คือ 0 กับ 1 อย่างเช่นเซตของคนที่จะถูกเรียกว่าสูง ดังตารางที่ 1 จะเห็นได้ว่าคนที่จะถูกเรียกว่าสูงนั้นจะมีค่า Crisp เท่ากับ 1 หรือสูงมากกว่า 180 cm ขึ้นไป ส่วนคนที่มีค่า Crisp เท่ากับ 0 คือคนที่ไม่สูง ซึ่งค่า 180 นี้เองที่เป็นเส้นแบ่งระหว่างคำว่าสูงกับไม่สูง ในความเป็นจริงนั้นเราไม่สามารถบอกได้ว่าคนที่มีความสูงมากกว่า 180 cm จะต้องเป็นคนที่สูงเพราะว่าความรู้สึกของคนเราไม่เท่ากัน บางคนอาจจะมองว่าคนที่สูง 170 cm เป็นคนที่สูง ดังนั้นการใช้ฟัซซีเซตเข้ามาช่วยทำให้สามารถแยกแยะได้ว่าแต่ละคนมีความสูงที่ระดับใดด้วยค่าความเป็นสมาชิก ที่มีค่าอยู่ในช่วง [0 – 1] จากตารางที่ 2.1 สามารถวิเคราะห์ได้ว่าแต่ละคนมีระดับความเป็นสมาชิกในเซตคนสูงที่ระดับใด เช่น David ที่มีระดับความเป็นสมาชิกอยู่ที่ 0.78 หรือ Chris และ Mark ที่มีความสูงแตกต่างกันคือ 208 และ 205 ตามลำดับ แต่มีระดับระดับความเป็นสมาชิกในเซตคนสูงที่ระดับเดียวกันคือ 1 ส่วน Peter มีค่าระดับระดับความเป็นสมาชิกในเซตคนสูงเท่ากับ 0 หรืออาจจะเรียกได้ว่าไม่สูง ข้อมูลจากตารางที่ 2.1 ส่วนของ Fuzzy แสดงเป็นกราฟได้ดังรูปที่ 2.1

ตารางที่ 2.1ตัวอย่างระดับความสูงของเซตทวินัยและฟัซซีเซต(He, 2014)

Name	Height, cm	Degree of Membership	
		Crisp	Fuzzy
Chris	208	1	1
Mark	205	1	1
John	198	1	0.98
Tom	181	1	0.82
David	179	0	0.78
Mike	172	0	0.24
Bob	167	0	0.15
Steven	158	0	0.06
Bill	155	0	0.01
Peter	152	0	0



รูปที่ 2.1 ฟังก์ชันสมาชิกของเซตทวินัยและฟัซซีเซต(He, 2014)

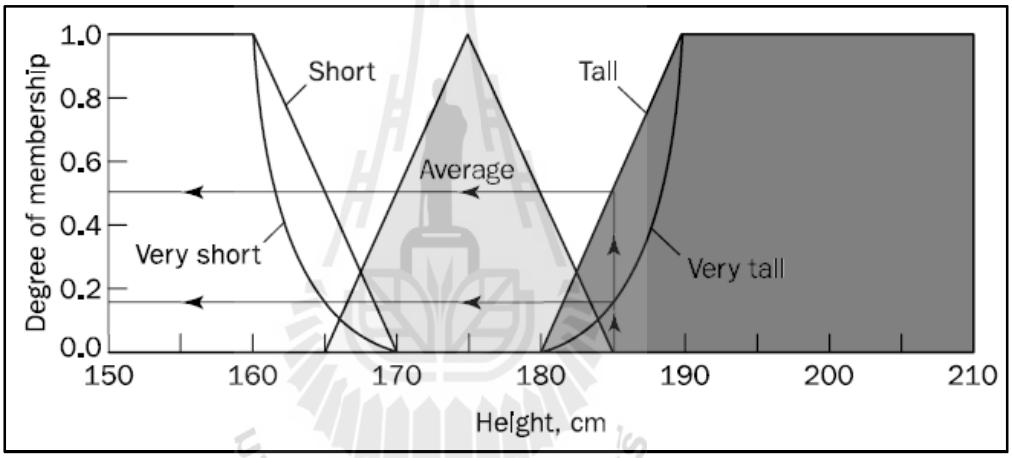
จากรูปที่ 2.1 นั้นแกน x หมายถึงค่าความสูง ส่วนแกน y หมายถึงค่าความเป็นสมาชิก และเราสามารถแสดงฟัซซีของคนที่สูงได้หลายรูปแบบ เช่น Tall-men = {(Chris,1), (Mark,1), (John,0.98), (Tom,0.82), (David,0.78), (Mike,0.24), (Bob,0.15), (Steven,0.06), (Bill,0.01), (Peter,0)} หรืออาจจะแทนด้วยรูปแบบของสมการที่ 2.1 (เครื่องหมายทางคณิตศาสตร์มิได้หมายถึงการหารหรือการบวก แต่เป็นการแสดงระดับความเป็นสมาชิกของแต่ละบุคคลในเซต)

$$\text{Tall-men} = \frac{1}{\text{Chris}} + \frac{1}{\text{Mark}} + \frac{0.98}{\text{John}} + \frac{0.82}{\text{Tom}} + \frac{0.78}{\text{David}} + \frac{0.24}{\text{Mike}} + \frac{0.15}{\text{Bob}} + \frac{0.06}{\text{Steven}} + \frac{0.01}{\text{Bill}} + \frac{0}{\text{Peter}} \quad (2.1)$$

จากสมการที่ 2.1 แสดงถึงค่าความเป็นสมาชิกของแต่ละบุคคลในฟัซซีTall-men ซึ่งสามารถตีความได้ว่า Chris มีระดับความเป็นสมาชิกของเซต Tall-men ที่ระดับ 1 และ Mark, John, Tom, …, Peter มีระดับความเป็นสมาชิกของเซต Tall-men เป็น 1, 0.98, 0.82, …, 0 ตามลำดับ

2.1.2 ตัวแปรเชิงภาษา(Linguistic Variable)

การทำฟัซซีสิ่งที่ต้องพบบ่อย ๆ คือการกำหนดตัวแปรเชิงภาษาที่จะบอกถึงลักษณะหรือระดับตามความรู้สึกของสิ่งต่าง ๆ เช่น ความสูง ความอ้วน อุณหภูมิ เป็นต้น ตัวแปรเชิงภาษาจะประกอบไปด้วยคำบ่งบอกปริมาณหรือความรู้สึกเช่น ความสูงมาก ความอ้วนน้อย อุณหภูมิหนาวที่สุด โดยที่คำว่า มาก (Very) ค่อนข้าง(Slightly) ที่สุด (Extremely) เป็นค่าตัวแปรทางภาษาที่ประกอบด้วย คำคุณศัพท์ (Adjective) หรือคำกริยาวิเศษณ์ (Adverb) ซึ่งในขั้นตอนการประมวลผลของฟังชันก์สมาชิก ค่าเหล่านี้จะถูกแปลงให้อยู่ในรูปแบบของตัวเลข และค่าตัวแปรเชิงภาษาแต่ละค่าเหล่านี้จะส่งผลต่อรูปร่างฟังชันก์สมาชิก ดังรูปที่ 2.2



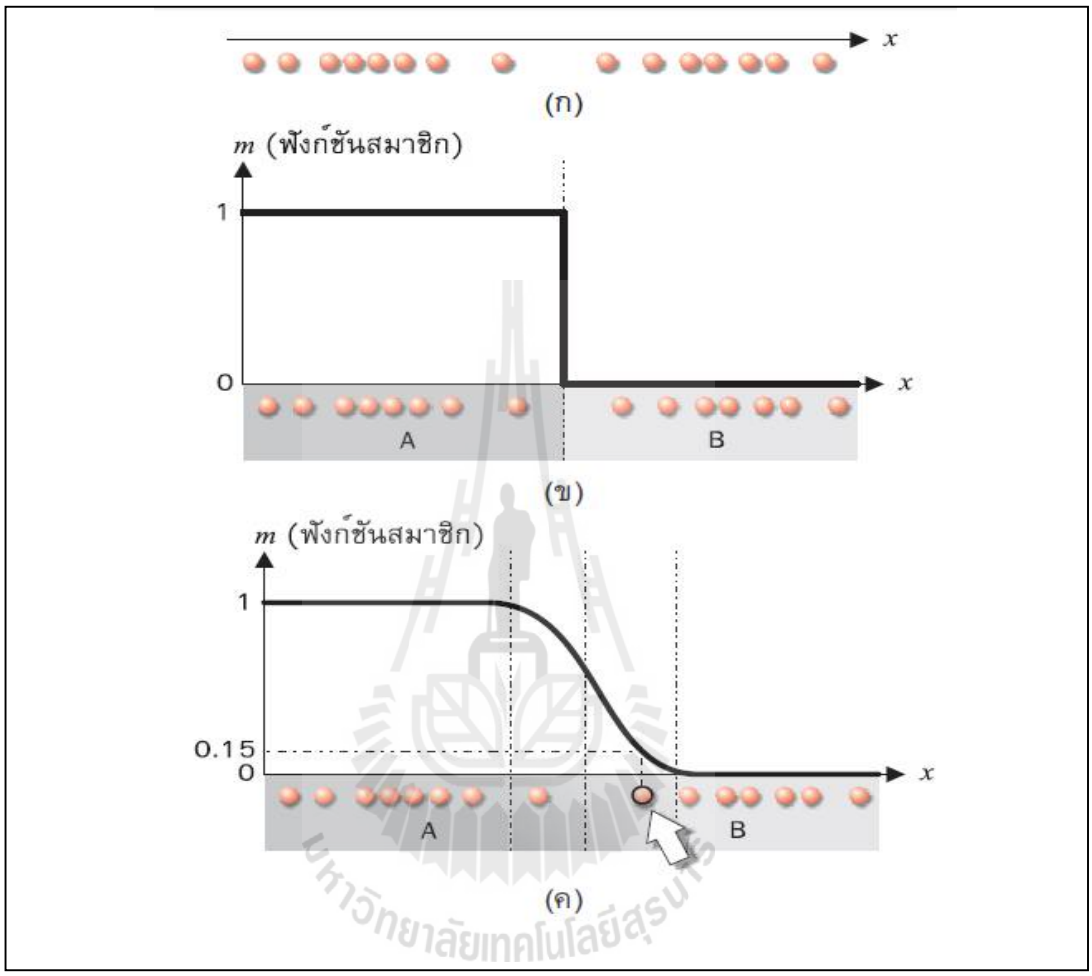
รูปที่ 2.2 ค่าของตัวแปรเชิงภาษาในแต่ละช่วงของความสูง(He, 2014)

จากรูปที่ 2.2 ผู้ชายที่สูง 185 cm เป็นสมาชิกของเซต Tall ด้วยค่าความเป็นสมาชิกเท่ากับ 0.5 และยังเป็นสมาชิกของเซต Very tall ด้วยค่าความเป็นสมาชิกเท่ากับ 0.15 ซึ่งจะเห็นได้ว่าผู้ชายคนดังกล่าวไม่เพียงแค่สูง (Tall) อย่างเดียวแต่ยังสูงมาก (Very tall) ด้วย ดังนั้นคำที่ใช้เป็นตัวแปรเชิงภาษาจึงจำเป็นต้องแตกต่างกันเพื่อให้สามารถบอกถึงระดับความแตกต่างของเซตต่าง ๆ ได้

2.1.3การบ่งบอกระดับความเป็นสมาชิกด้วยอัลกอริทึมฟัซซีซีมีน(FuzzyC-means:FCM)

ในงานวิทยานิพนธ์นี้ได้นำอัลกอริทึมฟัซซีซีมีนมาใช้ในการบ่งบอกระดับความเป็นสมาชิกของเซตที่มีค่า [0-1]ซึ่งจะแตกต่างจากอัลกอริทึม K-Means ที่บ่งบอกระดับความเป็นสมาชิกที่มีค่า 0 และ 1 เท่านั้น ดังรูปตัวอย่างที่ 2.3 แสดงการจัดกลุ่มข้อมูล 1 มิติ ของอัลกอริทึม K-Means และอัลกอริทึม FCMโดยที่ (ก) คือข้อมูลขนาด 1 มิติที่ยังไม่ได้จัดกลุ่ม (ข) การจัดกลุ่มเป็นกลุ่ม A และกลุ่ม B ด้วยอัลกอริทึมK-Means และ (ค) การจัดกลุ่มด้วยค่าความเป็นสมาชิกของอัลกอริทึม

FCM จะเห็นได้ว่าการจัดกลุ่มด้วย FCM ข้อมูลหนึ่งตัวสามารถอยู่ได้หลายกลุ่ม และขั้นตอนการประมวลผลของอัลกอริทึม FCM มีรายละเอียดดังนี้(Bezdeket al., 1984)



รูปที่ 2.3 การจัดกลุ่มข้อมูล 1 มิติ (ก) ของอัลกอริทึม K-Means (ข) และอัลกอริทึม FCM (ค) (อาทิตย์ ศรีแก้ว, 2552)

1.กำหนดจำนวนกลุ่มและค่าเริ่มต้นความเป็นสมาชิก

2.คำนวณหาค่าจุดศูนย์กลางของกลุ่มด้วยสมการที่2.2

$$c_j^{(t)} = \sum_{i=1}^{N} \mu_{ij}^{(t)m} x_i \,/\, \sum_{ij}^{N} \mu_{ij}^{(t)m} \qquad (2.2)$$

3.ทำการคำนวณและปรับค่าความเป็นสมาชิกของข้อมูลด้วยสมการที่ 2.3

$$\mu_{ij}^{(t+1)m} = 1/\sum_{k=1}^{C}\left(\frac{||x_i - c_j||}{||x_i - c_k||}\right)^{\frac{2}{(m-1)}} (2.3)$$

4.แต่ถ้า$\|\mu_{ij}^{(t+1)m} - \mu_{ij}^{(t)m}\|$ มีค่าน้อยกว่า β ซึ่งเป็นเกณฑ์ที่ตั้งไว้ล่วงหน้าให้หยุดทำงาน แต่ถ้ามากกว่าให้กลับไปที่ขั้นตอนที่สองและเพิ่มค่า t ด้วย 1

โดยที่สัญลักษณ์ต่าง ๆ มีความหมายดังนี้

x_i คือข้อมูลลำดับที่ iโดยข้อมูลจะอยู่ในรูปแบบของเวกเตอร์

mคือ สัมประสิทธิ์ความเป็นฟัซซี ซึ่งมีค่ามากกว่าหรือเท่ากับ 1 แต่น้อยกว่า อนันต์

μ_{ij} คือ ค่าความเป็นสมาชิกของ x_iที่กลุ่ม j

N คือ ข้อมูลทั้งหมดที่ใช้ในการจัดกลุ่ม

C คือ จำนวนกลุ่มทั้งหมดและ c_jคือ จุดศูนย์กลางกลุ่ม

tคือ จำนวนรอบ โดยจะมีค่าเริ่มต้นเท่ากับ 0

2.2 การประยุกต์ใช้ฟัซซีในการทำเหมืองข้อมูล

การทำเหมืองข้อมูลคือ การหารูปแบบ (Pattern) หรือโมเดล (Model) ของข้อมูล หรือเป็นการหาความสัมพันธ์(Association/Relation) ของข้อมูล เพื่อมาใช้ในการจำแนกประเภทข้อมูล (Data Classification) ใช้เพื่อการแบ่งกลุ่มข้อมูล (Clustering) และการหากฎความสัมพันธ์ของข้อมูล (Association Rule) โดยใช้หลักการของสถิติ และการเรียนรู้ของเครื่อง (Machine Learning) (Fayyad et al., 1996; Seifert, 2004) ซึ่งมีแนวคิดการทำเหมืองข้อมูลอย่างง่ายคือการนำข้อมูลในอดีตที่มีความสัมพันธ์กับข้อมูลปัจจุบัน มาสร้างโมเดลหรือรูปแบบที่มีประโยชน์ต่อการทำเหมืองข้อมูล (Data Archeology)เช่น การนำข้อมูลสภาพอากาศของอดีตมาวิเคราะห์เพื่อหารูปแบบของสภาพอากาศ เพื่อทำนายว่าพรุ่งนี้สภาพอากาศจะเป็นอย่างไรหรือการนำเอาข้อมูลการซื้อสินค้าของลูกค้ามาวิเคราะห์หารูปแบบของข้อมูลเพื่อจัดการชั้นวางสินค้าหรือจัดรายการแนะนำสินค้า เป็นต้น

การนำเทคนิคของฟัซซีมาใช้ในการทำเหมืองข้อมูลในแต่ละขั้นตอนมีประโยชน์ดังนี้

1. ขั้นตอนการเตรียมข้อมูล (Data Preparation) จะช่วยในการทำความสะอาดข้อมูลเช่น การกำจัดข้อมูลที่มีลักษณะผิดปกติ (Outliers) ด้วยการใช้การจัดกลุ่มแบบคลุมเครือ (Fuzzy Clustering) (Keller et al., 2005; Naucket al., 1997)หรือทำการแปลงข้อมูล (Transform) ให้อยู่ในรูปแบบของระดับความเป็นสมาชิกของเซต
2. ขั้นตอนการประมวลผล (Processing) ช่วยในการแก้ปัญหาการคำนวณข้อมูลที่มีลักษณะเชิงปริมาณ (Quantitative) ในการทำเหมืองข้อมูลแบบหากฎความสัมพันธ์ของ

ข้อมูล (Hong et al., 2003)หรือข้อมูลที่มีลักษณะเป็นค่าต่อเนื่อง ยกตัวอย่างเช่น งานวิทยานิพนธ์เล่มนี้ได้ใช้เทคนิคฟัซซีมาใช้ในการแบ่งแยกข้อมูลเพื่อบ่งบอกระดับความเป็นสมาชิก และมีการคำนวณเกณฑ์(Measurement) ต่าง ๆ เป็นแบบฟัซซี

3. ขั้นตอนการประเมินผล (Evaluation)ในการประเมินผลของโมเดลสามารถทำได้ง่าย เพราะการทำฟัซซีมีระบบอนุมานผล(Interpretable Systems) ในการตรวจคำตอบ (Kruse et al., 1999)

จากประโยชน์ดังกล่าวงานวิทยานิพนธ์เล่มนี้จึงได้นำเทคนิคฟัซซีมาใช้ในการสร้างกฎการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือเพื่อลดปัญหาข้อมูลที่มีลักษณะเป็นค่าต่อเนื่องเพื่อเพิ่มประสิทธิภาพในการจำแนกข้อมูลและผลที่ได้สามารถทำความเข้าใจได้ง่ายและสะดวกต่อการนำไปประยุกต์ใช้

2.3 การจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์

การจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์เป็นวิธีการสร้างกฎเพื่อนำมาใช้ในการทำนายข้อมูล ซึ่งกฎที่นำมาสร้างนั้นคือ กฎความสัมพันธ์ที่ได้มาจากการประมวลผลของการทำเหมืองข้อมูลเพื่อค้นหากฎความสัมพันธ์ (Association Rule Mining) หรืออธิบายการจำแนกข้อมูลด้วยกฎความสัมพันธ์อีกความหมายหนึ่งคือ การนำข้อดีของสองเทคนิคทางด้านเหมืองข้อมูลมารวมกันระหว่างเทคนิคการทำเหมืองข้อมูลเพื่อค้นหาความสัมพันธ์และเทคนิคการจำแนกประเภทข้อมูลโดยการนำเทคนิคการทำเหมืองข้อมูลเพื่อค้นหากฎความสัมพันธ์ มาช่วยในการสร้างกฎด้วยวิธีการหาความสัมพันธ์ของข้อมูลทั้งหมดจากฐานข้อมูล(Hanet al., 2006; Zhanget al., 2002) แต่กฎที่ได้เป็นผลลัพธ์มีจำนวนมากและยากต่อการนำไปใช้ในการจำแนกข้อมูล จึงต้องนำข้อดีของเทคนิคการจำแนกประเภทข้อมูลเข้ามาช่วยลดจำนวนกฎความสัมพันธ์และสร้างกฎความสัมพันธ์ให้สามารถใช้ในการจำแนกประเภทข้อมูลได้ ดังตัวอย่างข้อแตกต่างระหว่างกฎความสัมพันธ์และกฎจำแนกประเภทข้อมูลดังนี้

กฎที่(1) IF B then D and E

กฎที่(2) IF B then Class=Yes

จากตัวอย่าง กฎที่(1) คือกฎความสัมพันธ์ที่อ่านว่า ถ้า B เกิดขึ้น แล้ว D และ E จะเกิดขึ้นด้วย แต่ถ้าเป็นกฎที่(2) ซึ่งเป็นกฎการจำแนกประเภทข้อมูล อ่านว่า ถ้า B เกิดขึ้นแล้ว จะทำนายได้ว่าจะเป็นจริง (Yes) ซึ่งจะเห็นข้อแตกต่างของทั้งสองกฎนี้คือ กฎที่(1) สามารถมีแอททริบิวต์ (Attribute) ข้างหลัง then (Consequent)ได้มากกว่า 1 แอททริบิวต์คือ D และ E แต่กฎที่(2) สามารถ

มีแอททริบิวต์ข้างหลัง then ได้เพียงหนึ่งแอททริบิวต์ซึ่งแอททริบิวต์ดังกล่าวจะเป็นแอททริบิวต์คลาสเป้าหมาย (Class Target)

แนวคิดการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์นี้ได้ถูกนำเสนอครั้งแรกโดย Liuet al. (1998)และได้เสนออัลกอริทึมที่มีชื่อว่าClassification Based on Association Rules (CBA)ซึ่งกฎที่ได้จากการประมวลผลของการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์นี้เรียกว่า Class Association Rules (CARs)และในการสร้างกฎ CARsของงานวิจัยนี้ ได้ใช้ค่าสนับสนุน(Support) และค่าความเชื่อมั่น(Confidence) เป็นเกณฑ์ในการสร้างกฎ และในการทดสอบประสิทธิภาพของอัลกอริทึม CBA พบว่ามีความแม่นยำมากกว่าอัลกอริทึม C4.5 (Quinlan, 1993)ซึ่งเป็นอัลกอริทึมที่ได้รับความนิยมใช้กันอย่างแพร่หลายในปัจจุบัน ต่อมามีผู้วิจัยอีกจำนวนมากได้เสนอแนวคิดในการสร้างกฎ CARs ซึ่งในงานวิทยานิพนธ์เล่มนี้ได้นำเสนอแนวคิดการสร้างกฎ CARsของงานวิจัยที่เกี่ยวข้องไว้ที่หัวข้อ 2.6 และอธิบายทฤษฎีพื้นฐานของการทำเหมืองข้อมูลเพื่อค้นหาความสัมพันธ์และการทำเหมืองข้อมูลแบบจำแนกประเภทไว้ในหัวข้อ 2.3.1 และ 2.3.2 ตามลำดับ

2.3.1การทำเหมืองข้อมูลเพื่อค้นหาความสัมพันธ์(Association Mining)

การทำเหมืองข้อมูลเพื่อค้นหาความสัมพันธ์ถูกนำเสนอขึ้นเมื่อปี 1993โดย RakeshAgrawal และคณะเพื่อนำไปใช้ในการวิเคราะห์รูปแบบความสัมพันธ์ของข้อมูลหรือ การวิเคราะห์การซื้อสินค้าของลูกค้า(Market Basket Analysis) ซึ่งข้อมูลรายการซื้อของลูกค้าจะถูกเก็บไว้ในรูปแบบของทรานแซคชัน (Transactions) โดยที่ทรานแซคชันนี้เหมือนกับตะกร้าซื้อของที่ประกอบด้วยสินค้าหลาย ๆ ชิ้น เมื่อเก็บทรานแซคชันได้จำนวนมาก ก็สามารถนำทรานแซคชันเหล่านี้ไปประมวลผลเพื่อหากฎความสัมพันธ์และนำกฎความสัมพันธ์ที่ได้มาใช้ในการวางแผนการจัดวางชั้นสินค้าการแนะนำสินค้าหรือแผนจัดรายการกระตุ้นยอดขายสินค้า

ในการสร้างกฎความสัมพันธ์มีผู้เสนอวิธีการสร้างไว้หลายอัลกอริทึม แต่อัลกอริทึมที่เป็นที่นิยมมี 2 อัลกอริทึมคือ Apriori(Agrawalet al., 1994) และ FP-growth (Hanet al., 2000)โดยที่ Aprioriได้ใช้เทคนิค Support-based Pruning เพื่อช่วยตัดหรือลดจำนวนเซตคู่แข่ง(Candidate Itemsets) แทนที่จะแจกแจงทั้งหมดและใช้การค้นหาคำตอบแบบ Breadth-First-Search ส่วน FP-growth จะทำงานตัดหรือลดจำนวนเซตคู่แข่งด้วยวิธีการ Depth-First Search(Palanisamy, 2006) โดยงานวิทยานิพนธ์นี้ได้ใช้อัลกอริทึม Apriori ในการสร้างกฎความสัมพันธ์ เนื่องจากมีขั้นตอนที่ไม่ซับซ้อนและเข้าใจได้ง่ายกว่าอัลกอริทึม FP-growth ซึ่งมีรายละเอียดการค้นหากฎความสัมพันธ์ 2 ขั้นตอนดังนี้

- ขั้นตอนที่ 1 การสร้างไอเท็มเซตปรากฏบ่อย (Frequent Itemset)โดยไอเท็มเซตใด ๆ จะเป็นไอเท็มเซตปรากฏบ่อยจะต้องมีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ(Minimum Support) ซึ่งค่าสนับสนุนขั้นต่ำจะถูกกำหนดมาจากผู้ใช้ ส่วนค่าสนับสนุนของแต่ละไอเท็มเซตสามารถคำนวณได้จากสมการที่ 2.4และแสดงขั้นตอนวิธีการสร้างไอเท็มเซตปรากฏบ่อย ดังรูปที่ 2.4

การหาค่าสนับสนุนของไอเท็มA หาได้จาก

$$\text{Support (A)} = \frac{numberoftransactionsthatcontainA}{numberofalltransactions} \quad (2.4)$$

- ขั้นตอนที่ 2 การสร้างกฎความสัมพันธ์จากไอเท็มเซตปรากฏบ่อยโดยที่ไอเท็มเซตปรากฏบ่อยใด ๆ ที่นำมาสร้างนั้นจะต้องมีขนาดของเซตมากกว่า 1ไอเท็ม เช่น เซตของ {A, B}มีขนาด 2 ไอเท็ม เซตของ {A, B, C}มีขนาด 3 ไอเท็มเป็นต้น จากนั้นไอเท็มเซตปรากฏบ่อยที่ผ่านเงื่อนไขจะถูกนำมาสร้างกฎความสัมพันธ์ด้วยเกณฑ์ค่าความเชื่อมั่น โดยที่กฎความสัมพันธ์จะต้องมีค่าความเชื่อมั่นมากกว่าค่าความเชื่อมั่นขั้นต่ำ (Minimum Confidence) ซึ่งค่าความเชื่อมั่นขั้นต่ำจะถูกกำหนดมาจากผู้ใช้ และสมการที่ใช้หาค่าความเชื่อมั่นของกฎความสัมพันธ์แสดงดังสมการที่ 2.5และแสดงขั้นตอนการสร้างกฎความสัมพันธ์ ดังรูปที่ 2.5

การหาค่าความเชื่อมั่นของกฎ A→Bหาได้จาก

$$\text{Confidence(A}\rightarrow\text{B)} = \frac{support\ (A\ and\ B)}{support(A)} \quad (2.5)$$

Algorithm Apriori

//Input : Database D , Minimum_support.

//Output : L frequent itemsets in D.

(1) $L_1 = \text{find\_ frequent\_1itemset}(D)$

(2) $\text{for}(k = 2; L_{k-1} \neq \emptyset; k{+}{+})\{$

(3) $\quad C_k = \text{apriori\_gen}(L_{k-1}, \text{Minimum\_support});\}$

(4) for each transaction $t \in D$ { // scan D for counts

(5) $\quad C_1 = \text{subset}(C_k, t)$

(6) $\quad$ for each cadidate $c \in C_1$ {

(7) $\quad\quad$ c.count++ }

(8) $\quad$ }

(9) $L_k = \{c \in C_1 | \, c.\text{count} \geq \text{Minimum\_support}\}$

(10) }

(11) return $\cup_k L_k$

รูปที่ 2.4 การสร้างไอเท็มเซตปรากฏบ่อยของอัลกอริทึม Apriori(Agrawalet al., 1994)

Procedure generate_rule(L_k : frequent-items, Min_conf : Minimum confident, RHS : right hand side Items);

(1) For each l $\in L_k$// l is frequent-itemset.

(2) k = | l | // size offrequent itemset

(3) m = $|H_m|$ // size of right hand side Items

(4) For each $h_{m+1} \in H_{m+1}$ {

(5) If h_{m+1} = RHS {

(6) conf = $\sigma(f_k) / \sigma (f_k - h_{m+1})$;

(7) If conf>= Min_conf {

(8) Rule = rule($f_k - h_{m+1}$) $\Rightarrow h_{m+1}$

(9) } Else

(10) delete h_{m+1} from H_{m+1}

(11) }

(12) }

(13) return Rule

รูปที่ 2.5 การสร้างกฎความสัมพันธ์ (Agrawalet al., 1994)

ตัวอย่างการหากฎความสัมพันธ์ด้วยอัลกอริทึมApriori สามารถแสดงได้ดังนี้สมมติว่ามีลูกค้ามาซื้อของที่ร้านสะดวกซื้อจำนวน 5 คน (ตารางที่ 2.2)เจ้าของร้านสะดวกซื้อต้องการจัดชั้นวางสินค้า โดยใช้ข้อมูลลูกค้าทั้ง 5 คน มาวิเคราะห์เพื่อหารูปแบบการซื้อสินค้า โดยมีเงื่อนไขว่าสินค้าที่นำมาวางบนชั้นวางนั้นจะต้องมีลูกค้าซื้อไปแล้วอย่างน้อย 3 ชิ้น จากโจทย์จะเห็นได้ว่าเจ้าของร้านสะดวกซื้อต้องการหากฎความสัมพันธ์ของข้อมูลจากลูกค้า 5 คน แต่กฎความสัมพันธ์ที่เจ้าของร้านต้องการนั้น จะต้องมีค่าสนับสนุนขั้นต่ำมากกว่า 3 ซึ่งสามารถวิเคราะห์หากฎความสัมพันธ์ได้ดังนี้

ตารางที่ 2.2 รายการซื้อสินค้าของลูกค้า 5คน

TID	Items
001	Eggs, Pepsi, Noodle
002	Sushi , Pepsi, Toy
003	Eggs, Sushi , Pepsi, Toy
004	Sushi , Pepsi , Toy
005	Eggs, Noodle

จากรายการซื้อของของลูกค้า 5คนหรือเรียกว่า 5ทรานแซคชัน มีการซื้อสินค้าอยู่ 5 ชนิดหรือเรียกว่า 5 ไอเท็มคือ Eggs,Sushi, Pepsi, NoodleและToy จากข้อมูลตารางที่ 2.2 ทำการนับค่าสนับสนุนของแต่ละไอเท็มจะได้ดังรูปที่ 2.6(ก) และทำการคัดเลือกไอเท็มที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ คือ 3 ไอเท็ม จะได้ผลลัพธ์ดังรูปที่ 2.6(ข) ซึ่งเรียกว่าไอเท็มเซตปรากฏบ่อยขนาด 1 ไอเท็ม

Item	**Support**
Eggs	3
Sushi	3
Pepsi	4
~~Noodle~~	~~2~~
Toy	3

(ก) ไอเท็มเซตคู่แข่งขนาด 1 ไอเท็ม

Item	**Support**
Eggs	3
Sushi	3
Pepsi	4
Toy	3

(ข) ไอเท็มเซตปรากฏบ่อยขนาด 1 ไอเท็ม

รูปที่ 2.6การสร้างไอเท็มเซตคู่แข่งและไอเท็มเซตปรากฏบ่อยขนาด 1 ไอเท็ม

ต่อจากนั้นทำการสร้างเซตคู่แข่งขนาด 2 ไอเท็ม โดยการนำไอเท็มเซตปรากฏบ่อยขนาด 1 ไอเท็มไปจับคู่จะได้ดังรูปที่ 2.7 (ก) และทำการคัดเลือกไอเท็มเซตที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ คือ 3 ไอเท็ม เช่นเดียวกันกับตอนสร้างไอเท็มเซตปรากฏบ่อยขนาด 1 ไอเท็ม ซึ่งจะได้ผลลัพธ์ดังรูปที่ 2.7(ข) คือ ไอเท็มเซตปรากฏบ่อยขนาด 2 ไอเท็ม

Item	Support
~~Eggs, Sushi~~	~~1~~
~~Eggs, Pepsi~~	~~2~~
~~Eggs, Toy~~	~~1~~
Sushi, Pepsi	3
Sushi, Toy	3
Pepsi, Toy	3

→

Item	Support
Sushi, Pepsi	3
Sushi, Toy	3
Pepsi, Toy	3

(ก)ไอเท็มเซตคู่แข่งขนาด 2 ไอเท็ม (ข) ไอเท็มเซตปรากฏบ่อยขนาด 2 ไอเท็ม

รูปที่ 2.7การสร้างไอเท็มเซตคู่แข่งและไอเท็มเซตปรากฏบ่อยขนาด 2 ไอเท็ม

หลังจากได้ไอเท็มเซตปรากฏบ่อยขนาด 2 ไอเท็มแล้วขั้นตอนต่อไปก็จะเหมือนกันกับขั้นตอนการสร้างไอเท็มเซตปรากฏบ่อยขนาด 2 ไอเท็ม คือ การสร้างไอเท็มเซตปรากฏบ่อยขนาด 3 ไอเท็ม โดยการนำไอเท็มเซตปรากฏบ่อยขนาด 2 ไอเท็มไปจับคู่ ซึ่งจะได้เซตคู่แข่งขนาด 3 ไอเท็ม คือ {Sushi, Pepsi, Toy} เพียงไอเท็มเซตเดียวและมีค่าสนับสนุนเท่ากับ 3 ที่มีค่าเท่ากันกับค่าสนับสนุนขั้นต่ำ ทำให้ไอเท็มเซตดังกล่าวเป็นไอเท็มเซตปรากฏบ่อยขนาด 3 ไอเท็มด้วย ซึ่งต่อจากนี้เป็นการสร้างไอเท็มเซตคู่แข่งขนาด 4 ไอเท็ม แต่ไอเท็มเซตปรากฏบ่อยขนาด 3 ไอเท็ม มีเพียงไอเท็มเซตเดียวจึงไม่สามารถจับคู่ได้ ทำให้สิ้นสุดการสร้างไอเท็มเซตปรากฏบ่อยที่ขนาด 3 ไอเท็ม

วิธีการต่อไปคือการสร้างกฎความสัมพันธ์จากไอเท็มเซตปรากฏบ่อยขนาด 1 ไอเท็มเซตขึ้นไป ซึ่งมีอยู่ 4 ไอเท็มเซต คือ {Sushi, Toy}, {Sushi, Pepsi}, {Pepsi, Toy} และ{Sushi, Pepsi, Toy}โดยสามารถสร้างได้ 12 กฎความสัมพันธ์ดังนี้

1. Sushi ==> Pepsi (confidence = 100%)
2. Sushi ==> Toy (confidence = 100%)
3. Pepsi, Toy ==> Sushi (confidence = 100%)
4. Sushi, Toy ==> Pepsi (confidence = 100%)
5. Sushi, Pepsi ==> Toy (confidence = 100%)
6. Sushi ==> Pepsi, Toy (confidence = 100%)

7. Pepsi ==> Sushi (confidence = 75%)
8. Toy ==> Sushi (confidence = 75%)
9. Toy ==> Pepsi (confidence = 75%)
10. Pepsi ==> Toy (confidence = 75%)
11. Toy ==> Sushi, Pepsi (confidence = 75%)
12. Pepsi ==> Sushi, Toy (confidence = 75%)

จาก 4 ไอเท็มเซตปรากฏบ่อย สามารถสร้างกฎความสัมพันธ์ได้มากถึง 12 กฎ เพราะว่าในการสร้างกฎความสัมพันธ์นั้น หนึ่งไอเท็มเซตปรากฏบ่อยสามารถนำไอเท็มทั้งหมดในเซตมาเรียงสับเปลี่ยนเพื่อสร้างกฎความสัมพันธ์ได้มากกว่า 1 กฎความสัมพันธ์ ซึ่งคำนวณจำนวนกฎความสัมพันธ์ได้จาก ($2^N - 2$ =จำนวนกฎความสัมพันธ์โดยที่ N คือจำนวนไอเท็ม) เช่น เซตปรากฏบ่อย {A, B, C}สามารถสร้างกฎความสัมพันธ์ได้มากถึง 6 กฎความสัมพันธ์ คือ$2^3 - 2 = 6$ และแสดงรายละเอียดของแต่ละกฎความสัมพันธ์ดังรูปที่ 2.8

IF	A	then	B, C
IF	B	then	A, C
IF	C	then	A, B
IF	A, B	then	C
IF	A, C	then	B
IF	B, C	then	A

รูปที่ 2.8ตัวอย่าง กฎความสัมพันธ์

จากสาเหตุดังกล่าวทำให้กฎความสัมพันธ์ที่ได้รับนั้นมีจำนวนมากจึงทำให้การเลือกกฎความสัมพันธ์มาใช้งานค่อนข้างลำบาก ดังนั้นจากผลลัพธ์ทั้ง 12 กฎความสัมพันธ์ เจ้าของร้านสะดวกซื้อจะต้องทำการคัดเลือกกฎทั้ง 12 มาใช้ในการจัดวางชั้นสินค้าเองอีกทีด้วยประสบการณ์จัดวางสินค้าเจ้าของร้าน จากปัญหาดังกล่าวทำให้งานวิจัยนี้ได้เสนอวิธีการทำเหมืองข้อมูลแบบจำแนกประเภทข้อมูลจะสามารถลดจำนวนของกฎความสัมพันธ์ให้มีขนาดลดลงได้

2.3.2การจำแนกประเภทข้อมูล

จากหัวข้อ 2.3.1 ได้เสนอวิธีการสร้างกฎความสัมพันธ์ไว้ซึ่งจำนวนกฎความสัมพันธ์ที่ได้รับมีจำนวนมาก แต่ในหัวข้อนี้จะเสนอวิธีการทำเหมืองข้อมูลแบบจำแนกประเภทข้อมูลแบบง่าย

ๆ เพื่อนำมาสร้างกฎCARsโดยการใช้วิธีการคัดเลือกกฎความสัมพันธ์ด้วยการเลือกกฎที่มีแอททริบิวต์ที่อยู่หลัง then ที่เพียง 1 แอททริบิวต์มาเป็นกฎ CARs(Palanisamy, 2006)ซึ่งจากกฎความสัมพันธ์ทั้งหมด ดังรูปที่ 2.9 (ก)สามารถตัดกฎที่มีจำนวนแอททริบิวต์ข้างหลัง then มากกว่า 1 แอททริบิวต์ได้ 3 กฎและกฎที่เป็นCARsคือกฎที่อยู่ในรูปที่ 2.9(ข)

IF	A	then	B, C
IF	B	then	A, C
IF	C	then	A, B
IF	A, B	then	C
IF	A, C	then	B
IF	B, C	then	A

(ก) กฎความสัมพันธ์ทั้งหมด

IF	A, B	then	C
IF	A, C	then	B
IF	B, C	then	A

(ข) กฎความสัมพันธ์ที่ใช้เพื่อการจำแนกประเภทข้อมูล

รูปที่ 2.9 ตัวอย่างการสร้างกฎจำแนกประเภทข้อมูลอย่างง่าย

จะเห็นได้ว่าวิธีการนี้เป็นวิธีการที่ง่ายที่สามารถสร้างกฎ CARs ได้อย่างรวดเร็ว แต่ยังมีวิธีการอีกมากที่ใช้ในการสร้างกฎ CARsซึ่งงานวิทยานิพนธ์เล่มนี้ได้นำแนวคิดของอัลกอริทึม OneR(Holte, 1993) มาประยุกต์ใช้ในขั้นตอนการเลือกกฎความสัมพันธ์แบบคลุมเครือ

โดยแนวคิดและหลักการพื้นฐานของอัลกอริทึม OneRคือ การเลือกแอททริบิวต์ที่มีค่าความผิดพลาดในการทำนายข้อมูลน้อยที่สุดหนึ่งแอททริบิวต์(One-Attribute-Rule)เพื่อมาใช้เป็นโมเดลในการทำนายผล ยกตัวอย่างเช่น ตารางที่ 2.3ข้อมูลสภาพอากาศ ซึ่งเป็นข้อมูลที่นำไปใช้ในการแนะนำผู้เล่นกอล์ฟในการไปตีกอล์ฟว่าควรไปตีกอล์ฟหรือไม่โดยใช้สภาพอากาศเป็นปัจจัย จากข้อมูลจะเห็นได้ว่ามี 5 แอททริบิวต์คือ Outlook, Temperature, Humidity, Windy และ Play โดยที่แอททริบิวต์Play เป็นคลาสเป้าหมาย ในส่วนวิธีการของอัลกอริทึม OneRจะทำการนับค่าความผิดพลาดของแต่ละแอททริบิวต์ดังตารางที่ 2.4

ตารางที่ 2.3ข้อมูลสภาพอากาศ

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

ตารางที่ 2.4การนับค่าความผิดพลาดด้วยอัลกอริทึม OneR

Attribute	Rule	Error	Total Error
Outlook	Sunny ⟶ No	2/5	4/14
	Overcast ⟶ Yes	0/4	
	Rainy ⟶ Yes	2/5	
Temperature	Hot ⟶ No	2/4	5/14
	Mild ⟶ Yes	2/6	
	Cool ⟶ Yes	1/4	
Humidity	High ⟶ No	3/7	4/14
	Normal ⟶ Yes	1/7	
Windy	False ⟶ Yes	2/8	5/14
	True ⟶No	3/6	

จากตารางที่ 2.4 จะเห็นได้ว่าแอททริบิวต์Outlook และ Humidity มีค่าความผิดพลาดรวมเท่ากันคือ 4/14 ซึ่งเป็นค่าความผิดพลาดน้อยที่สุด จากหลักการของ OneRสามารถเลือกแอททริบิวต์ใดก็ได้จาก 2 แอททริบิวต์ดังกล่าวมาเป็นโมเดล ซึ่งวิทยานิพนธ์นี้ได้เลือกแอททริบิวต์Outlook เพื่อใช้แสดงเป็นโมเดลตัวอย่างโดยจะได้ 3 กฎ ดังนี้

IF Outlook = Sunny then No

IF Outlook = Overcastthen Yes

IF Outlook = Rainythen Yes

2.4 การจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือ

การสร้างกฎการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือหรือ FCARs (Fuzzy Classification with Association Rules)โดยทั่วไปแล้วจะประกอบไปด้วย 3 ขั้นตอนหลัก (Pachet al.,2008; Alcala-Fdezet al., 2011; Fazzolariet al., 2013) (รูปที่ 2.10) คือ 1.การแบ่งแยกตามระดับความเป็นสมาชิกข้อมูลหรือการจัดกลุ่ม 2. การสร้างไอเท็มเซตปรากฏบ่อยแบบคลุมเครือ 3. การสร้างกฎการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือซึ่งรายละเอียดของแต่ละขั้นตอนอธิบายได้ดังนี้

2.4.1 การแบ่งแยกข้อมูลตามระดับความเป็นสมาชิก

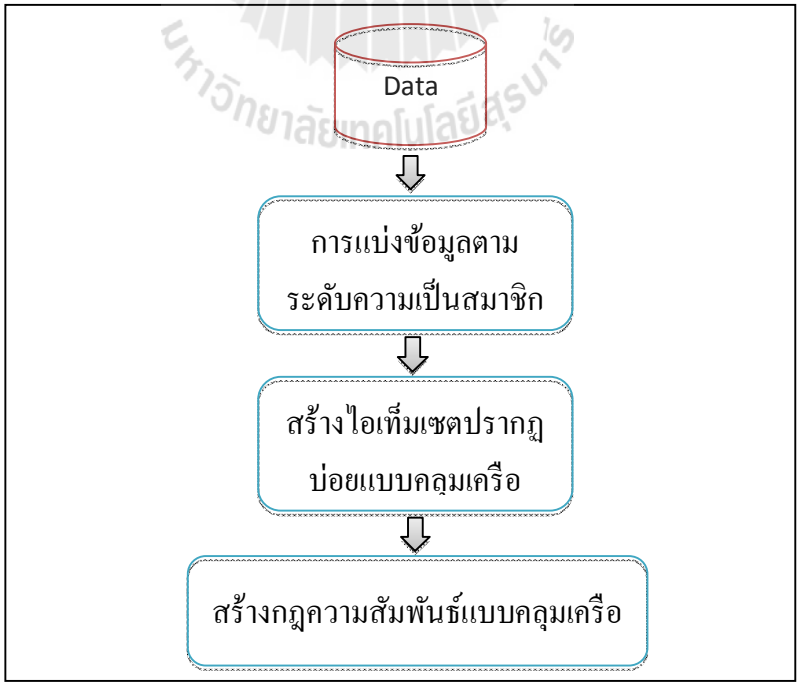
แนวคิดพื้นฐานของขั้นตอนนี้ คือ การนำทฤษฎีของฟัซซีเข้ามาใช้ในการประมวลผลในกรณีของการทำเหมืองข้อมูลแบบกฎความสัมพันธ์ที่เป็นข้อมูลตัวเลขต่อเนื่อง(Ishibuchiet al., 2001)โดยเทคนิคของฟัซซีจะทำการแปลงแอททริบิวต์ที่เป็นตัวเลขให้อยู่ในรูปแบบของค่าของตัวแปรเชิงภาษาเช่น แอททริบิวต์อายุ แปลงให้อยู่ในรูปแบบค่าของตัวแปรเชิงภาษาเป็นอายุน้อย อายุปานกลาง และอายุมากเป็นต้น และข้อมูลแต่ละแถวก็จะถูกแปลงให้อยู่ในระดับความเป็นสมาชิกของเซตนั้น ๆ ด้วยตัวอย่างเช่น ตารางที่ 2.5ทำการแปลงแอททริบิวต์Age ให้อยู่ในรูปแบบค่าของตัวแปรเชิงภาษาAge = Low, Age = Medium และ Age = High ดังตารางที่ 2.6 (การกำหนดจำนวนของระดับของฟัซซีขึ้นอยู่กับผู้ใช้ ซึ่งจากตัวอย่างนี้ได้กำหนดค่าของตัวแปรเชิงภาษาเป็น 3 ระดับ)และข้อมูลแต่ละแถวหมายถึงค่าความเป็นสมาชิกของค่าตัวแปรเชิงภาษานั้น ๆ เช่น คอลัมน์ Age = Low ของข้อมูลแถวแรกคือ 0.9863 ตัวเลขนี้หมายถึงระดับความเป็นสมาชิกของค่าตัวแปรเชิงภาษาAge = Low ซึ่งมีค่าความเป็นสมาชิกสูงกว่า Age = Medium และ Age = High ซึ่งมีค่าสมาชิกเป็น 0.0127 และ 0.001 ตามลำดับ

ตารางที่ 2.5ข้อมูลตัวอย่างก่อนทำการแบ่งแยกระดับความเป็นสมาชิกด้วย FCM

Id	Age
1	18
2	20
3	19
4	24
5	25

ตารางที่ 2.6ข้อมูลตัวอย่างหลังทำการแบ่งแยกระดับความเป็นสมาชิกด้วย FCM

Id	Age		
	Age = Low	Age = Medium	Age = High
1	0.9863	0.0127	0.001
2	0.0119	0.9862	0.0019
3	0.4996	0.49	0.0104
4	0.0074	0.0141	0.9785
5	0.0053	0.009	0.9857



รูปที่ 2.10ขั้นตอนการสร้างกฎการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือ

2.4.2 การสร้างไอเท็มเซตปรากฏบ่อยแบบคลุมเครือ

วิธีการสร้างไอเท็มเซตปรากฏบ่อยแบบคลุมเครือและกฎการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือ ในงานวิทยานิพนธ์นี้ได้ใช้วิธีการสร้างแบบเดียวกันกับอัลกอริทึม CFARC (Pachet al., 2008)ซึ่งมีพื้นฐานมาจากอัลกอริทึม Apriori โดยแสดงขั้นตอนการสร้างดังรูปที่ 2.11ซึ่งสามารถอธิบายได้ ดังนี้ บรรทัดที่ 1-2คือการกำหนดค่าสนับสนุนแบบคลุมเครือขั้นต่ำ (Minimum Fuzzy Support: γ)คำนวณได้จากการกระจายของคลาสเป้าหมายที่มีค่าน้อยที่สุด จะได้ดังสมการที่ 2.7โดยที่ min(Class) คือ แอททริบิวต์คลาสที่มีค่าจำนวนแถวน้อยที่สุด ส่วน N คือ แถวทั้งหมด บรรทัดที่ 3 ทำการสร้างไอเท็มเซตคู่แข่งแบบคลุมเครือ(Fuzzy Candidate Itemsets)ที่มีขนาด 1 ไอเท็มเซตบรรทัดที่ 4 ทำการนับค่าสนับสนุนแบบคลุมเครือ (Fuzzy Support: FS)ด้วยสมการที่ 2.6และทำการเลือกไอเท็มเซตปรากฏบ่อยแบบคลุมเครือ โดยที่ไอเท็มเซตคู่แข่งแบบคลุมเครือใด ๆ ที่จะเป็นไอเท็มเซตปรากฏบ่อยแบบคลุมเครือ จะต้องมีค่าสนับสนุนแบบคลุมเครือมากกว่าค่าสนับสนุนแบบคลุมเครือขั้นต่ำ(สมการที่ 2.7)บรรทัดที่ 5 สร้างไอเท็มเซตคู่แข่งแบบคลุมเครือที่มีขนาด n-1 ไอเท็ม(โดยใช้หลักการจับคู่ ดังสมการที่ 2.8) ต่อจากนั้นทำการเลือกไอเท็มเซตแบบคลุมเครือที่มีค่าสนับสนุนแบบคลุมเครือมากกว่าค่าสนับสนุนแบบคลุมเครือขั้นต่ำ

$$\text{FS}(<Z: A>) = \frac{\sum_{k=1}^{N} \prod_{<z_i:A_{i,j}>\in<Z:A>} t_k(z_i)}{\text{N}} \quad (2.6)$$

$$\gamma = \frac{(min(Class)/N)}{2} \quad (2.7)$$

$$\text{n-CandidateFuzzy Itemsets} = Number\_of(A \cap B) = n - 2 \text{ เมื่อ } n>2 \quad (2.8)$$

เมื่อ <Z: A>คือ ฟัซซีไอเท็มเซต (Fuzzy Item-Set) โดยที่ <Z: A>=[$<z_{i1}: A_{i,j}>$U$<z_{i2}: A_{i,j}>$U...U$<z_{iq}: A_{i_q,j}>$] เมื่อ $\text{q} \leq n + 1$

z_iคือตัวแปรเชิงภาษาเช่น Age, Income และ Balance เป็นต้น

$A_{i,j}$คือค่าของตัวแปรเชิงภาษาเช่น Low, Medium และ High เป็นต้น

t_k คือ ทรานแซคชันที่ k

N คือ จำนวนทรานแซคชันทั้งหมด

nคือ ขนาดของเซต

แสดงตัวอย่างการคำนวณค่าสนับสนุนของไอเท็มเซตแบบคลุมเครือได้ดังนี้ สมมติว่าต้องการหาค่าสนับสนุนของไอเท็มเซตแบบคลุมเครือ{Age = Medium, Income = Medium} โดยใช้ข้อมูลจากตารางที่ 2.7 จะได้ว่า

FS({Age = Medium, Income = Medium})=((0.0127*0)+(0.9862*0.9769)+(0.49*0.9773)+(0.0141*0.1849)+(0.009*0.0329))/5

FS({Age = Medium, Income = Medium})= 0.29

ตารางที่ 2.7 ข้อมูลตัวอย่างระดับความเป็นสมาชิกของ Age = Mediumและ Income = Medium

		class	
Age = Medium	**Income = Medium**	**yes**	**no**
0.0127	0	0	1
0.9862	0.9769	1	0
0.4900	0.9773	0	1
0.0141	0.1849	1	0
0.009	0.0329	1	0

Frequent fuzzy item set searching (an Apriori fuzzy implementation)

Input: DF fuzzy data

Output: the set of frequent fuzzy item set

Method:

1. Determine the supports of the classes by the distribution of classes;
2. Set the minimal fuzzy support (γ) to the half of the minimum frequency of classes;
3. Generate the 1- candidate fuzzy items;
4. Calculate FS values then select the frequent fuzzy items from the 1- candidate which has FS >γ, and n = 2;
5. While there existsome n-1 size frequent item sets:

   Generate the n–size candidate sets from n-1size frequents (and 1-size frequents);

รูปที่ 2.11การสร้างไอเท็มเซตปรากฏบ่อยแบบคลุมเครือ(Pachet al., 2008)

2.4.3 การสร้างกฎความสัมพันธ์แบบคลุมเครือ

วิธีการสร้างกฎความสัมพันธ์แบบคลุมเครือหรือFCARsจะคล้ายกันกับหัวข้อที่ 2.3.1 แตกต่างกันที่ทรานแซคชันที่นำมาประมวลผลจะต้องมีคลาสเป้าหมายเดียวกัน โดยวิธีการส่วนใหญ่จะเลือกกฎFCARs ที่มีค่าความเชื่อมั่นแบบคุมเครือ(Fuzzy Confidence: FC) มากกว่าค่าความเชื่อมั่นแบบคุมเครือขั้นต่ำเป็นเกณฑ์ในการสร้างกฎ FCARs (Bayardo et al., 1999; Hühn et al, 2009)ซึ่งสามารถคำนวณค่าความเชื่อมั่นได้จากสมการที่ 2.9(Pachet al., 2008)

$$FC(<X:A>\longrightarrow<Y:B>)=\frac{FS(<X:A>\cup<Y:B>)}{FS(<X:A>)} \quad (2.9)$$

เมื่อ <X:A> คือ ไอเท็มเซตแบบคลุมเครือที่อยู่ทางซ้ายมือของกฎ และ <Y:B> ไอเท็มเซตแบบคลุมเครือที่อยู่ทางขวามือของกฎซึ่งจากนิยามข้างต้นสามารถแสดงตัวอย่างการคำนวณได้ดังนี้ สมมติว่าต้องการหาค่า Fuzzy Confidence ของกฎ (Age=me, Inc=me) =>(Class = yes)โดยที่ใช้ข้อมูลจากตารางที่ 2.7

FC((Age=me, Inc=me) => Class = yes) =

$$\frac{FS(\ (0.9862*0.9769*1 + 0.0141*0.1849*1 + 0.009*0.0329*1)/5\ )}{(\ (0.0127*0) + (0.9862*0.9769) + (0.49*0.9773) + (0.0141*0.1849) + (0.009*0.0329)\ )/5}$$

= 0.66

เกณฑ์อื่น ๆ ที่นิยมใช้ในการสร้างกฎ FCARs(Pachet al., 2008) คือ FCORR หรือ Fuzzy Correlation(สมการที่2.10) เป็นเกณฑ์ที่ใช้บอกถึงระดับความสัมพันธ์ระหว่างแอททริบิวต์ที่อยู่ข้างหน้า then และแอททริบิวต์ที่อยู่ข้างหลัง then ของกฎ มีค่าอยู่ในช่วง [-1, 1] ซึ่งค่าบวกหมายถึงมีความสัมพันธ์ไปในทิศทางเดียวกัน ส่วนค่าลบหมายถึงมีความสัมพันธ์ตรงกันข้ามและศูนย์หมายถึงไม่มีความสัมพันธ์กันและ βหรือFiring Strength(สมการที่2.11) เป็นเกณฑ์ที่ใช้บ่งบอกถึงกำลังของกฎหรืออาจจะเรียกว่าค่าสนับสนุนของกฎ ซึ่งจะคล้ายกันกับเกณฑ์ Support

$$FCORR(<X:A>\longrightarrow<Y:B>) = \frac{FS(<X:A>\cup<Y:B>)-FS(<X:A>)\times FS(<Y:B>)}{\sqrt{FS(<X:A>)\times(1-FS(<X:A>))\times FS(<Y:B>)\times(1-FS(<Y:B>))}} \quad (2.10)$$

$$\beta(<X:A>\longrightarrow<Y:B>)=\sum_{k=1}^{N}\prod_{<z_i:A_{i,j}>\in<Z:A>} t_k(z_i) \quad (2.11)$$

เมื่อ X คือ แอททริบิวต์ที่มีค่าเป็น A

Y คือ แอททริบิวต์คลาสเป้าหมายที่มีค่าเป็น B

t_kคือ ทรานแซคชันที่ k

โดยแสดงรายละเอียดขั้นตอนการสร้างดังรูปที่ 2.12ซึ่งบรรทัดที่ 1 ทำการสร้างกฎความสัมพันธ์แบบคลุมเครือด้วยไอเท็มเซตปรากฏบ่อยแบบคลุมเครือที่มีคลาสเป้าหมายอยู่ทางขวามือของกฎ บรรทัดที่ 2 คำนวณค่าคะแนน (Score) ซึ่งหมายถึงค่าที่ได้จากสมการที่ 2.9หรือ2.10หรือ2.11ส่วนบรรทัดที่ 3 ทำการเลือกกฎความสัมพันธ์แบบคลุมเครือที่มีค่าคะแนนเป็นค่าบวกเพื่อสร้างเป็นกฎ FCARs(เงื่อนไขบรรทัดนี้เหมาะกับการใช้สมการที่ 2.10)

Fuzzy classification association rule (FCAR) generation

Input: a set of frequent fuzzy item sets

Output: positive correlated FCARs separated by size

Method:

1. Generate association rules with class label consequent from all the frequent item sets to consider the size of item sets
2. Calculate the Score values of all the rules;
3. Select rules with positive Score value for all size;

รูปที่ 2.12การสร้างกฎความสัมพันธ์แบบคลุมเครือ (Pachet al., 2008)

2.5.1 เกณฑ์ที่ใช้ในการวัดประสิทธิภาพของโมเดล

เกณฑ์ที่ใช้ประเมินผลในงานวิจัยนี้มีทั้งหมด 3 เกณฑ์ คือเกณฑ์ความถูกต้อง (Accuracy)เกณฑ์ความกะทัดรัดของกฎแบบปกติ(Normalized of Compact Value) และ เกณฑ์ความเหมาะสมของกฎ(Suitability of Rules)เกณฑ์ความถูกต้องเป็นเกณฑ์พื้นฐานทั่วไปที่นิยมใช้ในงานทำเหมืองข้อมูล ในงานวิทยานิพนธ์นี้ต้องการมาตรวัดที่มีความเจาะจงมากขึ้น เพื่อให้สามารถวัดความกะทัดรัดและความเหมาะสมต่อการนำไปใช้งานของโมเดล จึงได้เสนอเกณฑ์เพิ่มเติมอีก 2 เกณฑ์ คือ เกณฑ์ความกะทัดรัดของกฎแบบปกติ และเกณฑ์ความเหมาะสมของกฎ ซึ่งแต่ละเกณฑ์มีความหมายและความสำคัญดังต่อไปนี้

2.5.1 เกณฑ์ความถูกต้อง (Accuracy: Acc)

เกณฑ์ความถูกต้องเป็นเกณฑ์ที่ใช้บ่งบอกถึงระดับความความถูกต้องในการจำแนกประเภทข้อมูลของโมเดลที่ได้จากการประมวลผล ซึ่งคำนวณได้จากสมการที่ (2.12)

$$\text{Acc} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \quad (2.12)$$

เมื่อ True Positive: TP คือ จำนวนข้อมูลที่ทำนายถูกในเชิงบวกยกตัวอย่างเช่น การทำนายผู้ป่วยคนหนึ่งว่าเป็นโรคมะเร็งแล้วผลการตรวจสอบบอกว่าเป็นโรคมะเร็งจริง

True Negative: TN คือ จำนวนข้อมูลที่ทำนายถูกในเชิงลบยกตัวอย่างเช่น การทำนายผู้ป่วยคนหนึ่งว่าไม่เป็นโรคมะเร็งแล้วผลการตรวจสอบบอกว่าไม่เป็นโรคมะเร็งจริง

False Positive: FPคือ จำนวนข้อมูลที่ทำนายผิดในเชิงบวก ยกตัวอย่างเช่น การทำนายผู้ป่วยคนหนึ่งว่าเป็นโรคมะเร็งแล้วผลการตรวจสอบบอกว่าไม่เป็นโรคมะเร็ง

False Negative: FNคือ จำนวนข้อมูลที่ทำนายผิดในเชิงลบยกตัวอย่างเช่น การทำนายผู้ป่วยคนหนึ่งว่าไม่เป็นโรคมะเร็งแล้วผลการตรวจสอบบอกว่าเป็นโรคมะเร็ง

2.5.2 เกณฑ์ความกะทัดรัดของกฎแบบปกติ (Normalized of Compact Value: NCV)

เกณฑ์ความกะทัดรัดของกฎแบบปกติเป็นเกณฑ์ที่ใช้บ่งบอกความสัมพันธ์ของระดับความกะทัดรัดของกฎกับวิธีการอื่นว่าอยู่ระดับใด ซึ่งหาได้จากค่า Compact Value: CV เป็นค่าที่ได้จากการนับจำนวนกฎที่ได้จากการประมวลผล โดยเกณฑ์ความกะทัดรัดของกฎแบบปกติคำนวณได้จากสมการที่ (2.13)

$$\text{NCV}_{\text{AL}} = \frac{\text{Avg(CV)}_{\max} - \text{Avg(CV)}_{\text{AL}}}{\text{Avg(CV)}_{\max} - \text{Avg(CV)}_{\min}} \quad (2.13)$$

โดยที่ NCV_{AL}คือ ความกะทัดรัดของกฎแบบปกติของอัลกอริทึม AL ที่สนใจ $\text{Avg(CV)}_{\max}$ คือ ค่าเฉลี่ยของ CV ที่มีค่ามากที่สุด

$\text{Avg(CV)}_{\min}$ คือ ค่าเฉลี่ยของ CV ที่มีค่าน้อยที่สุด

$\text{Avg(CV)}_{\text{AL}}$ คือ ค่าเฉลี่ยของ CV ของอัลกอริทึมที่สนใจ

2.5.3 เกณฑ์ความเหมาะสมของกฎ (Suitability of Rules: SR)

เกณฑ์ความเหมาะสมของกฎเป็นเกณฑ์ที่ใช้บ่งบอกระดับความเหมาะสมของโมเดลที่ได้ว่ามีความถูกต้องและจำนวนกฎที่ได้มีความเหมาะสมเพียงใด โดยที่มีค่าอยู่ในช่วง[0, 1] ถ้าเป็นค่า 0 แสดงว่าโมเดลที่ได้ไม่เหมาะสมซึ่งมีค่าความถูกต้องน้อยและมีจำนวนกฎที่มาก แต่ถ้ามีค่าเป็น 1 แสดงว่าโมเดลที่ได้เหมาะสมมากซึ่งมีความหมายว่ามีค่าความถูกต้องมากและมีจำนวนกฎที่น้อยมาก การคำนวณค่าความเหมาะสมของกฎสามารถคำนวณได้จากสมการที่ (2.14)

$$SR_{AL} = \frac{Avg(Acc)_{AL} + NCV_{AL}}{2} \quad (2.14)$$

เมื่อ SR_{AL}คือ ค่าความเหมาะสมของกฎของอัลกอริทึม AL

$Avg(Acc)_{AL}$ คือ ค่าเฉลี่ย Accของอัลกอริทึม (AL) ที่สนใจ

2.6 งานวิจัยที่เกี่ยวข้อง

ในหัวข้อนี้จะประกอบด้วยงานวิจัยที่เกี่ยวข้อง 3 ประเภท คือ งานวิจัยที่เกี่ยวข้องกับการจำแนกประเภทข้อมูล งานวิจัยที่เกี่ยวข้องกับการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์และงานวิจัยที่เกี่ยวข้องกับการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือ ซึ่งสรุปได้ดังตารางที่ 2.8โดยมีรายละเอียดของงานวิจัยที่เกี่ยวข้องดังนี้

2.6.1 งานวิจัยที่เกี่ยวข้องกับการจำแนกประเภทข้อมูลโดยใช้วิธีแบบดั้งเดิม

Quinlan (1992)เสนออัลกอริทึม C4.5ซึ่งเป็นอัลกอริทึมที่รู้จักกันดีในการจำแนกประเภทข้อมูล เพราะเป็นอัลกอริทึมที่พัฒนามาจากอัลกอริทึม Id3 (Quinlan, 1986)ที่สามารถจัดการกับข้อมูลที่เป็นค่าต่อเนื่องได้ โดยใช้หลักการค้นหาแบบฮิวริสติกเพื่อสร้างต้นไม้ตัดสินใจ และใช้เทคนิคในการตัดกิ่งต้นไม้ซึ่งทำให้อัลกอริทึมนี้สามารถประมวลผลได้รวดเร็วและกฎที่ได้รับนั้นผู้ใช้สามารถทำความเข้าใจได้ง่าย

Cohen (1995)เสนออัลกอริทึมที่มีชื่อว่า RIPPER (Repeated Incremental Pruning to Produce Error Reduction)ที่พัฒนามาจากอัลกอริทึม IREP* โดยใช้หลักการ (Growing and Pruning) คือ การแบ่งกฎการจำแนกประเภทข้อมูลเป็นกลุ่มตามคลาสเป้าหมาย แล้วใช้หลักการตัดกิ่งเพื่อเลือกกฎที่มีประสิทธิภาพมากกว่าอัตราความผิดพลาด (Error Rate)ที่กำหนดมาใช้ในการทำนายข้อมูล

Holte (1993) เสนออัลกอริทึม OneRหรือ One-Rule เป็นอัลกอริทึมที่ทำความเข้าใจวิธีการสร้างกฎได้ง่ายและใช้งานง่าย โดยใช้หลักการสร้างกฎการจำแนกประเภทด้วยการเลือกแอ

ททริบิวต์ที่มีค่าความผิดพลาดน้อยที่สุดเพียงแอททริบิวต์เดียวมาเป็นตัวทำนายคลาสของข้อมูล ซึ่งทำให้ได้รับกฎการจำแนกประเภทข้อมูลที่มีจำนวนน้อยมาก

2.6.2 งานวิจัยที่เกี่ยวข้องกับการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์

Liu and Ma (1998)เสนออัลกอริทึมที่มีชื่อว่า CBA คืออัลกอริทึมที่ใช้ในการจำแนกข้อมูลและได้รับการทดสอบว่ามีความแม่นยำในการจำแนกที่สูงกว่าอัลกอริทึม C4.5 โดยอัลกอริทึม CBA เป็นอัลกอริทึมแรกที่นำแนวคิดของการทำเหมืองข้อมูลแบบกฎความสัมพันธ์และการทำเหมืองข้อมูลแบบจำแนกประเภทมาทำงานร่วมกัน โดยขั้นตอนการทำงานของ CBA จะประกอบด้วย 2 ส่วน คือ ส่วนที่สร้างกฎความสัมพันธ์ซึ่งเรียกว่า CBA-RG และส่วนที่สร้างกฎการจำแนกจากกฎความสัมพันธ์เรียกว่า CBA-CBซึ่งแนวคิดหลักของส่วนสร้างกฎความสัมพันธ์CBA-RG คือ สร้างกฎความสัมพันธ์ที่มีค่าสนับสนุนมากกว่าค่าสนับสนุนขั้นต่ำ และกฎความสัมพันธ์ที่ได้จะอยู่ในรูปแบบดังนี้<condset, y>ซึ่ง condset หมายถึง เซตของไอเท็ม และ y หมายถึงคลาสเป้าหมาย ส่วนแนวคิดของส่วนCBA-CB คือ ทำการคัดเลือกกฎความสัมพันธ์ที่มีค่าความผิดพลาดน้อยที่สุด

Chen and Zhang (2006)เสนออัลกอริทึมที่มีชื่อว่า GARC (Gain Based Association Rule Classification) เป็นอัลกอริทึมที่มีประสิทธิภาพในการจำแนกประเภทข้อมูลค่อนข้างสูง และกฎการจำแนกประเภทข้อมูลที่ได้รับจากการประมวลผลมีจำนวนน้อย เพราะว่าใช้เกณฑ์การได้ประโยชน์จากสารสนเทศ(Information Gain) ในการสร้างไอเท็มปรากกฎบ่อยและใช้วิธีการตัดกฎที่มีลักษณะซ้ำซ้อน (Redundancy) และขัดแย้งกัน (Conflict)

Hu and Li (2006)เสนออัลกอริทึมที่มีชื่อว่าOAC(Optimal Association Classifier)ที่มีประสิทธิภาพในการจำแนกข้อมูลที่ดี โดยใช้หลักการสร้างกฎความสัมพันธ์โดยอัลกอริทึม Aprioriและคัดเลือกกฎความสัมพันธ์ด้วยการใช้เงื่อนไขบังคับ (Bayardoet al., 1999)เพื่อทำการเลือกกฎความสัมพันธ์ที่มีคลาสเป้าหมายตามต้องการ จากนั้นทำการเลือกกฎที่เหมาะสมสำหรับการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์ด้วยวิธีการคัดเลือกกฎความสัมพันธ์ที่ให้ความแม่นยำสูงที่สุด(Optimal Class Association Rule Mining)(Li et al., 2002)

2.6.3งานวิจัยที่เกี่ยวข้องกับการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือ

Pachet al. (2008) เสนออัลกอริทึม CFARC (Compact Fuzzy Association Rule-Based Classifier) เพื่อจัดการกับปัญหา กฎการจำแนกประเภทข้อมูลที่มีจำนวนมากเกินไป ปัญหาการกำหนดค่าสนับสนุนขั้นต่ำและค่าความเชื่อมั่นขั้นต่ำที่เหมาะสมในการสร้างกฎความสัมพันธ์ และปัญหาการจัดการกับข้อมูลตัวเลขต่อเนื่อง ซึ่งวิธีการที่ใช้ในการแก้ปัญหาลดจำนวนกฎการจำแนกประเภทข้อมูลคือการทำการตัดกิ่งกฎที่มีค่า FCORR ที่มีค่าลบทิ้ง แล้วทำการเลือกกฎที่มีค่าคะแนนสูงมาใช้ในการทำนายข้อมูลในส่วนของการทดลองนั้นได้แสดงให้เห็นถึงผลลัพธ์ที่ได้เป็น

จำนวนกฎ FCARs ที่มีขนาดเล็กและจำนวนไม่มาก และยังมีความแม่นยำในการจำแนกประเภทข้อมูลอยู่ในเกณฑ์ที่ดี

Chen (2008) เสนออัลกอริทึม CFAR (Classification with Fuzzy Association Rules) อัลกอริทึมนี้สร้างกฎด้วยอัลกอริทึมAprioriแล้วทำการลบกฎ FCARs ที่มีลักษณะที่ซ้ำซ้อนและขัดแย้งกัน ด้วย 2 วิธีการคือ การเลือกกฎที่มีค่าความเชื่อมั่นสูงที่สุดไปใช้ในการจำแนกและเก็บกฎที่เหลือไว้ และการลบกฎที่เหลือที่มีค่าความเชื่อมั่นน้อยที่สุดแล้วกลับไปทำวิธีที่หนึ่งใหม่จนกระทั่งค่าความผิดพลาดมีค่าเพิ่มขึ้น

Hühn and Hüllermeier(2009)เสนออัลกอริทึม FURIA: An Algorithm For Unordered Fuzzy Ruleเป็นอัลกอริทึมที่มีพื้นฐานมาจากอัลกอริทึม RIPPERแต่ในอัลกอริทึมนี้จะใช้วิธีการไม่เรียงลำดับกฎ(Unordered Rule Set)เพื่อลดการลำเอียงของคลาสเป้าหมายหลัก แทนการเรียงลำดับกฎแบบเดิม ทำให้เกิดปัญหาขึ้น 2 ปัญหาคือกฎมีความขัดแย้งกัน และกฎที่ได้รับมีลักษณะไม่ครอบคลุม ซึ่งแก้ปัญหาด้วยวิธีการ Stretching เป็นวิธีการเลือกกฎที่มีค่าการครอบคลุมมากที่สุดและกฎที่ได้นั้นเรียกว่า Generalized Rules

จากการศึกษางานวิจัยที่เกี่ยวข้องพบว่ามีงานวิจัยจำนวนมากที่มุ่งเน้นในการพัฒนาขั้นตอนวิธีการ เพื่อให้โมเดลที่ได้รับจากการประมวลผลมีความแม่นยำในการจำแนกประเภทข้อมูลสูงและสามารถตีความหมายได้ดี ดังนั้นวิทยานิพนธ์นี้จึงได้เสนอการพัฒนาขั้นตอนวิธีเพื่อจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือที่กะทัดรัด เพื่อให้ได้ผลลัพธ์ดังกล่าวโดยการนำเทคนิคการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์ผสมผสานกับเทคนิคฟัซซีเซต และทำการเลือกกฎความสัมพันธ์ที่มาใช้ในการทำนายด้วยเทคนิคของ OneR(Holte, 1993) เพื่อทำให้กฎที่ได้รับมีจำนวนน้อยและครอบคลุมค่าที่เป็นไปได้ทั้งหมด

ตารางที่ 2.8สรุปเปรียบเทียบงานวิจัยที่เกี่ยวข้องกับการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์ที่มีขนาดกะทัดรัด

กระบวนการทำงาน	งานวิจัยที่เกี่ยวข้อง*									
	CLASS			CAR			FCAR			
	ก	ข	ค	ง	จ	ฉ	ช	ซ	ฌ	ญ*
จุดประสงค์ของการวิจัย										
พัฒนาความแม่นยำในการจำแนกประเภทข้อมูล	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ลดจำนวนกฎการจำแนกประเภท			✓		✓		✓	✓	✓	✓
ลดปัญหาการกำหนดค่าพารามิเตอร์ของเกณฑ์ต่าง ๆ			✓		✓		✓			✓
วิธีการจำแนกประเภทข้อมูลและเทคนิคที่ใช้										
จำแนกด้วยเกณฑ์ Information Gain	✓			✓	✓					
จำแนกด้วยเกณฑ์ Correlation							✓			✓
จำแนกด้วยเกณฑ์ Confidence				✓	✓	✓		✓		
จำแนกประเภทข้อมูลด้วยการไม่กำหนดค่าพารามิเตอร์ใด ๆ			✓		✓		✓			✓
ใช้อัลกอริทึม Apriori เป็นพื้นฐานในการสร้างกฎความสัมพันธ์				✓	✓	✓	✓	✓		✓
ใช้เทคนิค Growing and Pruning เพื่อสร้างกฎการจำแนกประเภทข้อมูล		✓							✓	
ข้อมูลที่ใช้ในการทดลอบ										
ข้อมูลสังเคราะห์										✓
ข้อมูลจริง	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

*งานวิจัยที่เกี่ยวข้องประกอบด้วย

CLASSแทนกลุ่มงานวิจัยที่เกี่ยวข้องกับการจำแนกประเภทข้อมูลประกอบด้วย

ก แทนงานวิจัยของQuinlan (1992) (อัลกอริทึม C4.5)

ข แทนงานวิจัยของ Cohen (1995) (อัลกอริทึม RIPPER)

ค แทนงานวิจัยของHolte (1993) (อัลกอริทึม OneR)

CAR แทนกลุ่มงานวิจัยที่เกี่ยวข้องกับการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์ประกอบด้วย

ง แทนงานวิจัยของ Liuet al. (1998)(อัลกอริทึม CBA)

จ แทนงานวิจัยของ Chen et al. (2006)(อัลกอริทึม GARC)

ฉ แทนงานวิจัยของ Hu and Li (2006)(อัลกอริทึม OAC)

FCAR แทนกลุ่มงานวิจัยที่เกี่ยวข้องกับการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือประกอบด้วย

ช แทนงานวิจัยของPachet al. (2008) (อัลกอริทึม CFARC)

ซ แทนงานวิจัยของChen et al. (2008) (อัลกอริทึม CFAR)

ฌ แทนงานวิจัยของHühnandHüllermeier.(2009) (อัลกอริทึม FURIA)

ญ*แทนงานวิจัยของวิทยานิพนธ์ฉบับนี้

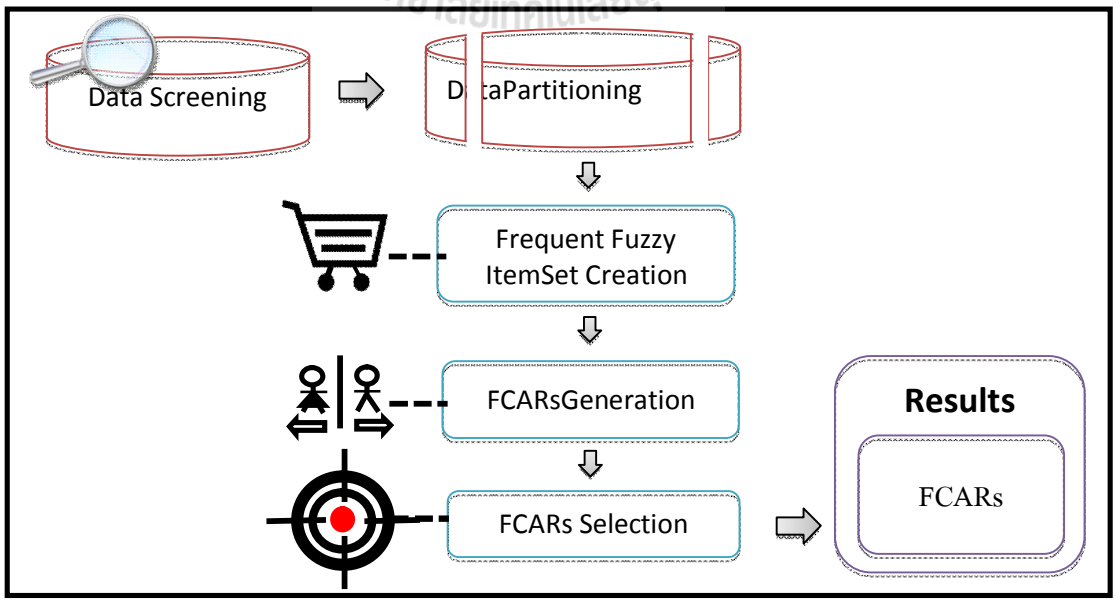
บทที่ 3

วิธีดำเนินการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาอัลกอริทึมที่ใช้ในการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือให้มีความถูกต้องในการจำแนกอยู่ในเกณฑ์ดีและจำนวนกฎที่ใช้ในการทำนายข้อมูลมีขนาดเล็กแต่ครอบคลุมทุกค่าของแอททริบิวต์ในบทนี้จะกล่าวถึง วิธีการวิจัยและกระบวนการต่าง ๆ ของการวิจัย โดยมีรายละเอียดดังนี้

3.1 กรอบแนวคิดของอัลกอริทึม Classification with Compact Fuzzy Association Rules (CCFAR)

แนวคิดหลักของงานวิจัยนี้คือ การสร้างกฎการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือที่มีจำนวนไม่มาก สามารถตีความได้ง่ายและมีประสิทธิภาพในการจำแนกประเภทข้อมูลสูง ซึ่งกรอบแนวคิดของงานวิจัยนี้ประกอบด้วยขั้นตอนการทำงาน 5 ส่วนคือ1. การตรวจสอบข้อมูลก่อนการประมวลผล2. การแบ่งแยกข้อมูล 3. การสร้างไอเท็มเซตปรากฏบ่อยแบบคลุมเครือ 4. การสร้างกฎ FCARs และ 5. การเลือกกฎ FCARs เพื่อนำไปใช้ทำนายข้อมูลโดยที่ส่วนที่ 2. ถึงส่วนที่ 4. ได้ใช้อัลกอริทึม CFARC (Pach et al., 2008) เป็นพื้นฐานในการพัฒนา ซึ่งแสดงขั้นตอนทั้งหมดนี้ดังรูปที่ 3.1



รูปที่ 3.1กรอบแนวคิดอัลกอริทึม CCFAR

3.1.1 การตรวจสอบข้อมูลก่อนการประมวลผล(DataScreening)

ข้อมูลที่นำเข้ามาประมวลผลจะต้องเป็นข้อมูลตัวเลข เนื่องจากว่าในการแปลงข้อมูลให้อยู่ในรูปแบบของฟัซซีเซตนั้น ฟัซซีจะทำการแปลงข้อมูลตัวเลขให้เป็นระดับของความเป็นสมาชิกของเซตนั้น ๆส่วนข้อมูลสูญหายหรือ Missing Value ข้อมูลลักษณะดังกล่าวระบบจะไม่สามารถประมวลผลได้จำเป็นจะต้องเข้าสู่กระบวนการเตรียมข้อมูลก่อน (Pre-Processing) ซึ่งข้อมูลที่จะสามารถนำไปประมวลผลได้นั้น จะต้องประกอบด้วยแอททริบิวต์เป้าหมาย (Class Target) โดยจะอยู่คอลัมน์สุดท้ายของตาราง แสดงตัวอย่างดังตารางที่ 3.1 จากตารางจะประกอบด้วยข้อมูล 5 แอททริบิวต์และ 5 แถว ในการประมวลผลจะใช้ข้อมูลเพียง 4 แอททริบิวต์ คือ Age, Income, Balanceและ Classซึ่งเป็นแอททริบิวต์เป้าหมาย แต่จะไม่ใช้แอททริบิวต์Id เพราะว่าแอททริบิวต์ดังกล่าวเป็นข้อมูลที่บ่งบอกถึงหมายเลขแถวเท่านั้น

ตารางที่ 3.1 ข้อมูลตัวอย่าง

Id	Age	Income	Balance	Class
1	18	10000	4000	2
2	20	18000	20000	1
3	19	17000	5000	2
4	24	20000	10000	1
5	25	22000	9000	1

3.1.2 การแบ่งแยกข้อมูล (Input Data Partitioning)

การแบ่งแยกข้อมูลเป็นการทำให้ข้อมูลที่นำเข้ามาประมวลผลนั้นมีลักษณะเป็นข้อมูลแบบฟัซซีเซต โดยงานวิจัยนี้ได้เลือกอัลกอริทึม FCM มาใช้เพื่อการแบ่งกลุ่มหรือแบ่งแยกข้อมูลให้มีลักษณะเป็นระดับของความเป็นสมาชิกในเซตต่าง ๆ ซึ่งจะช่วยแก้ปัญหาข้อมูลที่มีลักษณะเป็นตัวเลขค่าต่อเนื่องได้ โดยแสดงตัวอย่างการแบ่งแยกข้อมูลดังตารางที่ 3.2– 3.4ซึ่งใช้ข้อมูลจากตารางที่ 3.1 และได้กำหนดค่า k เท่ากับ 3 หรือ 3 ระดับคือ Low, Medium และ High

ตารางที่ 3.2 การแบ่งแยกข้อมูล Age ให้เป็น 3 ระดับ คือ Low, Medium และ High

Age		
Low	**Medium**	**High**
0.9863	0.0127	0.001
0.0119	0.9862	0.0019
0.4996	0.49	0.0104
0.0074	0.0141	0.9785
0.0053	0.009	0.9857

ตารางที่ 3.3การแบ่งแยกข้อมูล Income ให้เป็น 3 ระดับ คือ Low, Medium และ High

Income		
Low	**Medium**	**High**
1	0	0
0.0031	0.9769	0.0199
0.006	0.9773	0.0167
0.0111	0.1849	0.804
0.0045	0.0329	0.9626

ตารางที่ 3.4การแบ่งแยกข้อมูล Balance ให้เป็น 3 ระดับ คือ Low, Medium และ High

Balance		
Low	**Medium**	**High**
0.9909	0.0081	0.001
0	0	1
0.9866	0.0123	0.0011
0.0081	0.9894	0.0025
0.0122	0.9857	0.002

3.1.3 การสร้างไอเท็มเซตปรากฏบ่อยแบบคลุมเครือ (Frequent Fuzzy ItemSetCreation)

ขั้นตอนนี้จะใช้วิธีการสร้างแบบเดียวกันกับอัลกอริทึม CFARC (Pachet al., 2008) ซึ่งแสดงรายละเอียดในหัวข้อที่ 2.4.2 ประกอบด้วย 2 ขั้นตอน คือ การหาค่าสนับสนุนแบบคลุมเครือขั้นต่ำ และการค้นหาไอเท็มเซตปรากฏบ่อยแบบคลุมเครือที่มีค่าสนับสนุนแบบคลุมเครือมากกว่าค่าสนับสนุนแบบคลุมเครือขั้นต่ำ จาก 2 ขั้นตอนดังกล่าวนั้นสามารถแสดงตัวอย่างได้ดังนี้ โดยใช้ตัวอย่างจากตารางที่ 3.1 ทำการหาค่าสนับสนุนแบบคลุมเครือขั้นต่ำด้วยการแทนค่าในสมการที่ 2.7 จะได้ค่าสนับสนุนแบบคลุมเครือขั้นต่ำ คือ 0.2

$$\gamma = \frac{(2/5)}{2} = 0.2$$

จากนั้นทำการสร้างไอเท็มเซตคู่แข่งแบบคลุมเครือที่มีขนาด 1 ไอเท็มและทำการนับค่าสนับสนุนแบบคลุมเครือด้วยสมการที่ 2.6แสดงตัวอย่างการสร้างเซตปรากฏบ่อยแบบคลุมเครือโดยใช้ข้อมูลจากตารางที่ 3.2 -3.4 จะได้ดังตารางที่ 3.5

ตารางที่ 3.5ไอเท็มเซตคู่แข่งแบบคลุมเครือที่มีขนาด 1 ไอเท็ม

1-CandidateFuzzy Items	Fuzzy Support
Age=low	0.3021
Age=me	0.3024
Age=hi	0.3955
Inc=low	0.2050
Inc=me	0.4344
Inc=hi	0.3606
Bal=low	0.3996
Bal=me	0.3991
Bal=hi	0.2013

หลังจากนั้นทำการเลือกไอเท็มเซตแบบคลุมเครือที่มีค่าสนับสนุนแบบคลุมเครือมากกว่า 0.2ซึ่งจากตารางที่ 3.5 จะเห็นได้ว่าไอเท็มเซตแบบคลุมเครือทุกเซตมีค่าสนับสนุนแบบคลุมเครือมากกว่า 0.2 ดังนั้นไอเท็มเซตเหล่านี้จะเรียกว่า ไอเท็มเซตปรากฏบ่อยแบบคลุมเครือ ซึ่งจะ

นำไปสร้างไอเท็มเซตคู่แข่งแบบคลุมเครือที่มีขนาด 2 ไอเท็ม(โดยใช้หลักการจับคู่) จะได้ดังตารางที่ 3.6ต่อจากนั้นทำการเลือกไอเท็มเซตแบบคลุมเครือที่มีค่าสนับสนุนแบบคลุมเครือมากกว่า 0.2 ซึ่งจะได้ทั้งหมด 5 เซตดังตารางที่ 3.7แล้วทำการสร้างไอเท็มเซตแบบคลุมเครือขนาด 3 ไอเท็มโดยที่ไอเท็มเซตสองเซตจะทำการรวมกันได้จะต้องมีไอเท็มที่เหมือนกัน 1 ไอเท็มและใช้ข้อมูลจากตารางที่ 3.7 ผลที่ได้แสดงดังตารางที่ 3.8

ขั้นต่อไปทำการเลือกไอเท็มเซตแบบคลุมเครือที่มีค่าสนับสนุนแบบคลุมเครือมากกว่า 0.2 จะได้ดังตารางที่ 3.8 เช่นเดิม ทำให้ในการสร้างไอเท็มเซตแบบคลุมเครือที่มีขนาด 4 ไอเท็มเซตไม่สามารถสร้างได้เพราะเหลือไอเท็มเซตปรากฏบ่อยแบบคลุมเครือขนาด 3 ไอเท็มเพียงเซตเดียว

ตารางที่ 3.6ไอเท็มเซตคู่แข่งแบบคลุมเครือที่มีขนาด 2 ไอเท็ม

2-CandidateFuzzy Items	**Fuzzy Support**
Age=low, Inc=low	0.1979
Age=low, Inc=me	0.1003
Age=low, Inc=hi	0.0039
Age=me, Inc=low	0.0038
Age=me, Inc=me	0.2891
Age=me, Inc=hi	0.0096
Age=hi, Inc=low	0.0033
Age=hi, Inc=me	0.0451
Age=hi, Inc=hi	0.3472
Age=low, Bal=low	0.2941
Age=low, Bal=me	0.0053
Age=low, Bal=hi	0.0027
Age=me, Bal=low	0.0993
Age=me, Bal=me	0.0058
Age=me, Bal=hi	0.1974
Age=hi, Bal=low	0.0063

ตารางที่ 3.6ไอเท็มเซตคู่แข่งแบบคลุมเครือที่มีขนาด 2 ไอเท็ม (ต่อ)

2-CandidateFuzzy Items	Fuzzy Support
Age=hi, Bal=me	0.3880
Age=hi, Bal=hi	0.0013
Inc=low, Bal=low	0.1994
Inc=low, Bal=me	0.0047
Inc=low, Bal=hi	0.0008
Inc=me, Bal=low	0.1932
Inc=me, Bal=me	0.0455
Inc=me, Bal=hi	0.1957
Inc=hi, Bal=low	0.0070
Inc=hi, Bal=me	0.3489
Inc=hi, Bal=hi	0.0048

ตารางที่ 3.7ไอเท็มเซตปรากฏบ่อยแบบคลุมเครือที่มีขนาด 2 ไอเท็ม

2- Frequent Fuzzy Items	Fuzzy Support
Age=me, Inc=me	0.2891
Age=hi, Inc=hi	0.3472
Age=low, Bal=low	0.2941
Age=hi, Bal=me	0.3880
Inc=hi, Bal=me	0.3489

ตารางที่ 3.8ไอเท็มเซตคู่แข่งแบบคลุมเครือที่มีขนาด 3 ไอเท็ม

3-CandidateFuzzy Items	Fuzzy Support
Age=hi, Inc=hi, Bal=me	0.3427

ตารางที่ 3.9ไอเท็มเซตปรากฏบ่อยแบบคลุมเครือทั้งหมด

Frequent Fuzzy Items	Fuzzy Support
Age=low	0.3021
Age=me	0.3024
Age=hi	0.3955
Inc=low	0.2050
Inc=me	0.4344
Inc=hi	0.3606
Bal=low	0.3996
Bal=me	0.3991
Bal=hi	0.2013
Age=me, Inc=me	0.2891
Age=hi, Inc=hi	0.3472
Age=low, Bal=low	0.2941
Age=hi, Bal=me	0.3880
Inc=hi, Bal=me	0.3489
Age=hi, Inc=hi, Bal=me	0.3427

3.1.4 การสร้างกฎการจำแนกด้วยกฎความสัมพันธ์แบบคลุมเครือ(FCARs Generation: Fuzzy Classification Association Rule Generation)

จาก 3 ขั้นตอนที่ผ่านมา คือ การนำข้อมูลเข้า การแบ่งแยกข้อมูล และการสร้างไอเท็มเซตปรากฏบ่อยแบบคลุมเครือ ขั้นตอนที่ 4 นี้เป็นขั้นตอนที่นำเอาไอเท็มเซตปรากฏบ่อยแบบคลุมเครือทั้งหมดมาสร้างเป็นกฎความสัมพันธ์แบบคลุมเครือหรือFCARs มีขั้นตอนการสร้างดังรูปที่ 2.10โดยขั้นตอนนี้จะใช้วิธีการสร้างคล้ายกันกับอัลกอริทึม CFARC (Pachet al., 2008) ซึ่งจะสร้างกฎ FCARs จากไอเท็มเซตปรากฏบ่อยแบบคลุมเครือทั้งหมด (ตารางที่ 3.9) ด้วยคลาสเป้าหมาย (คลาส 1 และ 2)โดยที่กฎ FCARs จะต้องมีค่าคะแนน (Score) มากกว่า 0 และงานวิจัยนี้ได้ใช้สมการที่ 3.1 ในการคำนวณค่าคะแนนซึ่งจะแตกต่างจากวิธีของอัลกอริทึม CFARC (สมการคำนวณ FCORR, FC และFiring-strengthแสดงในบทที่ 2 สมการที่ 2.9 ถึง 2.11)

$$\text{Score} = \text{FCORR} \times \text{FC} \times \text{Firing\_strength} \quad (3.1)$$

จากตารางที่ 3.9 นำไปสร้างกฎ FCARs ด้วยการคำนวณคะแนนจะได้ดังตารางที่ 3.10 และทำการคัดเลือกกฎ FCARs ที่มีคะแนนมากกว่า 0 โดยกฎ FCARs ที่ถูกคัดเลือกทั้งหมดแสดงดังตารางที่ 3.11

ตารางที่ 3.10กฎ FCARs ทั้งหมดและคะแนน

FCARs	**Score**
Age=low -> 1	-3.1260e-04
Age=low -> 2	1.1459
Age=me -> 1	0.0611
Age=me -> 2	-0.0152
Age=hi -> 1	1.2725
Age=hi -> 2	-4.2833e-05
Inc=low -> 1	-2.0817e-04
Inc=low -> 2	0.5953
Inc=me -> 1	-0.0587
Inc=me -> 2	0.0393
Inc=hi -> 1	1.0602
Inc=hi -> 2	-9.2857e-05
Bal=low -> 1	-2.0367e-04
Bal=low -> 2	1.9225
Bal=me -> 1	1.2677
Bal=me -> 2	-1.3503e-04
Bal=hi -> 1	0.4088
Bal=hi -> 2	-1.7274e-06
Age=me, Inc=me -> 1	0.0577
Age=me, Inc=me ->2	-0.0142
Age=hi, Inc=hi -> 1	1.0330

ตารางที่ 3.10กฎ FCARs ทั้งหมดและคะแนน (ต่อ)

FCARs	Score
Age=hi, Inc=hi -> 2	-1.0426e-08
Age=low, Bal=low ->1	-8.3168e-09
Age=low, Bal=low -> 2	1.1619
Age=hi, Bal=me ->1	1.2608
Age=hi, Bal=me -> 2	-6.1730e-09
Inc=hi, Bal=me -> 1	1.0421
Inc=hi, Bal=me ->2	-1.4372e-08
Age=hi, Inc=hi, Bal=me ->1	1.0104
Age=hi, Inc=hi, Bal=me -> 2	-1.5702e-12

ตารางที่ 3.11กฎ FCARs ที่มีคะแนนมากกว่า 0

FCARs	Score
Age=low -> 2	1.1459
Age=me -> 1	0.0611
Age=hi -> 1	1.2725
Inc=low -> 2	0.5953
Inc=me -> 2	0.0393
Inc=hi -> 1	1.0602
Bal=low -> 2	1.9225
Bal=me -> 1	1.2677
Bal=hi -> 1	0.4088
Age=me, Inc=me -> 1	0.0577
Age=hi, Inc=hi -> 1	1.0330
Age=low, Bal=low -> 2	1.1619
Age=hi, Bal=me -> 1	1.2608
Inc=hi, Bal=me -> 1	1.0421
Age=hi, Inc=hi, Bal=me -> 1	1.0104

3.1.5 การเลือกกฎการจำแนกด้วยกฎความสัมพันธ์แบบคลุมเครือ (FCARs Selection: Fuzzy Classification Association Rule Selection)

ขั้นตอนที่ 5 นี้เป็นขั้นตอนการเลือกกฎ FCARs ไปใช้งานโดยมีวิธีการคัดเลือก 4 ขั้นตอนย่อยดังรูปที่ 3.2ซึ่งสามารถอธิบายได้ดังนี้

ขั้นตอนย่อยแรก บรรทัดที่ 1 เป็นการกำหนดค่าเริ่มต้นของตัวแปร k เพื่อใช้ในการเก็บจำนวนกฎที่อยู่ในบรรทัดที่ 5 ส่วนบรรทัดที่ 2 ทำการคัดเลือกกฎFCARs ที่มีค่าคะแนนมากที่สุดของแต่ละคลาสและแต่ละขนาดด้วยฟังก์ชัน find_top_rulesโดยจะใช้ตัวอย่างจากตารางที่ 3.11 ซึ่งจากวิธีการดังกล่าวจะทำให้ได้ผลลัพธ์ดังตารางที่ 3.12

ตารางที่ 3.12กฎFCARs ที่มีค่าคะแนนมากที่สุดของแต่ละคลาสและแต่ละขนาด

FCARs	**Score**
Age=hi -> 1	1.2725
Bal=low -> 2	1.9225
Age=low, Bal=low -> 2	1.1619
Age=hi, Bal=me -> 1	1.2608
Age=hi, Inc=hi, Bal=me -> 1	1.0104

ขั้นตอนย่อยที่สอง บรรทัดที่ 3 คือ การนับความถี่ของแต่ละแอททริบิวต์และทำการเลือกแอททริบิวต์ที่มีความถี่มากที่สุดด้วยฟังก์ชัน find_max_frequentแสดงตัวอย่างได้จากตารางที่ 3.12 ด้วยการนับความถี่ของแต่ละแอททริบิวต์จะได้ดังตารางที่ 3.13 จะเห็นได้ว่าในกรณีนี้มีค่าความถี่สูงสุด 2 แอททริบิวต์ที่มีความถี่เป็น 4 โปรแกรมจะทำการสุ่มเลือกมา 1 แอททริบิวต์ สมมติว่าตัวอย่างนี้สุ่มได้ แอททริบิวต์Balance ก็จะนำแอททริบิวต์ดังกล่าวนี้ไปใช้เป็นเกณฑ์คัดเลือกกฎในขั้นตอนย่อยถัดไป

AlgorithmFuzzy_Classification_Association_Rule_Selection

Input:

F: a set of FCARs.

FS: a set of size of FCARs.

FC: a set of class of FCARs.

FScore: a set of score of FCARs.

Attributes: a set of all attributes.

Output:

CF: the Compact FCARs

---

(1) k = 0

(2) TopF= find_top_rules(F, FS, FC, FScore)

(3) Attr = find_max_frequent(TopF) // Attr is the best attribute

(4) for each items ∈Attr { // Structure of Attr {attribute-low,//attribute-medium,attribute-high}

(5) BF_k = find_best_rules(items, F,FC, FScore, Attributes);

(6) k = k+1; //BF is an array}

(7) CF = remove_redundant_rules(BF) // remove rules that are superset in antecedent part (left-hand-side) with the same class as other rules

รูปที่ 3.2 ขั้นตอนการเลือกกฎ FCARsเพื่อนำไปใช้ในการทำนายข้อมูลด้วยอัลกอริทึม CCFAR

ตารางที่ 3.13 จำนวนความถี่ของแต่ละแอททริบิวต์

Attribute	Item	Frequency
Age	Age=hi, Age=low, Age=hi, Age=hi	4
Income	Inc=hi	1
Balance	Bal=low, Bal=low, Bal=me, Bal=me	4

ส่วนขั้นตอนย่อยที่สาม บรรทัดที่ 4-6 คือ การคัดเลือกกฎ FCARsที่ดีที่สุดด้วยการส่งแต่ละไอเท็ม(ตัวแปร items) ของตัวแปร Attr ที่ได้จากประมวลผลในขั้นตอนย่อยที่สอง ซึ่งจากตัวอย่างที่ได้ คือ แอททริบิวต์Balance ดังนั้นแต่ละไอเท็มของแอททริบิวต์Balance คือ Bal=lowBal=me และ Bal=hi จะถูกส่งผ่านฟังก์ชัน find_best_rulesเพื่อทำการคัดเลือกกฎ FCARsที่ดีที่สุด โดยมีขั้นตอนดังรูปที่ 3.5 และแสดงตัวอย่างได้จากตารางที่ 3.13ซึ่งแอททริบิวต์ที่มีความถี่มากที่สุดที่ได้จากแอททริบิวต์ในฟังก์ชัน find_max_frequent คือ แอททริบิวต์Balance โปรแกรมจะทำการจัดกลุ่มกฎ FCARs ที่ประกอบด้วย Bal=lowBal=me และ Bal=hi โดยจะได้ 3 กลุ่ม ดังตารางที่ 3.14-3.16 แต่ในกรณีที่ไม่มีกฎ FCARs ที่ประกอบด้วย Bal=low หรือ Bal=me หรือ Bal=hi อัลกอริทึมนี้จะทำการสร้างกฎดังกล่าวขึ้นมาใหม่ เช่น ไม่มีกฎ FCARs ที่ประกอบด้วยBal=hi อัลกอริทึมนี้จะสร้างกฎ 2 กฎ คือ กฎ Bal=hi -> 1 และ กฎ Bal=hi ->2 แต่จะทำการเลือกกฎที่มีค่าคะแนนสูงสุด และขนาดสั้นที่สุดของแต่ละกลุ่มจากตารางที่ 3.14 ถึง 3.15 แสดงตัวอย่างการเลือกกฎได้ดังตารางที่ 3.17 ซึ่งเป็นกฎผลลัพธ์ของอัลกอริทึมนี้ แต่ถ้ามีกฎ FCARs ที่มีปัจจัยเดียวกันเป็นสมาชิกของกฎ FCARs อื่น (ไอเท็มทางด้านซ้ายมือของกฎ) และมีคลาสเดียวกัน อัลกอริทึมนี้จะทำการเลือกกฎ FCARs ที่มีขนาดสั้นที่สุด ยกตัวอย่างเช่น มีกฎ Age=hi, Bal=me ->1 กับ กฎ Bal=me ->1 อัลกอริทึมนี้จะทำการเลือกกฎ Bal=me ->1 มาใช้ทำนายเพียงกฎเดียว

ตารางที่ 3.14กฎ FCARs ที่ประกอบด้วย Bal=low

FCARs	Score
Bal=low -> 2	1.9225
Age=low, Bal=low -> 2	1.1619

ตารางที่ 3.15กฎ FCARs ที่ประกอบด้วย Bal=me

FCARs	Score
Bal=me -> 1	1.2677
Age=hi, Bal=me -> 1	1.2608
Inc=hi, Bal=me -> 1	1.0421
Age=hi, Inc=hi, Bal=me -> 1	1.0104

ตารางที่ 3.16กฎ FCARs ที่ประกอบด้วย Bal=hi

FCARs	Score
Bal=hi -> 1	0.4088

ตารางที่ 3.17 กฎ FCARs ที่ใช้ในการทำนาย

FCARs	Score
Bal=low -> 2	1.9225
Bal=me ->1	1.2677
Bal=hi -> 1	0.4088

ส่วนสุดท้ายขั้นตอนย่อยที่ 4 บรรทัดที่ 7 เป็นการลบกฎที่มีลักษณะซ้ำซ้อน (Remove Redundant Rules) โดยกฎที่จะทำการลบไปนั้นจะต้องเป็นกฎที่มีปัจจัยซ้ำกันกับกฎอื่น (หรือเรียกว่ากฎที่เป็น Superset ของกฎอื่น) และกฎดังกล่าวมีคลาสเป้าหมายที่เหมือนกัน ยกตัวอย่างเช่น

กฎที่ 1: Age = low -> yes,

กฎที่ 2: Age = low and Bal = high -> yes

จากตัวอย่างประกอบไปด้วย 2 กฎ ซึ่งจะเห็นได้ว่าทั้ง 2 กฎจะมีปัจจัย Age = low และคลาสเป้าหมายเดียวกัน คือ yesซึ่งจากตัวอย่างใช้เพียงกฎที่ 1 ก็สามารถพยากรณ์ข้อมูลในอนาคตได้แล้วว่าถ้าข้อมูลในอนาคตเป็น Age = low ข้อมูลดังกล่าวจะต้องเป็นคลาส yes จากเหตุผลดังกล่าวทำให้งานวิจัยนี้ทำการคัดเลือกกฎที่ 1 ไว้และลบกฎที่ 2 ทิ้ง

หลังจากได้อธิบายกลไกหลักของการเลือกกฎการจำแนกด้วยกฎความสัมพันธ์แบบคลุมเครือทั้ง 4 ขั้นตอนย่อยแล้ว ต่อไปจะเป็นการอธิบายฟังก์ชันการทำงานที่มีการเรียกใช้งานในขั้นตอนย่อยทั้ง 3 ดังนี้

ฟังก์ชันfind_top_rulesดังรูปที่ 3.3 เป็นฟังก์ชันที่ทำการเลือกกฎFCARs ที่มีค่าคะแนนมากที่สุดของแต่ละคลาสและแต่ละขนาด โดยที่บรรทัดที่ 1-4 เป็นการกำหนดค่าต่าง ๆ ดังนี้ บรรทัดที่ 1 กำหนดให้ตัวแปรSF มีค่าเท่ากับจำนวนกฎ FCARs ทั้งหมด บรรทัดที่ 2 กำหนดให้ตัวแปรSR มีค่าเท่ากับขนาดของกฎที่มีค่ามากที่สุด บรรทัดที่ 3 กำหนดให้ตัวแปรMC มีค่าเท่ากับจำนวนคลาสที่เป็นเอกลักษณ์ และบรรทัดที่ 4 เป็นการกำหนดค่า k เพื่อใช้เป็นดรรชนีชี้ตำแหน่งของกฎ ส่วนบรรทัดที่ 5-7 เป็นการวนเพื่อดึงค่า คลาส ขนาดของกฎ และกฎ เพื่อนำมาตรวจสอบในบรรทัดที่ 8 ว่าถ้ากฎใด ๆ มีคลาสเดียวกัน ขนาดเท่ากัน และมีค่าคะแนนมากที่สุด กฎนั้นจะถูกเก็บไว้ในตัวแปร TopF

Procedurefind_top_rules

Input:

F: a set of FCARs.

FS: a set of size of FCARs.

FC: a set of class of FCARs.

FScore: a set of score of FCARs.

Output:

TopF: a set of FCARs that max score.

---

(1) SF = |F| //number of rules

(2) SR = max (FS) //max size rule

(3) MC = |unique(FC)| //number of class

(4) k = 0

(5) for(c = 1; c <= MC; c ++){

(6) for(s = 1; s <= SR; s ++){

(7) for(i = 1; i<= SF; i ++){

(8) If $FC_i == c \&\& FS_i == s \&\& F_i == \max(FScore_{c,s})${

(9) $TopF_k = F_i$; k = k+1;

(10) } } } }

(11) return TopF

รูปที่ 3.3 ฟังก์ชัน find_top_rules

ฟังก์ชันfind_max_frequent ดังรูปที่ 3.4 เป็นฟังก์ชันที่ทำการนับความถี่ของแต่ละแอททริบิวต์และทำการเลือกแอททริบิวต์ที่มีความถี่มากที่สุด โดยบรรทัดที่ 2-4 เป็นการดึงไอเท็มแบบคลุมเครือที่อยู่ทางซ้ายมือของกฎมาตรวจสอบในบรรทัดที่ 5 ว่าถ้าไอเท็มแบบคลุมเครือใด ๆ เป็นสมาชิกของแอททริบิวต์ใด ๆ จริง ให้ทำการเพิ่มค่าบวกหนึ่งแก่แอททริบิวต์นั้น เพื่อทำการหาค่าแอททริบิวต์ที่มีความถี่มากที่สุดและทำการส่งค่ากลับในบรรทัดที่ 10

Procedurefind_max_frequent

Input:

TopF: a set of FCARs that max score.

Attributes: a set of all attributes.

Output:

Attr: the best attribute

---

(1) count = 0

(2) for each rule R ∈ TopF { // R is rule

(3) for each LHS ∈ R { // LHS: left hand side Items of rules

(4) for each item ∈ LHS {

(5) If item ∈ a_k { // $a_1, a_2, \dots, a_k$ ∈ Atrributes

(6) $a_k.count$ +1

(7) }

(8) } } }

(9) Attr = max ($a_k.count$)

(10) return Attr

รูปที่ 3.4 ฟังก์ชัน find_max_frequent

ฟังก์ชันfind_best_rules ดังรูปที่ 3.5 เป็นฟังก์ชันที่ทำการคัดเลือกกฎ FCARsที่ดีที่สุดโดยบรรทัดที่ 2-3 ทำการดึงไอเท็มเซตแบบคลุมเครือที่อยู่ทางด้านซ้ายมือของกฎ มาตรวจสอบ (บรรทัดที่ 4) กับไอเท็มแบบคลุมเครือที่ได้จากฟังก์ชัน find_max_frequent ว่าถ้าไอเท็มแบบคลุมเครือ item เป็นสมาชิกของไอเท็มเซตแบบคลุมเครือที่อยู่ทางด้านซ้ายมือของกฎ LHS จริง ให้ทำการเพิ่มกฎเข้าไปในตัวแปร RGroupและเพิ่มค่าคะแนนเข้าไปในตัวแปรSGroup(บรรทัดที่ 5 และ 6) ในบรรทัดที่ 8-9 ทำการเลือกกฎที่มีค่าคะแนนสูงที่สุดในตัวแปรRGroup แต่ในกรณีที่ไม่มีกฎใดเลยในตัวแปร RGroup(บรรทัดที่ 10) ให้ทำการสร้างกฎที่ประกอบด้วยค่าของตัวแปร item เป็นสมาชิกใหม่ (บรรทัดที่ 11) โดยกำหนดค่าสนับสนุนขั้นต่ำเท่ากับ 0 ซึ่งกฎที่สร้างมาใหม่นั้นจะมีคุณสมบัติเป็นกฎที่ดีที่สุดหรือ BF ทันที

Functionfind_best_rules

Input:

items: attribute-partition (partition: low, medium, high)

F: a set of FCARs.

FScore: a set of score of FCARs.

Output:

BF is the best FCARs

---

(1) k = 0

(2) for each rule R_i ∈F { // R is rule : LHS -> RHS

(3) for each LHS∈R_i { // LHS:left hand side Items of rules

(4) If subset(LHS, item) {

(5) $RGroup_k = R_i$

(6) $SGroup_k = FScore_i$; k = k+1;

(7) }} }

(8) Index =max (SGroup)// find index of max score

(9) $BF = RGroup_{(Index)}$

(10) If BF = ⦰ {

(11) BF = create_rule(item) \\ create new rule that has the item \\ as subset and does not care minimum support.

(12) }

(13) return BF

รูปที่ 3.5ฟังก์ชัน find_best_rules

บทที่ 4

การทดสอบและอภิปรายผล

ในบทที่ 4 นี้ผู้วิจัยได้นำเสนอการทดสอบประสิทธิภาพของอัลกอริทึม CCFAR ที่ผู้วิจัยได้ทำการพัฒนา ซึ่งมีวิธีการทดสอบประสิทธิภาพของอัลกอริทึมไว้ 3 แบบ คือ การทดสอบหาเกณฑ์ที่เหมาะสมสำหรับการสร้างกฎ FCARs ของอัลกอริทึม CCFAR การทดสอบลักษณะการกระจายตัวของข้อมูลที่เหมาะสมสำหรับอัลกอริทึม CCFAR และการเปรียบเทียบประสิทธิภาพการจำแนกข้อมูลและจำนวนกฎที่ได้รับของอัลกอริทึม CCFAR กับอีก 9 อัลกอริทึมโดยมีรายละเอียดการทดสอบประสิทธิภาพของอัลกอริทึม CCFAR ของแต่ละแบบดังต่อไปนี้

4.1 การทดสอบหาเกณฑ์ที่เหมาะสมสำหรับการสร้างกฎ FCARs ของอัลกอริทึม CCFAR

การทดสอบหาเกณฑ์ที่เหมาะสมสำหรับการสร้างกฎ FCARs เพื่อคัดเลือกกฎที่มีประสิทธิภาพที่ส่งผลให้ได้รับค่าความถูกต้องในการจำแนกข้อมูลที่ดีที่สุด โดยเกณฑ์ดังกล่าวจะเรียกว่าคะแนน (Score) ซึ่งอยู่ในกระบวนการสร้างกฎความสัมพันธ์แบบคลุมเครือ FCARs ด้วยอัลกอริทึม CCFAR(รูปที่ 2.12)

วิธีการทดสอบจะทำการนำเกณฑ์ที่ใช้ในการสร้างกฎ FCARs แบบดั้งเดิม คือ เกณฑ์Fuzzy Confidence: FC(สมการที่ 2.9) เกณฑ์Fuzzy Correlation: FCORR(สมการที่2.10) และFiring Strength: β(สมการที่2.11) ซึ่งแต่ละเกณฑ์มีคุณสมบัติตามลำดับดังนี้ FC เป็นเกณฑ์ที่ใช้อธิบายความถูกต้องของกฎ FCORR เป็นเกณฑ์ที่ใช้อธิบายความสัมพันธ์ระหว่างไอเท็มเซตหลัง if และไอเท็มเป้าหมายหลัง then ของกฎและβเป็นเกณฑ์ที่ใช้อธิบายกำลังหรือค่าสนับสนุนของกฎ ซึ่งการทดลองจะนำเกณฑ์ทั้งสามมาหาสับเซตจะได้ทั้งหมด 7 เซตดังต่อไปนี้ (เกณฑ์ 4.7 เป็นเกณฑ์ที่งานวิจัยนี้สร้างขึ้นมาใหม่โดยใช้วิธีแบบ Heuristic)

1. {FCORR} หมายถึง ใช้เกณฑ์Fuzzy Correlationในการสร้างกฎ FCARs

2.{FC} หมายถึง ใช้เกณฑ์Fuzzy Confidence ในการสร้างกฎ FCARs

3.{β}หมายถึง ใช้เกณฑ์Firing Strengthในการสร้างกฎFCARs

4.{FCORR, FC}หมายถึง ใช้เกณฑ์Fuzzy Correlation คูณกับFuzzy Confidence ในการสร้างกฎ FCARs

5.{FC,β}หมายถึงใช้เกณฑ์Fuzzy Confidence คูณกับ Firing Strengthในการสร้างกฎ FCARs

6. {FCORR, β}หมายถึงใช้เกณฑ์Fuzzy Correlationคูณกับ Firing Strengthในการสร้างกฎ FCARs

7.{FCORR, FC,β} หมายถึงใช้เกณฑ์Fuzzy Correlation คูณกับFuzzy Confidence คูณกับ Firing Strengthในการสร้างกฎ FCARs

ข้อมูลที่ใช้ในการทดสอบเป็นข้อมูลตัวเลขที่ได้จาก UCI Machine Learning Repositoryมีทั้งหมด 5 ชุดข้อมูลซึ่งมีรายละเอียดดังตารางที่ 4.1 และผลที่ได้จากการทดสอบแสดงดังตารางที่ 4.2 และ 4.3 ส่วนการวัดประสิทธิภาพใช้ 10 Fold Cross-Validation

ตารางที่ 4.1 ข้อมูลที่ใช้ในการทดสอบประสิทธิภาพของอัลกอริทึม CCFAR

ชื่อชุดข้อมูล	จำนวนแถว	จำนวนคอลัมน์	จำนวนคลาส
Iris	150	4	3
Heart	270	13	2
Pima	768	8	2
Pupa	345	6	2
Transfusion	748	4	2

ตารางที่ 4.2 ผลการทดสอบประสิทธิภาพค่าความถูกต้องในการจำแนกข้อมูล เพื่อหาเกณฑ์ที่เหมาะสมสำหรับการสร้างกฎ FCARs ของอัลกอริทึม CCFAR

เกณฑ์ที่ใช้ในการทดสอบ	ชื่อชุดข้อมูล					
	Iris	Heart	Pima	Pupa	Transfusion	เฉลี่ย
{FCORR}	0.96	0.78182	0.7331	0.52479	0.62427	0.72479
{FC}	0.96	0.78182	0.69653	0.55437	0.61493	0.72153
{β}	0.94	0.71899	0.63546	0.57983	0.73872	0.7286
{FCORR, FC}	0.96	0.78182	0.74609	0.50193	0.62427	0.72282
{FCORR, β}	0.96	0.78182	0.66403	0.53924	0.70049	0.72911
{FC, β}	0.96	0.78182	0.62632	0.57983	0.73872	0.73733
{FCORR, FC,β}	0.96	0.78182	0.74609	0.53336	0.70049	0.74435

(หมายเหตุ ข้อมูลที่ขีดเส้นใต้ หมายถึงข้อมูลที่มีค่ามากที่สุดสำหรับชุดข้อมูลนั้น ๆ)

ตารางที่ 4.3 ผลการทดสอบประสิทธิภาพทางด้านจำนวนกฎที่ได้รับ เพื่อหาเกณฑ์ที่เหมาะสมสำหรับการสร้างกฎ FCARs ของอัลกอริทึม CCFAR

เกณฑ์ที่ใช้ในการทดสอบ	ชื่อชุดข้อมูล					
	Iris	Heart	Pima	Pupa	Transfusion	เฉลี่ย
{FCORR}	3	3	3.9	3	3	3.18
{FC}	3	3	3.2	3.8	3	3.2
{β},	3	3	3	3	3	3
{FCORR, FC}	3	3	4	3	3	3.2
{FCORR, β}	3	3	3.2	3	3	3.04
{FC, β}	3	3	4	3	3	3.2
{FCORR, FC,β}	3	3	4	3	3	3.2

ผลการทดสอบประสิทธิภาพทางด้านการจำแนกข้อมูลและด้านจำนวนกฎที่ได้รับ เพื่อหาเกณฑ์ที่เหมาะสมสำหรับการสร้างกฎ FCARs ของอัลกอริทึม CCFAR จากตารางที่ 4.2 และ 4.3 สามารถอภิปรายผลการทดลองของแต่ละชุดข้อมูลได้ดังนี้

- ชุดข้อมูล Iris เมื่อใช้เกณฑ์ $\{\beta\}$ในการประมวลผลจะทำให้ได้รับค่าความถูกต้องน้อยที่สุด คือ 0.94000 แต่เมื่อใช้เกณฑ์อื่น ๆ ในการประมวลผล ค่าความถูกต้องที่ได้จะมีค่าสูงสุดและเท่ากัน
- ชุดข้อมูล Heart เมื่อใช้เกณฑ์$\{\beta\}$ ในการประมวลผลจะทำให้ได้รับค่าความถูกต้องมากที่สุด คือ 0.71899 ซึ่งตรงกันข้ามกับชุดข้อมูล Iris แต่เมื่อใช้เกณฑ์อื่น ๆ ในการประมวลผล ค่าความถูกต้องที่ได้จะมีค่าต่ำสุดและเท่ากัน
- ชุดข้อมูล Pima เมื่อใช้เกณฑ์ {FC, β} ในการประมวลผลจะทำให้ได้รับค่าความถูกต้องน้อยที่สุด คือ 0.62632แต่เมื่อใช้เกณฑ์ {FCORR, FC} และ {FCORR, FC,β} ในการประมวลผลจะทำให้ได้รับค่าความถูกต้องมากที่สุด คือ 0.74609
- ชุดข้อมูล Pupa เมื่อใช้เกณฑ์ {FCORR, FC}ในการประมวลผลจะทำให้ได้รับค่าความถูกต้องน้อยที่สุด คือ 0.50193แต่เมื่อใช้เกณฑ์$\{\beta\}$ และ {FC, β} ในการประมวลผลจะทำให้ได้รับค่าความถูกต้องมากที่สุด คือ 0.57983
- ชุดข้อมูล Transfusion เมื่อใช้เกณฑ์ {FC} ในการประมวลผลจะทำให้ได้รับค่าความถูกต้องน้อยที่สุด คือ 0.61493แต่เมื่อใช้เกณฑ์$\{\beta\}$ และ {FC, β} ในการประมวลผลจะทำให้ได้รับค่าความถูกต้องมากที่สุด คือ 0.73872

ส่วนจำนวนกฎที่ได้รับจากทุกชุดข้อมูลมีจำนวนที่ไม่แตกต่างกันมาก จึงไม่มีผลต่อการนำไปใช้งาน ดังนั้นในการทดสอบประสิทธิภาพเพื่อหาเกณฑ์ที่เหมาะสมสำหรับการสร้างกฎ FCARs ของอัลกอริทึม CCFAR จึงได้ทำการเลือกเกณฑ์ที่มีค่าความถูกต้องในการจำแนกข้อมูลสูงสุดไปใช้ในการประมวลผล ซึ่งเกณฑ์ดังกล่าว คือ เกณฑ์ {FCORR, FC,β} ที่มีค่าความถูกต้องในการจำแนกข้อมูลเฉลี่ย คือ 0.74435

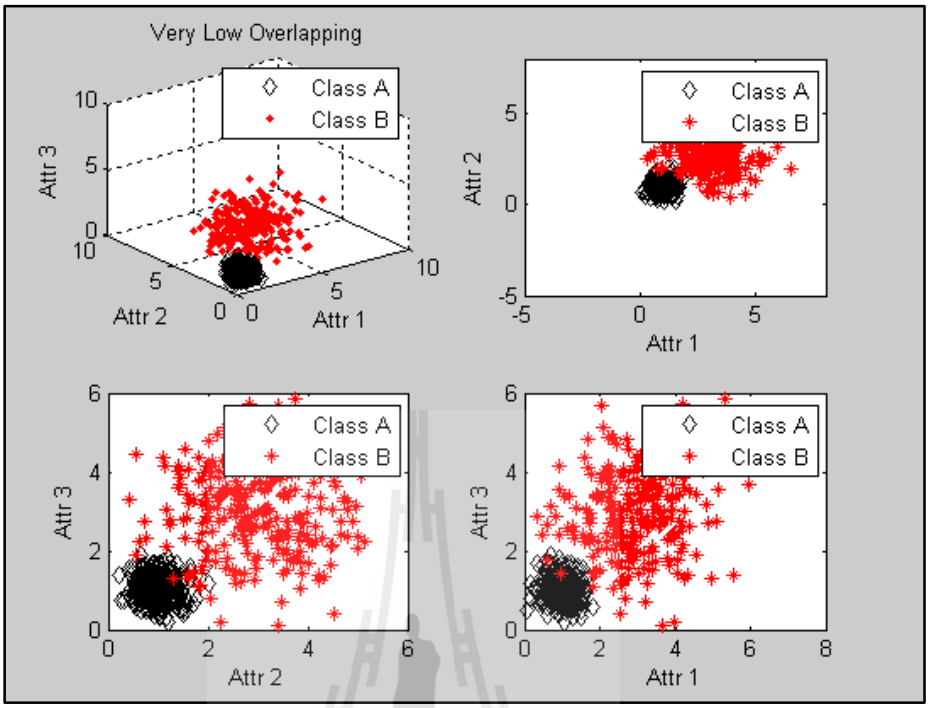
4.2 การทดสอบลักษณะการกระจายตัวของข้อมูลที่เหมาะสมสำหรับอัลกอริทึม CCFAR

ประสิทธิภาพในการจำแนกข้อมูลของแต่ละอัลกอริทึม อาจจะมีประสิทธิภาพในการจำแนกข้อมูลที่แตกต่างกันไป ทั้งนี้อาจจะขึ้นกับวิธีการจำแนกข้อมูลหรือลักษณะการกระจายของข้อมูลที่ใช้ในการจำแนก ดังนั้นในหัวข้อนี้จึงมุ่งเน้นไปที่ลักษณะของข้อมูลที่ใช้ในการจำแนก เพื่อหาการกระจายตัวของข้อมูลที่เหมาะสมสำหรับอัลกอริทึม CCFARและทำการเปรียบเทียบกับ 3 อัลกอริทึมที่ไม่ได้ใช้คุณสมบัติในการจำแนกข้อมูลด้วยฟัซซี คือ OneR J48 และ Ripper เพื่อแสดงจุดเด่นของการใช้เทคนิคฟัซซีด้วยอัลกอริทึม CCFAR ให้ชัดเจนยิ่งขึ้น

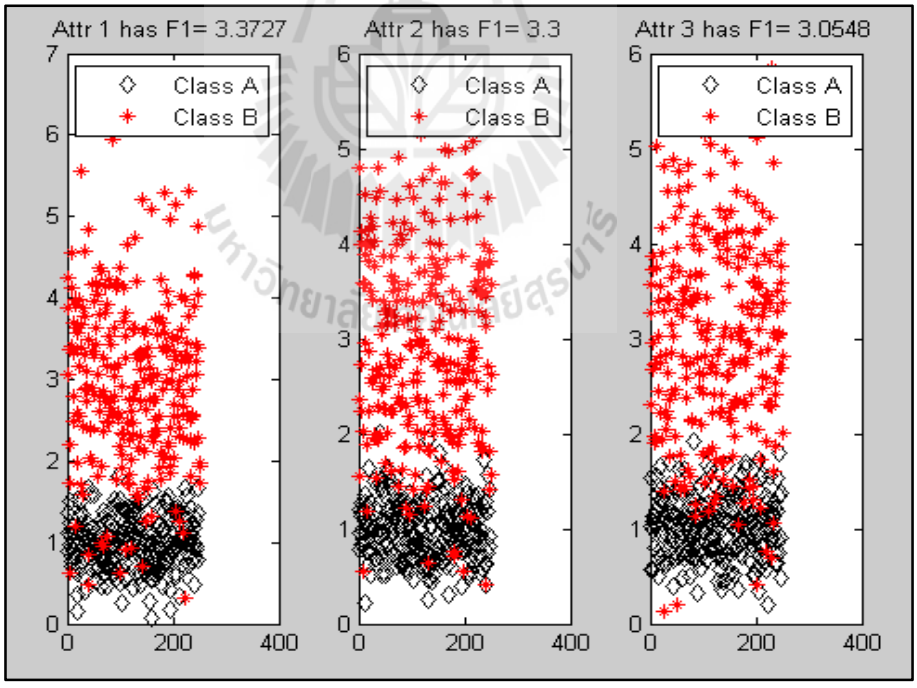
วิธีการทดสอบการกระจายตัวของข้อมูลจะทำการทดสอบการกระจายข้อมูล 5 แบบ(รูปที่ 4.1 ถึง 4.5) คือ ข้อมูลที่มีการซ้อนทับกันน้อยมาก ข้อมูลที่มีการซ้อนทับกันน้อย ข้อมูลที่มีการซ้อนทับกันปานกลาง ข้อมูลที่มีการซ้อนทับกันมาก และข้อมูลที่มีการซ้อนทับกันมากที่สุด ซึ่งแต่ละแบบจะมีจำนวนข้อมูล 500 แถว 3 คอลัมน์ และมีคลาส 2 คลาส โดยแบ่งเป็นคลาส A จำนวน 250 แถว ส่วนคลาส B แบ่งเป็นจำนวน 250 แถวเช่นกันการวัดการซ้อนทับกันของข้อมูลจะใช้ค่า Fisher's Ratio (Chen et al., 2004). ดังสมการที่ (4.1) ซึ่งถ้าค่านี้มีค่ามากจะหมายถึงมีการซ้อนทับกันของข้อมูลน้อย เช่น รูปที่ 4.1 (ข) จะเห็นได้ว่าค่า F1 หรือ ค่า Fisher's Ratioของแต่ละแอททริบิวต์ (Attr) จะมีค่ามากกว่าแอททริบิวต์ในรูปที่ 4.2 (ข)

$$\text{Fisher's Ratio} = \frac{(\text{m1}-\text{m2})^2}{v1+v2} \quad (4.1)$$

เมื่อ m1 และ m2 คือ ค่าเฉลี่ยของคลาส A และคลาส B ส่วน v1 และ v2 คือ ค่าความแปรปรวนของข้อมูลในคลาส A และคลาส B ตามลำดับ

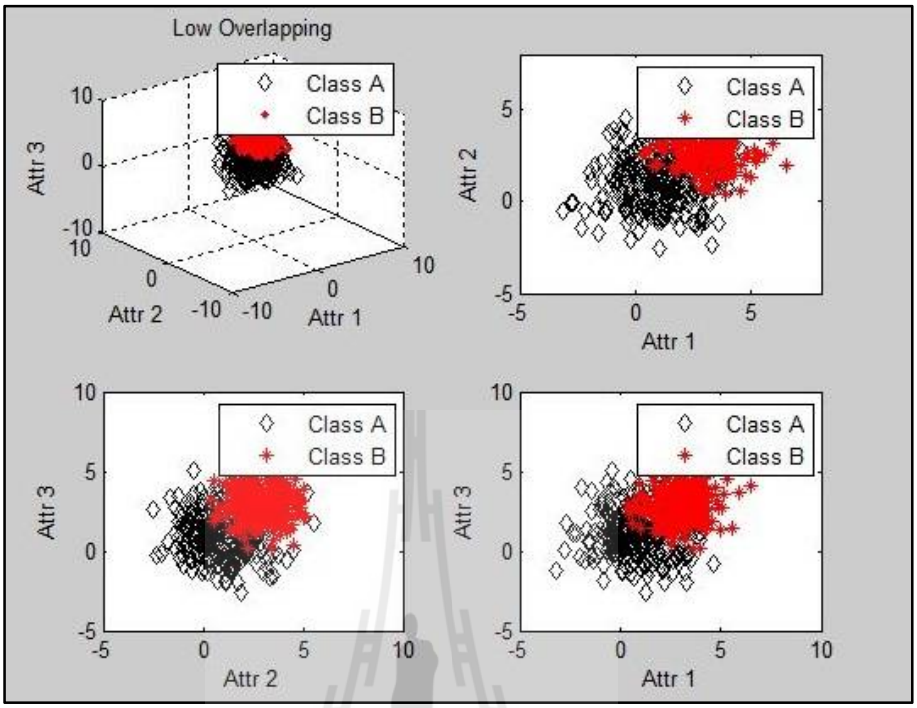


(ก) ลักษณะการซ้อนทับของข้อมูล

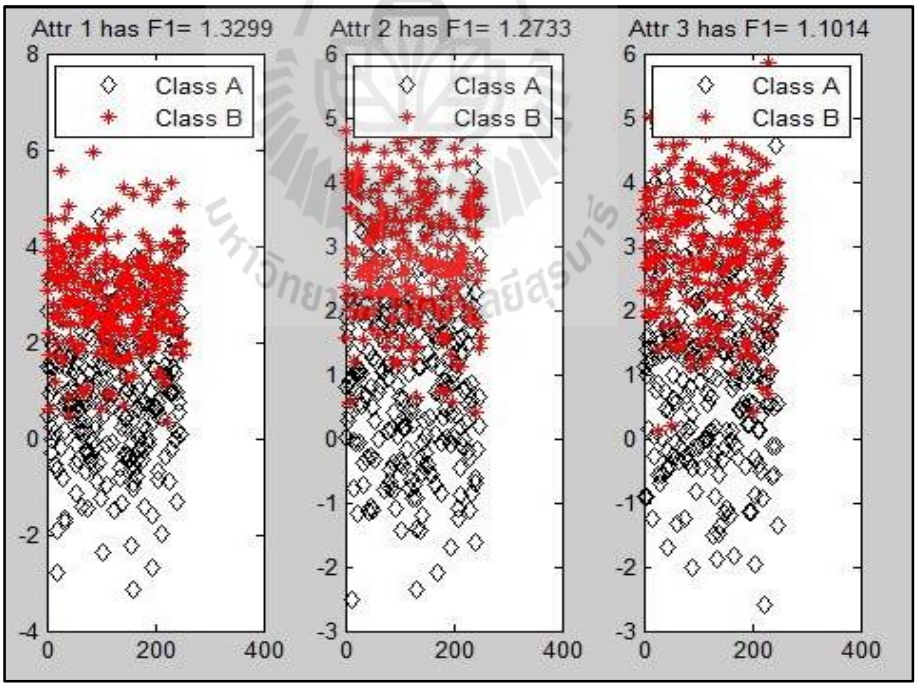


(ข) การซ้อนทับของข้อมูลในแต่ละมิติ

รูปที่ 4.1 การกระจายตัวของข้อมูลที่มีการซ้อนทับกันน้อยมาก

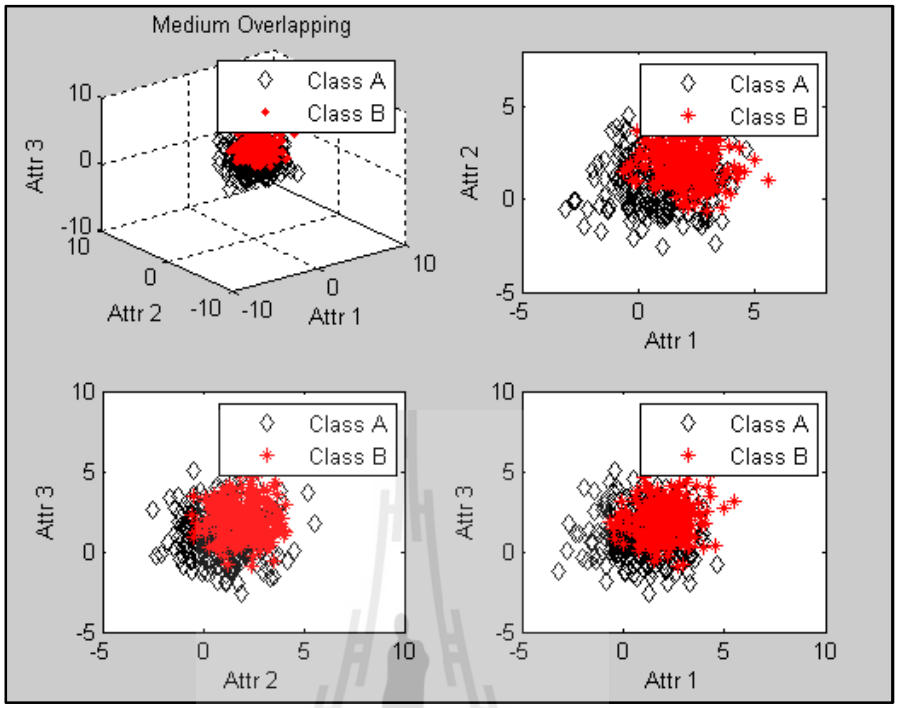


(ก) ลักษณะการซ้อนทับของข้อมูล

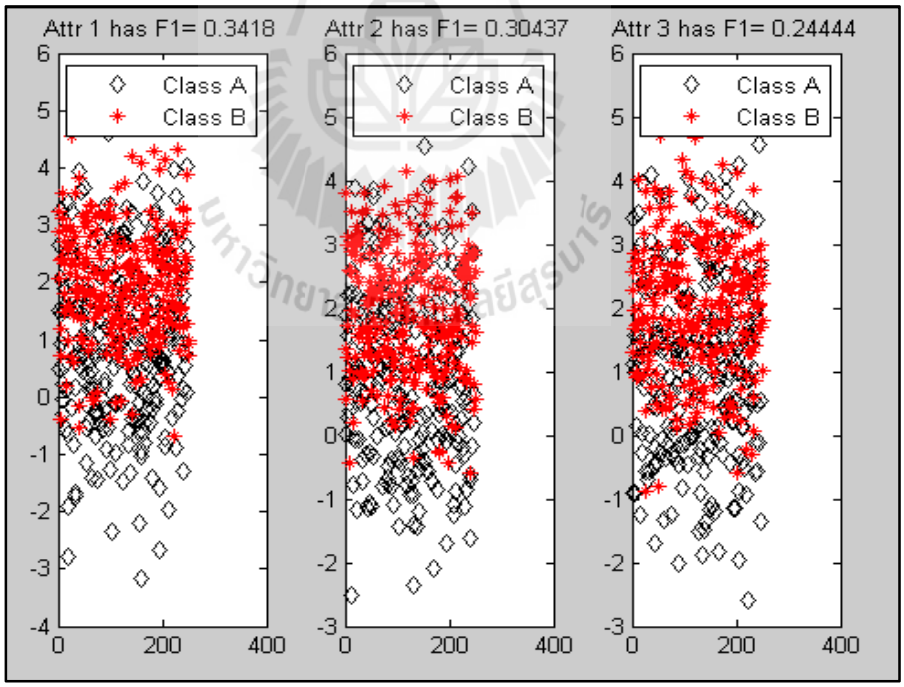


(ข) การซ้อนทับของข้อมูลในแต่ละมิติ

รูปที่ 4.2 การกระจายตัวของข้อมูลที่มีการซ้อนทับกันน้อย

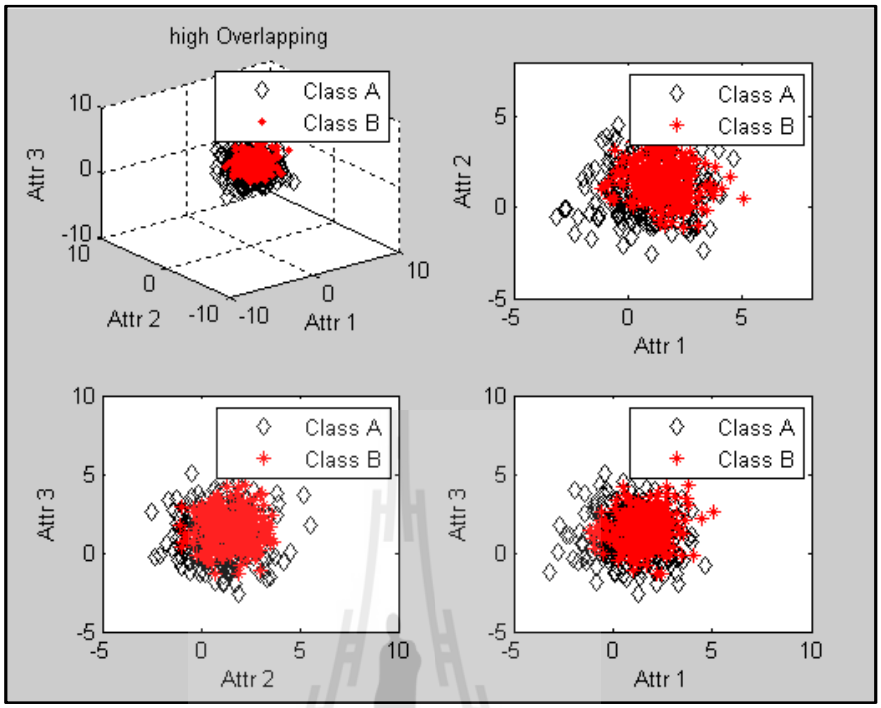


(ก) ลักษณะการซ้อนทับของข้อมูล

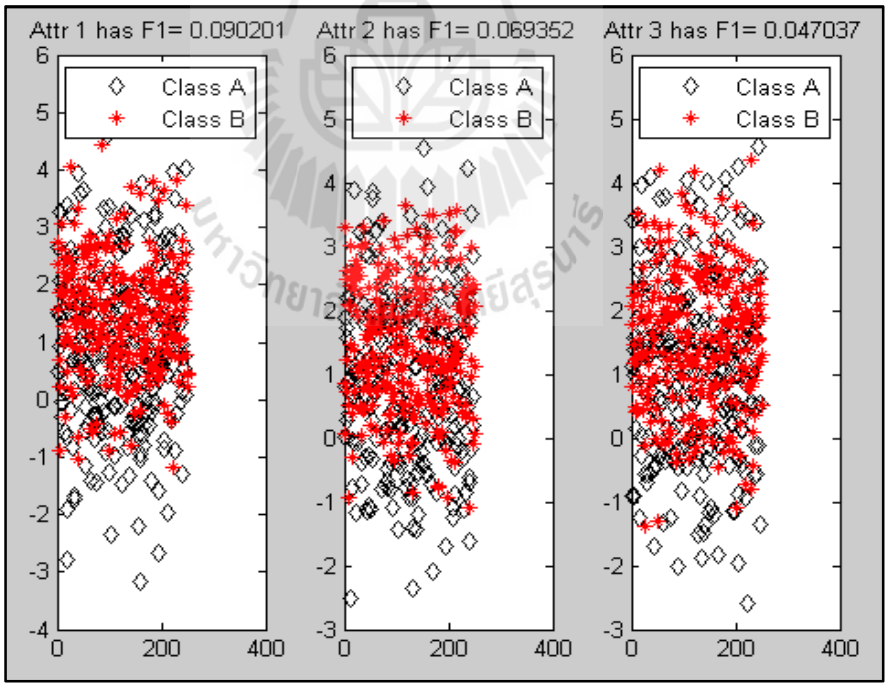


(ข) การซ้อนทับของข้อมูลในแต่ละมิติ

รูปที่ 4.3 การกระจายตัวของข้อมูลที่มีการซ้อนทับกันปานกลาง

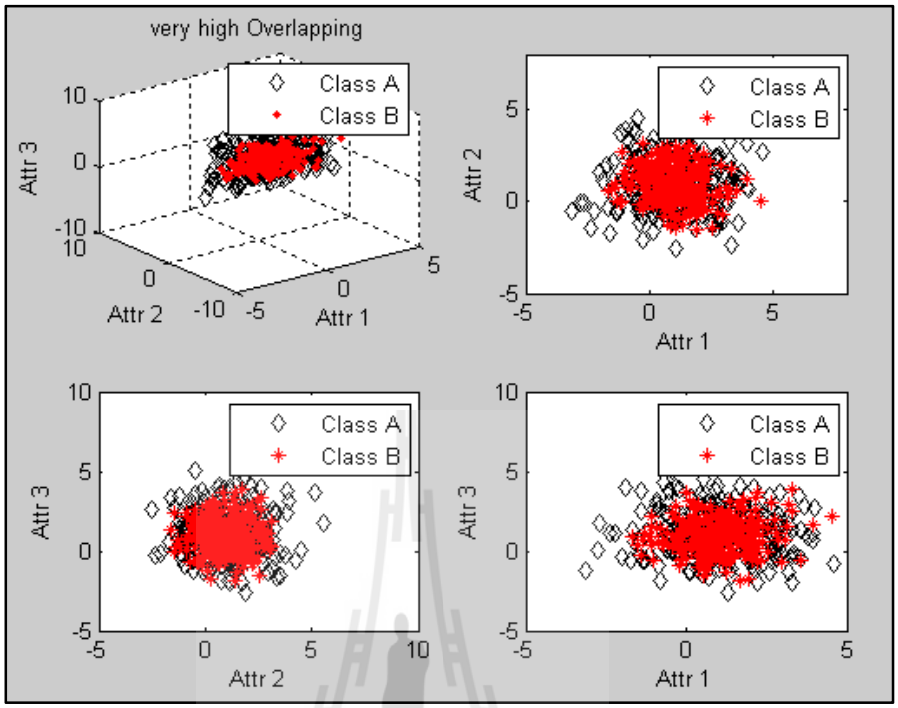


(ก) ลักษณะการซ้อนทับของข้อมูล

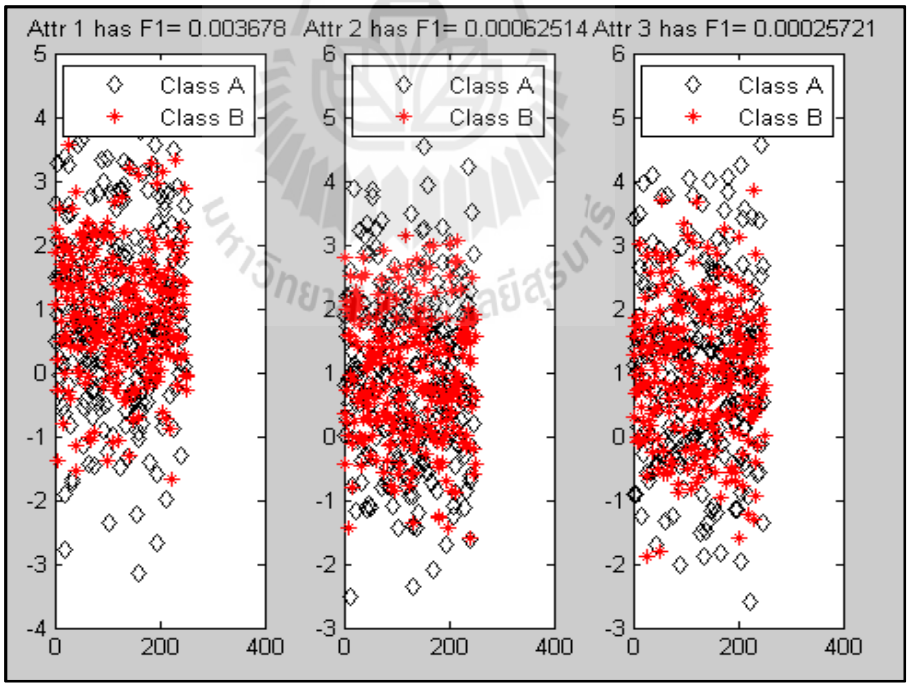


(ข) การซ้อนทับของข้อมูลในแต่ละมิติ

รูปที่ 4.4 การกระจายตัวของข้อมูลที่มีการซ้อนทับกันมาก



(ก) ลักษณะการซ้อนทับของข้อมูล



(ข) การซ้อนทับของข้อมูลในแต่ละมิติ

รูปที่ 4.5 การกระจายตัวของข้อมูลที่มีการซ้อนทับกันมากที่สุด

ตารางที่ 4.4ผลการทดสอบค่าความถูกต้องของการจำแนกข้อมูลตามลักษณะการกระจายข้อมูล

		อัลกอริทึม			
		OneR	J48	RIPPER	*CCFAR
ระดับการทับซ้อนของข้อมูล	น้อยมาก	0.950	0.982	<u>0.988</u>	0.920
	น้อย	0.796	<u>0.910</u>	0.890	0.730
	ปานกลาง	0.580	0.750	<u>0.760</u>	0.630
	มาก	0.500	<u>0.670</u>	0.650	0.600
	สูงมาก	0.530	0.550	<u>0.570</u>	0.560
ผลรวม	Avg(Acc)	0.671	<u>0.772</u>	0.771	0.688
	Rank	4	1	2	3

(หมายเหตุ ข้อมูลที่ขีดเส้นใต้ หมายถึงข้อมูลที่มีค่ามากที่สุดสำหรับชุดข้อมูลนั้น ๆ)

ตารางที่ 4.5ผลการทดสอบจำนวนกฎและความกะทัดรัดของกฎแบบปกติ

		อัลกอริทึม			
		OneR	J48	RIPPER	*CCFAR
ระดับการทับซ้อนของข้อมูล	น้อยมาก	2	7	2	3
	น้อย	10	27	7	3
	ปานกลาง	16	21	5	3
	มาก	29	11	5	3
	สูงมาก	27	17	4	3
ผลรวม	Avg(CV)	16.80	16.60	4.60	3
	ความกะทัดรัดของกฎแบบปกติ (NCV)	0	0.014	0.88	1
	Rank	4	3	2	1

ตารางที่ 4.6ผลการทดสอบค่าความเหมาะสมของกฎ

		อัลกอริทึม			
		OneR	J48	RIPPER	*CCFAR
ผลรวม	SR	0.335	0.393	0.827	0.844
	Rank	4	3	2	1

ผลการทดสอบการกระจายตัวของข้อมูลที่เหมาะสมสำหรับอัลกอริทึม CCFAR แสดงดังตารางที่ 4.4 ถึง 4.6จากตารางดังกล่าวประกอบด้วยสัญลักษณ์ต่าง ๆ ซึ่งมีความหมายดังนี้ Avg(Acc) คือค่าเฉลี่ยของความถูกต้องAvg(CV) คือ ค่าเฉลี่ยของจำนวนกฎ NCV คือ ค่าความกะทัดรัดของกฎแบบปกติ (แสดงรายละเอียดในหัวข้อ 2.5.2) และ SR คือ ค่าความเหมาะสมของกฎ (แสดงรายละเอียดในหัวข้อ 2.5.3)ส่วนการวัดประสิทธิภาพใช้ 10 Fold Cross-Validation

อภิปรายผลการทดสอบการกระจายข้อมูลของแต่ละรูปแบบได้ดังนี้

- เมื่อระดับการซ้อนทับของข้อมูลอยู่ที่ระดับน้อยมากประสิทธิภาพในการจำแนกข้อมูลของอัลกอริทึม RIPPER จะมีค่าสูงที่สุดและมีจำนวนกฎที่น้อยที่สุด คือ 0.988 และ 2 กฎตามลำดับส่วนอัลกอริทึม CCFAR ที่ผู้วิจัยพัฒนาจะมีประสิทธิภาพในการจำแนกข้อมูลที่ต่ำที่สุดและจำนวนกฎที่ได้รับอยู่ในอันดับที่ 3
- เมื่อระดับการซ้อนทับของข้อมูลอยู่ที่ระดับน้อยประสิทธิภาพในการจำแนกข้อมูลของอัลกอริทึม J48จะมีค่าสูงที่สุดแต่มีจำนวนกฎที่มากที่สุด คือ 0.910และ 27 กฎตามลำดับ ส่วนอัลกอริทึม CCFAR ที่ผู้วิจัยพัฒนาจะมีประสิทธิภาพในการจำแนกข้อมูลที่ต่ำที่สุดแต่จำนวนกฎที่ได้รับจะมีจำนวนน้อยที่สุด
- เมื่อระดับการซ้อนทับของข้อมูลอยู่ที่ระดับปานกลางประสิทธิภาพในการจำแนกข้อมูลของอัลกอริทึม RIPPER จะมีค่าสูงที่ คือ 0.760ส่วนอัลกอริทึม CCFAR ที่ผู้วิจัยพัฒนาจะมีประสิทธิภาพในการจำแนกข้อมูลอยู่ในอันดับที่ 3 แต่จำนวนกฎที่ได้รับจะมีจำนวนน้อยที่สุด
- เมื่อระดับการซ้อนทับของข้อมูลอยู่ที่ระดับมากประสิทธิภาพในการจำแนกข้อมูลของอัลกอริทึม J48 จะมีค่าสูงที่ คือ 0.670ส่วนอัลกอริทึม CCFAR ที่ผู้วิจัยพัฒนาจะมีประสิทธิภาพในการจำแนกข้อมูลอยู่อันดับที่ 3 แต่จำนวนกฎที่ได้รับจะมีจำนวนน้อยที่สุด
- เมื่อระดับการซ้อนทับของข้อมูลอยู่ที่ระดับสูงมากประสิทธิภาพในการจำแนกข้อมูลของอัลกอริทึมRIPPER จะมีค่าสูงที่ คือ 0.570ส่วนอัลกอริทึม CCFAR ที่ผู้วิจัยพัฒนาจะมีประสิทธิภาพในการจำแนกข้อมูลอยู่อันดับที่ 2 และจำนวนกฎที่ได้รับจะมีจำนวนน้อยที่สุด

จากการอภิปรายผลสามารถสรุปได้ว่าอัลกอริทึม CCFAR ที่ผู้วิจัยพัฒนา ไม่ได้มีประสิทธิภาพในการจำแนกข้อมูลที่ดีที่สุด แต่จำนวนกฎที่ได้รับมีประสิทธิภาพที่ดีที่สุด(มีจำนวนน้อยที่สุด) และข้อสังเกตอีกอย่างหนึ่งจะเห็นได้ว่า เมื่อข้อมูลมีการกระจายตัวแบบทับซ้อนกันมาก

อัลกอริทึม CCFAR จะให้ผลการจำแนกข้อมูลที่ไม่แตกต่างจากอัลกอริทึมที่ได้อันดับที่ 1 ในการจำแนกข้อมูล แต่จำนวนกฎที่ได้รับมีจำนวนน้อยที่สุด ซึ่งมีความหมายว่าถ้าใช้อัลกอริทึม CCFAR ในการประมวลผลข้อมูลที่มีการทับซ้อนกันมาก จะให้ค่าถูกต้องที่ดีและสามารถนำกฎที่ได้รับไปประยุกต์ใช้ได้ง่ายที่สุด เพราะได้รับจำนวนกฎจากการประมวลผลน้อยที่สุด อีกทั้งตารางที่ 4.6 แสดงให้เห็นว่าเมื่อนำค่าความถูกต้องและจำนวนกฎมาพิจารณาร่วมกัน ผลที่ได้คือ อัลกอริทึม CCFAR จะมีความเหมาะสมของกฎอยู่ในอันดับที่ 1

4.3 การเปรียบเทียบประสิทธิภาพการจำแนกข้อมูลและจำนวนกฎที่ได้รับของอัลกอริทึม CCFAR กับอีก 9 อัลกอริทึม

ในหัวข้อนี้ได้นำเสนอการทดสอบประสิทธิภาพของอัลกอริทึม CCFAR ที่ผู้วิจัยได้พัฒนาโดยใช้วิธีการเปรียบเทียบกับอีก 9 อัลกอริทึมที่เป็นที่นิยมในการจำแนกข้อมูล ซึ่งทั้ง 9 อัลกอริทึมนี้สามารถแบ่งตามวิธีการจำแนกได้ 3 กลุ่ม คือ

1. กลุ่ม CLASS หมายถึง อัลกอริทึมที่เกี่ยวข้องกับการจำแนกประเภทข้อมูลโดยใช้วิธีแบบดั้งเดิมประกอบไปด้วย อัลกอริทึม C4.5 อัลกอริทึม RIPPER และอัลกอริทึม OneR
2. กลุ่ม CAR หมายถึง อัลกอริทึมที่เกี่ยวข้องกับการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์ ประกอบไปด้วย อัลกอริทึม CBA อัลกอริทึม GARC และอัลกอริทึม OAC
3. กลุ่ม FCARหมายถึง อัลกอริทึมที่เกี่ยวข้องกับการจำแนกประเภทข้อมูลด้วยกฎความสัมพันธ์แบบคลุมเครือ ประกอบไปด้วย อัลกอริทึม FURIA อัลกอริทึม CFAR และอัลกอริทึม CFARC

เกณฑ์ที่ใช้ในการเปรียบเทียบ คือ ค่าความถูกต้องในการจำแนกข้อมูล (Acc) จำนวนของกฎที่ได้รับ และค่าความเหมาะสมของกฎ (SR) ซึ่งรายละเอียดของเกณฑ์ต่าง ๆ แสดงในหัวข้อ 2.5.2 ส่วนชุดข้อมูลที่ใช้ในการประมวลผล คือ ชุดข้อมูล Iris ชุดข้อมูล Heart และชุดข้อมูล Pima (รายละเอียดของแต่ละชุดข้อมูลแสดงในตารางที่ 4.1) และได้แสดงผลการทดสอบดังตารางที่ 4.7 ถึง 4.9 ส่วนการวัดประสิทธิภาพใช้ 10 Fold Cross-Validation

ตารางที่ 4.7ผลการทดสอบค่าความถูกต้องของการจำแนกข้อมูลด้วยอัลกอริทึม CCFAR และอีก 9 อัลกอริทึม

ชุดข้อมูล	CLASS			CAR			FCAR			
	C4.5	RIPPER	OneR	CBA	GARC	OAC	FURIA	CFAR	CFARC	CCFAR*
Iris	0.933	0.933	0.940	0.929	0.960	0.940	0.947	0.913	0.959	0.960
Heart	0.770	0.822	0.729	0.815	0.880	0.811	0.797	-	0.774	0.781
Pima	0.734	0.747	0.724	0.724	0.762	0.781	0.747	0.651	0.729	0.747
Avg (Acc)	0.812	0.834	0.797	0.822	0.867	0.844	0.830	0.782	0.820	0.828
Rank	8	3	9	6	1	2	5	10	7	4

ตารางที่ 4.8ผลการทดสอบจำนวนกฎและความกะทัดรัดของกฎแบบปกติด้วยอัลกอริทึม CCFAR และอีก 9 อัลกอริทึม

ชุดข้อมูล	CLASS			CAR			FCAR			
	C4.5	RIPPER	OneR	CBA	GARC	OAC	FURIA	CFAR	CFARC	CCFAR*
Iris	4.9	3.6	3.0	5.0	7.0	9.0	4.4	9.1	3	3
Heart	17.4	4.1	2.0	52.0	12.0	157.0	8.4	-	2.7	3
Pima	20.3	4.1	7.8	45.0	6.0	112.0	8.5	2.0	2	4
Avg (CV)	14.2	3.9	4.2	34	8.3	92.6	7.1	5.5	2.6	3.3
NCV	0.86	0.983	0.980	0.64	0.93	0	0.94	0.96	1	0.99
Rank	8	3	4	9	7	10	6	5	1	2

ตารางที่ 4.9ผลการทดสอบค่าความเหมาะสมของกฎด้วยอัลกอริทึม CCFAR และอีก 9 อัลกอริทึม

ชุดข้อมูล	CLASS			CAR			FCAR			
	C4.5	RIPPER	OneR	CBA	GARC	OAC	FURIA	CFAR	CFARC	CCFAR*
SR	.8400	.9080	.8880	.7350	.9000	.4220	.8890	.8730	.9100	.9104
Rank	8	3	6	9	4	10	5	7	2	1

ผลการทดสอบตารางที่ 4.7 - 4.8บางส่วนได้อ้างอิงจากงานวิจัยอื่น ซึ่งมีรายละเอียดดังนี้

1. ผลการทดสอบของอัลกอริทึม FURAI อ้างอิงจากงานวิจัยของ(Maet al., 2013)

2. ผลการทดสอบของอัลกอริทึมCFARอ้างอิงจากงานวิจัยของ(Alcala-Fdez et al., 2011)

3. ผลการทดสอบของอัลกอริทึมCBA GARC OAC และ CFARCอ้างอิงจากงานวิจัยของ (Pachet al., 2008)

จากผลการทดสอบตารางที่ 4.7 ถึง 4.9สามารถอภิปรายผลการทดสอบของแต่ละชุดข้อมูลได้ดังนี้

- ชุดข้อมูล Irisประสิทธิภาพในการจำแนกข้อมูลของอัลกอริทึม GARC และอัลกอริทึม CCFAR ที่ผู้วิจัยพัฒนาจะมีค่าความถูกต้องในการจำแนกข้อมูลสูงที่สุด คือ 0.96 และอัลกอริทึม CCFAR ยังมีจำนวนกฎที่ได้รับน้อยที่สุดอีกด้วย คือ จำนวน3 กฎส่วนอัลกอริทึมที่มีค่าความถูกต้องในการจำแนกข้อมูลน้อยที่สุด คือ CFAR ที่มีค่าความถูกต้องเท่ากับ 0.913
- ชุดข้อมูล Heart ประสิทธิภาพในการจำแนกข้อมูลของอัลกอริทึม GARCจะมีค่าความถูกต้องในการจำแนกข้อมูลสูงที่สุด คือ 0.880 และอัลกอริทึม OneRมีจำนวนกฎที่ได้รับน้อยที่สุด ส่วนอัลกอริทึม CCFAR ที่ผู้วิจัยพัฒนาจะมีค่าความถูกต้องในการจำแนกข้อมูลอยู่ที่อันดับ 6 และจำนวนกฎที่ได้รับอยู่ในอันดับที่ 3
- ชุดข้อมูล Pima ประสิทธิภาพในการจำแนกข้อมูลของอัลกอริทึม OACจะมีค่าความถูกต้องในการจำแนกข้อมูลสูงที่สุด คือ 0.781 แต่จำนวนกฎที่ได้รับมีมากถึง 112 กฎ ซึ่งจะแตกต่างกับอัลกอริทึม CCFAR ที่ผู้วิจัยพัฒนาที่มีค่าความถูกต้องในการจำแนกข้อมูลอยู่ที่อันดับ 3คือ0.747แต่จำนวนกฎที่ได้รับมีเพียง 4 กฎ

จากการอภิปรายผลสามารถสรุปได้ว่าอัลกอริทึม CCFAR ที่ผู้วิจัยพัฒนามีประสิทธิภาพในการจำแนกข้อมูลเฉลี่ยแล้วอยู่ในอันดับที่ 4 จากทั้งหมด 10 อัลกอริทึม และจำนวนกฎที่ได้รับมีปริมาณน้อยมากซึ่งอยู่ในอันดับที่ 2 แต่เมื่อนำค่าความถูกต้องและจำนวนกฎมาพิจารณาร่วมกัน ผลที่ได้คือ อัลกอริทึม CCFAR จะมีความเหมาะสมของกฎอยู่ในอันดับที่ 1คือ 0.9104ซึ่งมีค่าความเหมาะสมของกฎที่แตกต่างจากอัลกอริทึม CFARC ที่เป็นลำดับ 2 ไม่มาก แต่ถ้าเทียบความครอบคลุมค่าที่เป็นไปได้ทั้งหมดของข้อมูลแล้ว จะสังเกตได้ว่าอัลกอริทึมที่ผู้วิจัยพัฒนาจะสามารถทำนายค่าที่เป็นไปได้ทั้งหมดของข้อมูลคือ low, medium และ highเพราะกฎที่ได้รับมีจำนวนน้อยที่สุดคือ 3 กฎจากทุกชุดข้อมูล (ตารางที่ 4.8)ยกตัวอย่างเช่น กฎที่อัลกอริทึม CCFAR ได้รับทั้ง 3 กฎ คือ If low then A, If medium then B และ If highthen C ซึ่งถ้ามีข้อมูลใหม่เข้ามาทดสอบเป็น low, medium และ high จากทั้ง 3 กฎสามารถทำนายได้ทันทีว่าปัจจัยดังกล่าวจะเป็น A, B หรือ C แต่ถ้าเป็นอัลกอริทึม CFARC ที่เป็นลำดับ 2ของค่าเหมาะสมของกฎจะเห็นได้ว่า จากตารางที่ 4.8 กฎที่ได้รับมีจำนวนน้อยที่สุดคือ 2 กฎ ซึ่งหมายความว่าสามารถทำนายข้อมูลได้สูงสุดได้ 2 ปัจจัย เช่น กฎที่ได้รับคือ If low then A และ If medium then B และข้อมูลมีที่ต้องการทดสอบคือlow, medium และ high จากทั้งสองกฎจะไม่สามารถทำนายข้อมูลที่มีปัจจัยเป็น high ได้ จึงสรุปได้ว่าอัลกอริทึมที่ผู้วิจัยพัฒนาจะสามารถทำนายค่าที่เป็นไปได้ทั้งหมดของข้อมูล แต่อัลกอริทึมCFARC ไม่สามารถทำนายค่าที่เป็นไปได้ทั้งหมดของข้อมูลได้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

การจำแนกประเภทข้อมูลมีประโยชน์อย่างมากในการระบุปัจจัยที่จะทำให้เกิดบางสิ่งในอนาคต ยกตัวอย่างเช่น ปัจจัยที่จะทำให้เกิดไฟไหม้ป่าปัจจัยที่ทำให้เกิดแผ่นดินไหว หรือแม้กระทั่งปัจจัยที่จะทำให้เกิดโรคมะเร็งโดยปัจจัยเหล่านี้จะอยู่ในรูปแบบที่เรียกว่ากฎ เพราะฉะนั้นในการวิเคราะห์หากฎเหล่านี้มาใช้งาน จำเป็นจะต้องเลือกกฎที่มีความถูกต้องที่สูงและสามารถนำไปประยุกต์ใช้งานได้ซึ่งหมายถึงกฎที่ได้มานั้นไม่เยอะจนไม่สามารถนำมาใช้งานได้

ดังนั้นงานวิจัยนี้จึงมุ่งเน้นในการพัฒนาวิธีการเพื่อวิเคราะห์หาและเลือกกฎที่มีประสิทธิภาพในการจำแนกที่สูงและกฎที่ได้รับมีจำนวนน้อย ซึ่งอัลกอริทึมที่ผู้วิจัยพัฒนานี้มีชื่อว่า Classification with Compact Fuzzy Association Rules หรือ CCFARโดยเป็นอัลกอริทึมที่นำเทคนิคการหากฎความสัมพันธ์มาช่วยในการวิเคราะห์หาความสัมพันธ์กันของข้อมูล และใช้เทคนิคฟัซซีเซตมาช่วยในการแก้ปัญหาข้อมูลตัวเลขนำเข้าที่มีลักษณะเป็นค่าต่อเนื่อง อีกทั้งเทคนิคดังกล่าวยังเพิ่มความถูกต้องในการจำแนกข้อมูลที่มีการซ้อนทับกันมากอีกด้วย ซึ่งขั้นตอนวิธีการของงานวิจัยนี้ได้ใช้อัลกอริทึม CFARC (Pach et al., 2008) เป็นพื้นฐานในการพัฒนา โดยวิทยานิพนธ์นี้ได้เพิ่มเติมขั้นตอน การตรวจสอบข้อมูลก่อนการประมวลผล (ขั้นตอนที่ 1) และขั้นตอนการคัดเลือกกฎ (ขั้นตอนที่ 5) นอกจากนี้ยังได้ปรับเปลี่ยนเทคนิคการคำนวณคะแนนของกฎ FCARs ในขั้นตอนที่ 4 โดยสรุปแล้วอัลกอริทึม CCFAR ที่พัฒนาขึ้นประกอบด้วยขั้นตอนดังต่อไปนี้

1) การตรวจสอบข้อมูลก่อนการประมวลผล
2) ทำการแปลงข้อมูลตัวเลขค่าต่อเนื่องให้เป็นข้อมูลแบบฟัซซี (Pach et al., 2008)
3) สร้างไอเท็มเซตที่ปรากฏบ่อยแบบคลุมเครือ(Pach et al., 2008)
4) นำไอเท็มเซตที่ปรากฏบ่อยแบบคลุมเครือไปสร้างกฎความสัมพันธ์แบบคลุมเครือ และทำการเลือกกฎFCARsที่มีค่าคะแนนเป็นบวก (สมการที่ใช้คำนวณค่าคะแนนของกฎ FCARsผู้วิจัยงานนี้ได้ทำการพัฒนาสมการขึ้นมาใหม่)
5) โดยขั้นตอนนี้จะเป็นการคัดเลือกกฎไปใช้งาน ซึ่งจะประกอบด้วย 4 ส่วนย่อย คือ
   - การเลือกกฎFCARs ที่มีค่าคะแนนมากที่สุดของแต่ละคลาสและแต่ละขนาด

- การนับความถี่ของแต่ละแอตทริบิวต์จากกฎ FCARs ในส่วนย่อยที่ 1 และทำการเลือกแอตทริบิวต์ที่มีความถี่มากที่สุด
- การคัดเลือกกฎ FCARs ที่ดีที่สุดด้วยแอตทริบิวต์ที่มีความถี่มากที่สุด
- ส่วนสุดท้ายขั้นตอนย่อยที่ 4 เป็นการลบกฎที่มีลักษณะซ้ำซ้อน

การทดสอบประสิทธิภาพของอัลกอริทึม CCFAR มี 3 แบบ คือ การทดสอบหาเกณฑ์ที่เหมาะสมสำหรับการสร้างกฎ FCARs ของอัลกอริทึม CCFAR การทดสอบลักษณะการกระจายตัวของข้อมูลที่เหมาะสมสำหรับอัลกอริทึม CCFAR และการเปรียบเทียบประสิทธิภาพการจำแนกข้อมูลและจำนวนกฎที่ได้รับของอัลกอริทึม CCFAR กับอีก 9 อัลกอริทึม ซึ่งเกณฑ์ที่ใช้ในการทดสอบประสิทธิภาพ คือ ค่าความถูกต้องในการจำแนกข้อมูล จำนวนกฎที่ได้รับและค่าความเหมาะสมของกฎ

5.1 สรุปผลการวิจัย

จากผลการทดสอบประสิทธิภาพของอัลกอริทึมCCFAR ทั้ง 3 แบบ สามารถสรุปผลการทดสอบของแต่ละแบบได้ดังนี้

1) การทดสอบหาเกณฑ์ที่เหมาะสมสำหรับการสร้างกฎ FCARs ของอัลกอริทึม CCFAR ผลที่ได้ คือ จำนวนกฎที่ได้รับจากการทดสอบมีจำนวนที่แตกต่างกันน้อยมาก จึงไม่สามารถบ่งบอกถึงความแตกต่างได้อย่างชัดเจน ดังนั้นการทดสอบนี้จึงใช้ค่าความถูกต้องในการจำแนกข้อมูลเป็นหลักในการเลือกเกณฑ์ที่เหมาะสม ซึ่งผลการทดสอบแสดงให้เห็นว่า ถ้าใช้เกณฑ์ {FCORR, FC,β} ในการคำนวณค่าคะแนนจะทำให้ผลการจำแนกข้อมูลมีค่าถูกต้องมากที่สุด
2) การทดสอบลักษณะการกระจายตัวของข้อมูลที่เหมาะสมสำหรับอัลกอริทึม CCFAR ผลที่ได้แสดงให้เห็นว่าอัลกอริทึมCCFARไม่ได้มีความถูกต้องในการจำแนกข้อมูลสูงสุดเมื่อเทียบกับอัลกอริทึมที่ไม่ได้ใช้เทคนิคฟัซซีเซต แต่จำนวนกฎ FCARs ที่ได้รับมีจำนวนน้อยที่สุด และเมื่อเปรียบเทียบค่าความเหมาะสมของกฎอัลกอริทึม CCFAR จะมีค่าสูงสุด ส่วนข้อมูลที่กระจายตัวได้เหมาะสมกับอัลกอริทึม CCFARคือ ข้อมูลที่มีการกระจายตัวซ้อนทับกันมากเพราะเมื่อเปรียบเทียบกับอัลกอริทึมที่ได้อันดับ 1 ค่าความถูกต้องที่ได้มีความแตกต่างกันน้อยมากและกฎที่ได้มีจำนวนน้อยที่สุดเป็นอันดับที่ 1
3) การเปรียบเทียบประสิทธิภาพการจำแนกข้อมูลและจำนวนกฎที่ได้รับของอัลกอริทึม CCFAR กับอีก 9 อัลกอริทึมผลการทดสอบแสดงให้เห็นว่าอัลกอริทึมCCFARมี

ความถูกต้องในการจำแนกข้อมูลอยู่ในอันดับที่ 4 จากทั้งหมด 10 อัลกอริทึม และจำนวนกฎที่ได้รับมีปริมาณน้อยมากซึ่งอยู่ในอันดับที่ 2 แต่เมื่อเปรียบเทียบค่าความเหมาะสมของกฎ อัลกอริทึม CCFARจะมีค่าสูงสุดซึ่งอยู่ในอับดับที่ 1 จากทั้งหมด 10 อัลกอริทึม ซึ่งมีค่าความเหมาะสมของกฎที่แตกต่างจากอัลกอริทึม CFARC ที่เป็นลำดับ 2 ไม่มาก แต่ถ้าเทียบความครอบคลุมค่าที่เป็นไปได้ทั้งหมดของข้อมูลแล้วจะสังเกตได้ว่าอัลกอริทึมที่ผู้วิจัยพัฒนาจะสามารถทำนายค่าที่เป็นไปได้ทั้งหมดของข้อมูล แต่อัลกอริทึมCFARC ไม่สามารถทำนายค่าที่เป็นไปได้ทั้งหมดของข้อมูลได้ (รายละเอียดเพิ่มเติมติดตามได้ในหัวข้อ 4.3)

จากการสรุปทดสอบประสิทธิภาพของอัลกอริทึมCCFAR ทั้ง 3 แบบ แสดงให้เห็นว่าอัลกอริทึม CCFAR ที่ผู้วิจัยพัฒนาเหมาะสำหรับข้อมูลที่มีการกระจายตัวซ้อนทับกันมาก และเมื่อเปรียบเทียบประสิทธิภาพในการจำแนกข้อมูลกับอัลกอริทึมอื่น อัลกอริทึม CCFAR ไม่ได้มีความถูกต้องในการจำแนกข้อมูลสูงสุด แต่จำนวนกฎที่ได้รับมีจำนวนน้อยมาก และเมื่อนำค่าความถูกต้องและจำนวนกฎมาพิจารณาร่วมกันซึ่งเรียกว่าค่าความเหมาะสมของกฎ อัลกอริทึม CCFAR จะมีค่าความเหมาะสมของกฎที่มากที่สุด

5.2 ปัญหาและข้อเสนอแนะ

กฎ FCARs ที่ได้รับจากอัลกอริทึม CCFARจะอยู่ในรูปแบบของฟัซซีเซต เพราะฉะนั้นในการนำไปประยุกต์ใช้งานจะต้องแปลงข้อมูลดังกล่าวให้อยู่ในรูปแบบของช่วงข้อมูลเสียก่อน และในงานวิจัยนี้ได้กำหนดระดับความเป็นสมาชิกของฟัซซีเซตให้เป็น 3 ระดับ ได้แก่ ระดับ Low ระดับ Medium และระดับ High เพราะสามารถตีความได้ง่ายและสามารถนำไปเปรียบเทียบกับอัลกอริทึมอื่นได้ ซึ่งงานวิจัยในอนาคตอาจจะต้องเพิ่มความสามารถในส่วนนี้ที่ทำให้อัลกอริทึมสามารถกำหนดระดับความสมาชิกของฟัซซีเซตได้หลายระดับมากขึ้น

รายการอ้างอิง

อาทิตย์ ศรีแก้ว (2552).ฟัซซีลอจิก. **ปัญญาเชิงคำนวณ**.(หน้า 427-471).สาขาวิชาวิศวกรรมไฟฟ้า: มหาวิทยาลัยเทคโนโลยีสุรนารี.

Agrawal, R., Imieliński,T.and Swami, A. (1993). Mining association rules between sets of items in large databases. **ACM SIGMOD Record**.22(2): 207-216.

Agrawal, R.andSrikant, R. (1994).Fast algorithms for mining association rules.**In Proceedings of 20th International Conference on Very Large Data Bases**.1215: 487-499.

Alcala-Fdez, J., Alcala, R.and Herrera, F. (2011).A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning. **Fuzzy Systems**. 19(5): 857-872.

BayardoJr, R. J., Agrawal, R.and Gunopulos, D. (1999).Constraint-based rule mining in large, dense databases.**In Proceedings of 15th International Conference on Data Engineering**.:188-197.

Bezdek, J. C., Ehrlich, R. and Full, W. 1984. FCM: The fuzzy c-means clustering algorithm. **Computers and Geosciences**, 10(2): 191-203.

Chen, S., Hanzo, L., and Wolfgang, A. (2004).Kernel-based nonlinear beamforming construction using orthogonal forward selection with Fisher ratio class separability measure.**IEEE Signal Processing Letters**. 11(5): 478-481.

Chen, G., Liu, H., Yu, L., Wei, Q.and Zhang, X. (2006). A new approach to classification based on association rule mining. **Decision Support Systems**. 42(2): 674-689.

Chen, Z.and Chen, G. (2008). Building an associative classifier based on fuzzy association rules. **International Journal of Computational Intelligence Systems**. 1(3): 262-273.

Cohen, W. W. (1995). Fast effective rule induction. **In Proceedings of the Twelfth International Conference on Machine Learning, Lake Tahoe, California**.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P.andUthurusamy, R. (1996).**Advances in Knowledge Discovery and Data Mining.**

Fazzolari, M., Alcala, R., Nojima, Y., Ishibuchi, H.and Herrera, F. (2013). Improving a fuzzy association rule-based classification model by granularity learning based on heuristic

measures over multiple granularities. **In Proceedings of2013 IEEE International Workshop on Genetic and Evolutionary Fuzzy Systems (GEFS)**.: 44-51.

Han, J.andKamber, M. (2006). Data Mining, Southeast Asia Edition: Concepts and Techniques. **Morgan Kaufmann**.

Han, J., Jian P. and Yiwen, Y. (2000).Mining frequent patterns without candidate generation.**ACM SIGMOD Record**. 29(2): 1-12.

He, J.(2014).“Fuzzy expert systems” [Online].Available: **http://users.aber.ac.uk/jqh/csm6320**

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. **Machine Learning**. 11(1): 63-90.

Hong, T. P., Lin, K. Y.and Wang, S. L. (2003). Fuzzy data mining for interesting generalized association rules. **Fuzzy Sets and Systems**. 138(2): 255-269.

Hu, H.and Li, J. (2005). Using association rules to make rule-based classifiers robust. **In Proceedings ofthe 16th Australasian Database Conference**.39.: 47-54.

Hühn, J.andHüllermeier, E. (2009). FURIA: an algorithm for unordered fuzzy rule induction**. Data Mining and Knowledge Discovery**. 19(3): 293-319.

Ishibuchi, H., Nakashima, T.and Yamamoto, T. (2001). Fuzzy association rules for handling continuous attributes. In, 2001.**In Proceedings of International Symposium on Industrial Electronics**.1: 118-121.

Keller, J., Krisnapuram, R.and Pal, N. R. (2005).Fuzzy models and algorithms for pattern recognition and image processing.**The Handbooks of Fuzzy Sets, Vol. 4**.

Kruse, R., Nauck, D.andBorgelt, C. (1999). Data mining with fuzzy methods: status and perspectives. **In Proceedings of 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT’99)**.

Li, J., Shen, H.andTopor, R. (2002).Mining the optimal class association rule set.**Knowledge-Based Systems**. 15(7): 399-405.

Liu, B., Hsu, W., Ma, Y. (1998). Integrating classification and association rule mining.**In Proceedings of the 4th American Association for Artificial Intelligence.**

Ma, Y., Chen, G., and Wei, Q. (2013). A novel fuzzy associative classifier based on information gain and rule-covering. **In Proceedings of IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS**.: 490-495).

Nauck, D., Klawonn, F.and Kruse, R. (1997).**Foundations of Neuro-Fuzzy Systems**.John Wiley and Sons.

Pach, F. P., Gyenesei, A.andAbonyi, J. (2008).Compact fuzzy association rule-based classifier. **Expert Systems with Applications**. 34(4): 2406-2416.

Palanisamy, S. K. (2006). Association Rule Based Classification (Doctoral dissertation, Worcester Polytechnic Institute).

Quinlan, J. R. (1986). Induction of decision trees.**Machine Learning**. 1: 81-106.

Quinlan, J. R. (1993). C4.5: programs for machine learning. **Morgan Kaufmann.Vol.1**.

Quinlan, J. R.and Cameron-Jones, R. M. (1993). FOIL: A midterm report. **Machine Learning**. ECML 93: 1-20.

Seifert, J. W. (2004). Data mining: An overview. **National Security Issues**.201-217.

Zadeh, L. A. (1965). Fuzzy sets. **Information and Control**, 8(3): 338-353.

Zhang, C.and Zhang, S. (2002). Association rule mining: models and algorithms. **Springer VerlagBerlin**.: 228.

ภาคผนวก ก

รหัสต้นฉบับของโปรแกรม CCFAR

%**ฟังก์ชันหลัก**

```
%clear all;
clc
path(path,'C:\Program Files\MATLAB\R2013b\toolbox\FUZZCLUST');

%the data
filename = 'filename.txt';

delimiterIn = ',';
headerlinesIn = 1;
A = importdata(filename,delimiterIn,headerlinesIn);

data1 = A.data;
s = size(data1);
class = data1(:,end);

%# define class from your dataset
%# pima = 1 and 2, test = 0 , 1
defined_class = sort(unique(class));
fori=1:length(defined_class)
defclass{i} = defined_class(i);
end

%# fuzzyness with membership function
[testdata, traindata] = fuzzyness2(data1,s);

%# define Kfold
k =10;
```

```
cvFolds = crossvalind('Kfold', class, k);
%# start cross-validation
collect_acc = [];
collect_rule = 0;
fori = 1:k                                  %# for each fold
testIdx = (cvFolds == i);                   %# get indices of test instances
trainIdx = ~testIdx;                        %# get indices training instances
train_new = traindata(trainIdx,:);
train_row = size(train_new);
train_class = class(trainIdx);
   %# compute min-support
for j=1:length(defclass)
comp_class(j) = sum(train_class==defclass{j});
end
minclass = defclass{ comp_class == min(comp_class) };
min_sup = ( length(train_class(train_class==minclass))/train_row(1) )/2

%# disp('----------------------------main-----------------------------------------');
      [rules, num_rule] = CCFAR(train_new(:) , train_row(1), min_sup, s, class(trainIdx),
defclass, A.colheaders);
collect_rule = collect_rule+num_rule;

%# disp('--------------------------performance-------------------------------------');
test_new =testdata(testIdx,:);
test_class = class(testIdx);
Acc = performance(test_new, rules, test_class);
collect_acc = vertcat(collect_acc, Acc);
str = ['Accuracy =', num2str(Acc)];
disp(str);
end
```

```
%# disp('----------------------------Show ACC-------------------------------------');
collect_acc
collect_rule = (collect_rule/k)
str = ['Mean-Accuracy =', num2str(mean(collect_acc))];
disp(str);
```

%ฟังก์ชัน CCFAR

```
function [Bestrules, num_rule] = CCFAR (item , row, min_sup, s, class, defclass, Atrr_name)
fori=1:3*(s(2)-1)
   pack_1{i} = i;
end

[item_1, sup_1] = find_sup(item , pack_1, row); %# find support of 1-itemset
fre_1item = item_1(sup_1 >min_sup);                %# find frequent 1-itemset
sup_1item = sup_1(sup_1 >min_sup);
disp('fre_1item =');
disp(fre_1item);
disp(sup_1item);

[combi_2] = combi2item(s,fre_1item);
[item_2, sup_2] = find_sup(item , combi_2, row);%# find support of 2-itemset
fre_2item = item_2(sup_2 >min_sup);                %# find 2-frequent itemset
sup_2item = sup_2(sup_2 >min_sup);
disp('fre_2item =');
fre_2item{:}
disp(sup_2item);

pack{1} = zeros(1,10);
pack{1}(1) = 1;
freq{1} =fre_1item;
supp{1} =sup_1item;
freq{2} =fre_2item;
supp{2} =sup_2item;
```

```
%# find frequent itemset more than 2 item
fori=1:1
pack = combi_item(freq{i+1},i-1);
if pack{1}(1)~=0
      [citem, support] = find_sup(item , pack, row);
di = ['fre_',num2str((i+2)),'item ='];
disp(di);
freq{i+2} = citem(support >min_sup);
supp{i+2} = support(support >min_sup);
freq{i+2}{:}
disp('Support =');
disp(supp{i+2});
else
break;
end
end

%find Score + pruning step 1
%fixed 2 item
k = 1;
mm =0;
data_rule{1} = [];
data_score = [];
data_class = [];
fori=1:length(freq)
if ~isempty(freq{i})
   [positive_rule, positive_score, max_rule, max_score, max_class] = find_score(item ,
freq{i}, row, class, supp{i}, defclass);
   %# pack max score
if ~isempty(max_rule)
```

```
for ii=1:length(max_rule)
if ~isempty( max_rule )
data_rule{k} = max_rule{ii};
data_score = vertcat(data_score , max_score(ii));
data_class = vertcat(data_class , max_class(ii));
                k = k+1;
end
end
end
   %# pack positive rules
for iii=1:length(defclass)
kk =1;
ifi> 1
mm = length(all_positive_rule{iii});
end
if ~isempty(positive_rule{iii})
forjjj=1:length(positive_rule{iii})
all_positive_rule{iii}(:,:,kk+mm) = positive_rule{iii}(:,:,jjj);
all_positive_score{iii}(:,:,kk+mm) = positive_score{iii}(:,:,jjj);
kk = kk+1;
end
end
end
end
end
```

```
% #disp('----------My-Prunningstep----------');
[prule, pclass] = my_prunning(data_rule, all_positive_rule, all_positive_score, defclass, row,
item, class);
[bestR, bestclass] = cut_redundancy_rule(prule, pclass, defclass)
Bestrules = zeros(15,10);
m = 1;
k = 1;
fori=1:length(bestclass)
if ~isempty(bestR{i})
len = length(bestR{i}{:});
Bestrules(m,1: len) = bestR{i}{:}
      % convert rule form if 1,2 then 1 to if Age = low, Inc = Me then 1
      %transRule(m,1: len) = floor( bestR{i}{:}/3 )
for j=1:len
         %Change number to name-attr
Rule{m}{j} = Atrr_name( floor( (bestR{i}{1}(1,j)-1 )/3 ) + 1  ); %+2 cuase start from 0
         %degree of attribute
if  mod( bestR{i}{1}(1,j), 3 ) == 0
degreeRule2{m}{j} = 'High';
elseif mod( bestR{i}{1}(1,j), 3 ) == 1
degreeRule2{m}{j} = 'Low';
elseif mod( bestR{i}{1}(1,j), 3 ) == 2
degreeRule2{m}{j} = 'Medium';
end
end
Rule{m}{len+1} = {'then'};
Rule{m}{len+2} = num2str( bestclass{i} );
Rule{m}{:}
degreeRule2{m}{:}
Bestrules(m,end-1:end ) = [lenbestclass{i}];
```

```
  m=m+1;
end
end

%# pack rules
num_rule = m-1;
disp('Rules = ');
disp(Bestrules);
fori=1:length(Rule)
sprintf('Rules = %d', i)
frist = [];
for j=1:length(Rule{i})
if j == 1
frist = strcat(Rule{i}{j}, '=', degreeRule2{i}{j});
elseif j <= (length(Rule{i})- 2)
frist = strcat(frist,',', Rule{i}{j}, '=', degreeRule2{i}{j});
else
frist = strcat(frist, {' '} ,Rule{i}{j});
end
end
disp(frist);
end
end
```

%ฟังก์ชัน fuzzyness2

```
function [test,item] = fuzzyness2(data1,s)
item = [];
new_group = [];
fori=1:s(2)-1
      [prev_group,prev_data] = mbfunction(i,data1);
item = horzcat(item ,prev_data);
new_group = horzcat(new_group ,prev_group);
end

%# change groupdata Ex [1 3 2; 2 2 1] -> [1 6 8; 2 5 7]
fori=1:s(2)-1
for j=1:length(new_group)
test( j, i ) = new_group( j, i ) + ( (i-1)*3 );
end
end
end
```

%ฟังก์ชัน mbfunction

```
function [group,fuzzyset] = mbfunction(x,data1)
data.X = data1(:,[x]);
   %parameters
param.c=3;
param.m=1;
param.e=1e-4;
   %normalas = 1;
param.val=1;
data=clust_normalize(data,'range');
result = FCMclust(data,param);
param.c = result.data.f; %#ok<STRNU>
fuzzyset = [result.data.f(:,2) result.data.f(:,3) result.data.f(:,1)];
   %choose cluster
group = [];
fori=1:length(fuzzyset)
iffuzzyset(i,1) == max(fuzzyset(i,1:3))
group = vertcat(group,[1]);
elseiffuzzyset(i,2) == max(fuzzyset(i,1:3))
group = vertcat(group,[2]);
elseiffuzzyset(i,3) == max(fuzzyset(i,1:3))
group = vertcat(group,[3]);
end
end
end
```

```
%ฟังก์ชัน mbfunction
function [two_item,sup_2item] = find_sup(item , pack, row)

%find support
k =1;
fori=1:length(pack)
if pack{i}(1) ~= 0
      p = pack{i};
two_item{k} = pack{i};
for j=1:length(pack{1})

if  p(j) == 1
index_item(:,:,j) = item(p(j):row,1);
else
index_item(:,:,j) = item(((p(j)-1)*row)+1:p(j)*row,1);
end
end

   sup_2item(k) = sum(prod(index_item,3))/row ;
   k = k+1;
end
end
end
```

```
%ฟังก์ชัน find_sup
function [two_item,sup_2item] = find_sup(item , pack, row)

%find support
k =1;
fori=1:length(pack)
if pack{i}(1) ~= 0
      p = pack{i};
two_item{k} = pack{i};
for j=1:length(pack{1})

if  p(j) == 1
index_item(:,:,j) = item(p(j):row,1);
else
index_item(:,:,j) = item(((p(j)-1)*row)+1:p(j)*row,1);
end
end

   sup_2item(k) = sum(prod(index_item,3))/row ;
   k = k+1;
end
end
end
```

%ฟังก์ชัน combi2item

```
function pack = combi2item(s,f1)
num_attr = (s(2)-1)*3; %3 partition
c1 = [1:num_attr];
c2 = [1:num_attr];
c3 = [1:num_attr];
p_id = 1;
pack_1 = [];
fori=1:length(f1)
   pack_1 = vertcat(pack_1 ,f1{i});
end
sdiff = setdiff(c1,pack_1);
i = 1;
%combination 2 itemset, fuzzy 3 partition
%check pair ex. {1} {3} = {1,3}
whilei<num_attr
if ((mod(i,3))==0)
i = i+1;
else
c1(i+1) = 0;
c1(i+2) = 0;
      c1;
for j = i:c1(end)
if c1(j) ~= 0 && c1(i) ~= c1(j)
            newc1 = [c1(i) c1(j)];
pack{p_id} = newc1;
p_id = p_id+1;
end
end
c2(i) = 0;
```

```
c2(i+2) = 0;
      c2;
for k = i+1:c2(end)
if c2(k) ~= 0 && c2(i+1) ~= c2(k)
            newc2 = [c2(i+1) c2(k)];
pack{p_id} = newc2;
p_id = p_id+1;
end
end
c3(i) = 0; c3(i+1) = 0;  c3 ;
for m = i+2:c3(end)
if c3(m) ~= 0 && c3(i+2) ~= c3(m)
            newc3 = [c3(i+2) c3(m)];
pack{p_id} = newc3;
p_id = p_id+1;
end
end
pack{:};
i = i+2;
end
end
fori =1:length(pack)
for j =1:length(sdiff)
member = ismember(pack{i}, sdiff(j));
if member(1) == 1 || member(2) == 1
pack{i} = [0 , 0];
end
end
end
end
```

```
%ฟังก์ชัน combi_item
function pack = combi_item(fre_2item,checksubset)
com_bi3 =[];  k = 1;
fori =1:(length(fre_2item)-1)
for j = (i+1):length(fre_2item)
if sum(ismember(fre_2item{i}, fre_2item{j})) >checksubset
        combi_3 = union(fre_2item{i},fre_2item{j});
        com_bi3 = vertcat(com_bi3, combi_3);  k = k+1;
else
continue;
end
end
end
pack{1} = zeros(1,10);
% revmove duplicate set and add to list {} and
% check pair ex. {1,7} {1,8} = {1,7,8} ~= {1,6} {1,7} = {1,6,7}
com_bi3 = unique(com_bi3(1:end,:),'rows');
size_combi3 = size(com_bi3);
k = 1;
fori = 1:size_combi3(1)
for j = 1:length(com_bi3(i,:))
check(j) = floor( (com_bi3(i,j)-1 )/3 ); % 3 partition
end
if length(unique(check)) == length(com_bi3(i,:))
pack{k} = com_bi3(i,:); k = k+1;
else
continue;
end
end
end
```

%ฟังก์ชัน find_score

```
function [positive_rule, positive_score, max_rule, max_score, max_class] = find_score(item ,
pack, row, class, support, defclass)
%find support of class
fori=1:length(defclass)
support_class(i) = sum(class==defclass{i})/row;
end
k =1;
fori=1:length(pack)
if pack{i}(1) ~= 0
      p = pack{i};
itemset{k} = pack{i};
for j=1:length(pack{1})
if  p(j) == 1
index_item(:,:,j) = item(p(j):row);
for ii=1:length(defclass)
index_class{ii}(:,:,j) = index_item( class == defclass{ii}, :, j);
end
else
index_item(:,:,j) = item(((p(j)-1)*row)+1:p(j)*row);
for ii=1:length(defclass)
index_class{ii}(:,:,j) = index_item( class == defclass{ii}, :, j);
end
end
end
for iii=1:length(defclass)
%# Fuzzy Corelation
fcorr{iii}(:,:,k) =  ( (sum((prod(index_class{iii}(:,:,:),3))) / row)-
(support(i)*support_class(iii)) ) /sqrt(support(i)*(1-support(i))*support_class(iii)*(1-
support_class(iii))) ;
```

```
%# Fuzzy Confidence
fconf{iii}(:,:,k) = ( sum(prod(index_class{iii},3)) /row ) / support(i) ;


%# Score = fcorr * confidence * fringstreng
cor = fcorr{iii}(:,:,k);
conf =  fconf{iii}(:,:,k);
fri = sum(prod(index_class{iii}(:,:,:),3));
score{iii}(:,:,k) =  cor*conf*fri;
all_rules{iii}(:,:,k) = pack{i};
end
      k = k+1;
end
end


fori=1:length(defclass)
%# print show
   %str = ['----------------------------SCORE = class ::', num2str(i), ' ----------------------------' ];
   %disp(str);
   %disp(score{i}(:,:,:));


%# check positive Score
   k=1;
positive_rule{i}(:,:,k) = [];
positive_score{i}(:,:,k) =  [];
for j=1:length(score{i}(:,:,:))
if score{i}(:,:,j) > 0
positive_rule{i}(:,:,k) = pack(j);
positive_score{i}(:,:,k) =  score{i}(:,:,j);
          k=k+1;
end
```

```
end
end

%prunning step 1
max_rule{1} = [];
max_score = [];
max_class = [];
fori=1:length(defclass)
   if ~isempty( positive_rule{i}(positive_score{i}(:,:,:) == max(positive_score{i}(:,:,:))) )
max_rule{i} = positive_rule{i}( positive_score{i}(:,:,:) == max(positive_score{i}(:,:,:)));
max_score(i) = max(positive_score{i}(:,:,:));
max_class(i) = defclass{i};
end
end
end
```

```
%ฟังก์ชัน my_prunning
function [bestR, bestclass] = my_prunning(data_rule, all_positive_rule, all_positive_score,
defclass, row, item, class_original)

%# prunning by oneAttr
k = 1;
fori=1:length(data_rule)
if ~isempty(data_rule{i})
checkRule = data_rule{i}{:};
for j=1:length(checkRule)
check(k) = floor( (checkRule(j)-1 )/3 );
        k = k+1;
end
end
end
chk = tabulate(check)
s_chk = size(chk)
%# findOneR
fori=1:s_chk(1)
ifchk(i,end) == max(chk(:,end))
keep = chk(i,1);
end
end
oneR = [(keep*3)+1: (keep*3)+3]
%# group all rules with oneR (such as :[4 5 6])
fori=1:length(oneR)
group{i}(:,:,1) = [];
mm =1;
nn =0;
for j=1:length(defclass)
```

```
if j > 1
nn = length(group{i});
end

for k=1:length(all_positive_rule{j})
set = all_positive_rule{j}(:,:,k) ;
scr = all_positive_score{j}(:,:,k) ;
ifismember(oneR(i) ,set{:} )
group{i}(:,:,mm+nn) = set;
score{i}(:,:,mm+nn) = scr;
class{i}(:,:,mm+nn) = j;
mm = mm+1;
end
end
end
end

%# find rulesthat max score
k=1;
fori=1:length(group)
if ~isempty(group{i})
bestR{k} = group{i}(score{i} == max(score{i}));
bestclass{k} = class{i}(score{i} == max(score{i}));
      k=k+1;
      if ~( sum(bestR{k-1}{:} == oneR(i)) == length(bestR{k-1}{:}) )
         %chekตรงนี้กรณี ไม่มี supper-set เช่นมีแต่ [7 11] ไม่มี [7]
         [R_oneR, sup_oneR] = find_sup(item , {oneR(i)}, row);
         [positive_rule, positive_score, max_rule, max_score, max_class] = find_score(item ,
R_oneR, row, class_original, sup_oneR, defclass);
```

```
for ii = 1:length(positive_score)
ifisempty(positive_score{ii})
p_score(ii) = 0;
else
p_score(ii) = positive_score{ii};
end
end
bestR{k} = {oneR(i)};
bestclass{k} = max_class( p_score == max(p_score) );
        k=k+1;
end
%# ในกรณีไม่มีกฎเลยบางกลุ่ม ให้ทำการส่งเลขตัวนั้นไปหา score
%# แล้วเลือกคลาสมาใช้เลย เช่น 6 ไม่มี ให้เอา 6 ไปสร้างกฎมาใช้ทำนายเลย
else
      [R_oneR2, sup_oneR2] = find_sup(item , {oneR(i)}, row);
      [positive_rule, positive_score2, max_rule, max_score, max_class2] = find_score(item ,
R_oneR2, row, class_original, sup_oneR2, defclass);
for ii = 1:length(positive_score2)
ifisempty(positive_score2{ii})
            p_score2(ii) = 0;
else
            p_score2(ii) = positive_score2{ii};
end
end
bestR{k} = {oneR(i)};
bestclass{k} = max_class2( p_score2 == max(p_score2) );
        k=k+1;
end
end
```

```
%ฟังก์ชัน cut_redundancy_rule
function [br, cr] = cut_redundancy_rule(bestR, bestclass, defclass)

%# จัดกลุ่มกฎที่มีคลาสเดียวกัน
fori=1:length(defclass)
   k=1;
for j =1:length(bestclass)
ifbestclass{j} == cell2mat(defclass(i))
        %# จัดกลุ่มกฎที่มีคลาสเดียวกัน
groupRules{i}(:,:,k) = bestR{j};
groupClass{i}(:,:,k) = bestclass{j};
          k=k+1;
end
end
end

% นับความยาวของกฎ
fori=1:length(groupRules)
y1{i} = [];
for j=1:length(groupRules{i})
      y1{i} = vertcat(y1{i} , length(groupRules{i}{:,:,j}));
end
end

% sort กฎ ของแต่ละกลุ่ม
fori=1:length(groupRules)
y2{i} = sort(y1{i});
y2{i} = unique(y2{i});
   k=1;
```

```
for j=1:length(y2{i})
for m=1:length(y1{i})
if y1{i}(m) == y2{i}(j)
xnew{i}{1,1,k} = groupRules{i}{:,:,m};
xclass{i}{1,1,k} = groupClass{i}(:,:,m);
            k = k+1;
end
end
end
end

%ตัดกฎที่ซ้ำซ้อน
%หลักการ ถ้า |X intersect Y| == min(X,Y) แสดงว่ามี Supperset
fori=1:length(xnew)
for j=1:length(xnew{i})-1
if ~isempty(xnew{i}{1,1,j})
for m=j+1:length(xnew{i})
if ~isempty(xnew{i}{1,1,m})
minirule = min([length(xnew{i}{1,1,m}), length(xnew{i}{1,1,j})]);
            if length(intersect(xnew{i}{1,1,m}, xnew{i}{1,1,j})) == minirule
xnew{i}{1,1,m} = [];
xclass{i}{1,1,m} = [];
end
end
end
end
end
end
```

```
k = 1;
fori=1:length(xnew)
for j=1:length(xnew{i})
if ~isempty(xnew{i}{1,1,j})
rules{k} = xnew{i}{1,1,j};
class{k} = xclass{i}{1,1,j};
        k = k+1;
end
end
end

%sort max-length to min-length for check accuracy
y1 = [];
fori=1:length(rules)
   y1 = vertcat(y1 , length(rules{i}));
end

% sort กฎจากมากไปน้อย
y2 = unique(y1);
y2 = sort(y2, 'descend');
k = 1;
fori=1:length(y2)
for j=1:length(rules)
if y1(j) == y2(i)
br{k} = {rules{j}};
cr{k} = class{j};
        k = k+1;
end
end
end
```

```
%ฟังก์ชัน performance
functionAcc = performance(test_new, rules, test_class)
ts = size(test_new);
   g = 1;
fori=1:ts(1)
for j=1:length(rules)
inters = intersect(test_new(i, 1:ts(2)), rules(j, 1:rules(j, end-1)) );
s_inters = size(inters);
if  s_inters(2) == rules(j, end-1)
res(g) = rules(j, end);
break;
end
end
if res(g) == 0
res(g) = 999;
end
      g=g+1;
          end

%# test
   cm = confusionmat(test_class, res(:))          %# confusion matrix
   N = sum(cm(:));
err = ( N-sum(diag(cm)) ) / N;            %# testing error
Acc = 1-err
end
```

ภาคผนวก ข

บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่

รายชื่อบทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่

ไพชยนต์ คงไชย, นิตยา เกิดประสพ และกิตติศักดิ์ เกิดประสพ. 2556. **Dissimilar Rule Mining and Ranking Technique for Associative Classification**. ในงานประชุมวิชาการ IMECS 2013 the International Multi-Conference of Engineers and Computer Scientists 2013, Vol. I, ประเทศฮ่องกง. 13-15 มีนาคม 2556.

ไพชยนต์ คงไชย, กีระชาติ สุขสุทธิ์, รัฐพงษ์ สุธรรมมา, ศักย์ เพิ่มหรรษา, กิตติศักดิ์ เกิดประสพ และนิตยา เกิดประสพ. 2558. **The Compact Fuzzy Association Rules For Data Classification.** ในงานประชุมวิชาการ ICIAE'2015 the 3rd International Conference on Industrial Application Engineering 2015, The Institute of Industrial Applications Engineers, ประเทศญี่ปุ่น. 28-31 มีนาคม 2558.

Dissimilar Rule Mining and Ranking Technique for Associative Classification

PhaichayonKongchai*, NittayaKerdprasop, and KittisakKerdprasop

Abstract—**This research presents an associative classification with dissimilar rules(ACDR) algorithm to discover association rules with the highest priority and the top frequency. The proposed algorithm has the abillity to reduce redundant rules and to sort rules in decreasing order by their priorities. The results aredissimilarrules thatcan be used to predict information in the future. This algorithm can be appliedas an associative classification technique and then sortedthe results by interestingness measures. We develop the program with Rstudio, which is a very popular software package in statistical analysis and data mining. In the experimentation, we used the post-operative patientsdatasetto evaluate efficiency of the algorithm. The results confirm effectiveness of the ACDR algorithm by discovering a minimal but powerful set of association rules.**

Index Terms—**R Language, Association Rule, Algorithm Apriori, Associative Classification**

I. Introduction

Association rule mining is tofind the relationsamong data items from large database. The results can be used to predict future information or explain current relation. Apriori algorithm [1] is a popular method for association rule mining.This algorithm was developed based on AIS algorithm and focusedon the pruning infrequent item sets.Many open-sources software can be used to discover the frequent patterns such as WEKA,which is software that can import data into the program and the final results are association rules, RapidMinerthat has many tools for data mining and users can use operator chaining technique for mining with many algorithms in a single execution. But in this research we select the Rstudio for mining association rules because with this software, users can implement and extend algorithm easier than WEKA and RapidMiner that are Java implementation.

Rstudiois a suite of program environment to run theR language program, which is commonly language used to compute the statistics applications. This program environment provides several types of graphical display and has many libraries for discovering classification and association rules.In this research, we use the library arules because it can find the patternswith only a few lines of code.Moreover, this library was designed to allow users to specify the mining for association rules with the constraints. Withthe constraint mining feature, it was thus easier and faster to find associative patterns with the proposed ACDR (Associative Classification with DissimilarRules) algorithm.

The main contribution of this research is proposing the ACDR algorithm. It can be used to discover dissimilar rules for classification.The algorithm has 5 main steps: searching for association rules, categorizing rules into target association rules and general association rules, classifying rules into groups by their right-hand-side item (RHS), analyzing with selected agent of each group, and sorting rules.

The proposed algorithm works with any dataset, but for the demonstration purpose, we apply the algorithm to the post-operative patients dataset.

II. Related work

This research aims to reduce the number of association rules that are redundant and retain the remaining rules that are important for predictingthe future events.Kannan and Bhaskaran [4] proposed algorithm for reducing redundant rules by clustering association rules into many groups then cut redundant rules by interestingness measures.Mutter et al. [5] used CBA (Confidence-Based Association Rule Mining) algorithm to reduce the number of

Manuscript received November 30, 2012; revised January 10, 2013. This work was supported in part by grant from Suranaree University of Technology through the funding of Data Engineering Research Unit.

P. Kongchai is a doctoral student with the School of Computer Engineering, Suranaree University of Technology, Thailand, (email: zaguraba_ii@hotmail.com).

N. Kerdprasop is anassociate professor with the School of Computer Engineering, Suranaree University of Technology, Thailand.

K. Kerdprasop is anassociate professor with the School of Computer Engineering, Suranaree University of Technology, Thailand.

association rules. They ranked rules by confidence values then output rules for top hundred association patterns. Our work presented in this paper is differentfrom others in that we used associative classification technique to rank and reduce association rules.

Associative classification technique is an integrated of classification rules and association rules. The goal of this technique is to search for the results having the format "If one item or more items have occurred, then another item must occur". It is like the classification rules. Hranchotchuang et al. [3] used associative classification technique for predicting unknown class label by guessing the class label with association rules then the results will be classified with classification rules. Tang and Liao [7] proposed a new Class Based Associative Classification algorithm (CACA).Their algorithm tried to reduce the searching space and results are better accuracy of classification models.

Further this research also does the top ranking after the discovery of important association rules. The ranking technique is to sort rules in decreasing order by their priorities. There are many researches which focus on sorting rules [2], [6], [9], [10]. In this research, we use four criteria to rank priorities of the association rules. The four certeria are the size of the association rules, confidence, support and target rules.

III. METHODOLOGY

In this section wepresentACDR algorithm for discovering association rules with the highest priority and the top frequency in descending order.The process of ACDR of two main parts, (1) to mine for association rules, and (2) to analyze association rules for finding important rules. We do the ranking priorities of the association rules with RStudio program. The details of ACDR algorithm, are shown in Fig. 1. Its diagrammatic flow is presented in Fig.2. Each subsection, A to E, is explanation of ACDR algorithm through the simple running example.

Algorithm ACDR
Input: Dataset D, Target items T.
Output: Dissimilar RulesDR.

(1)$R_{rhs=1}$ = apriori(D) # $R_{rhs=1}$ = RHS #equal 1 item
(2) For each R ∈$R_{rhs=1}$ {
(3) If RHS == T {
(4) G_1= group(R)
(5) } else G_2= group(R)
(6) }
(7) R_{merge} = merge(RevDup(G_1), RevDup(G_2))
(8) MG = group_by_RHS(R_{merge})
(9) For each G∈MG {
(10) agent = find_agent(G)
(11) }
(12) DR = sort_by_4condition(agent)
(13) return DR

Fig. 1 ACDR (Associative Classification with Dissimilar Rules) algorithm.

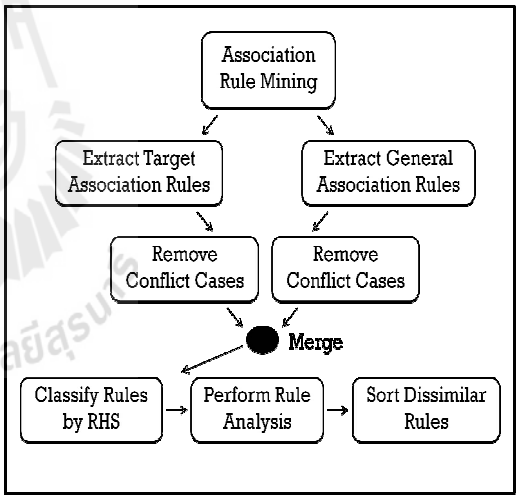


Fig. 2The process of ACDR algorithm.

A. Association Rules Mining

This research uses apriorialgorithm[1] as a basis for further extension because its association rule mining steps are simple but highly efficient pruning strategy to remove infrequent item sets with minimum support measure (eq1). Support measure of item A is proportion of number of transactions that contain A to the total number of transactions in the database.

$$\text{support}(A) = \frac{|A|}{|\text{transactions}|} \quad (1)$$

The results are frequent item sets that can be used furtherassociation rules constrained by the minimum confidence measure (eq2).

$$\text{confidence}(A \rightarrow B) = \frac{\text{support}(A \cap B)}{\text{support}(A)} \quad (2)$$

To implement the proposed methodology we are developed a program with R language which is suitable for data mining and the R system has many libraries for discovering association rules. For example to find association rules,the R code is as simple as the one show in Fig.3.

```
library(arules) # call libraryarules
Tr<read.transactions("test.txt",format="bas
ket" ,sep=",")

# read file and storing data in format
transaction.

rules<- apriori(Tr, parameter=
list(supp=0.1, conf=0.6, minlen = 2))

  # association rule mining by apriori
algorithm and set parameter with minimum
support as 0.1, minimum confidence as 0.6
and the size of rules to contain at least 2
items.
inspect(rules)

  # show all rules
```

Fig. 3 The R code for association rule mining.

From the commands in Fig.3 and the data as shown in Table 1, the result of program execution displayed in Table 2. With the simple six transactions given as the input, the output is a set of 17 association rules displayed in Table 2. These association rules have been constrained to contain exactly 1 item in the consequent part (or right-hand-side, RHS). This constraint is for later pruning the association rules.

B. Extract Target Association Rules and General Association Rules

This ACDR algorithm aims to predict or make decision on data that may occur in the future. Therefore,we applied a technique to include classification rules and association rules and call this technique is an associative classification. For our associative classification technique,we divided association rules into two groups, Thefirst group is the rules to be defined by users contain target items (called Target Rules), and the second group is the rules not defined to contain target items (called General Rules).Target and general rule extraction is the step between lines 2-6 in Fig.1. Suppose the target item defined by users are item C and D, then the extracted target association rules are those illustrated in Table 3, whereas the rest (Table 4) is a set of general association rules.

ID	Item
1	A, B, C
2	B, C
3	A, B, D
4	A, B, C, D
5	A
6	B

TABLE 1
EXAMPLE TRANSACTION DATABASE

TABLE 2
ASSOCIATION RULES WITH A SINGLE ITEM IN THEIR RHS

NO.	Rules	Support	Confidence
1	{D} => {A}	0.333	1
2	{D} => {B}	0.333	1
3	{C} => {A}	0.333	0.667
4	{C} => {B}	0.5	1
5	{B} => {C}	0.5	0.6
6	{A} =>{B}	0.5	0.75
7	{B} => {A}	0.5	0.6
8	{C,D} => {A}	0.167	1
9	{C,D} => {B}	0.167	1
10	{A,D} => {B}	0.333	1
11	{B,D} => {A}	0.333	1
12	{A,B} => {D}	0.333	0.667
13	{A,C} => {B}	0.333	1
14	{B,C} => {A}	0.333	0.667
15	{A,B} => {C}	0.333	0.667
16	{A,C,D} => {B}	0.167	1
17	{B,C,D} =>	0.167	1

	{A}		

TABLE 3
TARGET ASSOCIATION RULES THAT CONTAIN THE TARGET ITEMS C AND D IN THE RHS

NO.	Rules	Support	Confidence
5	{B} =>{C}	0.5	0.6
12	{A,B} => {D}	0.333	0.667
15	{A,B} => {C}	0.333	0.667

The rules in Tables 3 and 4 may contain conflicting cases such as rule number 12 and 15 have exactly the same antecedent parts, but they predict different consequences. We call such case a conflict. At line 7 of the ACDR algorithm (Fig.1), we remove conflicting cases from both the target and general association rules. The remaining rules are shown in Tables 5 and 6.

TABLE 4
GENERAL ASSOCIATION RULES

NO.	Rules	Support	Confidence
1	{D} => {A}	0.333	1
2	{D} => {B}	0.333	1
3	{C} => {A}	0.333	0.667
4	{C} => {B}	0.5	1
6	{A} => {B}	0.5	0.75
7	{B} => {A}	0.5	0.6
8	{C,D} => {A}	0.167	1
9	{C,D} => {B}	0.167	1
10	{A,D} => {B}	0.333	1
11	{B,D} => {A}	0.333	1
13	{A,C} => {B}	0.333	1
14	{B,C} => {A}	0.333	0.667
16	{A,C,D} => {B}	0.167	1
17	{B,C,D} => {A}	0.167	1

TABLE 5
TARGET RULES AFTER REMOVING CONFLICT CASES

NO.	Rules	Support	Confidence
5	{B}=>{C}	0.5	0.6

TABLE 6
GENERAL RULES AFTER REMOVING CONFLICT CASES

NO.	Rules	Support	Confidence
6	{A} => {B}	0.5	0.75
7	{B} =>{A}	0.5	0.6
10	{A,D} => {B}	0.333	1
11	{B,D} => {A}	0.333	1
13	{A,C} => {B}	0.333	1
14	{B,C} => {A}	0.333	0.667
16	{A,C,D} => {B}	0.167	1
17	{B,C,D} => {A}	0.167	1

C. Classify Rules by RHS(Right-Hand-Side) Item

The step at line 8 of the ACDR algorithm is to allocate rules into groups according to the items appeared in the RHS of the rules. The rule classifying strategy s as follow:

1. All items on the right hand side of association rules must be the same items. For example, from Table 6 rules 6, 10, 13 and 16 have the same item on their right hand side, which is B. Therefore, they are allocated as the same group.
2. Items on the right hand side are not the same, they will be allocated to the different groups.

From Tables 5 and 6, target and general rules are then classified into groups and the results are three groups as shown in Tables 7-9.

TABLE 7
GROUP C OF ASSOCIATION RULES

NO.	Rules	Support	Confidence
5	{B}=>{C}	0.5	0.6

TABLE 8
GROUP B OF ASSOCIATION RULES

NO.	Rules	Support	Confidence
6	{A} => {B}	0.5	0.75
10	{A,D} => {B}	0.333	1
13	{A,C} => {B}	0.333	1
16	{A,C,D} => {B}	0.167	1

TABLE 9
GROUP A OF ASSOCIATION RULES

NO.	Rules	Support	Confidence
7	{B} => {A}	0.5	0.6
11	{B,D} => {A}	0.333	1
14	{B,C} => {A}	0.333	0.667
17	{B,C,D} => {A}	0.167	1

D. Rule Analysis

After classifying rules into groups, the next step is to select agent of each group (Fig. 1 line 9-11). These agents are for rule ranking and selecting. The criteria for rule selection are:

1. Select association rules with the longest size. The reason is that they can describe the complex conditions. For example, the patient who had a first degree of the tumor, had irradiated, had surgery and a healthy body then decision is thatthe patient is recovered from cancer.
2. Select association rules with the shortest size for describing the causes that may incur the damage. For Example, the patient who had the tumor and is in the final stage then the patient is cancerous.

From the rules in Tables 7-9,after analyzing rules with two criteria, we obtain the results as shown in Tables 10 and 11. Note that a single rule in group C remains the same one as shown in Table 7.

TABLE 10
GROUP B AFTER RULE SELECTION

NO.	Rules	Support	Confidence
6	{A} => {B}	0.5	0.75
16	{A,C,D} => {B}	0.167	1

TABLE 11
GROUP A AFTER RULE SELECTION

NO.	Rules	Support	Confidence
7	{B} => {A}	0.5	0.6
17	{B,C,D} => {A}	0.167	1

E. SortDissimilar Rules

The final process is to combine the three groups into one group and then sort the rules by the following criteria (Fig. 1 line 12).

1. If association rule was the shortest size, it will then be in the first order. Iftherulesarethe same size, they will be considered by the next criterium.
2. If association rule is defined target item, it will be in the first order.
3. If association rule has the maximum confidence value, it will be in the first order. But iftherules have the same confidence value, they will be ranked by the next criterium.
4. If association rule has the maximum support value, it will be in the first order. But Iftherules have the same support value, they will be ranked by order number.

The rules in Tables 7, 10 and 11 will be merged and then sortedwith thefourcriteria. The results are shown in Table 12.

TABLE 12
ASSOCIATION RULES AFTER SORTING

NO.	Rules	Support	Confidence
5	{B} => {C}	0.5	0.6
6	{A} => {B}	0.5	0.75
7	{B} => {A}	0.5	0.6
16	{A,C,D} => {B}	0.167	1
17	{B,C,D} => {A}	0.167	1

From Table 12 association rules NO. 5 contains defined items by user (item C and D), thus it is ranked first. Association rules NO. 6 and 7 are rules of the same size, they must be ranked by confidence value.Rule NO. 6 hashigher confidence value than rule NO. 7, it is therefore ranked preceding rule No.7. Association rules NO. 16 and 17 are the same size and also the same confidence value and support value, they will be ranked according to the order number. The result is that NO. 16 has been ranked preceding rule NO. 17.

IV. EXPERIMENT

This research experimented with the post-operativepatients dataset obtained from the UCI Machine Learning Repository [8].The dataset has 8 attributes (explained in Table 13) and 90 transactions.

To perform the experiment, we developed a program usingRstudio environment and coding with R language for discovery association rules by apriori algorithm.We set minimum support and minimum confidence to be 0.01 and we define target items as ADM-DECS=I, ADM-DECS=S andADM-DECS=A.

The objectives of this experiment are to observe a decrease in the number of rules in each step of pruning associative classification rules and the efficiency of ranking important rules process (Fig. 4 and Table 14).

TABLE 13
DESCRIPTION OF POST-OPERATIVEPATIENTS' DATASET

Attribute	Description
L-CORE	patient's internal temperature in degree celsius: high (> 37), mid (>= 36 and <= 37), low (< 36)
L-SURF	patient's surface temperature in degree celsius: high (> 36.5), mid (>= 36.5 and <= 35), low (< 35)
L-O2	oxygen saturation in % excellent (>= 98), good (>= 90 and < 98), fair (>= 80 and < 90), poor (< 80)
L-BP	last measurement of blood pressure high (> 130/90), mid (<= 130/90 and >= 90/70), low (< 90/70)
SURF-STBL	stability of patient's surface temperature : stable, mod-stable, unstable
CORE-STBL	stability of patient's core temperature : stable, mod-stable, unstable
BP-STBL	patient's perceived comfort at discharge, measured asan integer between 0-10 and 11-20
ADM-DECS	discharge decision : I (patient sent to Intensive Care Unit), S (patient prepared to go home), A (patient sent to general hospital floor)

TABLE 14
THE PROCESS OF ACDR ALGORITHM AND NUMBER OF RULES AFTER PERFORMING EACH PROCESS

Process	Number of rules(Rules)
1. Association Rule Mining	88,423
2. Extracting Target Association Rules and General Association Rules	5,231
3. Classifying Rules by RHS items	5,231
4. Performing Rule Analysis	1,048
5. Sorting Dissimilar rules	1,048

The results from Table 14 are important rules discovery with five sub-processes. The first sub-process is association rule with the consequent part containing 1 item and the resultcontains 88,423 rules. The second sub-process is to find target rules and general rules and also removing conflicting cases. The result contains 5,231 rules. The third sub-process is classifying rules by their RHS, the result is the same set of rules because this step classifies rules then inserts into group but does not remove any rules. The fourth sub-process isanalyzing and selecting association rules by their sizes. The results are 1,048 rules. The last sub-process is sorting association rules, and the results are 1,048 rules.

```
1. {L-BP=low} => {ADM-DECS=A}
2. {CORE-STBL=mod-stable} => {ADM-DECS=A}
3. {BP-STBL=stable,CORE-STBL=unstable} => {ADM-
DECS=S}
4. {BP-STBL=stable,L-CORE=high} => {ADM-DECS=S}
5. {COMFORT=?,L-CORE=low} => {ADM-DECS=I}
6. {BP-STBL=stable,COMFORT=?} => {ADM-DECS=I}
7. {COMFORT=?,L-O2=good} => {ADM-DECS=I}
8. {COMFORT=?,L-SURF=mid} => {ADM-DECS=I}
9. {COMFORT=[11 - 20],CORE-STBL=unstable} => {ADM-
DECS=S}
10. {CORE-STBL=unstable,L-BP=high} => {ADM-DECS=S}
11. {CORE-STBL=unstable,L-O2=good} => {ADM-DECS=S}
12. {BP-STBL=stable,COMFORT=[0 - 10],CORE-STBL=stable,L-
BP=mid,L-CORE=mid,L-O2=excellent,L-SURF=mid,SURF-
STBL=unstable} => {ADM-DECS=A}
13. {BP-STBL=stable,COMFORT=[0 - 10],CORE-STBL=stable,L-
BP=high,L-CORE=mid,L-O2=excellent,L-SURF=mid,SURF-
STBL=stable} => {ADM-DECS=A}
14. {BP-STBL=mod-stable,COMFORT=[0 - 10],CORE-
STBL=stable,L-BP=high,L-CORE=low,L-O2=excellent,L-
SURF=mid,SURF-STBL=stable} => {ADM-DECS=A}
15. {BP-STBL=stable,COMFORT=[0 - 10],CORE-STBL=stable,L-
BP=mid,L-CORE=low,L-O2=excellent,L-SURF=low,SURF-
STBL=stable} => {ADM-DECS=A}
...
77. {BP-STBL=stable,COMFORT=[0 - 10],CORE-STBL=stable,L-
BP=mid,L-O2=excellent,L-SURF=mid,SURF-STBL=unstable}
=> {L-CORE=mid}
78. {BP-STBL=stable,COMFORT=[0 - 10],CORE-STBL=stable,L-
CORE=mid,L-O2=excellent,L-SURF=mid,SURF-STBL=unstable}
=> {L-BP=mid}
79. {BP-STBL=stable,COMFORT=[11 - 20],L-BP=mid,L-
CORE=mid,L-O2=good,L-SURF=mid,SURF-STBL=unstable} =>
{CORE-STBL=stable}
80. {BP-STBL=stable,COMFORT=[11 - 20],CORE-STBL=stable,L-
BP=mid,L-O2=good,L-SURF=mid,SURF-STBL=unstable} =>
{L-CORE=mid}
```

Fig. 4The results from ACDR algorithm.

The authors proposed an associative classification with dissimilar rules algorithm to discover association rules with the highest priority and the top frequency. The experimental results are composing of target rules and general rules. Target rules are the rules number 1-76 and general rules are the rules number 77-1,048. The rule number 1 can be interpreted as "if last measurement of blood pressure is low then discharge decision is to send the patient to general hospital floor". The symbol "?" in rule number 5 means that the attribute comfort has some effect tothe decisionADM-DECS=I but we do not know the value.

V. Conclusion

This research introduces a design approach called ACDR (Associative Classification with Dissimilar Rules) to reduce redundant target and general association rules, then the results can be used to predict information as most classification rules. The ACDR algorithm consistsof 5 main steps which are (1) finding association rules, (2) clustering target association rules and general association rules into two groups then removing redundant rules, (3) classifying rules into groups by their RHS item, (4) performing rule analysis with selected agent of each group, and (5) sorting rules according to proposed criteria. The dataset for algorithm evaluation is the post-operative patients dataset. The final result after processing the dataset through the five main steps of the ACDR algorithm is a minimal rule set containing 1,048 rules,which are significantly decreased from the original 88,423 rules.

References

[1] R. Agrawal, R. Srikant,"Fast algorithms for mining association rules,"*In Proceedings of the International Conference on Very Large Data Bases*,1994, pp. 487-499.

[2] S.Bouker,R.Saidi, S. B. Yahia, M. E. Nguifo. "Ranking and selecting association rules based ondominance relationship," *In Proceedings of the 24th IEEE International Conference on Tools with Artificial Intelligence*, 2012.

[3] W.Hranchotchuang, T. Rakthanmanon,K. Waiyamai, "Using maximal frequent itemsets for improving associative classification,"*In Proceedings of the1st National Conference on Computing and Information Technology,*2005, pp 24–25.

[4] S.Kannan; R.Bhaskaran,"Association Rule Pruning based on interestingness measures with clustering,"*IJCSI International Journal of Computer Science Issues*, V.6,2009, pp. 35-43.

[5] S.Mutter,M.Hall,E.Frank,"Using classification to evaluate the output of confidence-based association rule mining,"*In Proceedings of Australian Conference on Artificial Intelligence*,2004, pp. 538-549.

[6] G.Peyman,M. R. Sepehri,B.Azade,F. Nezam,"Ranking discovered rules fromdata mining by MADM method," *Journal of Computing Issue 11*. V.3.2011, pp. 64.

[7] Z. Tang, Q. Liao, "A new class based associative classification algorithm," *IAENG International Journal of Applied Mathematics*, 2007, Advance online publication.

[8] The UCI Repository Of Machine LearningDatabases and Domain Theories [Online]. Available:http://www.ics.uci.edu/~mlearn/MLRepository.html

[9] R.Sá. De.C,C.Soares,M. A. Jorge,P.Azevedo,J. Costa, "Mining association rules for label ranking," *In Proceedings of the 15th Pacific-Asia conference on*

Advances in knowledge discovery and data mining,2011, pp. 432-443.

The Compact Fuzzy AssociationRulesFor Data Classification

PhaichayonKongchai*, KeerachartSuksut, RattaphongSutamma, SakPhoemhansa, NittayaKerdprasop andKittisakKerdprasop

Data Engineering Research Unit, School of Computer Engineering, Suranaree University of Technology111 University Avenue, NakhonRatchasima 30000 THAILAND

*Corresponding Author:Zaguraba_ii@hotmail.com

Abstract

Data classification mining is a method to find data generalization in a form of rulesthen used these rules to predict some unknown value in the future data. But in actual applications, the rules may be of low accuracy and the number of rules may be so overwhelmedthat users could not efficiently apply them. Therefore, this research proposes the development of data classification algorithm with compact fuzzy association rules to optimize accuracy and interpretability of the model. To evaluate the performance of the proposed method, this research will compare accuracy of the classification model and the number of rules against9 different data classification algorithms. The results showed that our CCFAR algorithm is comparable in terms of accuracy. When considering both accuracy and size of model, our algorithm is the best one.

Keywords: Data Classification, Associative Classification, Fuzzy Classification Association Rule, Fuzzy Set.

1. Introduction

Data classification technique is one of a data mining widely used to predict the future data by inducing the model from sophisticated data. For instance, in a medical science the model is used to predict a patient who is risky of having a breast cancer[1].In a commercial bank, the model is used to screen credit requests and evaluate the credit rating of consumer. For its potential benefits, many researches have long been concentrating on improving the efficiency of data classification by proposing algorithms like C4.5[2] and OneR[3].

But these algorithms have a low predictive accuracy. Therefore, Liu and Ma[4]proposed a new method call the "associative classification" which is a combination between the association rule mining technique and the data classification technique.Their proposed algorithm was called the Classification based on Association Rules (CBA). The CBA has been tested its predictive performance by comparing with the C4.5, and it turns out that has high accuracy than C4.5 algorithm. This CBAhas drawn attention from many researches to develop algorithm based on associative classification technique, such as GARC[5]algorithm, OAC[6]algorithm, andmany more.However, the inherent problem of associative classification is that the association process can handle only symbolic and binary values. It cannot process continuous values. To solve this problem, we propose to use a fuzzy set to transform the continuous value to the degree of membership[7-9].

In this paper, we present the idea and the development of an algorithm called data classification with compact fuzzy association rules(CCFAR).Our main focus is to optimize both an accuracy and interpretability of the model. In addition, we applied the concept of OneR[3]algorithmto select the best rules and reduce the number of rules in the final result.

2. Related Works

Classification task is the mainstream of many researches are concentrating on developing algorithms for increasing the accuracy and reducing the number of rules. We can summarize the researches that are related to our works into 3 groups, that are group of data classification, group of associative classification, and group of fuzzy associative classification. The details of 3 groups are as follows:

Firstly, the group of data classification is the initiative concept of data mining to classify target variable with some related features. Examples of this group are as follows:

- Quinlan[2]proposed C4.5algorithm which is well known in the classification because the algorithm is able to process quickly and the model is easilyunderstandable. The main concept of this algorithm is the use of heuristic search to construct a decision tree and prunetree.

- Cohen[9]proposed an algorithm called RIPPER (Repeated Incremental Pruning to Produce Error Reduction) developed from the IREP* algorithm by the principle of growing and pruning techniques to select the low error rate rules.

- Holte[3]proposed an algorithm named OneR or One-Attribute Rule, which is an algorithm that easy-to-understand algorithm and the model is a compact set of rules. OneR is simple because it select a single attribute with the fewest error to build the model.

Secondly, the group of associative classificationuses the association mining to build theassociation rules, then using many techniques of data classification to make association rules appropriate for classification.Examples of this group are as follows:

- Liu and Ma[4]proposed CBA algorithm, which producedis a high accuracy for the data classification. The CBA composed of 2 steps: to build the model with CBA-RG (to create the association rules), and CBA-CB (to make association rules for classification by selecting the low error rate rules).

- Chen and Zhang[5]proposed an algorithm called GARC (Gain Based Association Rule Classification) with a efficacyin classifying dataused the model is a compact. Because of the GARC was using the information gain threshold to create frequent item-sets and then it used redundant and conflictive techniques to build the class association rules.

- Hu and Li[6]proposed an algorithm named OAC (Optimal Association Classifier) with a good efficacy in classifying data. The principle of OAC was to generate association rules by using the constraintof apriori[10]algorithm, then it selected the class association rules by the method call OCARM (Optimal Class Association Rule Mining[11]).

Finally, the group of fuzzy associative classificationuses the fuzzy set to controll the continuous value in the association rule mining processing. Examples of this group are as follows:

- Pach et al.[7]proposed an algorithm called CFARC (Compact Fuzzy Association Rule-Based Classifier) to resolve the problem of how to define the minimum support and minimum confidence in association rule mining byapplying fuzzy correlation threshold. In their experimental results,they showed a good accuracy and a compact set of model.

- Chen[8]proposed an algorithm named

CFAR (Classification with Fuzzy Association Rules). It generated rules by using the apriori algorithm, then it selected the rules that have highest confidence value and deleted remaining rules with lowest confidence value.

- Hühn and Hüllermeier[12]proposed an algorithm called FURIA (An Algorithm For Unordered Fuzzy Rule) based on the Ripper algorithm. But the FURIA used unordered rule set to reduce bias of the target class and it used stretching technique to obtain the generalized rules.

3. Preliminaries

In this section, we introduce the basic definitions of fuzzy sets, fuzzy association rules and fuzzy associative classification rules.

3.1 Fuzzy Sets

Fuzzy sets are sets that cannot explain something clearly.For example, an explanation of the people who are very tall, someone say that the people who are 180 cm. high are very tall, but othersmay say that the people who are 185 cm.high are very tall. So, we can see from this example that it is impossible to tell exactly regarding who is very tall.Therefore,to solve such problem Zadeh[13]proposed the concepts of fuzzy sets to explain something that is not clear with the degree of membership function (rang value of the degree is [0, 1]). For instance, the people who are 180 cm.high; they are medium tall at the degree of 0.7 and very tall at the degree of 0.3. The people who are 185 cmhigh, they are medium tall at the degree of0.1 and very tall at the degree of0.9. From this example, we can see the people who are 180 cm.high and 185 cm.high would be moderately tall and very tall,respectively with the different degrees.

Therefore, this research applied the concept of fuzzy sets to explain the continuous value of numeric data.We used fuzzy partitions with fuzzy c-means (FCM[14]) algorithm.

3.1.1 Fuzzy partitions

In this section, we present the important step of our CCFAR algorithmingfor transformthe continuous value(Fig. 1 a)to partitionswith the FCM algorithm. The FCM is used to indicate the degree of membership in a set with the value ranging from 0 to 1(Fig. 1 c), which is different from the k-means algorithm that indicates the degree of membership in a set with the discrete value (Fig. 1 b).

The processes of FCM arecomposed four main steps.

1) Initialize C and μ_{ij}

2) Calculate the center vectors by Eq.(1)

$$c_j^{(t)} = \sum_{i=1}^{N} \mu_{ij}^{(t)m} x_i \,/\, \sum_{ij}^{N} \mu_{ij}^{(t)m} \quad (1)$$

3) Compute and update membership of data by Eq.(2)

$$\mu_{ij}^{(t+1)m} = 1/\sum_{k=1}^{C} \left(\frac{||x_i - c_j||}{||x_i - c_k||}\right)^{\frac{2}{(m-1)}} \quad (2)$$

4) If $\|\mu_{ij}^{(t+1)m} - \mu_{ij}^{(t)m}\| < \varepsilon$ then STOP; otherwise repeat step 2.

Where$x_i$isdata vector, m is fuzziness that can be any real number equals or greater than 1,$\mu_{ij}$is the degree of membership of$x_i$to be in the cluster j, N is data for clustering, C is number of clusters and c_j is center of cluster j.

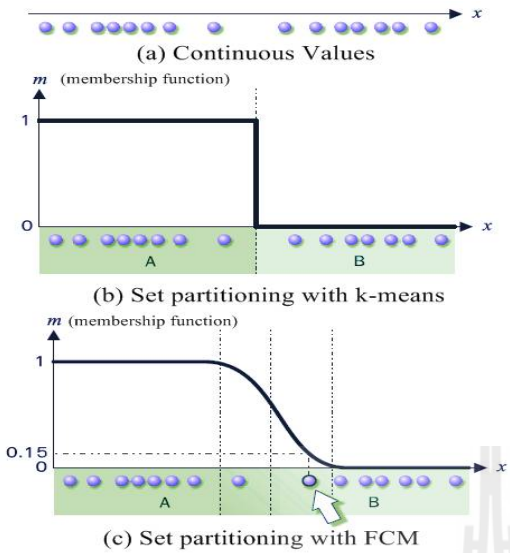


Fig. 1.The membership functions of k-means and FCM.

3.2 Fuzzy Association Rule

Association rule is originally an analysis of customer purchases (also called Market Basket Analysis) by storing the items in the basket of the customer into transaction. Then, it analyzed these transactions to discover the association rules. The association rules are expressions of the type X → Y, where X andY are sets of items. This means that if customers buy item X, then customers buy item Y together. But the processing of the association rule mining is unable to handle the continuous value (Table 1). So, we used fuzzy set to convert the continuous value into a degree of belonging to a set. For example, the Table 2shows the degree of attribute age associated with linguistic values low, medium and high.

In recent years, many researches have proposed methods to mining fuzzy association rules from continuous value[15,16]. The processing of fuzzy association rulesare composed of 2 steps. First step, create frequent Item-sets by counting item-sets that have fuzzy support (Eq.3) higher than the minimum fuzzy support.Second step, create fuzzy association rules from frequent item-sets that have fuzzy confidence (Eq.4) higher than the minimum fuzzy confidence.

$$FS(<Z:A>) = \frac{\sum_{k=1}^{N} \prod_{<z_i:A_{i,j}>\in<Z:A>} t_k(z_i)}{N} \quad (3)$$

$$FC(<X:A> \rightarrow <Y:B>) = \frac{FS(<X:A> \cup <Y:B>)}{FS(<X:A>)} \quad (4)$$

Let <Z: A>bea fuzzy item-setdenoted as<Z: A> = [$<z_{i1}: A_{i,j}> \cup <z_{i2}: A_{i,j}> \cup \ldots \cup <z_{iq}: A_{i_q,j}>$], where $q \leq n+1, z_i$ is an attribute, such as Age, Income andBalance, $A_{i,j}$ is a fuzzy interval, such as Low, Medium andHigh, t_k is a transaction, and N is the number of transactions.

Table 1.The example of data.

Id	Age
1	18
2	20
3	19
4	24
5	25

Table 2.The degrees of attribute age with FCM algorithm.

Id	Age		
	Age = Low	Age = Medium	Age = High
1	0.9863	0.0127	0.0010
2	0.0119	0.9862	0.0019
3	0.4996	0.4900	0.0104
4	0.0074	0.0141	0.9785
5	0.0053	0.0090	0.9857

3.3 Fuzzy Associative Classification Rule

Fuzzy associative classificationrule and Fuzzy association rule are similar techniques in terms of processing steps, but they differ at consequent part of rule. The fuzzy associative classificationrule must contain only one class label at consequent part of rule ($C = \{C_1, .., C_k\}$). Therefore, the fuzzy confidence of

fuzzy associative classification rule can be defined as follows:

FS(<Z:A>)

$$= \frac{\sum_{k=1}^{N} \prod_{<z_{i,C_k}:A_{i,C_k,j}>\in<Z:A>} t_k(z_{i,C_k})}{N} (5)$$

$$FC(<X:A> \rightarrow <Y:C>) = \frac{FS(<X:A> \cup <Y:C>)}{FS(<X:A>)} (6)$$

4. Our Proposed Methodology: CCFAR

In this section, we described our algorithmthat has been named data classification algorithm with compact fuzzy association rules (CCFAR) to obtain the fuzzy associative classification rules with optimum combination of accuracy and interpretability of the model.Our methodology is composed of five steps (Fig 2).

1)Data Screening: The data characteristic of CCFAR must be numeric data because of the transformation of fuzzy set that will convert numeric data only.At current stage, our algorithm assumes that there is no missing value in the data.

2) Data Partitioning: This step is transforming the numeric data to fuzzy intervals or fuzzy sets. In the CCFAR algorithm, we used FCM to makethe transformation because of it easy-to-understand property and high performance. As an example, we used the data from Table 1 and we defined fuzzy intervalsto be3 partitions; the results were represented in table 2.

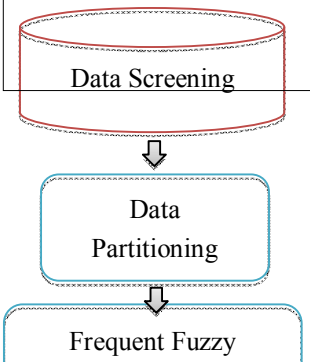


Fig. 2.The development of data classification algorithm with compact fuzzy association rules.

3)Frequent Fuzzy Item-Sets Searching: The fuzzy frequent item-sets are fuzzy item-sets that have fuzzy support more than minimum fuzzy support (γ). At this step, the process are the same asCFARC[7]algorithm (Fig. 3).

4) Fuzzy classification association rule (FCARs)generation: This step is similar to theCFARC[7]algorithm(Fig. 4),but it differed in the computation of score values of all rules. In this research we computed score values of all rules by using our new metric as shown in Eq.9 fuzzy correlation (Eq.7), fuzzy confidence (Eq.6) and firing strength(β) (Eq.8). The intuitive idea is that we need a high correlated, high confidence, as well as high support rules.

FCORR (<X: A> → <Y: C>) =(7)

$$\frac{FS(<X:A> \cup <Y:C>) - FS(<X:A>) \times FS(<Y:C>}{\sqrt{FS(<X:A>) \times (1 - FS(<X:A>)) \times FS(<Y:C>) \times (1 - FS}}$$

$$\beta(<X:A> \rightarrow <Y:C>) = \sum_{k=1}^{N} \prod_{<z_i:A_{i,j}>\in<Z:A>} t_k(z_i) (8)$$

$$SCORE = FCOR * FC * Firing\_strength(\beta) (9)$$

Frequent fuzzy item set searching (an Apriori fuzzy implementation)
Input: DF fuzzy data
Output: the set of frequent fuzzy item set
Method:

1. Determine the supports of the classes by the distribution of classes;
2. Set the minimal fuzzy support (γ) to the half of the minimum frequency of classes;
3. Generate the 1- candidate fuzzy items;
4. Calculate FS values then select the frequent fuzzy items from the 1- candidate which has FS > γ, and n = 2;
5. While there exist some n-1 size frequent item sets: Generate the n–size candidate sets from n-1 size frequents (and 1-size frequents);

Fig. 3.The frequent fuzzy item set searching[7].

Fuzzy classification association rulegeneration
Input: a set of frequent fuzzy item sets
Output: positive correlated FCARs separated by size
Method:

1. Generate association rules with class label consequent from allthe frequent item sets to consider the size of item sets
2. Calculate the Score values of all the rules;
3. Select rules with positive SCORE

Fig. 4.The fuzzy classification association rule generation[7]

5) Fuzzy Classification Association Rule Selection: This step is selecting the FCARs to be used in the data prediction. It's composing of4 parts as follows (Fig.5):

- Frist part:The rules that have the same class label and the same size will be grouped together. After that, this algorithm will select rule with the highest SCORE from each group.

- Secondpart:The ruleswith the highest SCORE in each groupwill be counted the frequency of attributes in antecedent of rule.For example:

From Table 3, the frequency of attribute Age is 5 (No. 1, 2, 3, 4 and 5), attribute Inc is 3 (No. 3, 5 and 6) andattribute Bal is 1 (No. 2). Thus, the highest frequent attributeis the attribute Age. *If frequencies of attributes are equal, attributes will be randomlyselected.

- Thirdpart:A set of FCARs with the highest SCORE and shortest rule from eachfuzzy interval is selected. For examples, the rules from Table 3 that contain attribute Age (the highest frequency) are No.1, 2, 3, 4 and 5. After that, it selects the FCARs from the rules No. 1-5 with the highest SCORE and the shortest rule, which are rule No. 1(Shortest rule of Age = high), 2 (Highest SCORE of Age = high), and 4 (highest SCORE and shortest rule of Age = medium). But this selection is incomplete became the rule that contains Age = low is missing. Therefore, this algorithm will create new rule that contains Age = low with don't cared support and SCORE values, such as:

Rule 1: Age = low → yes, SCORE = -0.12
Rule 2:Age = low → no, SCORE = 0.12

In this algorithm, the Rule 2 will be selected because it has positive SCORE value. Thus, we can conclude the results as follows:

1. Age = high → yes
2. Age = high and Bal = low → no
3. Age = medium → no
4. Age = low → no

- Fourth part: we will remove rules that are superset in antecedent part (left-hand-side) with the same class as other rules. For instance,

Rule 1: Age = low → yes,
Rule 2:Age = low and Bal = high→ yes

From example, we removed the Rule 2

because of it is superset in antecedent part and has same class of Rule 1.

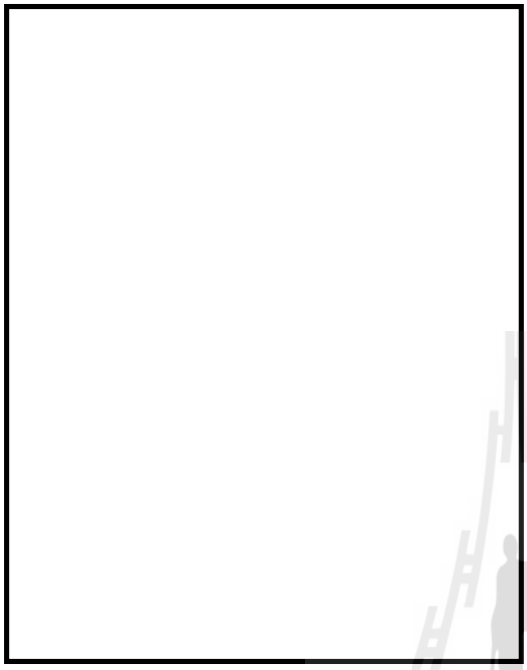


Fig. 5.The fuzzy classification association rule selection.

Table 3.The FCARs with SCORE.

No.	FCARs	SCORE
1	Age = high → yes	1.2725
2	Age = high and Bal = low → no	1.9225
3	Age = high and Inc = me → no	1.1619
4	Age = medium → no	1.2608
5	Age = medium, Inc = high → no	0.8432
6	Inc = high → yes	1.1232

Table 4.Data sets.

Data Set	**Size**	**# Attribute**	**# Class**
Iris	150	4	3
Heart	270	13	2
Pima	768	8	2

5. Experimentation and Results

In this section, we evaluated our CCFAR algorithm using three measurements accuracy: Acc (Eq.10), number of the models (Compact Value: CV) that we normalized it to be the Normalization of Compact Value: NCV (Eq.11), and the combination ofAcc and NCV in which we call SuitableRule: SR (Eq.12).Our algorithm was tested on three different data sets and compared with ninedifferent algorithms.These algorithms are grouped as: data classification, associative classification and fuzzy associative classification. The nine algorithms are namely C4.5, RIPPER, OneR, CBA, GARC, OAC, FURIA, CFAR, and CFARC. The data are taken from the UC Irvine Machine Learning Repository, namelyIris, Heart, and Pima (details shown in Table 4). The classification performances of all algorithms were measured by ten-fold cross validation.

$$\text{Acc}=\frac{\text{TP+TN}}{\text{TP + TN + FP + FN}} \quad (10)$$

Where TP is the number of true positive examples, FP is the number of false positive examples, TN is the number of true negative examples and FN is the number of false negative examples.

$$\text{NCV}_{\text{AL}} = \frac{\text{Avg(CV)}_{\max}-\text{Avg(CV)}_{\text{AL}}}{\text{Avg(CV)}_{\max}-\text{Avg(CV)}_{\min}} \quad (11)$$

$$\text{SR}_{\text{AL}} = \frac{\text{Avg(Acc)}_{\text{AL}}+\text{NCV}_{\text{AL}}}{2} \quad (12)$$

Let Avg(CV)is an average of compact value, AL is an Algorithm.

Table5.The Accuracy (Acc)of classification ofCCFAR algorithm and other algorithms.

Data Set	CLASS			AC			FCAR			
	C4.5	RIP PER	OneR	CBA	GAR C	OAC	FU RIA	CFA R	CFA RC	CCFA R*
Iris	0.933	0.933	0.940	0.929	0.960	0.940	0.947	0.913	0.959	0.960
Heart	0.770	0.822	0.729	0.815	0.880	0.811	0.797	-	0.774	0.781
Pima	0.734	0.747	0.724	0.724	0.762	0.781	0.747	0.651	0.729	0.747
Avg(Acc)	0.812	0.834	0.797	0.822	0.867	0.844	0.830	0.782	0.820	0.828
Rank.	8	3	9	6	1	2	5	10	7	4

Table 6.The number of classification rules (Compact Value) ofCCFAR algorithm and other algorithms.

Data Set	CLASS			AC			FCAR			
	C4.5	RIP PER	One R	CBA	GAR C	OAC	FU RIA	CFA R	CFA RC	CCFA R*
Iris	4.9	3.6	3.0	5.0	7.0	9.0	4.4	9.1	3	3
Heart	17.4	4.1	2.0	52.0	12.0	157.0	8.4	-	2.7	3
Pima	20.3	4.1	7.8	45.0	6.0	112.0	8.5	2.0	2	4
Avg(CV)	14.2	3.9	4.2	34	8.3	92.6	7.1	5.5	2.6	3.3
NCV	0.86	0.983	0.980	0.64	0.93	0	0.94	0.96	1	0.99
Rank.	8	3	4	9	7	10	6	5	1	2

Table 7.The complete rules ofCCFAR algorithm and other algorithms.

Data Set	CLASS			AC			FCAR			
	C4.5	RIP PER	OneR	CBA	GAR C	OAC	FU RIA	CFA R	CFA RC	CCFA R*
SR	0.840	0.908	0.888	0.735	0.900	0.422	0.889	0.873	0.910 0	0.9104
Rank.	8	3	6	9	4	10	5	7	2	1

The results in Tables5-7 were summarized and discussed as follows:

- Table 5 shows the accuracy of classificationof CCFARalgorithm compared to the9 algorithms. The symbol "-" means no available published result. GARC algorithm shows the highest accuracy (0.867) in all dataset and it has been ranked number 1 in the comparison among classifier data from 10 algorithms. Our proposed CCFAR algorithm was ranked number 4 (0.828) in classification accuracy.

- Table 6 represents the number of classification rulesof CCFARalgorithm and the other9 algorithms. The measure NCV is a normalization of the compact values computed as in Eq.8. The CFARC algorithm gives the smallest number of rules (2.6) comparing from 10 algorithms. Our CCFAR algorithm was ranked number 2 with average number of rules equals (3.3).

- From table 7, we showed the combination of accuracy and normalization of compact value

whichis calledthe SRvalues of CCFARalgorithm and other nine algorithms. We can see that the bestSR is our presented method CCFAR (0.9104). It means our algorithm is a good classifier in terms of both accuracy and good

compact of fuzzy associative classification rules considered all together.

6. Conclusions

This paper proposes the development of data classification algorithm with compact fuzzy association rulescalled (CCFAR) to optimize both an accuracy and interpretability of the classification model. To evaluate the performance of the proposed method, our algorithmwas tested on three different data sets and compared the results with the other ninedifferent algorithms. The results showed thatour proposed CCFAR algorithm was ranked number 3based onaccuracy of classification, and it was ranked number 2 based on the smallest number of rules.CCFAR is the best algorithm when we combine accuracy and compact value together.

Acknowledgment

The first author has been funded by scholarship from the SuranareeUniversity of Technology.

References

(1) BellaachiaAbdelghani, and GuvenErhan: "Predicting breast cancer survivability using data mining techniques", In Proceedings of 2nd International Conference on Software Technology and Engineering (ICSTE), Vol. 58, No. 13, pp. 10-110, 2006

(2) Quinlan J. Ross: "C4.5: programs for machine learning", Morgan Kaufmann, Vol. 1, 1992

(3) Holte Robert C: "Very simple classification rules perform well on most commonly used datasets", Machine learning, Vol. 11, No. 1, pp. 63-90, 1993

(4) Bing Liu, Wynne Hsu, and Yiming Ma:"Integrating classification and association rule mining", In Proceedings of the 4th American Association for Artificial Intelligence, 1998

(5) Guoqing Chen, Hongyan Liu, Lan Yu, Qiang Wei, and Xing Zhang: "A new approach to classification based on association rule mining", Decision Support Systems. 42(2), pp. 674-689, 2006

(6) Hu Hong, and Li Jiuyong: "Using association rules to make rule-based classifiers robust", In Proceedings of the 16th Australasian database conference, Vol. 39, pp. 47-54, 2005

(7) FerencPéterPach,Attila Gyenesei b, and Janos Abonyi: "Compact fuzzy association rule-based classifier", Expert systems with applications, Vol. 34, No. 4, pp. 2406-2416, 2008

(8) Chen Zuoliang. and GuoqingChen: "Building an associative classifier based on fuzzy association rules", International Journal of Computational Intelligence Systems, 1(3), pp. 262-273, 2008

(9) CohenW.William: "Fast effective rule induction", In Proceedings of the Twelfth International Conference on Machine Learning, pp. 1995

(10) BayardoJr. Roberto, RakeshAgrawal, and DimitriosGunopulo: "Constraint-based rule mining in large, dense databases", In Proceedings of 15th International Conference on Data Engineering, pp. 188-197, 1999

(11) Li Jiuyong Hong Shen, and Rodney Topor: "Mining the optimal class association rule set", Knowledge-Based Systems. 15(7), pp. 399-405, 2002

(12) HühnJens, and EykeHüllermeier: "FURIA: an algorithm for unordered fuzzy rule induction", Data Mining and Knowledge Discovery, 19(3), pp. 293-319, 2009

(13) LotfaliAskarZadeh: Fuzzy sets. Information and control, Vol. 8, No. 3: pp. 338-353, 1965

(14) James C. Bezdek, Robert Ehrich. and William Full: "FCM: The fuzzy c-means clustering algorithm", Computers and

Geosciences, Vol. 10, No. 2, pp. 191-203, 1984

(15) Miguel Delgado, NicolásMarín, Daniel Sánchez, and María-Amparo Vila: "Fuzzy association rules:General model and applications," IEEE Trans. Fuzzy Syst, Vol. 11, No. 2, pp. 214–225, 2003

(16) JesúsAlcalá-Fdez, Rafael Alcalá, María José Gacto, and Francisco Herrera: "Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms," Fuzzy Sets Syst., Vol. 160, No. 7, pp. 905–921, 2009.

ประวัติผู้เขียน

นายไพชยนต์ คงไชย เกิดเมื่อวันที่ 27 มิถุนายน พ.ศ. 2531 ที่ อำเภอเมือง จังหวัดอำนาจเจริญ เริ่มเข้าศึกษาระดับชั้นอนุบาล 1 ที่โรงเรียนอนุบาลนพเก้า อำเภอเมือง จังหวัดอำนาจเจริญ หลังจากนั้นได้ย้ายไปศึกษาต่อในระดับชั้นอนุบาล 2 ถึงชั้นประถมปีที่ 6 ที่โรงเรียนอนุบาลอำนาจเจริญ อำเภอเมือง จังหวัดอำนาจเจริญและได้เข้าศึกษาต่อในระดับมัธยมศึกษาตอนต้นและตอนปลาย ที่โรงเรียนอำนาจเจริญ อำเภอเมือง จังหวัดอำนาจเจริญ ปีการศึกษา 2553 ได้เข้าศึกษาต่อระดับปริญญาตรีในสาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี และสำเร็จการศึกษาเมื่อปี พ.ศ. 2553 ภายหลังสำเร็จการศึกษาในระดับปริญญาตรี ได้เข้าศึกษาในระดับปริญญาโท สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ในปี 2554ซึ่งสำเร็จการศึกษาใน พ.ศ. 2555และได้เข้าศึกษาในระดับปริญญาเอกต่อในสาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ในปีเดียวกัน ระหว่างการศึกษาได้รับความอนุเคราะห์อย่างดียิ่งจากอาจารย์ประจำวิชา Database System และวิชา Knowledge Discovery and Data Mining ได้รับความไว้วางใจให้เป็นผู้ช่วยสอนปฏิบัติการ ได้รับความอนุเคราะห์จากสำนักวิชาวิศวกรรมศาสตร์ให้เป็นครูสอนพิเศษ (Tutor) ในรายวิชา Computer Programmingและได้รับการตีพิมพ์เผยแพร่บทความวิชาการซึ่งรายละเอียดสามารถดูได้ที่ภาคผนวก ก