กลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง

นายนันทวุฒิ คะอังกุ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2557

A MECHANISM TO DISCOVER AND INTEGRATE ASSOCIATION RULES FROM MULTIPLE SOURCES

Nuntawut Kaoungku

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy in Computer Engineering

Suranaree University of Technology

Academic Year 2014

กลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

_____

(รศ. ดร.กิตติศักดิ์ เกิดประสพ)
ประธานกรรมการ

_____

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

_____

(ผศ. ดร.ศุภกฤษฎิ์ นิวัฒนากูล)
กรรมการ

_____

(ผศ. ดร.สายสุนีย์ จับโจร)
กรรมการ

_____

(ดร.ขุนเสก เสกขุนทด)
กรรมการ

_____ _____

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

นันทวุฒิ คะอังกุ : กลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง (A MECHANISM TO DISCOVER AND INTEGRATE ASSOCIATION RULES FROM MULTIPLE SOURCES) อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร.นิตยา เกิดประสพ, 93 หน้า.

งานวิจัยนี้ได้ศึกษาปัญหาการหากฎความสัมพันธ์แบบกระจาย โดยการหากฎความสัมพันธ์แบบดั้งเดิมจะทำกับข้อมูลที่ถูกรวบรวมไว้ในแหล่งข้อมูลเพียงแหล่งเดียว ซึ่งสามารถหากฎความสัมพันธ์ได้แบบตรงไปตรงมา แต่ถ้าข้อมูลถูกกระจายตัวกันอยู่ตามแหล่งต่าง ๆ และด้วยข้อจำกัดของทรัพยากรทางด้านคอมพิวเตอร์ ไม่สามารถรวบรวมข้อมูลที่กระจายตัวกันอยู่ไว้ในแหล่งข้อมูลเพียงแหล่งเดียวเพื่อนำไปหากฎความสัมพันธ์ได้ ทำให้การหากฎความสัมพันธ์จะเป็นในลักษณะของการหากฎความสัมพันธ์แบบกระจาย แต่การหากฎความสัมพันธ์เพื่อให้ได้แหล่งความรู้เพียงแหล่งเดียวในลักษณะนี้ทำได้ยาก เนื่องจากขั้นตอนการรวมกฎความสัมพันธ์นั้นอาจทำให้ได้กฎความสัมพันธ์ที่ขัดแย้งกันเอง หรือได้จำนวนกฎความสัมพันธ์ที่มากจนเกินไป หรือเกิดการขาดไปของกฎความสัมพันธ์ที่สำคัญ

ดังนั้น งานวิจัยนี้ได้เสนอแนวทางแก้ไขปัญหาการหากฎความสัมพันธ์แบบกระจาย โดยในขั้นตอนการรวมกฎความสัมพันธ์จะนำมาเฉพาะกฎความสัมพันธ์ที่ปรากฏขึ้นบ่อยในทุก ๆ แหล่งความรู้ แล้วนำเฉพาะกฎความสัมพันธ์ที่ได้ไปตรวจสอบความขัดแย้งและในขั้นตอนนี้สามารถสร้างกฎความสัมพันธ์ใหม่จากกฎความสัมพันธ์เดิมที่มีอยู่ด้วยวิธีการอนุมานเชิงตรรกศาสตร์ ซึ่งสามารถเติมเต็มในส่วนของกฎความสัมพันธ์ที่ขาดหายไปได้ สุดท้ายจะได้กฎความสัมพันธ์ที่มีประสิทธิภาพเพียงพอสำหรับการนำไปทำนายผลข้อมูลในอนาคตและไม่เกิดความขัดแย้งกันเอง

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนักศึกษา_____

ปีการศึกษา 2557 ลายมือชื่ออาจารย์ที่ปรึกษา_____

NUNTAWUT KAOUNGKU : A MECHANISM TO DISCOVER AND INTEGRATE ASSOCIATION RULES FROM MULTIPLE SOURCES. THESIS ADVISOR : ASSOC. PROF. NITTAYA KERDPRASOP, Ph.D., 93 PP.

DISTRIBUTED DATA/ASSOCIATION RULE MINING/ LOGICAL INFERENCE /NATURAL LANGUAGE

In this research, we study the problem of distributed association rule mining method. The data in traditional storage is centralized; therefore, it is relatively straightforward to perform association rule mining. But if the data are distributed among various sources and computer memory is limited, it is almost impossible to collect all data from difference sources as a centralized data set for association rule mining. The association rule mining has to be distributed, but it is difficult to combine many knowledge bases at one location because the process of combining association rules may lead to inconsistent rules, too many number of association rules, and missing of significant association rules.

We thus propose in this research the distributed association rule mining method. In the combining process, association rules that appear frequently in all knowledge bases are combined and then checked for inconsistency of rules. This process can generate new association rules from original association rule set with the inference feature of first-order logic. Finally, the efficient and consistent association rules are obtained.

School of Computer Engineering Student's Signature_____

Academic Year 2014 Advisor's Signature_____

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดี ผู้วิจัยขอกราบขอบพระคุณ บุคคล และกลุ่มบุคคลต่างๆที่ได้กรุณาให้คำปรึกษา แนะนำ ช่วยเหลืออย่างดียิ่ง ทั้งในด้านวิชาการ และด้านการดำเนินงานวิจัย ดังต่อไปนี้

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

ผู้ช่วยศาสตราจารย์ ดร.พิชโยทัย มหัทธนาภิวัฒน์ ผู้ช่วยศาสตราจารย์ ดร.คะชา ชาญศิลป์ ผู้ช่วยศาสตราจารย์ สมพันธ์ ชาญศิลป์ และผู้ช่วยศาสตราจารย์ ดร.ปรเมศวร์ ห่อแก้ว อาจารย์ประจำสาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

คุณกัลญา พับโพธิ์ เลขานุการสาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ให้ความช่วยเหลือในการประสานงานด้านเอกสารระหว่างศึกษา

คุณภาสพิชญ์ ชูใจ คุณไพชยนต์ คงไชย คุณทิพยา ถินสูงเนิน คุณกิตติพงศ์ ชมบุญและนักศึกษาบัณฑิตสาขาวิชาวิศวกรรมคอมพิวเตอร์ ทุกท่านที่ให้คำปรึกษาและช่วยเหลือด้วยดีมาโดยตลอด

นอกจากนี้ขอขอบคุณครู อาจารย์ทั้งในอดีตและปัจจุบันที่ให้ความรู้แก่ผู้วิจัยจนประสบความสำเร็จในชีวิต

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

นันทวุฒิ คะอังกุ

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ **หน้า**

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

สารบัญรูป

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหาการวิจัย

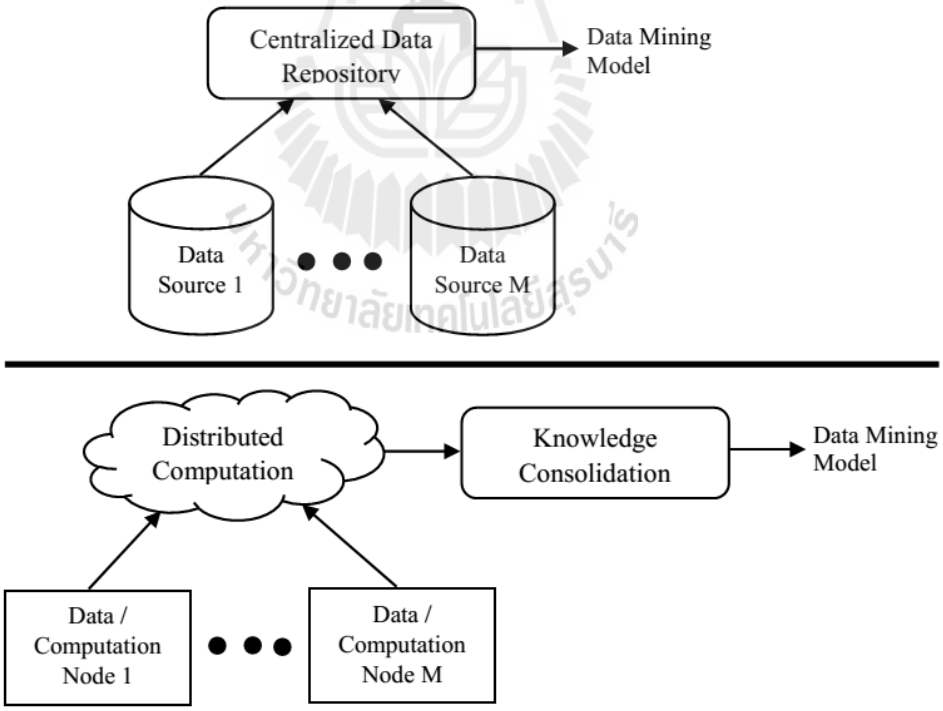
ปัจจุบันการจัดเก็บข้อมูลมีความสำคัญสำหรับแต่ละองค์กรหรือหน่วยงานต่าง ๆ เนื่องจากสามารถที่จะสกัดความรู้จากข้อมูลเหล่านั้นเพื่อนำไปพัฒนาองค์กรหรือหน่วยงานนั้น ๆ ได้ ทำให้การพัฒนาเทคโนโลยีในการจัดเก็บและการวิเคราะห์ข้อมูลได้มีการพัฒนาอย่างต่อเนื่องมาเรื่อย ๆ แต่ด้วยเทคโนโลยีเหล่านี้ทำให้การจัดเก็บข้อมูลสามารถทำได้ง่ายและรวดเร็ว ทำให้ข้อมูลที่ถูกจัดเก็บนั้นมีขนาดที่ใหญ่ขึ้นหรือข้อมูลไม่ได้ถูกจัดเก็บไว้ในแหล่งข้อมูลเพียงแหล่งเดียว ซึ่งการสกัดความรู้จากข้อมูลในลักษณะนี้นั้นทำได้ยาก

การทำเหมืองข้อมูล (Data Mining) เป็นเทคนิคการวิเคราะห์ข้อมูลที่ถูกนำมาใช้อย่างแพร่หลายในปัจจุบัน ซึ่งเป็นการสกัดความรู้จากฐานข้อมูลที่มีขนาดใหญ่เพียงแหล่งเดียวโดยอ้างอิงพื้นฐานทางด้านสถิติ เช่น รูปแบบของข้อมูล แนวโน้มของข้อมูล หรือความสัมพันธ์ภายในกลุ่มข้อมูล เป็นต้น โดยได้มีการนำไปประยุกต์ใช้เพื่อวิเคราะห์ข้อมูลหลากหลายด้าน เช่น ข้อมูลทางการแพทย์เพื่อนำไปวิเคราะห์โรคต่าง ๆ ข้อมูลทางด้านการตลาดเพื่อวิเคราะห์แนวโน้มของการตลาด เป็นต้น แต่ด้วยลักษณะของข้อมูลในปัจจุบันที่มีขนาดใหญ่หรือข้อมูลมีการกระจายตัวกันอยู่ ทำให้การทำเหมืองข้อมูลต้องมีการปรับเปลี่ยนรูปแบบการวิเคราะห์ข้อมูลจากเดิมเพื่อให้เหมาะกับลักษณะของข้อมูลในยุคปัจจุบันมากยิ่งขึ้น

เพื่อให้การทำเหมืองข้อมูลเหมาะสำหรับลักษณะของข้อมูลในปัจจุบันนั้น ได้มีการเสนอแนวคิดในการทำเหมืองข้อมูลแบบกระจาย (Distributed Data Mining) คือ การสกัดความรู้จากข้อมูลที่กระจายกันอยู่ตามแหล่งข้อมูลต่าง ๆ เพื่อให้ได้ฐานความรู้เพียงหนึ่งเดียว จากรูปที่ 1.1 แสดงตัวอย่างการทำเหมืองข้อมูลแบบเดิมและการทำเหมืองข้อมูลที่กระจาย ซึ่งจากแนวคิดนี้สามารถนำไปประยุกต์ใช้เพื่อแก้ไขปัญหาของการไม่สามารถสกัดความรู้จากข้อมูลที่มีขนาดใหญ่ได้ ด้วยการแบ่งข้อมูลออกเป็นชุด ๆ เพื่อเลียนแบบการทำเหมืองข้อมูลแบบกระจาย ทำให้มีหลากหลายงานวิจัยที่นำแนวคิดนี้ไปประยุกต์ใช้กับอัลกอริทึมต่าง ๆ ทางด้านการทำเหมืองข้อมูล เช่น การจำแนกข้อมูล (Classification) การจัดกลุ่มข้อมูล (Clustering) เป็นต้น

การหากฎความสัมพันธ์ (Association Rule Mining) เป็นอัลกอริทึมหนึ่งของการทำเหมืองข้อมูล ซึ่งใช้สำหรับหาความสัมพันธ์ของเหตุการณ์หรือวัตถุที่เกิดขึ้นร่วมกันหรือพร้อมกัน โดยยังปรากฏงานวิจัยอยู่น้อยมากสำหรับการทำเหมืองข้อมูลแบบกระจาย ซึ่งงานวิจัยที่ปรากฏอยู่นั้นจะเป็นในลักษณะของการหากฎความสัมพันธ์แบบขนาน หรือการหากฎความสัมพันธ์จากความคล้ายคลึงของข้อมูล แต่ยังไม่มีงานวิจัยที่จะสามารถตอบโจทย์ของการทำเหมืองข้อมูลแบบกระจายได้อย่างสมบูรณ์แบบ เนื่องจากในขั้นตอนของการรวมความรู้ที่กระจายกันอยู่เพื่อให้มีประสิทธิภาพสำหรับการนำไปทำนายผลในอนาคตนั้นทำได้ยาก

จากที่กล่าวข้างต้น เพื่อให้การทำเหมืองข้อมูลสามารถสกัดความรู้ที่ได้จากข้อมูลที่มีขนาดใหญ่หรือข้อมูลที่กระจายตัวอยู่นั้นจำเป็นต้องนำเทคนิคการทำเหมืองข้อมูลแบบกระจายมาปรับใช้และอัลกอริทึมทางด้านการทำเหมืองข้อมูลเพื่อหากฎความสัมพันธ์ที่ทำงานกับข้อมูลในหลายแหล่งยังปรากฏงานวิจัยอยู่น้อย ดังนั้นผู้วิจัยจึงได้เสนอวิธีการหากฎความสัมพันธ์แบบกระจาย (Distributed Association Rule Mining) ด้วยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง



รูปที่ 1.1 ตัวอย่างการทำเหมืองข้อมูลแบบเดิมและการทำเหมืองข้อมูลที่กระจายกันอยู่

1.2 วัตถุประสงค์ของการวิจัย

1) เพื่อศึกษาและพัฒนากลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง

2) เพื่อรวมกฎความสัมพันธ์ที่ได้จากการหากฎความสัมพันธ์ด้วยข้อมูลที่มีขนาดใหญ่และเป็นข้อมูลที่กระจายกันได้

3) เพื่อศึกษาและพัฒนาการทดสอบประสิทธิภาพของการรวมกฎความสัมพันธ์ ให้สามารถนำไปทำนายผลในอนาคตได้

1.3 ข้อตกลงเบื้องต้น

1) ข้อมูลที่ใช้ทดสอบประสิทธิภาพของการหากฎความสัมพันธ์แบบกระจายเป็นข้อมูลสังเคราะห์ที่สร้างจากโปรแกรมคอมพิวเตอร์และข้อมูลจริงจากแหล่งข้อมูล UCI Machine Learning Repository (http://archive.ics.uci.edu/ml)

2) ข้อมูลที่กระจายกันอยู่ และใช้ทดสอบประสิทธิภาพของการรวมกฎความสัมพันธ์แบบกระจายจะต้องมีโครงสร้างของข้อมูลที่ไม่แตกต่างกันมาก

3) งานวิจัยนี้เลือกใช้ภาษาหรือเครื่องมือในการพัฒนาโปรแกรมและทดสอบอัลกอริทึม ได้แก่ Python, Attempto Controlled English, Protégé Editor และ FaCT++ Reasoner

1.4 ขอบเขตของการวิจัย

1) การหากฎความสัมพันธ์จะเลือกใช้อัลกอริทึมเอไพรออริ (Apriori)

2) งานวิจัยนี้ใช้ภาษาไพทอน (Python language) ในส่วนของการรวมกฎความสัมพันธ์ และใช้ Attempto Controlled English, Protégé Editor และ FaCT++ Reasoner ในส่วนของการตรวจสอบความขัดแย้งและสร้างความรู้ใหม่จากความรู้เดิม

1.5 ประโยชน์ที่จะได้รับ

1) สามารถรวมกฎความสัมพันธ์จากการหากฎความสัมพันธ์ด้วยข้อมูลที่มีขนาดใหญ่หรือข้อมูลที่กระจายกันได้

2) สามารถนำอัลกอริทึมสำหรับการรวมกฎความสัมพันธ์มาประยุกต์ใช้ได้ง่าย

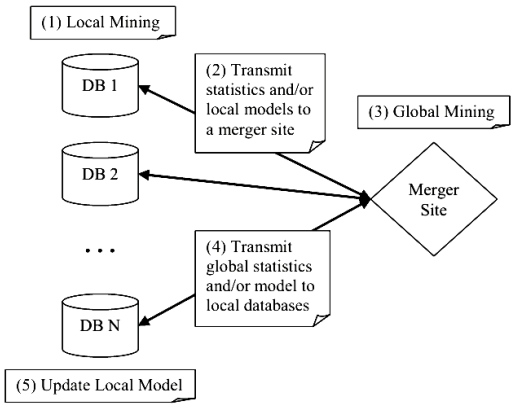
3) สามารถใช้คอมพิวเตอร์หลาย ๆ เครื่อง ในการหากฎความสัมพันธ์โดยที่กระบวนการไม่ขึ้นต่อกันเพื่อให้ได้ฐานความรู้เพียงหนึ่งเดียว

บทที่ 2
ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง โดยมีรายละเอียดของการทำเหมืองข้อมูลแบบกระจาย (Distributed Data Mining) การหากฎความสัมพันธ์ (Association Rule Mining) การหากฎความสัมพันธ์แบบกระจาย (Distributed Association Rule Mining) ภาษาควบคุม (Controlled Natural Language) ฐานความรู้ออนโทโลยี (Ontology Knowledge Base) การอนุมานเชิงตรรกะ (Logical Inference) และงานวิจัยที่เกี่ยวข้อง

2.1 การทำเหมืองข้อมูลแบบกระจาย

ปัจจุบันข้อมูลไม่ได้ถูกจัดเก็บไว้อยู่ในแหล่งข้อมูลเพียงแหล่งเดียวเนื่องจากจำเป็นต้องใช้พื้นที่สำหรับการจัดเก็บเป็นจำนวนมาก ซึ่งทำให้ต้องกระจายข้อมูลออกไปอยู่ตามแหล่งข้อมูลต่าง ๆ หรือข้อมูลบางชนิดมีการกระจายของข้อมูลอยู่แล้ว ซึ่งลักษณะข้อมูลแบบนี้ทำให้การทำเหมืองข้อมูลนั้นต้องถูกปรับเปลี่ยนไปจากเดิมที่ต้องใช้ข้อมูลจากแหล่งข้อมูลเพียงชุดเดียวเท่านั้น จากรูปที่ 2.1 แสดงสถาปัตยกรรมการทำเหมืองข้อมูลแบบกระจาย โดยขั้นตอนแรกจะหาความรู้ในแต่ละฐานข้อมูล และขั้นตอนที่ 2 ทำการส่งความรู้ที่ได้ในแต่ละฐานข้อมูลไปยังขั้นตอนที่ 3 เพื่อทำการรวมความรู้ที่ส่งมาจากแต่ละฐานข้อมูล ขั้นตอนที่ 4 ส่งความรู้ใหม่ที่ได้จากขั้นตอนที่ 3 ไปแต่ละฐานข้อมูลเพื่อนำไปใช้ต่อไป และขั้นตอนสุดท้ายในแต่ละฐานข้อมูลทำการปรับปรุงโมเดลที่ได้จากขั้นตอนที่ 4 (Tsoumakas et al., 2009)



รูปที่ 2.1 สถาปัตยกรรมการทำเหมืองข้อมูลแบบกระจาย (Tsoumakas et al., 2009)

2.2 การหากฎความสัมพันธ์

การหากฎความสัมพันธ์ คือ การค้นหาและเรียนรู้ความสัมพันธ์ของเหตุการณ์หรือวัตถุที่เกิดขึ้นร่วมกันหรือพร้อมกัน หรือเพื่อหารูปแบบที่เกิดขึ้นบ่อย (Frequent Pattern) เพื่อนำไปใช้ในการวิเคราะห์ หรือทำนายปรากฏการณ์ต่าง ๆ ซึ่งการหากฎความสัมพันธ์นั้นสามารถนำไปใช้งานได้หลายรูปแบบ เช่น การวิเคราะห์พฤติกรรมซื้อสิ้นค้า เช่น ถ้าลูกค้าซื้อสินค้า A แล้วมักจะซื้อ B ตามไปด้วย เป็นต้น โดยจะอยู่ในรูปแบบ **IF** condition **Then** result (Agrawal et al., 1993) เกณฑ์ที่ใช้คัดเลือกข้อมูลในขั้นตอนการหากฎความสัมพันธ์ประกอบด้วย

- ค่าสนับสนุน (Support) คือ ค่าที่บอกว่าความถี่ที่ข้อมูลในกฎนั้น ๆ เกิดขึ้นบ่อยมากน้อยแค่ไหน สามารถหาได้จากสมการที่ 2.1

$$Support(A \rightarrow B) = P(A \wedge B) \tag{2.1}$$

- ค่าความเชื่อมั่น (Confidence) คือ ค่าที่บอกว่าโอกาสที่จะเกิดกฎความสัมพันธ์นั้นขึ้นมีมากน้อยเพียงใด เช่น ถ้ามี condition เกิดขึ้น โอกาสที่จะเกิด result มีมากน้อยแค่ไหน สามารถหาได้จากสมการที่ 2.2

$$Confidence(A \rightarrow B) = \frac{Supp(A \rightarrow B)}{Supp(A)} \tag{2.2}$$

ตัวอย่างการหากฎความสัมพันธ์ สมมติให้มีข้อมูลรายการซื้อสินค้าของลูกค้า แล้วนำไปผ่านการหาความถี่ของการซื้อสินค้าของลูกค้าในแต่ละชิ้นของสินค้าดังตารางที่ 2.1 เพื่อหาความสัมพันธ์ของสินค้าแต่ละอย่างซึ่งจะแสดงได้ดังตารางที่ 2.2 หลังจากนั้นก็จะนำสินค้าที่มีความถี่สูงไปสร้างกฎความสัมพันธ์ดังตารางที่ 2.3

ตารางที่ 2.1 รายการซื้อสินค้าของลูกค้าทั้งหมด

รายการสินค้า	นม	น้ำ	ขนม	ไส้กรอก
1	1	1	0	0
2	0	1	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0

ตารางที่ 2.2 ความถี่ของการซื้อสินค้าของลูกค้า เพื่อหาความสัมพันธ์ของสินค้าแต่ละอย่าง

	นม	**น้ำ**	**ขนม**	**ไส้กรอก**
นม	2*	2	1	0
น้ำ	2	4*	2	0
ขนม	1	1	2*	0
ไส้กรอก	0	0	0	1*

ตารางที่ 2.3 ตัวอย่างกฎความสัมพันธ์

กฎความสัมพันธ์	**Support**	**Confidence**
ขนม => นม	0.2	0.5
นม => ขนม	0.2	0.5
ขนม => น้ำ	0.4	1.0
น้ำ => ขนม	0.4	0.5
นม => น้ำ	0.4	1.0
น้ำ => นม	0.4	0.5
นม,ขนม => น้ำ	0.2	1.0
น้ำ,ขนม => นม	0.2	0.5
นม,น้ำ => ขนม	0.2	0.5

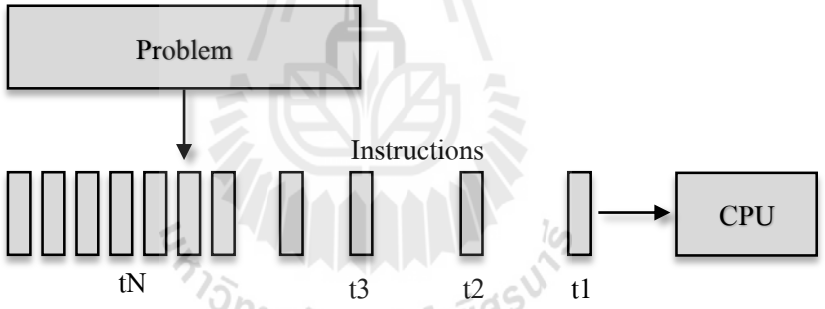
2.3 การหากฎความสัมพันธ์แบบกระจาย

การหาความสัมพันธ์ของเหตุการณ์หรือวัตถุที่เกิดขึ้นร่วมกันหรือพร้อมกันจากข้อมูลที่มีขนาดใหญ่หรือข้อมูลที่กระจายกันอยู่นั้นเป็นไปได้ค่อนข้างยาก เนื่องจากการหาความสัมพันธ์แบบเดิมนั้นถูกพัฒนามาไว้สำหรับการหาความสัมพันธ์จากข้อมูลเพียงชุดเดียว ทำให้จำเป็นต้องรวมข้อมูลจากหลาย ๆ แหล่งมารวมเป็นข้อมูลเพียงชุดเดียว แต่ข้อมูลที่ได้นั้นอาจมีขนาดใหญ่เกินไปสำหรับการนำไปหากฎความสัมพันธ์ ดังนั้นจึงทำให้มีเทคนิคที่จะมาช่วยแก้ไขปัญหาในการหาความสัมพันธ์แบบกระจาย

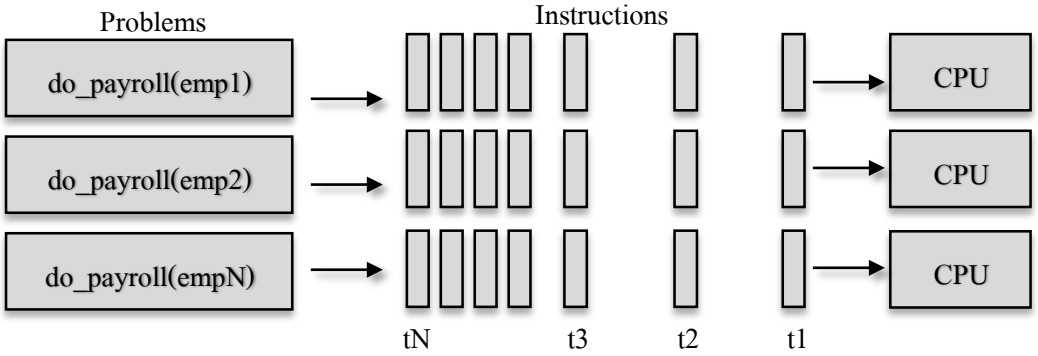
2.3.1 การหากฎความสัมพันธ์แบบขนาน (Parallel Association Rule Mining)

การประมวลผลทางด้านคอมพิวเตอร์ในสมัยก่อนจะเป็นในรูปแบบการทำงานแบบอนุกรม (Serial Processing) ที่มีเพียงเหตุการณ์เดียวเกิดขึ้นในเวลาเดียว ดังรูปที่ 2.2 จะเห็นได้ว่าในการประมวลผลสามารถทำได้ครั้งละ Problem เท่านั้น ซึ่งทำให้ไม่เหมาะสำหรับการนำไปประมวลผลข้อมูลที่มีขนาดใหญ่ได้ แต่ด้วยปัจจุบันเทคโนโลยีทางด้านคอมพิวเตอร์ได้ถูกพัฒนาขีดความสามารถขึ้นมาเพียงพอที่จะสามารถแก้ไขปัญหาในจุดนี้ได้ และระบบคอมพิวเตอร์ดังกล่าวก็มีความสามารถในการประมวลผลในลักษณะที่เราเรียกว่า การประมวลผลแบบขนาน (Parallel Processing) คือเหตุการณ์มากกว่า 1 เหตุการณ์ เกิดขึ้นพร้อมกันในเวลาเดียวกัน (Leighton, 1992) ดังรูปที่ 2.3

ตัวอย่างเปรียบการทำงานของการทำงานแบบอนุกรมและแบบขนาน *"ถ้าต้องการสร้างบ้าน 1 หลังต้องใช้เวลา 1 เดือนและคนงานทั้งหมด 10 คนในการสร้าง แต่ถ้าต้องการสร้างบ้าน 10 หลังให้เสร็จภายใน 1 เดือนจะทำอย่างไร? การทำงานแบบขนานจะต้องจ้างคนงานเพิ่มมาเป็น 100 คน โดยให้คนงาน 10 คน สร้างบ้าน 1 หลัง"*



รูปที่ 2.2 ตัวอย่างการประมวลผลแบบอนุกรม (Leighton, 1992)



รูปที่ 2.3 ตัวอย่างการประมวลผลแบบขนาน (Leighton, 1992)

การนำเทคนิคการทำงานแบบขนานมาประยุกต์ใช้ในการหาความสัมพันธ์ของข้อมูลนั้นเป็นทางเลือกที่ดี เนื่องจากสามารถแบ่งข้อมูลออกเป็นส่วน ๆ แล้วนำไปหาความสัมพันธ์แบบขนานได้ตามโครงสร้างของการทำงานแบบขนาน โดยวิธีการหากฎความสัมพันธ์ด้วยเทคนิคแบบขนานส่วนมากจะเป็นการหากฎความสัมพันธ์ด้วยอัลกอริทึมเอไพรออริซึ่งได้รับความนิยมกันอย่างแพร่หลายเนื่องจากเป็นอัลกอริทึมที่สามารถเข้าใจได้ง่าย

การทำงานแบบขนานนั้นยังมีข้อจำกัดตรงที่กระบวนการหรือขั้นตอนที่จะสามารถนำมาใช้กับการทำงานแบบขนานได้นั้น จำเป็นต้องเป็นขั้นตอนที่ไม่มีความเกี่ยวเนื่องกัน ดังนั้นในการหากฎความสัมพันธ์ด้วยอัลกอริทึมเอไพรออรินั้นจะไม่ใช่การทำงานแบบขนานของกระบวนการทั้งหมด แต่จะเป็นขั้นตอนการทำงานข้างในอัลกอริทึมเอไพรออริ ซึ่งมีเพียงบางกระบวนการเท่านั้นที่สามารถประยุกต์ใช้เพื่อการทำงานแบบขนานได้ จากรูปที่ 2.4 แสดงตัวอย่างการหากฎความสัมพันธ์ด้วยอัลกอริทึมเอไพรออริแบบอนุกรม จะเห็นได้ว่าในขั้นตอนของการนับเพื่อหาความถี่ของแต่ละไอเท็มเซต (Item Set) นั้นไม่มีความเกี่ยวเนื่องกัน ดังนั้นสามารถประยุกต์ในส่วนของตรงนี้ให้เป็นการทำงานแบบขนานได้ดังตัวอย่างในรูปที่ 2.5 จากตาราง Transaction จะแบ่งข้อมูลออกเป็น P_0 P_1 และ P_2 แล้วนำไปหาความถี่ในแต่ละไอเท็มเซตตาม Step 1 ซึ่งใน Step 1 นี้ก็คือการทำงานแบบขนาน หลังจากนั้นจะทำการรวมความถี่ในแต่ละไอเท็มที่ขั้นตอน P_0 P_1 และ P_2 หามาได้ตาม Step 2 และ 3

D

Transaction
ACD
BCE
ABCE
BE

1-pass →

C_1

1- itemset	Count
A	2
B	3
C	3
D	1
E	3

→

L_1

Large 1-itemset	Count
A	2
B	3
C	3
E	3

C_2

2- itemset
AB
AC
AE
BC
BE
CE

2-pass →

C_2

2- itemset	Count
AB	1
AC	2
AE	1
BC	2
BE	3
CE	2

→

L_2

Large 2-itemset	Count
AC	2
BC	2
BE	3
CE	2

C_3

3- itemset
BCE

3-pass →

C_3

3- itemset	Count
BCE	2

→

L_3

Large 3-itemset	Count
BCE	2

Result →

Large itemsets	Count
BCE	2
AC	2

รูปที่ 2.4 ตัวอย่างการหากฎความสัมพันธ์ด้วยอัลกอริทึมเอไพรออริแบบอนุกรม (Agrawal et al., 1993)

TID	Items Bought	Processor Number
1	A, B, C, D, E	
2	F, B, D, E, G	-> P_0
3	B, D, A, E, G	
4	A, B, F, G, D	
5	B, F, D, G, K	-> P_1
6	A, B, F, G, D	
7	A, R, M, K, O	
8	B, F, G, A, D	-> P_2
9	A, B, F, M, O	

Transactional database example

Item	Counters		
	P_0	P_1	P_2
A	2	2	3
B	3	3	2
C	1	0	0
D	3	3	1
E	3	0	0
F	1	3	2
G	2	3	1
K	0	1	1
R	0	0	1
M	0	0	2
O	0	0	2

Step 1

Proc. #	Item	Global Counter
P_0	A	7
	B	8
	C	1
	D	7
P_1	E	3
	F	6
	G	6
	K	2
P_2	R	1
	M	2
	O	2

Step 2

Item	Global Counter
A	7
B	8
D	7
F	6
G	6

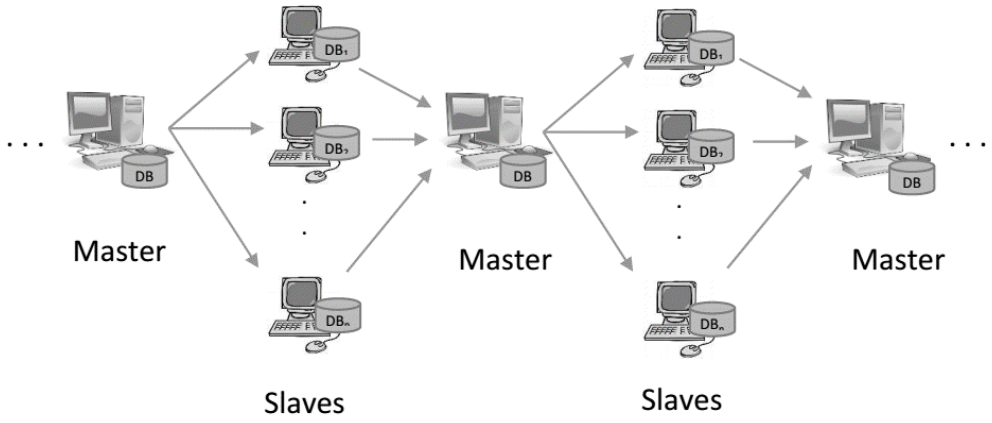
Step 3

Item	Global Counter
B	8
A	7
D	7
F	6
G	6

Step 4

รูปที่ 2.5 ตัวอย่างการหากฎความสัมพันธ์ด้วยอัลกอริทึมเอไพรออริแบบขนาน

จากรูปที่ 2.6 เป็นตัวอย่างการการหากฎความสัมพันธ์ด้วยอัลกอริทึมเอไพรออริแบบขนาน จะเห็นได้ว่าเริ่มต้นมาสเตอร์ (Master) จะแบ่งข้อมูลออกเป็นชุด ๆ คือ DB_1, DB_2, … , DB_n ส่งไปให้สลาฟ (Slaves) แต่ละตัวสำหรับการนำไปหาความถี่ หลังจากนั้นจะรวมความถี่ที่ได้จากแต่ละชุดสลาฟมาไว้ที่มาสเตอร์ แล้วนำไปตัดไอเท็มเซตที่มีค่าน้อยกว่าค่าสนับสนุนขั้นต่ำ (Minimum Support) ที่กำหนดไว้ แล้วทำการแบ่งข้อมูลออกเป็นชุด ๆ คือ DB_1, DB_2, … , DB_n ส่งไปให้สลาฟสำหรับการจับคู่ของแต่ละไอเท็มเซตหลังจากนั้นจะรวมการจับคู่ของแต่ละไอเท็มเซตที่ได้จากแต่ละชุดสลาฟมาไว้ที่มาสเตอร์และจะกลับไปที่ขั้นตอนที่ 1 ทำแบบนี้ไปเรื่อย ๆ จนกว่าไอเท็มเซตไม่เปลี่ยนแปลง (Agrawal and Srikant 1994; Agrawal and Shafer, 1996; Cheung et al., 1996; Cheung et al., 1996; Cheung et al., 2002; Yu et al., 2010)



รูปที่ 2.6 ตัวอย่างการหากฎความสัมพันธ์ด้วยอัลกอริทึมเอไพรออริแบบขนาน

2.3.2 การหากฎความสัมพันธ์แบบกระจายจากความคล้ายคลึง (Similarity Based Distributed Association Rule Mining)

การหากฎความสัมพันธ์โดยทั่วไปสามารถหาได้จากข้อมูลเพียงชุดเดียว แต่ด้วยข้อมูลบางประเภทที่ไม่ได้มีการจัดเก็บข้อมูลไว้ในแหล่งข้อมูลเพียงแหล่งเดียว ทำให้การนำข้อมูลเหล่านี้มาหากฎความสัมพันธ์นั้นจำเป็นต้องรวมข้อมูลที่ถูกกระจายกันอยู่ให้เป็นข้อมูลเพียงชุดเดียวเพื่อสามารถนำไปหากฎความสัมพันธ์ได้ แต่เนื่องจากข้อมูลที่กระจายกันอยู่นั้นมีการจัดเก็บข้อมูลที่มีลักษณะแตกต่างกันออกไป ซึ่งอาจทำให้การนำข้อมูลเหล่านี้มารวมกันแล้วนำไปหากฎความสัมพันธ์นั้นได้ประสิทธิภาพที่ไม่ดีพอสำหรับการนำไปทำนายผลในอนาคต ดังนั้นข้อมูลที่กระจายกันอยู่ตามแหล่งต่าง ๆ นั้นจะต้องมีลักษณะข้อมูลที่มีความคล้ายคลึงกัน โดยสามารถวัดความคล้ายคลึงกันของข้อมูลได้จากมาตรวัดความคล้ายคลึง (Similarity Measure) (Li et al., 2003) ซึ่งสามารถหาได้จากสมการที่ 2-3

$$Sim(A,B) = \frac{2I_3}{I_1 + I_2} \tag{2-3}$$

โดยกำหนดให้

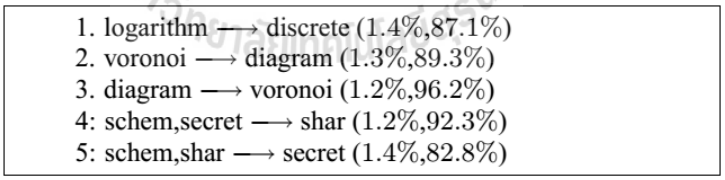
$$I_1 = \sum_{i,j} \frac{\left|A_i \cap A_j\right|}{\left|A_i \cup A_j\right|} \log\left(1 + \frac{\left|A_i \cap A_j\right|}{\left|A_i \cup A_j\right|}\right) \min\left\{C_{Ai}, C_{Aj}\right\},$$

$$I_2 = \sum_{i,j} \frac{\left|B_i \cap B_j\right|}{\left|B_i \cup B_j\right|} \log\left(1 + \frac{\left|B_i \cap B_j\right|}{\left|B_i \cup B_j\right|}\right) \min\left\{C_{Bi}, C_{Bj}\right\},$$

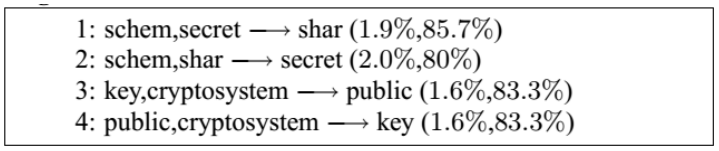
$$I_3 = \sum_{i,j} \frac{\left|A_i \cap B_j\right|}{\left|A_i \cup B_j\right|} \log\left(1 + \frac{\left|A_i \cap B_j\right|}{\left|A_i \cup B_j\right|}\right) \min\left\{C_{Ai}, C_{Bj}\right\},$$

จากสมการที่ 2-3 จะเป็นการวัดความคล้ายคลึงของข้อมูล 2 ข้อมูล คือ ข้อมูล A และ B ซึ่งค่าที่ได้ออกมาจะอยู่ระหว่าง 0 ถึง 1 โดยถ้ามีค่ามากแสดงว่าข้อมูล A และ B มีความคล้ายคลึงกันมาก แต่ถ้าค่าที่ได้ออกมามีค่าน้อยแสดงว่าข้อมูล A และ B มีความคล้ายคลึงกันน้อย ตัวอย่างการนำมาตรวัดความคล้ายคลึงกันไปใช้สำหรับการกฎความสัมพันธ์โดยข้อมูลที่นำมาใช้ได้แก่ Computational Geometry (COMPGEOM), Conference on Cryptography (CRYPT) และ European Conference on Cryptography (EUROCRYPT) ซึ่งจะหาค่าความคล้ายคลึงของข้อมูลกับข้อมูลชุดอื่น ๆ หลังจากนั้นจะนำข้อมูลที่มีความคล้ายคลึงกันมากที่สุดมารวมกันแล้วนำไปหากฎความสัมพันธ์ โดยใช้ Minimum support = 1% และ Minimum confidence = 80%

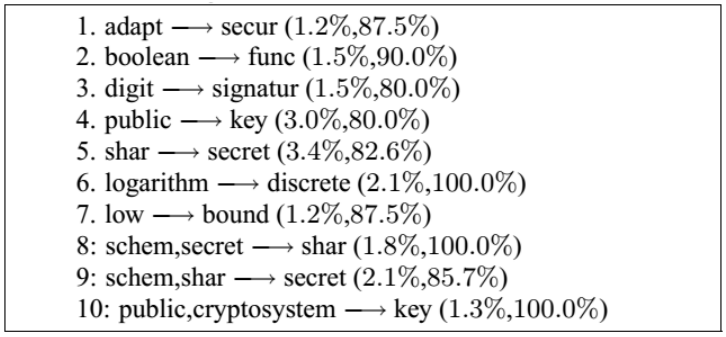
จากรูปที่ 2.8 , 2.9 และ 2.10 แสดงกฎความสัมพันธ์ที่ได้จากข้อมูล CRYPT, EUROCRYPT และ COMPGEOM ตามลำดับ ซึ่งจะสังเกตเห็นได้ว่าบางกฎความสัมพันธ์ระหว่างข้อมูล CRYPT และ EUROCRYPT เหมือนกัน และเมื่อมาหากฎความสัมพันธ์จากค่าความคล้ายคลึงผลลัพธ์ที่ได้คือข้อมูล CRYPT และ EUROCRYPT ให้ค่ามากที่สุด ดังนั้นการหากฎความสัมพันธ์สามารถแบ่งออกได้เป็น 2 กลุ่ม คือ การหากฎความสัมพันธ์ระหว่างข้อมูล CRYPT และ EUROCRYPT กับ การหากฎความสัมพันธ์จากข้อมูล COMPGEOM จากรูปที่ 2.7 แสดงกฎความสัมพันธ์ที่ได้จากข้อมูล CRYPT, EUROCRYPT และ COMPGEOM ซึ่งจะเห็นได้ว่าจำนวนกฎความสัมพันธ์ที่ได้ออกมานั้นน้อยกว่าเมื่อเปรียบเทียบจากรูปที่ 2.10 และ 2.11 ที่เป็นกฎความสัมพันธ์ที่ได้จากการวัดค่าความคล้ายคลึง ซึ่งบางกฎความสัมพันธ์ที่เพิ่มขึ้นมานั้นอาจเป็นกฎความสัมพันธ์ที่สำคัญก็เป็นไปได้



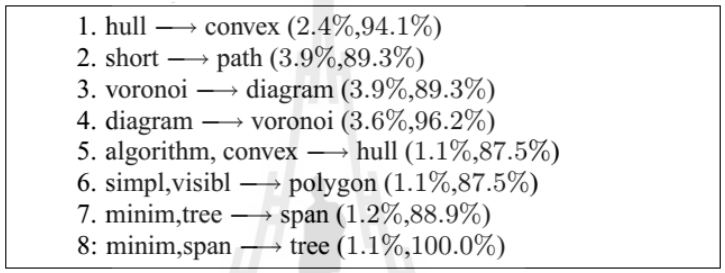
รูปที่ 2.7 กฎความสัมพันธ์ที่ได้จากข้อมูล COMPGEOM, CRYPT และ EUROCRYPT (Li et al., 2003)



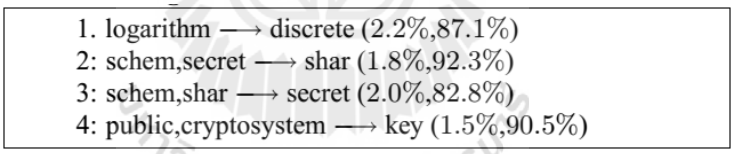
รูปที่ 2.8 กฎความสัมพันธ์ที่ได้จากข้อมูล CRYPT (Li et al., 2003)



รูปที่ 2.9 กฎความสัมพันธ์ที่ได้จากข้อมูล EUROCRYPT (Li et al., 2003)



รูปที่ 2.10 กฎความสัมพันธ์ที่ได้จากข้อมูล COMPGEOM (Li et al., 2003)



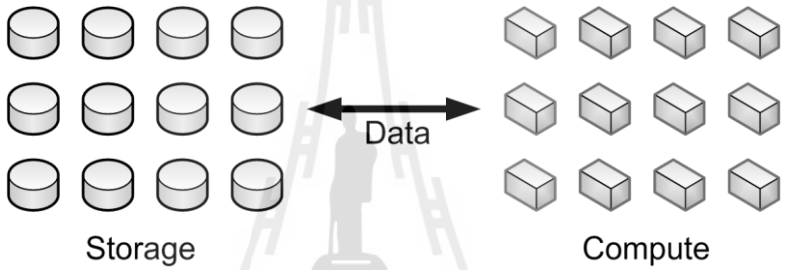
รูปที่ 2.11 กฎความสัมพันธ์ที่ได้จากข้อมูล CRYPT และ EUROCRYPT (Li et al., 2003)

2.3.3 การทำเหมืองข้อมูลจากข้อมูลแบบกลุ่มเมฆ (Data Mining from Data Clouds)

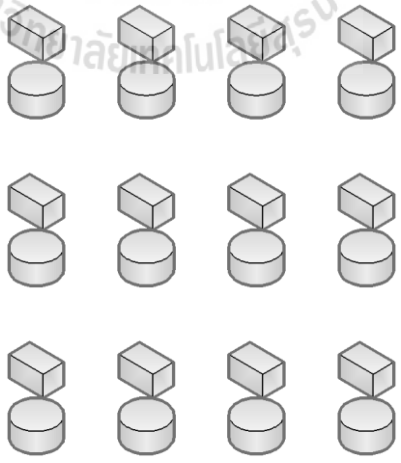
ลักษณะข้อมูลในปัจจุบันมีขนาดที่ใหญ่ขึ้นตามเทคโนโลยีที่ถูกพัฒนามาเพื่อสำหรับการจัดเก็บข้อมูล แต่สิ่งที่ตามมาคือการสกัดความรู้จากข้อมูลในลักษณะนี้ทำได้ยาก เนื่องจากต้องใช้คอมพิวเตอร์ที่มีประสิทธิสูง เช่น ซุปเปอร์คอมพิวเตอร์ (Super Computer) (Chattratichat et al., 1999) ดังรูปที่ 2.12 แสดงการใช้ซุปเปอร์คอมพิวเตอร์ในการสกัดความรู้จากข้อมูลที่มีขนาดใหญ่ เป็นต้น แต่ด้วยราคาของซุปเปอร์คอมพิวเตอร์มีราคาสูงมากทำให้มีเฉพาะหน่วยงานหรือองค์กรใหญ่ ๆ เท่านั้นที่สามารถมีกำลังทรัพย์ในการนำมาใช้ ซึ่งการแก้ไขปัญหาตรงนี้สำหรับการช่วยลดค่าใช้จ่ายในการจัดหาซุปเปอร์คอมพิวเตอร์มาใช้คือการนำคอมพิวเตอร์หลาย ๆ เครื่องมาช่วยในการ

ประมวลผลดังรูปที่ 2.13 แสดงการใช้คอมพิวเตอร์หลาย ๆ เครื่องในการประมวลผล แต่การทำงานในลักษณะนี้ค่อนข้างที่จะจัดการยากพอสมควร (Gu and Grossman, 2009)

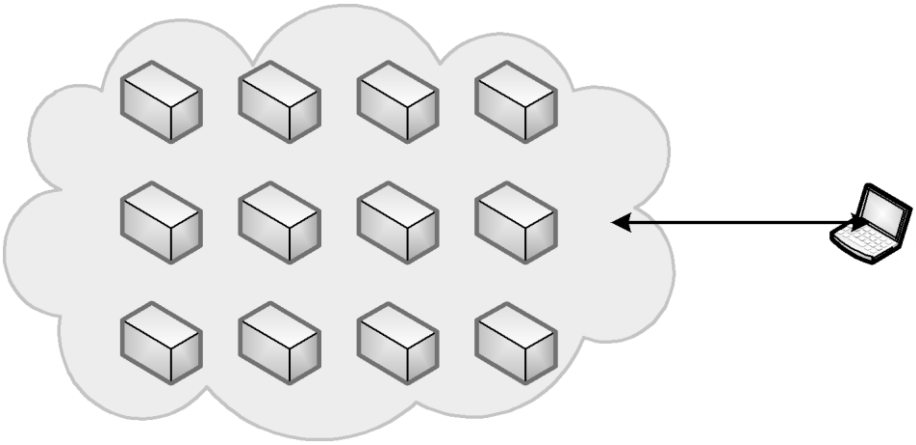
แต่ในปัจจุบันเทคโนโลยีได้ก้าวหน้าไปอย่างรวดเร็วทำให้การประมวลผลกับงานในลักษณะที่กล่าวมาข้างต้นสามารถทำได้ง่าย และสามารถช่วยลดค่าใช้จ่ายในการจัดหาซุปเปอร์คอมพิวเตอร์หรือคอมพิวเตอร์จำนวนมากได้ ซึ่งการประมวลผลแบบกลุ่มเมฆ (Cloud Computing) (Armbrust et al., 2010) เป็นการให้บริการทรัพยากรที่ใช้ในการประมวลผลข้อมูล โดยผู้ใช้เพียงแค่ร้องขอทรัพยากรที่ต้องการไปยังระบบแล้วระบบจะจัดสรรทรัพยากรให้กับผู้ใช้ ดังรูปที่ 2.14 แสดงการจัดการกับทรัพยากรในระบบแบบกลุ่มเมฆ ซึ่งหลังจากนั้นสามารถนำทรัพยากรที่ได้ไปประมวลกับข้อมูลที่มีขนาดใหญ่ได้ (Grossman และ Gu, 2008)



รูปที่ 2.12 การใช้ซุปเปอร์คอมพิวเตอร์ในการสกัดความรู้จากข้อมูลที่มีขนาดใหญ่ (Grossman and Gu, 2008)



รูปที่ 2.13 การใช้คอมพิวเตอร์หลาย ๆ เครื่องในการประมวลผล (Grossman and Gu, 2008)



รูปที่ 2.14 การจัดการกับทรัพยากรในระบบแบบกลุ่มเมฆ (Grossman and Gu, 2008)

2.4 Attempto Controlled English (ACE)

Attempto Controlled English หรือ ACE คือภาษาควบคุมที่มีลักษณะเป็นภาษาอังกฤษอย่างง่าย มีไวยากรณ์ที่จำกัด ถูกพัฒนาขึ้นมาโดยการรวมเอาข้อดีของภาษารูปนัย (Formal Language) และภาษาธรรมชาติเข้าด้วยกัน โดยออกแบบมาเพื่อให้ผู้เชี่ยวชาญสามารถเขียน หรืออ่านแทนองค์ความรู้ได้ง่ายด้วยประโยคภาษาอังกฤษทั่วไป จากรูปที่ 2.15 แสดงตัวอย่างการเปรียบเทียบระหว่าง FOL, DL, OWL, UML และ ACE ซึ่งผู้เชี่ยวชาญไม่จำเป็นต้องมีความรู้ทางด้านภาษาคอมพิวเตอร์ (Fuchs et al., 2006) รูปแบบการเขียนนั้นมีการกำหนดกฎของไวยากรณ์ คำศัพท์ และรูปแบบประโยคไว้ เพื่อไม่ให้เกิดความหลากหลายของตัวภาษา ACE สามารถนำองค์ความรู้ที่ได้จากผู้เชี่ยวชาญไปตรวจสอบความกำกวมและสามารถตีความองค์ความรู้ออกมาได้ (Fuchs et al., 2006)

first-order logic	$\forall X(protein(X) \rightarrow \exists Y(terminus(Y) \wedge has(X,Y)))$
DL	$Protein \sqsubseteq \exists has.Terminus$

OWL (RDF/XML)	`<owl:Class rdf:ID="Protein">`
	`  <rdfs:subClassOf>`
	`  <owl:Restriction>`
	`  <owl:onProperty rdf:resource="#has"/>`
	`  <owl:someValuesFrom rdf:resource="#Terminus"/>`
	`  </owl:Restriction>`
	`  </rdfs:subClassOf>`
	`</owl:Class>`

UML	
ACE	Every protein has a terminus.

รูปที่ 2.15 ตัวอย่างการเปรียบเทียบระหว่าง FOL, DL, OWL, UML และ ACE (Fuchs et al., 2006)

Every pet is an animal.

Dog is a pet.

Cat is a pet.

รูปที่ 2.16 ตัวอย่างประโยคในการแทนความรู้ใน ACE

If X is a pet then X is an animal.

If X is a dog then X is a pet.

If X is a cat then X is a pet.

รูปที่ 2.17 ตัวอย่างการแปลงกฎความสัมพันธ์ให้อยู่ในรูปแบบของ ACE

จากรูปที่ 2.16 แสดงตัวอย่างประโยคในการแทนความรู้ใน ACE ซึ่งสามารถเขียนในรูปแบบของประโยคภาษาอังกฤษอย่างง่าย งานวิจัยนี้จะนำเอากฎความสัมพันธ์มาเขียนในรูปแบบของ ACE ซึ่งสามารถเขียนได้ดังรูปที่ 2.17 เนื่องจากต้องการตรวจสอบความขัดแย้งและตีความของกฎความสัมพันธ์ที่ได้จากการหากฎความสัมพันธ์แบบกระจาย

2.4.1 ภาษาควบคุม

ภาษาควบคุมเป็นส่วนหนึ่งของภาษาธรรมชาติ (Natural Language) ซึ่งมีการกำหนดกฎการเขียนตามไวยากรณ์ คำศัพท์ และรูปแบบของภาษาไว้อย่างชัดเจน เพื่อช่วยลดความกำกวมและความซับซ้อนของภาษาธรรมชาติ (Kuhn, 2008) โดยมีการนำไปประยุกต์ใช้กับงานหลากหลายด้าน ตัวอย่างเช่น ใช้สำหรับการติดต่อสื่อสารกันระหว่างคอมพิวเตอร์กับมนุษย์ ใช้สำหรับการเขียนข้อกำหนดซอฟต์แวร์ (Software Specification) ใช้เพื่อสนับสนุนกระบวนการแสวงหาความรู้ (Knowledge Acquisition) และการแทนองค์ความรู้ (Knowledge Representation) ตัวอย่างของภาษาควบคุมได้แก่ Attempto Controlled English (Fuchs et al., 1999), PENG Processable English (Rolf, 2002), Common Logic Controlled English (Sowa, 2004), และ Boeing's Computer-Processable Language (Clark et al., 2005) เป็นต้น

2.4.2 กฎเกณฑ์การเขียนภาษาธรรมชาติด้วย ACE (ACE Construction Rules)

การเขียนภาษาธรรมชาติในรูปแบบของ ACE จำเป็นต้องมีข้อกำหนดในการเขียน ซึ่งมีข้อกำหนดหรือกฎเกณฑ์ในการเขียนประโยคมีรายละเอียดดังต่อไปนี้

2.4.2.1 คำ (Words)

ฟังก์ชันของคำและวลีใน ACE ได้ถูกกำหนดรูปแบบไว้แล้ว โดยผู้ใช้ไม่สามารถทำการเปลี่ยนแปลงหรือแก้ไขได้ ซึ่งประกอบด้วย คำนำหน้านาม (Determiners) คำบอกปริมาณ (Quantifiers) คำบุพบท (Prepositions) คำประสาน (Coordinators) เช่น ‘and’ ‘or’ เป็นต้น คำปฏิเสธ (Negation Words) เช่น ‘no’ ‘not’ ‘does not’ ‘is not’ ‘do not’ ‘are not’ เป็นต้น คำสรรพนาม (Pronouns) และ คำสอบถาม (Query Words) วลีที่มีการกำหนดไว้ล่วงหน้าคือ there is/are…such that และ it is false that… (Kaljurand, 2007)

2.4.2.2 วลี (Phrases)

วลีใน ACE ประกอบด้วยนามวลี และกริยาวลี โดยมีรายละเอียดดังนี้

- นามวลี (Noun Phrases) ประกอบด้วยคำนามวลีที่มีรูปเป็นเอกพจน์ (Singular Countable Noun Phrases) เช่น ‘a card’ ‘the card’ ‘1 card’ ‘one card’ ‘no card’ ‘every card’ ‘each card’ ‘not every card’ และ ‘not each card’ เป็นต้น คำนามวลีที่มีรูปเป็นพหูพจน์ (Plural Countable Noun Phrases) เช่น ‘the cards’ ‘some cards’ ‘all cards’ ‘no cards’ ‘nothing but cards’ และ ‘3 cards’ เป็นต้น ชื่อเฉพาะ(Proper Names) เช่น ‘John’ ‘Mr-Miller’ และ ‘Pi’ เป็นต้น ตัวเลขและข้อความ เช่น ‘12’ ‘-2’ ‘3.141’ และ ‘this is a string!’ เป็นต้น สรรพนามไม่สะท้อน (Non-Reflexive Pronouns) เช่น ‘it’ ‘he’ ‘she’ ‘he/she’ ‘they’ ‘him’ ‘her’ ‘him/her’ และ ‘them’ เป็นต้น สรรพนามสะท้อน (Reflexive Pronouns) เช่น ‘itself’ ‘himself’ ‘herself’ ‘himself/herself’ และ ‘themselves’ เป็นต้น คำสรรพนามที่ใช้แทนคำนามได้ทั่วไป (Indefinite Pronouns) เช่น ‘someone’ ‘somebody’ ‘something’ ‘no one’ ‘nobody’ ‘nothing’ ‘everyone’ ‘everybody’ ‘everything’ และ ‘not every one’ เป็นต้น ตัวแปร (Variables) เช่น ‘X’ ‘X1’ ‘Y’ และ ‘Y1’ เป็นต้น (Kaljurand, 2007)

- กริยาวลี (Verb Phrases) ใน ACE ประกอบด้วยอกรรมกริยา (Intransitive Verbs) เช่น ‘wait’ เป็นต้น สกรรมกริยา (Transitive Verbs) เช่น ‘enter something’ เป็นต้น และ กริยาทวิกรรม (Ditransitive Verbs) เช่น ‘give something to somebody’ หรือ ‘give someone something’ เป็นต้น (Kaljurand, 2007)

2.4.2.3 ประโยคบอกเล่า (Declarative Sentences)

ประโยคบอกเล่าอย่างง่ายนั้นประกอบด้วยนามวลีตามด้วยกริยาวลีและจบด้วยเครื่องหมายจุด ตัวอย่างเช่น (Kaljurand, 2007)

A customer enters a card.

Every customer enters a card.

นอกเหนือจากโครงสร้างประโยคทั่วไปแล้ว สามารถสร้างประโยคที่อยู่ในรูปแบบของคำนามวลีที่มีรูปเป็นเอกพจน์ โดยใช้วลี ‘there is/are’ นำหน้าประโยค ตัวอย่างเช่น (Kaljurand, 2007)

There is a customer that enters a card.

There are more than 6 customers.

ส่วนประกอบของประโยคบอกเล่าที่ถูกสร้างขึ้นจากประโยคง่าย ๆ หลายประโยคนั้นสามารถสร้างได้จากรูปแบบที่กำหนดไว้ เช่น คำประสาน คำปฏิเสธ คำบอกปริมาณ if-then สำหรับเชื่อมประโยคเข้าด้วยกัน เป็นต้น และจบประโยคด้วยเครื่องหมายจุด ตัวอย่างเช่น (Kaljurand, 2007)

There is a man X, and every woman likes the man X or every woman hates the man X.

ประโยคปฏิเสธสามารถใช้ ‘it is false that’ ไว้ในตำแหน่งต้นประโยคเพื่อแสดงว่าประโยคนั้น ๆ เป็นประโยคปฏิเสธที่สมบูรณ์ ตัวอย่างเช่น (Kaljurand, 2007)

It is false that John likes Mary.

It is false that John is a manager and that John likes Mary.

ประโยคเงื่อนไขสามารถใช้ ‘if’ และ ‘then’ ซึ่งทั้งสองจะต้องตามด้วยประโยค ตัวอย่างเช่น (Kaljurand, 2007)

If John enters a card then the clerk accepts it

2.4.2.4 ประโยคคำถาม (Interrogative Sentences)

ประโยคคำถามใน ACE นั้นยอมให้ใช้ใน 2 รูปแบบประโยคอย่างง่าย คือ ประโยคคำถามที่ต้องการคำตอบ ‘yes’ หรือ ‘no’ ต้นประโยคจะเริ่มต้นด้วย ‘does’ ‘do’ ‘is’ และ ‘are’ เป็นต้น และประโยคคำถามที่เริ่มต้นด้วย ‘who’ หรือ ‘what’ ซึ่งทุกประโยคคำถามจะจบประโยคด้วยเครื่องหมายคำถาม ตัวอย่างเช่น (Kaljurand, 2007)

Does John enter a card?

Is John a manager?

2.4.3 กฎเกณฑ์การตีความหมายภาษาธรรมชาติด้วย ACE (ACE Interpretation Rules)

ประโยคใน ACE นั้นสามารถตีความความหมายได้เพียงหนึ่งความหมาย แต่ในการตีความทางด้านภาษาอังกฤษนั้นประโยคใน ACE สามารถมีความหมายได้มากกว่าหนึ่งความหมาย ซึ่งจุดมุ่งหมายของการตีความใน ACE นั้นคือการตีความหมายที่เป็นไปได้ในประโยค ACE และโดยทั่วไปแล้วความต้องการของการสร้างเครื่องมือในการตีความของ ACE นั้น เพื่อให้เกิดความชัดเจนในด้านภาษาอังกฤษและในเอกสารของผู้ใช้โดยไม่จำเป็นต้องอธิบายในขั้นตอนของการจับคู่ประโยคเข้ากับรูปแบบเชิงตรรกะ (Kaljurand, 2007)

2.4.3.1 การบ่งปริมาณและขอบเขต (Quantifiers and their scope)

ACE มีตัวบ่งปริมาณอยู่ 3 ประเภทได้แก่ ตัวบ่งปริมาณทั้งหมด (Universal Quantifiers) คือ ‘every’ (alternatively ‘each’, ‘all’) และ ‘no’ ตัวบ่งปริมาณมีอย่างน้อยหนึ่ง (Existential Quantifiers) คือ ‘a’ (alternatively ‘an’, ‘some’) และตัวบ่งปริมาณตัวเลข เช่น ‘at least 2’ โดยตัวบ่งปริมาณเหล่านี้ใช้แทนคำนำหน้านาม นอกจากนี้ยังสามารถใช้ตัวบ่งปริมาณทั้งหมดเขียนไว้ที่ต้นประโยค คือ ‘for every’ (‘for each’, ‘for all’) และตัวบ่งปริมาณมีอย่างน้อยหนึ่ง คือ ‘there is/are … such that’ และเพื่อไม่ให้เกิดความกำกวมขึ้นได้มีตัวบ่งปริมาณที่ไม่ได้รับการอนุญาต เช่น ‘a’ ไม่สามารถใช้กับการบ่งปริมาณทั้งหมดได้แม้ว่าในภาษาอังกฤษบางครั้งสามารถทำได้ (Kaljurand, 2007)

2.4.3.2 การประสาน (Coordination)

ACE ได้ให้ตัวประสานมา 2 ตัว คือ ‘and’ และ ‘or’ ซึ่งใช้สำหรับการควบคุมผลลัพธ์ที่มีความคลุมเครือของระดับความผูกพัน เช่น ‘and’ มีความผูกพันที่แข็งแรงกว่า ‘or’ อย่างไรก็ตามสามารถทำย้อนกลับกันได้โดยเขียนเครื่องหมายจุลภาคนำหน้า ‘and’ และ ‘or’ ดังนั้นจึงใช้คำสั่งดังต่อไปนี้สำหรับบอกความผูกพัน คือ ‘a ≻ b’ ย่อมาจาก “a มีความผูกพันที่แข็งแรงกว่า b” (Kaljurand, 2007)

and ≻ or ≻, and ≻, or

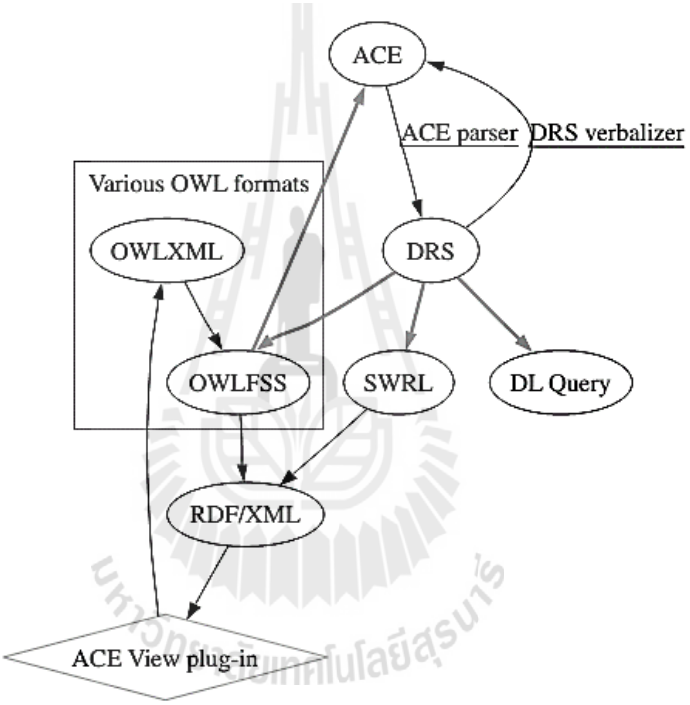
จากตัวอย่างแสดงขอบเขตของประโยคการเลือก ซึ่งแสดงอยู่ภายในวงเล็บ

A client {enters a UBS-card or enters a ZKB-card}, and types a code.

2.4.4 การแปลงรูปแบบของ ACE ให้อยู่ในรูปแบบของ OWL/SWRL

ตัวแปลงรูปแบบจาก ACE ไปเป็น OWL/SWRL นั้นจะใช้ ACE Parsing Engine (APE) ในการแปลงข้อความ ACE ให้อยู่ในรูปแบบของ Discourse Representation Structures (DRS) และถ้าการแปลงสำเร็จโดยที่ไม่มีความผิดพลาดของไวยากรณ์ของ ACE เกิดขึ้น แล้วจะทำ

การแปลงรูปแบบ DRS ให้อยู่ในรูปแบบของ Web Ontology Language (OWL) แต่ถ้าเกิดความล้มเหลวในขั้นตอนการแปลงไปเป็น OWL จะพยายามแปลงให้อยู่ในรูปแบบของ Semantic Web Rule Language (SWRL) แต่ถ้ายังเกิดความผิดพลาดขึ้นอีกจะทำการส่งค่าความผิดพลาดกลับไปให้ยังผู้ใช้รับทราบ ซึ่งถ้ากระบวนการที่กล่าวมาข้างต้นเสร็จสมบูรณ์ OWL หรือ SWRL จะถูกแปลงให้อยู่ในรูปแบบของภาษาที่ใช้สำหรับแลกเปลี่ยนข้อมูลระหว่าง ACE view และ OWL หรือ SWRL โดยใช้ Resource Description Framework (RDF) หรือ Extensible Markup Language (XML) ดังรูปที่ 2.18 แสดงแผนภาพการแปลง ACE ให้อยู่ในรูปแบบของ OWL/SWRL (Kaljurand, 2007)



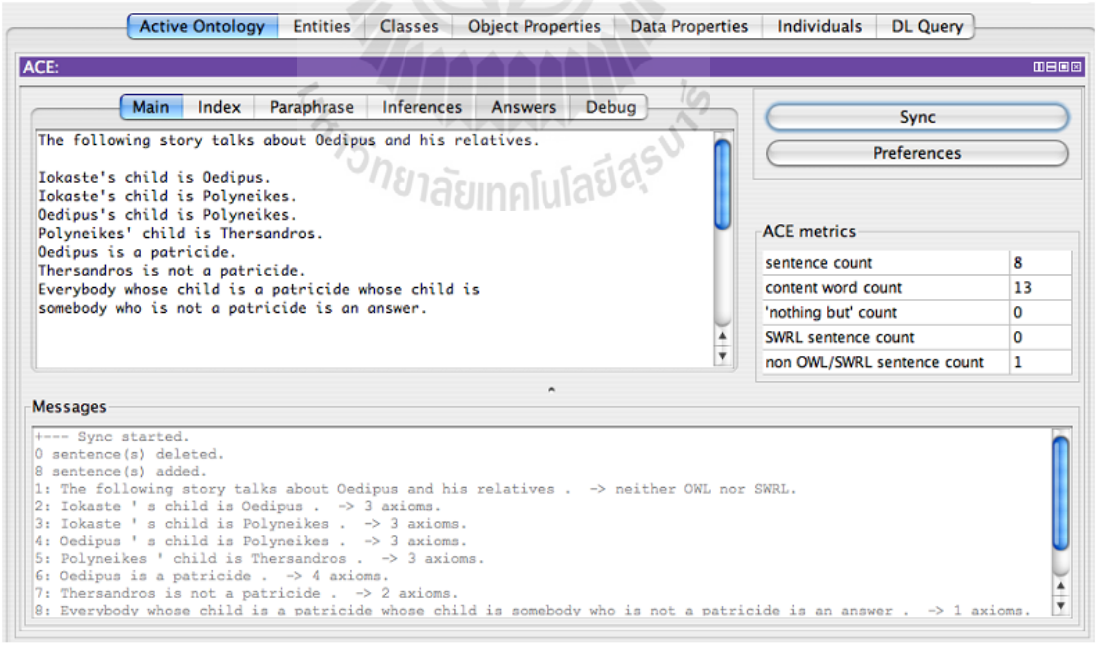
รูปที่ 2.18 แผนภาพการแปลงรูปแบบของ ACE ให้อยู่ในรูปแบบของ OWL/SWRL (Kaljurand, 2007)

2.4.5 การเชื่อมต่อของ ACE เพื่อทำงานบน Protégé

โปรแกรมประยุกต์ที่ทำหน้าที่เชื่อมต่อ ACE ให้สามารถทำงานอยู่บนโปรแกรม Protégé Editor เรียกว่า ACE View ซึ่งเป็นนวัตกรรมใหม่สำหรับการทำออนโทโลยีและการแก้ไขกฎ โดยเป้าหมายคือการลดความซ้ำซอนของการสำรวจและแก้ไขในรูปแบบของออนโทโทยี OWL และ SWRL โดยใช้ส่วนต่อประสานกับผู้ใช้ (User Interface) บน ACE ดังนั้น ACE View จึงมีความแตกต่างจากเครื่องมือแก้ไข OWL และ SWRL ทั่วไปที่มีความซับซ้อนในการใช้งาน ซึ่งอาจทำให้

เกิดความเข้าใจผิดในการใช้งานของผู้เชี่ยวชาญที่ไม่มีพื้นฐานทางด้านนี้ก็เป็นได้ และ ACE View ได้รวมเอา 2 เทคนิคการแปลงรูปแบบไว้กับปลั๊กอินบน Protégé Editor คือ ACE→OWL/SWRL และ OWL→ACE โดยจากรูปที่ 2.19 แสดงส่วนประกอบของ ACE View ปลั๊กอินบน Protégé Editor ซึ่งสามารถแบ่งการทำงานออกไปเป็นดังนี้ (Kaljurand, 2007)

- Main tab คือส่วนที่ใช้สำหรับการแทนความรู้ด้วยประโยคภาษาอังกฤษที่อยู่ในรูปแบบของ ACE เพื่อนำไปสร้างเป็นออนโทโลยี
- Index tab คือส่วนที่แสดงโครงสร้างของความรู้ที่ถูกแทนลงไปให้อยู่ในรูปแบบข้อความ และใช้ HTML ในการแสดงผลและนำทาง
- Paraphrase tab คือส่วนที่แสดงการแปลความหมายของความรู้ที่แทนเข้ามาในรูปแบบของ ACE
- Inferences tab คือส่วนที่แสดงข้อสรุปหรือความรู้ใหม่ที่ได้จากความรู้ที่ถูกแทนลงไป
- Answers tab คือส่วนที่ใช้สำหรับให้ผู้ใช้ถามและแสดงคำตอบ
- Debug tab คือส่วนที่ได้ตรวจสอบข้อผิดพลาดของความรู้ที่ถูกแทนลงไปในด้านเทคนิค และนำเสนอแนวทางแก้ไขให้กับผู้ใช้



รูปที่ 2.19 ตัวอย่างหน้าจอ ACE View ปลั๊กอินบน Protégé Editor (Kaljurand, 2007)

2.5 ฐานความรู้ออนโทโลยี (Ontology Knowledge Base)

ฐานความรู้ออนโทโลยี คือการบรรยายแนวคิดหรือข้อกำหนดของแนวคิดที่เราสนใจ โดยอดีตข้อมูลที่ถูกจัดเก็บมีแต่เพียงมนุษย์ที่เข้าใจความหมายแต่เครื่องจักรไม่สามารถเข้าใจความหมายนั้นได้ เพราะข้อมูลที่ถูกจัดเก็บนั้นยังขาดโครงสร้าง แต่ในปัจจุบันข้อมูลถูกจัดเก็บในรูปแบบของความสัมพันธ์เชิงความหมายมากยิ่งขึ้น นั้นคือ ฐานความรู้ออนโทโลยี คือ การเก็บข้อมูลที่ถูกอธิบายด้วยโครงสร้างความสัมพันธ์ ซึ่งจะอยู่ในรูปแบบของคลาส (Class) และสับคลาส (Sub-Class) (Gruber, 2014) โดยองค์ประกอบของฐานความรู้ออนโทโลยีประกอบด้วย ดังนี้

- แนวคิด (Concept) คือ ขอบเขตของฐานความรู้ออนโทโลยี ซึ่งสามารถนำไปอธิบายหรือตีความหมายได้
- คุณสมบัติ (Properties) คือ สิ่งที่ใช้อธิบายความรู้ที่นำมาใช้
- ความสัมพันธ์ (Relationships) คือ ความสัมพันธ์ระหว่างความรู้
- ข้อกำหนดของความสัมพันธ์ (Axioms) คือ เงื่อนไขสำหรับการแสดงความสัมพันธ์ระหว่างแนวคิดกับแนวคิด

2.6 FaCT++ Reasoner

FaCT++ Reasoner คือโปรแกรมประยุกต์ที่ถูกพัฒนามาจากอัลกอริทึม FaCT โดยใช้ภาษา C++ ในการพัฒนา ซึ่งมีพื้นฐานมาจาก Description Logics (DL) ซึ่งเป็นเครื่องมือสำหรับการตรวจสอบความขัดแย้งของออนโทโลยีที่ถูกสร้างขึ้นมา โดยการสร้างออนโทโลยีขึ้นมานั้นจะต้องไม่เกิดความขัดแย้งกันเอง และสามารถตีความจากออนโทโลยีเพื่อให้ได้ความรู้ใหม่ได้ (Tsarkov and Ian, 2006)

จากรูปที่ 2.20 แสดงตัวอย่างประโยคที่ขัดแย้งกันเอง จะเห็นได้ว่าประโยคที่ 1 และ 2 จากการตีความด้วย FaCT++ Reasoner จะได้ความรู้ใหม่ คือ “John is a human” ซึ่งขัดแย้งกับประโยคที่ 3 ที่บอกว่า “John is not a human” ทำให้ 3 ประโยคนี้เกิดความขัดแย้งกันเอง ไม่สามารถนำไปสร้างเป็นออนโทโลยีได้

1. If X is a man then X is a human.
2. If X is a John then X is a man.
3. If X is a John then X is not a human

รูปที่ 2.20 ตัวอย่างประโยคที่ขัดแย้งกันเอง

2.7 การอนุมานเชิงตรรกะ (Logical Inference)

การอนุมานเชิงตรรกะคือการกระทำหรือกระบวนการในการหาข้อสรุปจากความรู้ที่เราเข้าใจ ซึ่งเป็นสิ่งที่ไม่ได้บอกออกมาอย่างชัดเจนหรือระบุไว้โดยตรง แต่ความรู้ที่สามารถอนุมานได้นั้นมักจะมีเหตุที่ให้เราสามารถอนุมานหรือจะเข้าใจได้ว่าเป็นเช่นนั้น (Wikipedia, 2014b) ตัวอย่างเช่น

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal.

จากตัวอย่างประโยคที่ 1 และ 2 บอกว่าผู้ชายทุกคนเป็นมนุษย์และ Socrates คือผู้ชาย เราสามารถอนุมานจากสองประโยคแรกได้ว่า Socrates คือมนุษย์ แต่ในการอนุมานควรระวังในส่วนของความรู้หรือประโยคที่เป็นจริง แต่นำไปสู่การสรุปหรือการอนุมานที่ผิดพลาดได้ ตัวอย่างเช่น

- All apples are fruit. (Correct)
- Bananas are fruit. (Correct)
- Therefore, bananas are apples. (Wrong)

จากตัวอย่างประโยคที่ 1 และ 2 บอกว่าแอปเปิลทุกลูกคือผลไม้เป็นจริง และกล้วยคือผลไม้เป็นจริง ดังนั้นสามารถอนุมานได้ว่ากล้วยคือแอปเปิล ซึ่งจากการอนุมานจะเห็นได้ว่าเป็นเท็จเนื่องจากความเป็นจริงแล้วกล้วยไม่ใช่แอปเปิล

การอนุมานเชิงตรรกะสามารถแบ่งออกได้เป็น 2 แบบ ได้แก่ การอนุมานเชิงอุปนัย (Inductive Inference) (Angluin and Smith, 1983) คือ การหาข้อสรุปจากข้อสังเกตหรือผลการทดลอง และการอนุมานเชิงนิรนัย (Deductive Inference) (MacGregor, 1991) คือ การหาข้อสรุปจากหลักความเป็นจริง โดยการอนุมานเชิงตรรกะได้ถูกนำไปใช้งานทางคอมพิวเตอร์หลายด้าน เช่น ด้านระบบผู้เชี่ยวชาญ (Expert System) ด้านเว็บเชิงความหมาย (Semantic Web) เป็นต้น

2.8 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้องกับการหากฎความสัมพันธ์แบบกระจายนั้นยังปรากฏน้อยมากเมื่อเปรียบเทียบกับงานวิจัยทางด้านการจำแนกและการจัดกลุ่มข้อมูลแบบกระจาย เนื่องจากการหากฎความสัมพันธ์แบบดั้งเดิมจำเป็นต้องวิเคราะห์ข้อมูลทั้งหมดพร้อม ๆ กัน ซึ่งเป็นเรื่องยากที่จะกระจายข้อมูลหรือนำข้อมูลที่กระจายกันอยู่มาหากฎความสัมพันธ์ โดยงานวิจัยที่ปรากฏอยู่ประกอบด้วยงานในส่วนการนำเสนอแนวคิดการหากฎความสัมพันธ์แบบขนาน การปรับปรุงอัลกอริทึมการหากฎความสัมพันธ์แบบขนานให้รวดเร็ว การนำเสนอแนวคิดการหากฎ

ความสัมพันธ์จากข้อมูลแบบกลุ่มเมฆ และผู้วิจัยได้ทำการศึกษาค้นคว้างานวิจัยที่มีความเกี่ยวข้องกับงานวิจัยที่จะทำโดยมีรายละเอียดสรุปได้ดังนี้

Agrawal และ Shafer (1996) ได้เสนอการหากฎความสัมพันธ์แบบขนานโดยอาศัยพื้นฐานจากการนับความถี่แบบกระจาย (Count Distribution) และการกระจายข้อมูล (Data Distribution) ด้วยเหตุที่ข้อมูลมีการเจริญเติมโตขึ้นตามเทคโนโลยีทำให้ข้อมูลมีขนาดใหญ่ ซึ่งทำให้การหากฎความสัมพันธ์จากข้อมูลลักษณะนี้ต้องใช้เวลานานในการประมวลผล ดังนั้นการหากฎความสัมพันธ์แบบขนานสามารถเข้ามาช่วยแก้ไขปัญหาตรงนี้ได้ โดยขั้นตอนจะแบ่งข้อมูลออกเป็นชุด ๆ เพื่อนำข้อมูลที่ได้ไปกระจายให้กับโปรเซสเซอร์ในการนับความถี่ของแต่ละไอเท็มเซต ซึ่งในขั้นตอนนี้จะเป็นการทำงานแบบขนานที่สามารถช่วยลดเวลาในการหากฎความสัมพันธ์ แต่สิ่งที่ต้องเสียเพิ่มขึ้นจากกระบวนการนี้คือจำนวนหน่วยความจำที่จะต้องใช้ (Memory Used) จำนวนของโปรเซสเซอร์ (Amount of Processor) และจำนวนช่องทางในการติดต่อสื่อสาร (Amount of Communication)

Li et al. (2003) ได้เสนอการทำเหมืองข้อมูลแบบกระจายโดยพิจารณาจากความคล้ายคลึงด้วยเทคโนโลยีการจัดเก็บข้อมูลที่มีการพัฒนามาอย่างต่อเนื่อง ทำให้การจัดเก็บข้อมูลต่าง ๆ ไว้ในฐานข้อมูลนั้นทำได้ง่าย ซึ่งความสามารถนี้เองทำให้ข้อมูลได้ถูกจัดเก็บไว้ในแหล่งต่าง ๆ ซึ่งเป็นเรื่องยากสำหรับการสกัดความรู้จากข้อมูลเหล่านี้ เนื่องจากข้อมูลไม่ได้ถูกจัดเก็บไว้ในแหล่ง ๆ เดียวที่สามารถสกัดความรู้แบบตรงไปตรงมา ดังนั้นงานวิจัยของทีมนี้จึงได้นำเสนอเทคนิคการวัดความคล้ายคลึงเพื่อนำไปวัดความคล้ายคลึงระหว่างข้อมูลแต่ละแหล่ง ถ้าข้อมูลมีความคล้ายคลึงกันมากก็จะนำข้อมูลเหล่านั้นมายุบรวมกันแล้วนำผลลัพธ์ที่ได้ไปสกัดความรู้ด้วยเทคนิคต่าง ๆ เช่น การจำแนกข้อมูล การจัดกลุ่มข้อมูล และการหากฎความสัมพันธ์ เป็นต้น

Yu et al. (2010) ได้เสนออัลกอริทึม Distributed Parallel Apriori (DPA) เป็นการปรับปรุงการหากฎความสัมพันธ์แบบขนาน โดยขั้นตอนหนึ่งในอัลกอริทึมการหากฎความสัมพันธ์ คือการนับความถี่ที่ปรากฏขึ้นบ่อยมีโครงสร้างของการทำงานที่ไม่ขึ้นต่อกัน ซึ่งในขั้นตอนนี้สามารถนำไปปรับปรุงให้มีการทำงานแบบขนานเพื่อเพิ่มประสิทธิภาพการหากฎความสัมพันธ์แบบดั้งเดิมให้รวดเร็วยิ่งขึ้น แต่ด้วยเทคนิคในลักษณะนี้อาจจะเกิดการว่างงานของแต่ละโปรเซสเซอร์ขึ้น ดังนั้นงานวิจัยนี้จึงได้นำเสนอเทคนิคโหลดบาลานซ์ (Load Balance) เพื่อมาปรับปรุงอัลกอริทึมเดิมเพื่อช่วยลดเวลาในการหากฎความสัมพันธ์ โดยการทดสอบประสิทธิภาพอัลกอริทึมของทีมวิจัยนี้ใช้การเปรียบเทียบประสิทธิภาพกับอัลกอริทึมของงานวิจัยอื่น และมีการสรุปผลการทดลองว่าอัลกอริทึมของทีมวิจัยนี้มีประสิทธิภาพดีกว่างานวิจัยอื่นเมื่อใช้กับค่าสนับสนุนขั้นต่ำที่น้อย

Tseng et al. (2010) ได้เสนอการหากฎความสัมพันธ์ด้วยเทคนิคการจัดกลุ่มของคอมพิวเตอร์หลาย ๆ เครื่อง เพื่อให้สามารถทำงานได้เสมือนกับเป็นคอมพิวเตอร์เครื่องเดียวกัน

(De-Clustering) เนื่องจากงานวิจัยส่วนมากที่ทำการหากฎความสัมพันธ์จากข้อมูลที่มีขนาดใหญ่เพื่อให้ใช้เวลาที่น้อยลง แต่ใช้จำนวนช่องทางในการติดต่อสื่อสารกันระหว่างกระบวนการนั้นมาก ดังนั้นทีมวิจัยนี้จึงได้เสนอการหากฎความสัมพันธ์ด้วยเทคนิคการจัดกลุ่มของคอมพิวเตอร์หลาย ๆ เครื่อง ด้วยอัลกอริทึม Shortest Spanning Path (SSP) ซึ่งจะสามารถช่วยลดการติดต่อสื่อสารกันระหว่างกระบวนการและลดการติดต่อสื่อสารกันระหว่างอัลกอริทึมและฐานข้อมูล โดยการทดลองของทีมวิจัยนี้ใช้ข้อมูลที่สังเคราะห์ขึ้นเอง ซึ่งจะเปรียบเทียบในสองแง่มุม คือ ขนาดของข้อมูลที่แตกต่างกัน และจำนวนกลุ่มที่แตกต่างกัน โดยทั้งหมดจะเปรียบเทียบด้วยค่าความถูกต้อง (Precision) และค่าความครบถ้วน (Recall)

Elayyadi et al. (2014) ได้เสนอการหากฎความสัมพันธ์จากข้อมูลแบบกลุ่มเมฆ เนื่องจากข้อมูลที่ถูกจัดเก็บอยู่บนกลุ่มเมฆอาจเกิดการขาดหายไปของข้อมูล (Missing Value) ซึ่งเมื่อนำข้อมูลในลักษณะดังกล่าวไปหากฎความสัมพันธ์อาจทำให้เกิดการขาดหายไปของค่าความเชื่อมั่น (Missing Confidence) ในบางกฎความสัมพันธ์ ดังนั้นงานวิจัยนี้จึงได้นำเสนอเทคนิคปริมาณเทนเซอร์ (Tensor) สำหรับจัดเก็บค่าความเชื่อมั่นของแต่ละกฎความสัมพันธ์ เพื่อนำผลลัพธ์ที่ได้ไปประมาณค่าความเชื่อมั่นที่ขาดหายไปด้วยอัลกอริทึม Conjugate Gradient โดยการทดลองของทีมวิจัยนี้ใช้ข้อมูลที่สังเคราะห์ขึ้นเองทั้งหมด 10 ชุด โดยแต่ละชุดจะมีข้อมูลที่ขาดหายไปแล้วนำไปหากฎความสัมพันธ์ ซึ่งการเปรียบเทียบประสิทธิภาพจะพิจารณาจากค่าความผิดพลาดของการประมาณค่าความเชื่อมั่นของแต่ละชุดข้อมูล

จากการศึกษางานวิจัยที่เกี่ยวข้องพบว่าการหากฎความสัมพันธ์กับข้อมูลที่มีขนาดใหญ่นั้นใช้เวลามากในการประมวลผล และงานวิจัยส่วนมากที่ปรากฏอยู่จะเป็นการหากฎความสัมพันธ์แบบขนานเพื่อช่วยลดเวลาในการหากฎความสัมพันธ์จากข้อมูลที่มีขนาดใหญ่ ซึ่งจะทำเฉพาะในส่วนของการนับความถี่แต่ละไอเท็มเซต แต่ไม่ได้หมายถึงกระบวนการทั้งหมดของการหากฎความสัมพันธ์ ในส่วนนี้อาจทำให้เกิดปัญหาคอขวดได้ คือการรอให้ทุกโปรเซสทำงานเสร็จก่อนถึงจะสามารถทำงานในส่วนต่อไปได้ และการหากฎความสัมพันธ์แบบขนานไม่สามารถใช้กับกรณีที่ข้อมูลถูกกระจายกันอยู่ตามแหล่งต่าง ๆ อยู่ก่อนแล้ว ซึ่งข้อมูลในลักษณะนี้ยังปรากฏงานวิจัยอยู่น้อยมากเนื่องจากอัลกอริทึมการหากฎความสัมพันธ์แบบดั้งเดิมถูกออกแบบมาใช้สำหรับข้อมูลเพียงชุดเดียว ในงานวิจัยนี้จึงได้ออกแบบและพัฒนากลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งจากการหากฎความสัมพันธ์ด้วยข้อมูลที่มีขนาดใหญ่หรือข้อมูลที่กระจายกันอยู่ตามแหล่งต่าง ๆ โดยสาระสำคัญในงานวิจัยนี้เมื่อเปรียบเทียบกับงานวิจัยอื่นสรุปได้ดังตารางที่ 2.4

ตารางที่ 2.4 สรุปเปรียบเทียบงานวิจัยที่เกี่ยวข้องกับกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง

กระบวนการทำงาน	งานวิจัยที่เกี่ยวข้อง					
	ก	ข	ค	ง	จ	ฉ*
อัลกอริทึมการหาความความสัมพันธ์แบบกระจาย						
อัลกอริทึม Apriori	✓	✓	✓	✓	✓	✓
การหากฎความสัมพันธ์แบบขนาน	✓		✓			
การหากฎความสัมพันธ์ด้วยเทคนิค De-Clustering				✓		
ข้อมูลหลายชุดสคีมาเดียวกัน	✓		✓	✓	✓	✓
ข้อมูลหลายชุดสคีมาต่างกัน		✓				
หากฎความสัมพันธ์ใหม่จากกฎความสัมพันธ์เดิม						✓
เกณฑ์การประเมินประสิทธิภาพกฎความสัมพันธ์						
Support	✓	✓	✓			✓
Confidence	✓	✓	✓		✓	✓
F-Measure				✓		
ความถูกต้องของกฎความสัมพันธ์						✓
ตรวจสอบความขัดแย้ง						✓
จำนวนกฎความสัมพันธ์						✓
เวลาที่ใช้ในการหากฎความสัมพันธ์	✓		✓	✓		
ขอบเขตของการวิจัย						
วิจัยเพื่อทดสอบประสิทธิภาพ	✓	✓	✓	✓	✓	✓
วิจัยเพื่อเสนอแนวคิดใหม่	✓	✓	✓	✓	✓	✓
มีการประยุกต์ใช้กับข้อมูลจริง	✓	✓	✓			✓

หมายเหตุ งานวิจัยที่เกี่ยวข้อง ประกอบด้วย

ก แทนงานวิจัยของ Agrawal and Shafer (1996)

ข แทนงานวิจัยของ Li et al. (2003)

ค แทนงานวิจัยของ Yu et al. (2010)

ง แทนงานวิจัยของ Tseng et al. (2010)

จ แทนงานวิจัยของ Elayyadi et al. (2014)

ฉ* แทนงานวิจัยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง (งานวิจัยของวิทยานิพนธ์ฉบับนี้)

บทที่ 3
วิธีดำเนินการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อเสนอกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง ในบทนี้จะกล่าวถึง ขั้นตอนการดำเนินงานวิจัย วิธีการวิจัย เครื่องมือที่ใช้ในการวิจัย และกระบวนการต่าง ๆ ของการวิจัย โดยมีรายละเอียดดังนี้

3.1 ขั้นตอนการดำเนินงานวิจัย

งานวิจัยนี้ได้ศึกษาและพัฒนาอัลกอริทึมกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งขั้นตอนของงานวิจัยนี้แบ่งออกเป็น

1) การศึกษาการหากฎความสัมพันธ์แบบกระจาย ซึ่งงานวิจัยส่วนมากได้เสนอแนวคิดเกี่ยวกับการหากฎความสัมพันธ์แบบกระจายที่สามารถช่วยลดเวลาในการหากฎความสัมพันธ์ นั้นคือการหากฎความสัมพันธ์แบบขนาน แต่ต้องใช้คอมพิวเตอร์ประสิทธิภาพสูงในการประมวลผล และจำเป็นต้องปรับปรุงอัลกอริทึมเดิมของการหากฎความสัมพันธ์เพื่อให้มีความเหมาะสมกับเทคโนโลยี

2) การศึกษาภาษาธรรมชาติและการอนุมานเชิงตรรกะ ซึ่งการรวมกฎความสัมพันธ์จากหลายแหล่งนั้นอาจทำให้เกิดความขัดแย้งกันเองของกฎ และกฎความสัมพันธ์ที่ได้อาจยังมีบางกฎความสัมพันธ์ที่ขาดหายไป ดังนั้นในงานวิจัยนี้ใช้เทคนิคการอนุมานเชิงตรรกะมาช่วยแก้ไขปัญหาตรงนี้ โดยจำเป็นต้องแปลงรูปแบบของกฎความสัมพันธ์ให้อยู่ในรูปแบบของออนโทโลยีโดยใช้ภาษาธรรมชาติในการแทนความรู้เพื่อนำไปสร้างเป็นออนโทโลยี

3) การออกแบบอัลกอริทึมรวมกฎความสัมพันธ์จากแหล่งความรู้ต่าง ๆ โดยจะนำกฎความสัมพันธ์ที่ปรากฏทุกแหล่งความรู้มาเก็บไว้ในแหล่งความรู้เพียงแหล่งเดียว เนื่องกฎความสัมพันธ์เหล่านี้ปรากฏขึ้นบ่อยคือกฎความสัมพันธ์ที่น่าสนใจ และมีประสิทธิภาพพอสำหรับการนำไปทำนายผลในอนาคต

4) การออกแบบอัลกอริทึมการแปลงรูปแบบกฎความสัมพันธ์ทั่วไปให้อยู่ในรูปแบบภาษาธรรมชาติ เนื่องจากการตรวจสอบความขัดแย้งข้อมูลจำเป็นต้องอยู่ในรูปแบบของออนโทโลยี ดังนั้นการที่จะแทนความรู้ด้วยกฎความสัมพันธ์จำเป็นต้องแปลงกฎความให้อยู่ในรูปแบบของ

ภาษาธรรมชาติ โดยจะใช้เทคนิคการค้นหาและแทนที่คำที่ต้องการเพื่อให้กฎความสัมพันธ์อยู่ในรูปแบบของภาษาธรรมชาติยอมรับได้

5) การออกแบบอัลกอริทึมตรวจสอบความขัดแย้งของกฎความสัมพันธ์ที่ได้จากแหล่งความรู้ต่าง เนื่องจากกฎความสัมพันธ์ที่ได้จากหลายแหล่งอาจเกิดความขัดแย้งเกิดขึ้นได้ ดังนั้นจำเป็นต้องตรวจสอบความขัดแย้งของกฎความสัมพันธ์ เมื่อเกิดความขัดแย้งเกิดขึ้นจะทำการลบกฎความสัมพันธ์นั้น ๆ ทิ้งไป และสิ่งที่ได้หลังจากกระบวนการนี้นอกจากกฎความสัมพันธ์ที่ไม่ขัดแย้งกันเองแล้ว คือกฎความรู้ใหม่ที่สามารถนำไปเพิ่มเติมกฎความสัมพันธ์ที่ขาดหายไปได้

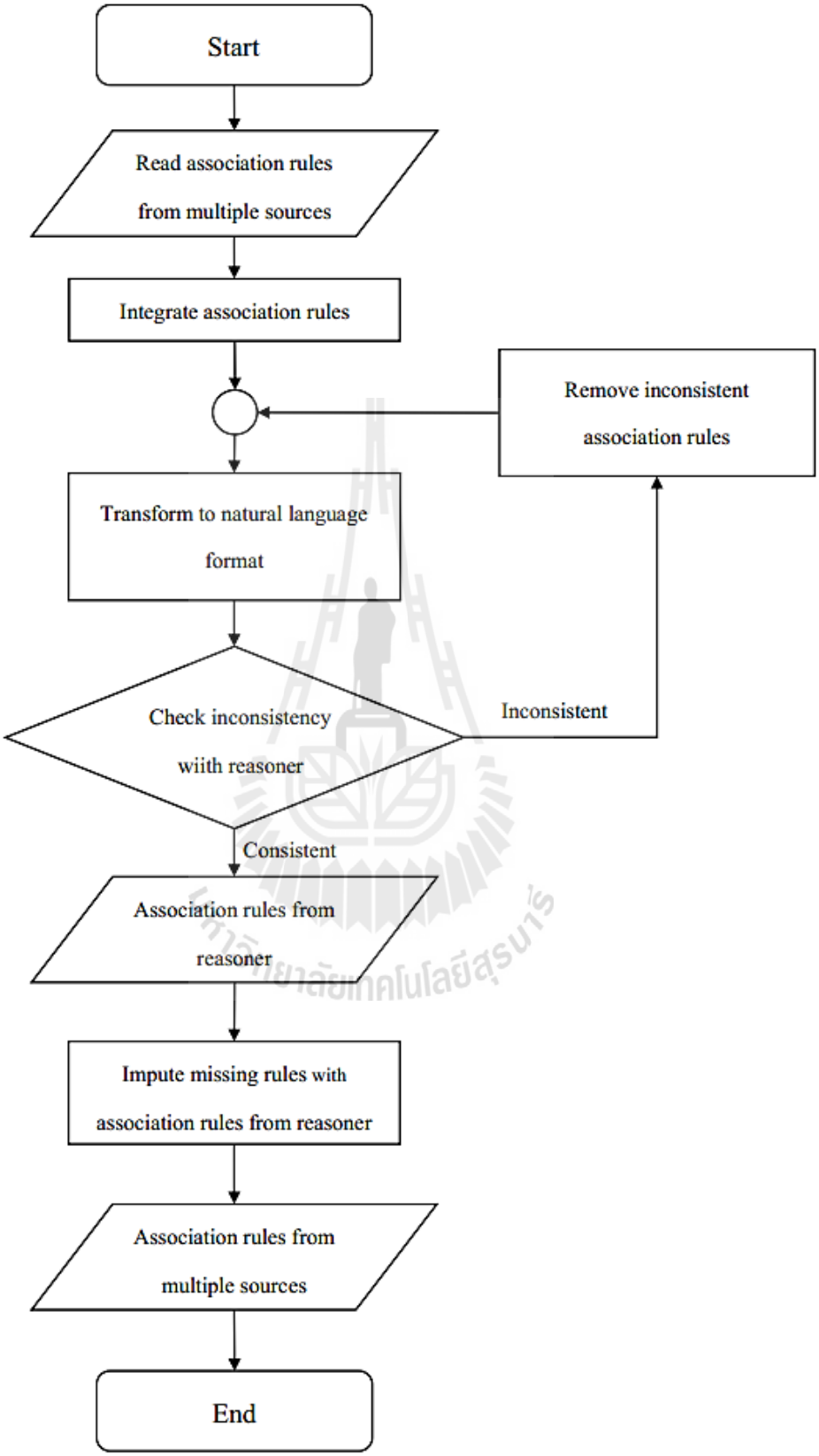
3.2 กรอบแนวคิดของการวิจัย

กรอบแนวคิดของงานวิจัยนี้คือกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง ซึ่งสามารถแบ่งลักษณะของข้อมูลออกไปเป็น 2 แบบ คือ ข้อมูลที่มีการกระจายกันอยู่แล้ว และข้อมูลขนาดใหญ่ในแหล่งเดียวที่ถูกกระจายข้อมูลออกเป็นหลายชุดย่อย ข้อมูลตามลักษณะที่กล่าวมานี้ไม่สามารถหากฎความสัมพันธ์ได้แบบตรงไปตรงมาเหมือนกับการหากฎความสัมพันธ์แบบดั้งเดิมที่รวมข้อมูลไว้เพียงแหล่งเดียวแล้วสามารถหากฎความสัมพันธ์ทั้งหมดในคราวเดียว ซึ่งในส่วนที่ทำให้การหากฎความสัมพันธ์แบบกระจายนั้นทำได้ยาก คือการรวมกฎความสัมพันธ์ที่ได้จากแหล่งความรู้ต่าง ๆ ให้อยู่ในฐานความรู้เพียงชุดเดียวแล้วมีประสิทธิภาพการค้นหากฎใกล้เคียงกับการหากฎความสัมพันธ์แบบดั้งเดิมนั้นทำได้ยาก ดังนั้นงานวิจัยนี้จึงเสนอกรอบแนวคิดกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง โดยมีกรอบแนวคิดดังรูปที่ 3.1

จากกรอบแนวคิดกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งตามรูปที่ 3.1 สามารถแบ่งการทำงานออกเป็น 5 ขั้นตอน คือ

ขั้นตอนที่ 1 คือหากฎความสัมพันธ์จากข้อมูลที่กระจายกันอยู่ ซึ่งข้อมูลในส่วนนี้จะมาจากข้อมูลที่มีการกระจายกันอยู่แล้ว หรือข้อมูลขนาดใหญ่ที่ถูกกระจายข้อมูลอออกเป็นชุด ๆ โดยในขั้นตอนนี้กระบวนการหากฎความสัมพันธ์ในแต่ละชุดข้อมูลนั้นจะไม่ขึ้นต่อกัน ผลลัพธ์ที่ได้จากขั้นตอนนี้ คือฐานความรู้ที่เก็บกฎความสัมพันธ์จากข้อมูลที่กระจายกันอยู่ ซึ่งจำนวนของฐานความรู้ที่จะได้นั้นจะขึ้นอยู่กับจำนวนของชุดข้อมูลที่กระจายกันอยู่

ขั้นตอนที่ 2 คือการรวมกฎความสัมพันธ์ที่ได้จากขั้นตอนที่ 1 ซึ่งจะดึงกฎความสัมพันธ์ที่มีลักษณะเหมือนกันออกมาจากฐานความรู้ที่กระจายกันอยู่เพื่อให้ได้ฐานความรู้เพียงชุดเดียว $C = R_1 \cap R_2 \cap ... \cap R_i$ โดย $i = 1,2,3,...,n$



รูปที่ 3.1 กรอบแนวคิดกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง

ขั้นตอนที่ 3 เนื่องจากกฎความสัมพันธ์ที่ได้จากขั้นตอนที่ 2 เป็นกฎความสัมพันธ์ที่ได้จากการหากฎความแบบกระจาย ปัญหาที่อาจจะตามมาจากการหากฎความสัมพันธ์ในลักษณะนี้ คืออาจเกิดความขัดแย้งกันเองของกฎความสัมพันธ์ และอาจเกิดการขาดหายไปของกฎความสัมพันธ์เมื่อเทียบกับการหากฎความสัมพันธ์แบบดั้งเดิม ดังนั้นขั้นตอนนี้จะเป็นการแปลงรูปแบบของกฎความสัมพันธ์จากรูปแบบทั่วไปให้อยู่ในรูปแบบของภาษาธรรมชาติเพื่อนำไปใช้ในขั้นตอนต่อไป

ขั้นตอนที่ 4 กฎความสัมพันธ์ที่อยู่ในรูปแบบของภาษาธรรมชาติที่ได้จากขั้นตอนที่ 3 จะนำไปตรวจสอบความขัดแย้ง โดยผลลัพธ์ที่ได้จากขั้นตอนนี้คือสามารถบอกได้ว่ากฎความสัมพันธ์ที่ได้จากกระบวนการก่อนหน้านี้มีความขัดแย้งกันหรือไม่ ถ้าเกิดความขัดแย้งเกิดขึ้นจริงจะทำการลบกฎความสัมพันธ์นั้นทิ้งไปและทำการตรวจสอบความขัดแย้งใหม่ และจะได้ความรู้ใหม่จากกฎความสัมพันธ์ที่มีอยู่เดิม

ขั้นตอนที่ 5 จากความรู้ใหม่ที่ได้จากกระบวนการของขั้นตอนที่ 4 สามารถนำไปสร้างเป็นกฎความสัมพันธ์แล้วนำไปเพิ่มเติมจากกฎความสัมพันธ์ของเดิมที่มีอยู่แล้วได้ ซึ่งในส่วนนี้สามารถชดเชยกฎความสัมพันธ์ที่ขาดหายไปจากที่กล่าวมาแล้วในข้างตนได้ สุดท้ายจะได้ฐานความรู้ของกฎความสัมพันธ์เพียงชุดเดียวที่มาจากการหากฎความสัมพันธ์แบบกระจายที่มีประสิทธิภาพใกล้เคียงกับการหากฎความสัมพันธ์แบบดั้งเดิม

3.3 การออกแบบอัลกอริทึม

3.3.1 อัลกอริทึมการรวมกฎความสัมพันธ์จากแหล่งความรู้ต่าง ๆ

การรวมกฎความสัมพันธ์จากแหล่งความรู้ต่าง ๆ นั้นจะรวมโดยนำมาเฉพาะกฎความสัมพันธ์ที่ปรากฏขึ้นในทุกแหล่งความรู้ เนื่องจากกฎความสัมพันธ์ที่ปรากฏในทุกแหล่งความรู้นั้น หมายถึงกฎความสัมพันธ์นั้น ๆ มีความสำคัญและมีประสิทธิภาพในการนำไปทำนายผลข้อมูลในอนาคต แต่การนำกฎความสัมพันธ์มารวมกันทั้งหมดอาจได้กฎความสัมพันธ์ที่มากและบางกฎความสัมพันธ์ที่ได้อาจไม่มีประสิทธิภาพเพียงพอสำหรับการนำไปทำนายผลข้อมูลในอนาคต ดังนั้นการรวมกฎความสัมพันธ์จะทำการดึงมาเฉพาะกฎความสัมพันธ์ที่ปรากฏทุก ๆ แหล่งความรู้ จากตารางที่ 3.1 แสดงตัวอย่างข้อมูลผู้ป่วยโรคมะเร็งเต้านมจำนวน 6 คอลัมน์ และนำไปแบ่งข้อมูลออกเป็น 3 ชุด สำหรับการนำไปหากฎความสัมพันธ์แล้วนำกฎความสัมพันธ์ที่ได้ไปสู่ขั้นตอนของการรวมกฎความสัมพันธ์ ดังรูปที่ 3.3 แสดงตัวอย่างการรวมกฎความสัมพันธ์จากแหล่งความรู้ต่าง ๆ ซึ่งมีแหล่งความรู้ทั้ง 3 แหล่ง คือ R_1, R_2, R_3 แล้วนำมารวมให้เป็นฐานความรู้เพียงชุดเดียวได้เป็น C

จากรูปที่ 3.2 แสดงคำสั่งเทียมขั้นตอนการรวมกฎความสัมพันธ์จากแหล่งความรู้ต่าง ๆ โดยจะรับข้อมูลเป็นกฎความสัมพันธ์ที่ได้จากแหล่งความรู้ต่าง ๆ คือ $R_1 \cap R_2 \cap ... \cap R_N$ โดย $N = 1,2,3,...,i$ ซึ่งแหล่งข้อมูลที่นำมานี้อาจจะถูกจัดเก็บอยู่บนคอมพิวเตอร์หรือฐานข้อมูลบนคอมพิวเตอร์แม่ข่าย หรืออาจจะอยู่ในรูปแบบของไฟล์ก็ได้ แล้วนำไปดึงกฎความสัมพันธ์ที่ปรากฏในทุก ๆ แหล่งความรู้ คือ $C = R_1 \cap R_2 \cap ... \cap R_N$ โดย $N = 1,2,3,...,i$ ผลลัพธ์ที่ได้คือฐานความรู้กฎความสัมพันธ์เพียงชุดเดียวที่ได้จากการรวมกฎความสัมพันธ์จากแหล่งความรู้ต่าง ๆ

Algorithm Integrate Association Rules

//Input: {R_1, R_2, …, R_N}, association rules in all nodes.

//Output: C, an association rule set.

1. Create C as empty list;
2. For i =1 ← to N do
3. C = intersection (R_i)
4. End for
5. Return C;

รูปที่ 3.2 คำสั่งเทียมขั้นตอนการรวมกฎความสัมพันธ์จากแหล่งความรู้ต่าง ๆ

ตารางที่ 3.1 ตัวอย่างข้อมูลผู้ป่วยโรคมะเร็งเต้านมจำนวน 6 คอลัมน์

age	menopause	tumor-size	inv-nodes	node-caps	Class
40-49	premeno	15-19	0-2	yes	recurrence-events
50-59	ge40	15-19	0-2	no	no-recurrence-events
50-59	ge40	35-39	0-2	no	recurrence-events
50-59	ge40	15-19	0-2	no	no-recurrence-events
40-49	premeno	15-19	0-2	yes	recurrence-events

R_1	R_2	R_3
{Class=no-recurrence-events}=>{inv-nodes=0-2}	-	-
{inv-nodes=0-2}=>{irradiat=no}	{inv-nodes=0-2}=>{irradiat=no}	{inv-nodes=0-2}=>{irradiat=no}
{irradiat=no}=>{node-caps=no}	{irradiat=no}=>{node-caps=no}	{irradiat=no}=>{node-caps=no}

R
{inv-nodes=0-2}=>{irradiat=no}
{irradiat=no}=>{node-caps=no}

รูปที่ 3.3 ตัวอย่างการรวมกฎความสัมพันธ์จากแหล่งความรู้ต่าง ๆ

3.3.2 อัลกอริทึมการแปลงรูปแบบกฎความสัมพันธ์ทั่วไปให้อยู่ในรูปแบบภาษาธรรมชาติ

การหากฎความสัมพันธ์แบบกระจายนั้นไม่สามารถหากฎความสัมพันธ์ได้แบบตรงไปตรงมา เมื่อเปรียบเทียบกับการหากฎความสัมพันธ์แบบดั้งเดิม ดังนั้นการรวมกฎความสัมพันธ์ที่ได้จากการหากฎความสัมพันธ์แบบกระจาย ซึ่งเป็นกระบวนการที่ไม่ขึ้นต่อกันเลย อาจทำให้ได้กฎความสัมพันธ์ที่เกิดความขัดแย้งกันเอง ตัวอย่างเช่น

1. If X is a man then X is a human.
2. If X is a John then X is a man.
3. If X is a John then X is not a human

จากตัวอย่างกฎความสัมพันธ์จะเห็นได้ว่าเกิดความขัดแย้งของกฎความสัมพันธ์ที่ 2 และ 3 ทำให้กฎความสัมพันธ์ข้างต้นนี้ไม่เหมาะสมสำหรับการนำไปทำนายผลข้อมูลในอนาคต ดังนั้นจำเป็นต้องมีเครื่องมือสำหรับการตรวจสอบความขัดแย้ง แต่ก่อนการจะนำไปสู่ในขั้นตอนนั้นจะต้องแปลงกฎความสัมพันธ์ให้อยู่ในรูปแบบที่สามารถนำไปใช้กับเครื่องมือนั้น ๆ ได้ ซึ่งในงานวิจัยนี้จะแปลงรูปแบบกฎความสัมพันธ์ทั่วไปให้อยู่ในรูปแบบของภาษาธรรมชาติ โดยใช้เทคนิคการค้นหาและแทนที่ข้อความหรือตัวอักษรที่ต้องเพื่อทำให้กฎความสัมพันธ์ในรูปแบบทั่วไป เช่น ‘{A=B} => {B=C}’ เป็นต้น ให้อยู่ในรูปแบบภาษาธรรมชาติ เช่น ‘if X is a n:A_equal_B then X is a n:B_equal_C’ เป็นต้น ซึ่งข้อความหรือตัวอักษรดังกล่าวนั้นมีดังต่อไปนี้

‘{’	แทนที่ด้วย	‘if X is a n:’
‘=>’	แทนที่ด้วย	‘then X is a’
‘=’	แทนที่ด้วย	‘_equal_’
‘,’	แทนที่ด้วย	‘and X is a n:’
‘{’	แทนที่ด้วย	‘n:’
‘}’	ตัดทิ้ง	

โดยภาษาธรรมชาติในที่นี้คือประโยคภาษาอังกฤษทั่วไป แต่จะเขียนโดยมีข้อกำหนดของภาษาเพิ่มขึ้นมา ซึ่งในงานวิจัยนี้ใช้ ACE ซึ่งเป็นภาษาธรรวมชาติชนิดหนึ่งที่สามารถแทนความรู้แล้วนำไปสร้างเป็นออนโทโลยีได้ ดังตารางที่ 3.2 แสดงตัวอย่างการแปลงรูปแบบกฎความสัมพันธ์ทั่วไปให้อยู่ในรูปแบบภาษาธรรมชาติ จากรูปที่ 3.4 แสดงคำสั่งเทียมขั้นตอนการแปลงรูปแบบกฎความสัมพันธ์ทั่วไปให้อยู่ในรูปแบบภาษาธรรมชาติ โดยจะรับข้อมูลเป็นกฎความสัมพันธ์ที่ได้จากการรวมกฎความสัมพันธ์จากแหล่งความรู้ต่าง ๆ ซึ่งการแปลงรูปแบบกฎความสัมพันธ์นี้จะใช้เทคนิคของการค้นหาและแทนที่ในการค้นหารูปแบบของประโยคหรือสัญลักณ์ที่ต้องการแทนที่ด้วยประโยคหรือสัญลักณ์ที่ต้องแทนที่ลงไป ผลลัพธ์ที่ได้คือกฎความสัมพันธ์ที่อยู่ในรูปแบบของภาษาธรรมชาติหรือประโยคภาษาทั่วไปนั้นเอง ที่สามารถนำไปแทนความรู้เพื่อนำไปสร้างเป็นออนโทโลยีได้

ตารางที่ 3.2 ตัวอย่างการแปลงรูปแบบกฎความสัมพันธ์ทั่วไปให้อยู่ในรูปแบบภาษาธรรมชาติ

Original association rules	**Association rules in ACE**
{Class=no-recurrence-events}=>{inv-nodes=0-2}	If X is a n: Class_equal_no-recurrence-events then X is a n:inv-nodes_equal_0-2.
{inv-nodes=0-2}=>{irradiat=no}	If X is a n:inv-nodes_equal_0-2 then X is a n:irradiat_equal_no.
{irradiat=no}=>{node-caps=no}	If X is a n:irradiat_equal_no then X is a n: node-caps_equal_no.

Algorithm Transform to Natural Language Format

//Input: C, an association rule set.

//Output: CACE, an association rule set in the form of natural language.

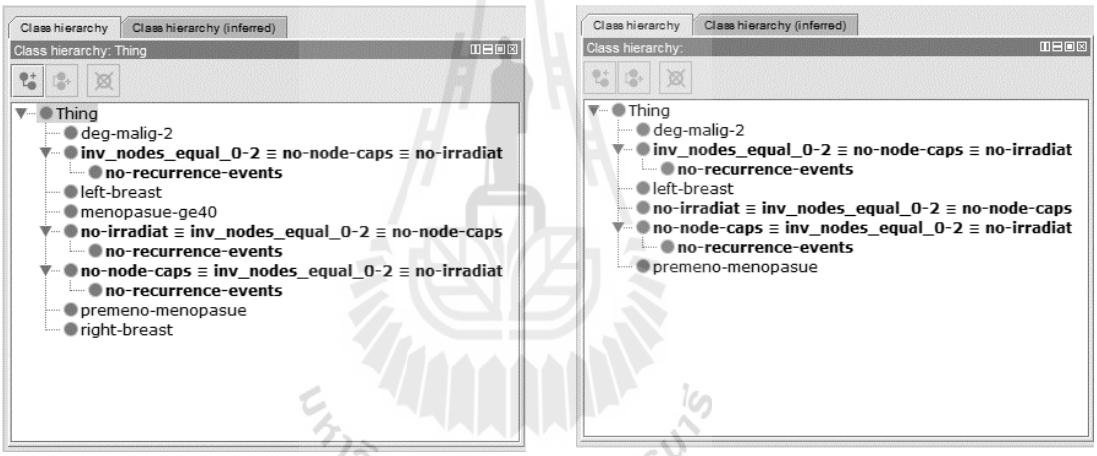
```
1. Create CACE as empty list;
2. Create D as dictionary = {       '\A{' :'if X is a n:'
3.                                  '=>' : 'then X is a',
4.                                  '=' : '_equal_',
5.                                  ',' : ' and X is a n:',
6.                                  '{' : 'n:',
7.                                  '}' : '',
8. }
9. For i =1 ← to length(C) do
10.       RN = multiple_replace(D, C_i);
11.       add RN to CACE;
12. End for
13. Return CACE;
```

รูปที่ 3.4 คำสั่งเทียมขั้นตอนการแปลงรูปแบบกฎความสัมพันธ์ทั่วไปให้อยู่ในรูปแบบภาษาธรรมชาติ

3.3.3 อัลกอริทึมตรวจสอบความขัดแย้งของกฎความสัมพันธ์ที่ได้จากแหล่งความรู้ต่าง ๆ

จากอัลกอริทึมก่อนหน้านี้จะเป็นการแปลงรูปแบบกฎความสัมพันธ์ทั่วไปให้อยู่ในรูปแบบภาษาธรรมชาติ เพื่อใช้สำหรับการนำกฎความสัมพันธ์ไปสร้างเป็นออนโทโลยีหรือฐานข้อมูลที่มีการระบุความสัมพันธ์ของข้อมูลนั้นเอง เนื่องจากรูปแบบข้อมูลที่ใช้ในเครื่องมือตรวจสอบความขัดแย้งนั้นต้องอยู่ในรูปแบบของออนโทโลยี ดังรูปที่ 3.5 และนอกเหนือจากผลลัพธ์ที่ได้ว่ากฎความสัมพันธ์เกิดความขัดแย้งกันหรือไม่ ยังสามารถสร้างความรู้ใหม่หรือกฎความสัมพันธ์ใหม่จากกฎความสัมพันธ์ที่มีอยู่เดิมได้ ดังตารางที่ 3.3 แสดงตัวอย่างความรู้ใหม่ที่ได้จากการตรวจสอบความขัดแย้งของกฎความสัมพันธ์ที่ได้จากแหล่งความรู้ต่าง ๆ

จากรูปที่ 3.6 แสดงคำสั่งเทียมขั้นตอนการตรวจสอบความขัดแย้งของกฎความสัมพันธ์ที่ได้จากแหล่งความรู้ต่าง ๆ โดยจะรับข้อมูลเป็นกฎความสัมพันธ์ที่อยู่ในรูปแบบของภาษาธรรมชาติ แล้วนำแต่ละกฎความสัมพันธ์ไปสร้างเป็นออนโทโลยีสำหรับการนำไปใช้กับเครื่องมือตรวจสอบความขัดแย้ง หลังจากได้กฎความสัมพันธ์ที่อยู่ในรูปแบบของออนโทโลยีแล้วจะนำไปตรวจสอบความขัดแย้งด้วย FaCT++ Reasoner (Tsarkov and Ian, 2006) ซึ่งถ้าเกิดความขัดแย้งจะทำการลบกฎความสัมพันธ์นั้น ๆ โดยผลลัพธ์ที่ได้คือสามารถบอกได้ว่ากฎความสัมพันธ์ที่ได้จากการรวมกฎความสัมพันธ์จากแหล่งความรู้ต่าง ๆ นั้นขัดแย้งกันหรือไม่ และกฎความสัมพันธ์ใหม่ที่ได้จากกฎความสัมพันธ์เดิมเพื่อนำไปเพิ่มเติมในส่วนของกฎความสัมพันธ์ที่ขาดหายไป สุดท้ายจะได้ฐานความรู้ของกฎความสัมพันธ์เพียงชุดเดียวที่มาจากการหากฎความสัมพันธ์แบบกระจายที่มีประสิทธิภาพใกล้เคียงกับการหากฎความแบบดั้งเดิม



รูปที่ 3.5 ตัวอย่างออนโทโลยีที่สร้างจากกฎความสัมพันธ์

ตารางที่ 3.3 ตัวอย่างความรู้ใหม่ที่ได้จากการตรวจสอบความขัดแย้งของกฎความสัมพันธ์ที่ได้จากแหล่งความรู้ต่าง ๆ

Entailment	ACE If-then
Every inv_nodes_equal_0-2 is a no-irradiat that is a no-node-caps.	If X is a n:inv_nodes_equal_0-2 and X is a n:no-irradiat then X is a n:no-node-caps .
Every no-irradiat is an inv_nodes_equal_0-2 that is a no-node-caps.	If X is a n:no-irradiat and X is a n:inv_nodes_equal_0-2 then X is a n:no-node-caps .
Every no-node-caps is an inv_nodes_equal_0-2 that is a no-irradiat.	If X is a n:no-node-caps and X is a n:inv_nodes_equal_0-2 then X is a n:no-irradiat .

Algorithm Check Inconsistency with Reasoner

```
//Input: CACE, an association rule set in the form of natural language.
//Output:  I, Inconsistency of Association Rules as true or false.
           E, Association rules entailed from reasoner.

1. I = true
2. Ontology = ACE_views (CACE);
2. While I = true{
3.        (I, E) = Reasoner (Ontology)
4.        If I = true
5.              Ontology = Remove_rules_inconsistent (CACE)
6. }
7. Return (I, E);
```

รูปที่ 3.6 คำสั่งเทียมขั้นตอนการการตรวจสอบความขัดแย้งของกฎความสัมพันธ์ที่ได้จากแหล่งความรู้ต่าง ๆ

บทที่ 4
การทดสอบและอภิปรายผล

การทดสอบประสิทธิภาพของระบบนั้น จะทดสอบประสิทธิภาพกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งเปรียบเทียบกับการหากฎความสัมพันธ์แบบดั้งเดิมในกรณีที่ค่าสนับสนุนที่แตกต่างกัน โดยจะพิจารณาจากจำนวนกฎความสัมพันธ์ โดยมีรายละเอียดของข้อมูลที่ใช้ในการทดสอบ การทดสอบประสิทธิภาพกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง การเปรียบเทียบผลการทดสอบกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งด้วยค่าสนับสนุนที่แตกต่างกัน ผลการทดสอบการหากฎความสัมพันธ์แบบดั้งเดิม และอภิปรายผล

4.1 ข้อมูลที่ใช้ในการทดสอบ

การทดสอบกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งจะใช้ข้อมูลมาตรฐานจาก UCI Machine Learning Repository ซึ่งเป็นข้อมูลเกี่ยวกับผู้รอดชีวิตจากเรือไททานิค (Titanic) มีข้อมูลทั้งหมด 2201 แถว ประกอบไปด้วย 4 คอลัมน์ ข้อมูลผู้ป่วยโรคมะเร็งเต้านม (Breast Cancer) มีข้อมูลทั้งหมด 286 แถว ประกอบไปด้วย 9 คอลัมน์ และข้อมูลผู้ป่วยโรคหัวใจ (Heart Disease) มีข้อมูลทั้งหมด 303 แถว ประกอบไปด้วย 14 คอลัมน์ สามารถดาวน์โหลดได้ที่ http://repository.seasr.org/Datasets/UCI/arff/ โดยมีรายละเอียดตัวอย่างข้อมูลดังตารางที่ 4.1 4.2 และ 4.3

ตารางที่ 4.1 ตัวอย่างข้อมูลผู้รอดชีวิตจากเรือไททานิค

CLASS	AGE	SEX	SURVIVED
first	adult	male	yes
first	adult	male	yes
first	adult	male	yes
first	adult	female	no
first	adult	female	no
first	child	male	yes

ตารางที่ 4.2 ตัวอย่างข้อมูลผู้ป่วยโรคมะเร็งเต้านม

age	menopause	tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad	irradiat	Class
40-49	premeno	15-19	0-2	yes	3	right	left_up	no	recurrence-events
50-59	ge40	15-19	0-2	no	1	right	central	no	no-recurrence-events
50-59	ge40	35-39	0-2	no	2	left	left_low	no	recurrence-events
50-59	ge40	15-19	0-2	no	1	right	central	no	no-recurrence-events
40-49	premeno	15-19	0-2	yes	3	right	left_up	no	recurrence-events

ตารางที่ 4.3 ตัวอย่างข้อมูลผู้ป่วยโรคหัวใจ

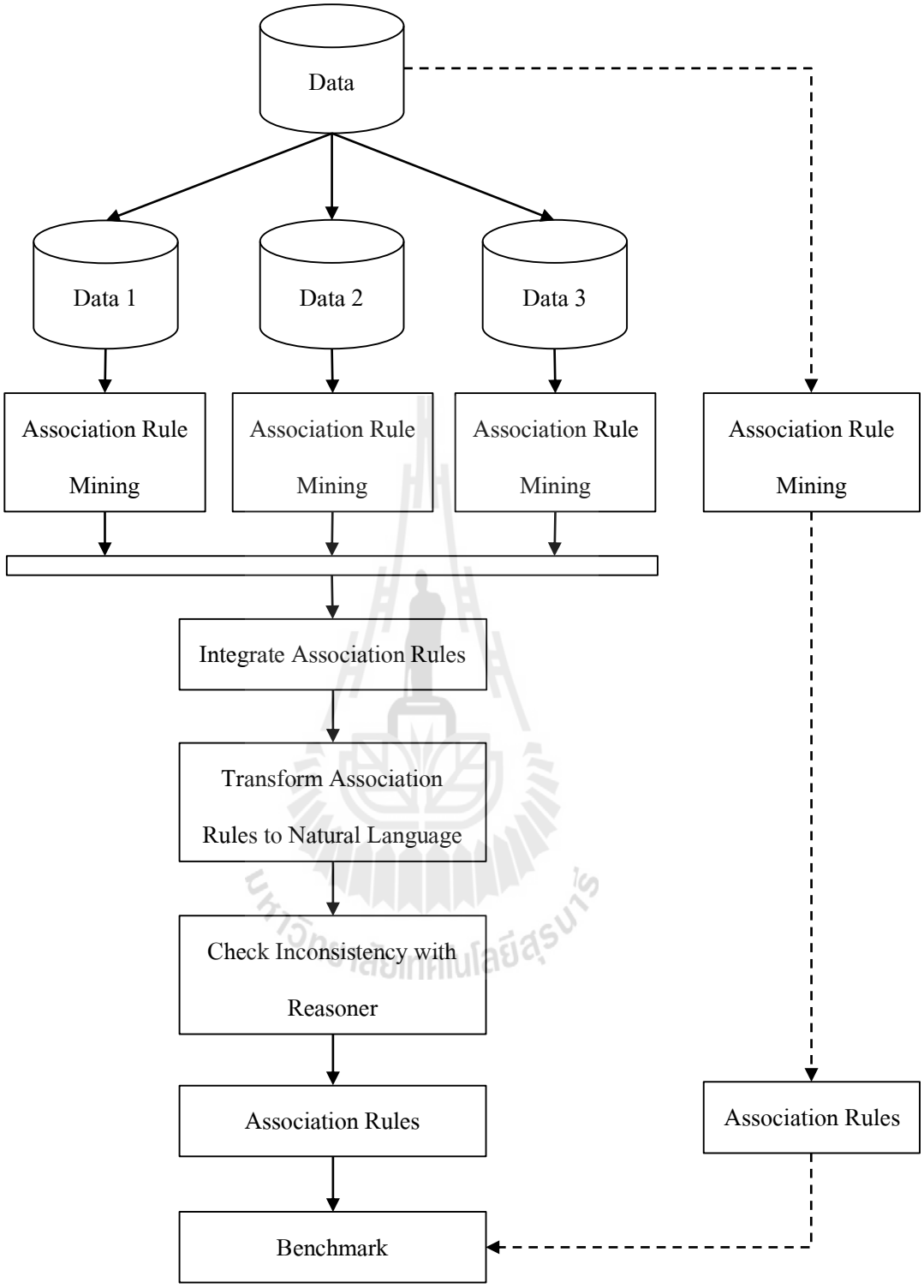
age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	num
61-77	male	typ_angina	129-164	126-272	t	left_vent_hyper	115-158	no	2	down	0	fixed_defect	<50
61-77	male	asympt	129-164	273-418	f	left_vent_hyper	71-114	yes	1	flat	3	normal	>50_1
61-77	male	asympt	94-129	126-272	f	left_vent_hyper	115-158	yes	2	flat	2	reversable_defect	>50_1
29-45	male	non_anginal	129-164	126-272	f	normal	159-202	no	3	down	0	normal	<50
61-77	male	asympt	129-164	273-418	f	left_vent_hyper	71-114	yes	1	flat	3	normal	>50_1

4.2 การทดสอบประสิทธิภาพกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง

การทดสอบประสิทธิภาพกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งจะเป็นการจำลองการหากฎความสัมพันธ์แบบกระจาย ซึ่งข้อมูลไม่ได้ถูกจัดเก็บไว้ในแหล่งข้อมูลเพียงแหล่งเดียวเพื่อใช้ในการหากฎความสัมพันธ์ แต่ข้อมูลจะถูกแบ่งออกเป็นชุด ๆ เพื่อนำไปหากฎความสัมพันธ์โดยที่กระบวนการของการทำงานจะไม่ขึ้นต่อกัน และใช้จำนวนกฎความสัมพันธ์ที่เหมือนกันกับการหากฎความสัมพันธ์แบบดั้งเดิมเป็นตัวชี้วัดประสิทธิภาพ

จากรูปที่ 4.1 แสดงแผนภาพวิธีการทดสอบประสิทธิภาพกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง โดยจะแบ่งข้อมูลออกเป็น 3 ชุด คือ Data1 Data2 และ Data 3 แล้วนำข้อมูลทั้ง 3 ชุดไปหากฎความสัมพันธ์ ซึ่งในส่วนนี้การหากฎความสัมพันธ์ของข้อมูลแต่ละชุดจะไม่ขึ้นต่อกัน ซึ่งผลลัพธ์ที่ได้คือชุดกฎความสัมพันธ์ของข้อมูลแต่ละชุด หลังจากนั้นจะดึงกฎความสัมพันธ์ที่ได้จากข้อมูลแต่ละชุดมารวมให้เป็นกฎความสัมพันธ์เพียงชุดเดียวด้วยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง ผลลัพธ์สุดท้ายที่ได้คือชุดกฎความสัมพันธ์เพียงชุดเดียวที่ได้จากการรวมกฎความสัมพันธ์จากหลายแหล่ง และการวัดประสิทธิภาพของกฎความสัมพันธ์ที่ได้นั้นจะเปรียบเทียบจากกฎความสัมพันธ์ที่ได้จากการหากฎความสัมพันธ์ด้วยข้อมูลดั้งเดิมที่ไม่ได้ถูกแบ่งออกเป็นชุด ๆ ซึ่งจะพิจารณาจากจำนวนกฎความสัมพันธ์ที่เหมือนกันระหว่างกฎความสัมพันธ์ที่ได้จากกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งและการหากฎความสัมพันธ์แบบดั้งเดิม

การทดสอบประสิทธิภาพกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งจะเป็นการจำลองการหากฎความสัมพันธ์จากหลายแหล่งนั้น นอกจากจะใช้จำนวนกฎความสัมพันธ์ที่เหมือนกันกับการหากฎความสัมพันธ์แบบดั้งเดิมเป็นตัวชี้วัดประสิทธิภาพแล้วนั้น ในขั้นตอนของการหากฎความสัมพันธ์นั้นจะเลือกใช้ค่าสนับสนุนที่แตกต่างกันออกไป เพื่อต้องการทดสอบว่าเมื่อใช้ค่าสนับสนุนที่แตกต่างกันออกไปจะได้จำนวนกฎความสัมพันธ์ที่เหมือนกันกับการหากฎความสัมพันธ์แบบดั้งเดิมเป็นอย่างไร โดยค่าสนับสนุนที่เลือกใช้ในการทดสอบได้แก่ ค่าสนับสนุนที่ 0.1 0.2 0.3 0.4 0.5 และ 0.6 เหตุผลที่เลือกใช้ค่าสนับสนุนถึงเพียงแค่ 0.6 เนื่องจากข้อมูลที่นำมาทดสอบเมื่อนำไปหากฎความสัมพันธ์โดยใช้ค่าสนับสนุนตั้งแต่ 0.7 ขึ้นไปจะไม่ปรากฏกฎความสัมพันธ์ ซึ่งผลการทดสอบของงานวิจัยนี้มีรายละเอียดดังต่อไปนี้



รูปที่ 4.1 วิธีการทดสอบประสิทธิภาพกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง

4.2.1 ผลการทดสอบกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งด้วยค่าสนับสนุนที่ 0.1

ผลทดสอบการหากฎความสัมพันธ์จากข้อมูลทั้งหมด 3 ชุด ด้วยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งด้วยค่าสนับสนุนที่ 0.1 ซึ่งผลลัพธ์ที่ได้คือชุดกฎความสัมพันธ์เพียงหนึ่งเดียวที่ได้จากการรวมกฎความสัมพันธ์หลายแหล่ง ผลการทดลองที่ได้คือ จำนวนกฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิมและจำนวนกฎความสัมพันธ์จากการหากฎความสัมพันธ์ด้วยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง ดังตารางที่ 4.4 แสดงผลการเปรียบเทียบจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์แบบดั้งเดิมและการหากฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยค่าสนับสนุนที่ 0.1 ซึ่งผลการทดสอบที่ได้คือ ข้อมูลผู้รอดชีวิตเรือไททานิคได้กฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิม จำนวน 26 กฎ และกฎความสัมพันธ์จากการหากฎความสัมพันธ์จากข้อมูลหลายแหล่ง จำนวน 25 กฎ ข้อมูลผู้ป่วยโรคมะเร็งเต้านมได้กฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิม จำนวน 883 กฎ และกฎความสัมพันธ์จากการหากฎความสัมพันธ์จากข้อมูลหลายแหล่ง จำนวน 380 กฎ และข้อมูลผู้ป่วยโรคหัวใจได้กฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิม จำนวน 7,245 กฎ และกฎความสัมพันธ์จากการหากฎความสัมพันธ์จากข้อมูลหลายแหล่ง จำนวน 2,160 กฎ

ตารางที่ 4.5 แสดงกฎความสัมพันธ์ใหม่ที่ได้จากกฎความสัมพันธ์เดิมด้วยค่าสนับสนุนที่ 0.1 ซึ่งเป็นกฎความสัมพันธ์ใหม่ที่ได้จากกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง โดยสามารถนำไปเพิ่มเติมในส่วนของกฎความสัมพันธ์ที่ขาดหายไปได้ ผลการทดสอบที่ได้คือข้อมูลผู้รอดชีวิตเรือไททานิคไม่ปรากฏกฎความสัมพันธ์ใหม่ ข้อมูลผู้ป่วยโรคมะเร็งเต้านมได้กฎความสัมพันธ์ใหม่ จำนวน 8 กฎ และข้อมูลผู้ป่วยโรคหัวใจได้กฎความสัมพันธ์ใหม่ จำนวน 2 กฎ

ตารางที่ 4.4 เปรียบเทียบจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์แบบดั้งเดิมและการหากฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยค่าสนับสนุนที่ 0.1

Results / Data sets	# Original association rules	# Association rules from multiple sources without reasoner	# Association Rules from Reasoner
Titanic	26	25	25
Breast Cancer	883	372	380
Heart Disease	7,245	2,158	2,160

ตารางที่ 4.5 กฎความสัมพันธ์ใหม่ที่ได้จากกฎความสัมพันธ์เดิมด้วยค่าสนับสนุนที่ 0.1

Result / Datasets	Entailment	If-Then Rules
Titanic	-	-
Breast Cancer	Every age_equal_50-59 is an inv-nodes_equal_0-2 .	If age=50-59 Then inv-nodes=0-2
	Every inv-nodes_equal_0-2 is an irradiat_equal_no that is a node-caps_equal_no .	If inv-nodes=0-2 and irradiat=no Then node-caps=no
	Every irradiat_equal_no is an inv-nodes_equal_0-2 that is a node-caps_equal_no .	If irradiat=no and inv-nodes=0-2 Then node-caps=no
	Every node-caps_equal_no is an inv-nodes_equal_0-2 that is an irradiat_equal_no .	If node-caps=no and inv-nodes=0-2 Then irradiat=no
	Every deg-malig_equal_1 is a Class_equal_no-recurrence-events .	If deg-malig=1 Then Class=no-recurrence-events
	Every Class_equal_no-recurrence-events is an inv-nodes_equal_0-2 .	If Class=no-recurrence-events Then inv-nodes=0-2
	Every age_equal_50-59 is a node-caps_equal_no .	If age=50-59 Then node-caps=no
	Every Class_equal_no-recurrence-events is an irradiat_equal_no .	If Class=no-recurrence-events Then irradiat=no
Heart Disease	Every sex_equal_female is a thal_equal_normal.	If sex=female Then thal=normal
	Every cp_equal_atyp_angina is a fbs_equal_f .	If cp=atyp_angina Then fbs=f

4.2.2 ผลการทดสอบกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งด้วยค่าสนับสนุนที่ 0.2

ผลทดสอบการหากฎความสัมพันธ์จากข้อมูลทั้งหมด 3 ชุด ด้วยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งด้วยค่าสนับสนุนที่ 0.2 ซึ่งผลลัพธ์ที่ได้คือชุดกฎความสัมพันธ์เพียงหนึ่งเดียวที่ได้จากการรวมกฎความสัมพันธ์หลายแหล่ง ผลการทดลองที่ได้คือจำนวนกฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิมและจำนวนกฎความสัมพันธ์จากการหากฎความสัมพันธ์ด้วยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง ดังตารางที่ 4.6 แสดงผลการเปรียบเทียบจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์แบบดั้งเดิมและการหากฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยค่าสนับสนุนที่ 0.2 ซึ่งผลการทดสอบที่ได้คือ

ข้อมูลผู้รอดชีวิตเรือไททานิคได้กฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิม จำนวน 17 กฎ และกฎความสัมพันธ์จากการหากฎความสัมพันธ์จากข้อมูลหลายแหล่ง จำนวน 17 กฎ ข้อมูลผู้ป่วยโรคมะเร็งเต้านมได้กฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิม จำนวน 203 กฎ และกฎความสัมพันธ์จากการหากฎความสัมพันธ์จากข้อมูลหลายแหล่ง จำนวน 126 กฎ และข้อมูลผู้ป่วยโรคหัวใจได้กฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิม จำนวน 726 กฎ และกฎความสัมพันธ์จากการหากฎความสัมพันธ์จากข้อมูลหลายแหล่ง จำนวน 283 กฎ

ตารางที่ 4.7 แสดงกฎความสัมพันธ์ใหม่ที่ได้จากกฎความสัมพันธ์เดิมด้วยค่าสนับสนุนที่ 0.3 ซึ่งเป็นกฎความสัมพันธ์ใหม่ที่ได้จากกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง โดยสามารถนำไปเพิ่มเติมในส่วนของกฎความสัมพันธ์ที่ขาดหายไปได้ ผลการทดสอบที่ได้คือข้อมูลผู้รอดชีวิตเรือไททานิคไม่ปรากฏกฎความสัมพันธ์ใหม่ ข้อมูลผู้ป่วยโรคมะเร็งเต้านมได้กฎความสัมพันธ์ใหม่ จำนวน 7 กฎ และข้อมูลผู้ป่วยโรคหัวใจได้กฎความสัมพันธ์ใหม่ จำนวน 1 กฎ

ตารางที่ 4.6 เปรียบเทียบจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์แบบดั้งเดิมและการหากฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยค่าสนับสนุนที่ 0.2

Results / Data sets	# Original association rules	# Association rules from multiple sources without reasoner	# Association Rules from Reasoner
Titanic	17	17	17
Breast Cancer	203	119	126
Heart Disease	726	282	283

ตารางที่ 4.7 กฎความสัมพันธ์ใหม่ที่ได้จากกฎความสัมพันธ์เดิมด้วยค่าสนับสนุนที่ 0.2

Result / Datasets	Entailment	If-Then Rules
Titanic	-	-
Breast Cancer	Every age_equal_50-59 is an inv-nodes_equal_0-2 .	If age=50-59 Then inv-nodes=0-2
	Every inv-nodes_equal_0-2 is an irradiat_equal_no that is a node-caps_equal_no .	If inv-nodes=0-2 and irradiat=no Then node-caps=no
	Every irradiat_equal_no is an inv-nodes_equal_0-2 that is a node-caps_equal_no .	If irradiat=no and inv-nodes=0-2 Then node-caps=no
	Every node-caps_equal_no is an inv-nodes_equal_0-2 that is an irradiat_equal_no .	If node-caps=no and inv-nodes=0-2 Then irradiat=no
	Every Class_equal_no-recurrence-events is an inv-nodes_equal_0-2 .	If Class=no-recurrence-events Then inv-nodes=0-2
	Every age_equal_50-59 is a node-caps_equal_no .	If age=50-59 Then node-caps=no
	Every Class_equal_no-recurrence-events is an irradiat_equal_no .	If Class=no-recurrence-events Then irradiat=no
Heart Disease	Every num_equal_less_than_50 is a thal_equal_normal .	If num=< 50 Then thal=normal

4.2.3 ผลการทดสอบกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งด้วยค่าสนับสนุนที่ 0.3

ผลทดสอบการหากฎความสัมพันธ์จากข้อมูลทั้งหมด 3 ชุด ด้วยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งด้วยค่าสนับสนุนที่ 0.3 ซึ่งผลลัพธ์ที่ได้คือชุดกฎความสัมพันธ์เพียงหนึ่งเดียวที่ได้จากการรวมกฎความสัมพันธ์หลายแหล่ง ผลการทดลองที่ได้คือจำนวนกฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิมและจำนวนกฎความสัมพันธ์จากการหากฎความสัมพันธ์ด้วยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง ดังตารางที่ 4.8 แสดงผลการเปรียบเทียบจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์แบบดั้งเดิมและการหากฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยค่าสนับสนุนที่ 0.3 ซึ่งผลการทดสอบที่ได้คือข้อมูลผู้รอดชีวิตเรือไททานิคได้กฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิม จำนวน 13 กฎ และกฎความสัมพันธ์จากการหากฎความสัมพันธ์จากข้อมูลหลายแหล่ง จำนวน 19 กฎ ข้อมูล

ผู้ป่วยโรคมะเร็งเต้านมได้กฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิม จำนวน 62 กฎ และกฎความสัมพันธ์จากการหากฎความสัมพันธ์จากข้อมูลหลายแหล่ง จำนวน 35 กฎ และข้อมูลผู้ป่วยโรคหัวใจได้กฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิม จำนวน 118 กฎ และกฎความสัมพันธ์จากการหากฎความสัมพันธ์จากข้อมูลหลายแหล่ง จำนวน 42 กฎ

ตารางที่ 4.9 แสดงกฎความสัมพันธ์ใหม่ที่ได้จากกฎความสัมพันธ์เดิมด้วยค่าสนับสนุนที่ 0.3 ซึ่งเป็นกฎความสัมพันธ์ใหม่ที่ได้จากกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง โดยสามารถนำไปเพิ่มเติมในส่วนของกฎความสัมพันธ์ที่ขาดหายไปได้ ผลการทดสอบที่ได้คือข้อมูลผู้รอดชีวิตเรือไททานิคไม่ปรากฏกฎความสัมพันธ์ใหม่ ข้อมูลผู้ป่วยโรคมะเร็งเต้านมได้กฎความสัมพันธ์ใหม่ จำนวน 5 กฎ และข้อมูลผู้ป่วยโรคหัวใจได้กฎความสัมพันธ์ใหม่ไม่ปรากฏกฎความสัมพันธ์ใหม่

ตารางที่ 4.8 เปรียบเทียบจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์แบบดั้งเดิมและการหากฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยค่าสนับสนุนที่ 0.3

Results / Data sets	# Original association rules	# Association rules from multiple sources without reasoner	# Association Rules from Reasoner
Titanic	13	9	9
Breast Cancer	62	30	35
Heart Disease	118	42	42

ตารางที่ 4.9 กฎความสัมพันธ์ใหม่ที่ได้จากกฎความสัมพันธ์เดิมด้วยค่าสนับสนุนที่ 0.3

Result / Datasets	Entailment	If-Then Rules
Titanic	-	-
Breast Cancer	Every inv-nodes_equal_0-2 is an irradiat_equal_no that is a node-caps_equal_no .	If inv-nodes=0-2 and irradiat=no Then node-caps=no
	Every irradiat_equal_no is an inv-nodes_equal_0-2 that is a node-caps_equal_no .	If irradiat=no and inv-nodes=0-2 Then node-caps=no
	Every node-caps_equal_no is an inv-nodes_equal_0-2 that is an irradiat_equal_no .	If node-caps=no and inv-nodes=0-2 Then irradiat=no
	Every Class_equal_no-recurrence-events is an inv-nodes_equal_0-2 .	If Class=no-recurrence-events Then inv-nodes=0-2
	Every Class_equal_no-recurrence-events is an irradiat_equal_no .	If Class=no-recurrence-events Then irradiat=no
Heart Disease	-	-

4.2.4 ผลการทดสอบกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งด้วยค่าสนับสนุนที่ 0.4

ผลทดสอบการหากฎความสัมพันธ์จากข้อมูลทั้งหมด 3 ชุด ด้วยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งด้วยค่าสนับสนุนที่ 0.4 ซึ่งผลลัพธ์ที่ได้คือชุดกฎความสัมพันธ์เพียงหนึ่งเดียวที่ได้จากการรวมกฎความสัมพันธ์หลายแหล่ง ผลการทดลองที่ได้คือจำนวนกฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิมและจำนวนกฎความสัมพันธ์จากการหากฎความสัมพันธ์ด้วยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง ดังตารางที่ 4.10 แสดงผลการเปรียบเทียบจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์แบบดั้งเดิมและการหากฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยค่าสนับสนุนที่ 0.4 ซึ่งผลการทดสอบที่ได้คือข้อมูลผู้รอดชีวิตเรือไททานิคได้กฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิม จำนวน 6 กฎ และกฎความสัมพันธ์จากการหากฎความสัมพันธ์จากข้อมูลหลายแหล่ง จำนวน 5 กฎ ข้อมูลผู้ป่วยโรคมะเร็งเต้านมได้กฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิม จำนวน 24 กฎ และกฎความสัมพันธ์จากการหากฎความสัมพันธ์จากข้อมูลหลายแหล่ง จำนวน 24 กฎ และข้อมูลผู้ป่วยโรคหัวใจได้กฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิม จำนวน 18 กฎ และกฎความสัมพันธ์จากการหากฎความสัมพันธ์จากข้อมูลหลายแหล่ง จำนวน 10 กฎ

ตารางที่ 4.11 แสดงกฎความสัมพันธ์ใหม่ที่ได้จากกฎความสัมพันธ์เดิมด้วยค่าสนับสนุนที่ 0.4 ซึ่งเป็นกฎความสัมพันธ์ใหม่ที่ได้จากกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง โดยสามารถนำไปเพิ่มเติมในส่วนของกฎความสัมพันธ์ที่ขาดหายไปได้ ผลการทดสอบที่ได้คือข้อมูลผู้รอดชีวิตเรือไททานิคไม่ปรากฏกฎความสัมพันธ์ใหม่ ข้อมูลผู้ป่วยโรคมะเร็งเต้านมได้กฎความสัมพันธ์ใหม่ จำนวน 5 กฎ และข้อมูลผู้ป่วยโรคหัวใจได้กฎความสัมพันธ์ใหม่ไม่ปรากฏกฎความสัมพันธ์ใหม่

ตารางที่ 4.10 เปรียบเทียบจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์แบบดั้งเดิมและการหากฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยค่าสนับสนุนที่ 0.4

Results / Data sets	# Original association rules	# Association rules from multiple sources without reasoner	# Association Rules from Reasoner
Titanic	6	5	5
Breast Cancer	24	19	24
Heart Disease	18	10	10

ตารางที่ 4.11 กฎความสัมพันธ์ใหม่ที่ได้จากกฎความสัมพันธ์เดิมด้วยค่าสนับสนุนที่ 0.4

Result / Datasets	Entailment	If-Then Rules
Titanic	-	-
Breast Cancer	Every inv-nodes_equal_0-2 is an irradiat_equal_no that is a node-caps_equal_no .	If inv-nodes=0-2 and irradiat=no Then node-caps=no
	Every irradiat_equal_no is an inv-nodes_equal_0-2 that is a node-caps_equal_no .	If irradiat=no and inv-nodes=0-2 Then node-caps=no
	Every node-caps_equal_no is an inv-nodes_equal_0-2 that is an irradiat_equal_no .	If node-caps=no and inv-nodes=0-2 Then irradiat=no
	Every Class_equal_no-recurrence-events is an inv-nodes_equal_0-2 .	If Class=no-recurrence-events Then inv-nodes=0-2
	Every Class_equal_no-recurrence-events is an irradiat_equal_no .	If Class=no-recurrence-events Then irradiat=no
Heart Disease	-	-

4.2.5 ผลการทดสอบกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งด้วยค่าสนับสนุนที่ 0.5

ผลทดสอบการหากฎความสัมพันธ์จากข้อมูลทั้งหมด 3 ชุด ด้วยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งด้วยค่าสนับสนุนที่ 0.5 ซึ่งผลลัพธ์ที่ได้คือชุดกฎความสัมพันธ์เพียงหนึ่งเดียวที่ได้จากการรวมกฎความสัมพันธ์หลายแหล่ง ผลการทดลองที่ได้คือจำนวนกฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิมและจำนวนกฎความสัมพันธ์จากการหากฎความสัมพันธ์ด้วยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง ดังตารางที่ 4.12 แสดงผลการเปรียบเทียบจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์แบบดั้งเดิมและการหากฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยค่าสนับสนุนที่ 0.5 ซึ่งผลการทดสอบที่ได้คือข้อมูลผู้รอดชีวิตเรือไททานิคได้กฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิม จำนวน 5 กฎ และกฎความสัมพันธ์จากการหากฎความสัมพันธ์จากข้อมูลหลายแหล่ง จำนวน 5 กฎ ข้อมูลผู้ป่วยโรคมะเร็งเต้านมได้กฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิม จำนวน 24 กฎ และกฎความสัมพันธ์จากการหากฎความสัมพันธ์จากข้อมูลหลายแหล่ง จำนวน 15 กฎ และข้อมูลผู้ป่วยโรคหัวใจได้กฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิม จำนวน 4 กฎ และกฎความสัมพันธ์จากการหากฎความสัมพันธ์จากข้อมูลหลายแหล่ง จำนวน 4 กฎ

ตารางที่ 4.13 แสดงกฎความสัมพันธ์ใหม่ที่ได้จากกฎความสัมพันธ์เดิมด้วยค่าสนับสนุนที่ 0.5 ซึ่งเป็นกฎความสัมพันธ์ใหม่ที่ได้จากกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง โดยสามารถนำไปเพิ่มเติมในส่วนของกฎความสัมพันธ์ที่ขาดหายไปได้ ผลการทดสอบที่ได้คือข้อมูลผู้รอดชีวิตเรือไททานิคไม่ปรากฏกฎความสัมพันธ์ใหม่ ข้อมูลผู้ป่วยโรคมะเร็งเต้านมได้กฎความสัมพันธ์ใหม่ จำนวน 5 กฎ และข้อมูลผู้ป่วยโรคหัวใจได้กฎความสัมพันธ์ใหม่ไม่ปรากฏกฎความสัมพันธ์ใหม่

ตารางที่ 4.12 เปรียบเทียบจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์แบบดั้งเดิมและการหากฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยค่าสนับสนุนที่ 0.5

Results / Data sets	# Original association rules	# Association rules from multiple sources without reasoner	# Association Rules from Reasoner
Titanic	5	5	5
Breast Cancer	24	10	15
Heart Disease	4	4	4

ตารางที่ 4.13 กฎความสัมพันธ์ใหม่ที่ได้จากกฎความสัมพันธ์เดิมด้วยค่าสนับสนุนที่ 0.5

Result / Datasets	Entailment	If-Then Rules
Titanic	-	-
Breast Cancer	Every inv-nodes_equal_0-2 is an irradiat_equal_no that is a node-caps_equal_no .	If inv-nodes=0-2 and irradiat=no Then node-caps=no
	Every irradiat_equal_no is an inv-nodes_equal_0-2 that is a node-caps_equal_no .	If irradiat=no and inv-nodes=0-2 Then node-caps=no
	Every node-caps_equal_no is an inv-nodes_equal_0-2 that is an irradiat_equal_no .	If node-caps=no and inv-nodes=0-2 Then irradiat=no
	Every Class_equal_no-recurrence-events is an inv-nodes_equal_0-2 .	If Class=no-recurrence-events Then inv-nodes=0-2
	Every Class_equal_no-recurrence-events is an irradiat_equal_no .	If Class=no-recurrence-events Then irradiat=no
Heart Disease	-	-

4.2.6 ผลการทดสอบกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งด้วยค่าสนับสนุนที่ 0.6

ผลทดสอบการหากฎความสัมพันธ์จากข้อมูลทั้งหมด 3 ชุด ด้วยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งด้วยค่าสนับสนุนที่ 0.6 ซึ่งผลลัพธ์ที่ได้คือชุดกฎความสัมพันธ์เพียงหนึ่งเดียวที่ได้จากการรวมกฎความสัมพันธ์หลายแหล่ง ผลการทดลองที่ได้คือจำนวนกฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิมและจำนวนกฎความสัมพันธ์จากการหากฎความสัมพันธ์ด้วยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง ดังตารางที่ 4.14 แสดงผลการเปรียบเทียบจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์แบบดั้งเดิมและการหากฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยค่าสนับสนุนที่ 0.6 ซึ่งผลการทดสอบที่ได้คือ ข้อมูลผู้รอดชีวิตเรือไททานิคได้กฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิม จำนวน 5 กฎ และกฎความสัมพันธ์จากการหากฎความสัมพันธ์จากข้อมูลหลายแหล่ง จำนวน 3 กฎ ข้อมูลผู้ป่วยโรคมะเร็งเต้านมได้กฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิม จำนวน 9 กฎ และกฎความสัมพันธ์จากการหากฎความสัมพันธ์จากข้อมูลหลายแหล่ง จำนวน 9 กฎ และข้อมูลผู้ป่วยโรคหัวใจได้กฎความสัมพันธ์จากการหากฎความสัมพันธ์แบบดั้งเดิม จำนวน 1 กฎ และกฎความสัมพันธ์จากการหากฎความสัมพันธ์จากข้อมูลหลายแหล่ง จำนวน 0 กฎ

ตารางที่ 4.15 แสดงกฎความสัมพันธ์ใหม่ที่ได้จากกฎความสัมพันธ์เดิมด้วยค่าสนับสนุนที่ 0.6 ซึ่งเป็นกฎความสัมพันธ์ใหม่ที่ได้จากกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง โดยสามารถนำไปเพิ่มเติมในส่วนของกฎความสัมพันธ์ที่ขาดหายไปได้ ผลการทดสอบที่ได้คือข้อมูลผู้รอดชีวิตเรือไททานิคไม่ปรากฏกฎความสัมพันธ์ใหม่ ข้อมูลผู้ป่วยโรคมะเร็งเต้านมได้กฎความสัมพันธ์ใหม่ จำนวน 3 กฎ และข้อมูลผู้ป่วยโรคหัวใจได้กฎความสัมพันธ์ใหม่ไม่ปรากฏกฎความสัมพันธ์ใหม่

ตารางที่ 4.14 เปรียบเทียบจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์แบบดั้งเดิมและการหากฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยค่าสนับสนุนที่ 0.6

Results / Data sets	# Original association rules	# Association rules from multiple sources without reasoner	# Association Rules from Reasoner
Titanic	5	3	3
Breast Cancer	9	6	9
Heart Disease	1	0	0

ตารางที่ 4.15 กฎความสัมพันธ์ใหม่ที่ได้จากกฎความสัมพันธ์เดิมด้วยค่าสนับสนุนที่ 0.6

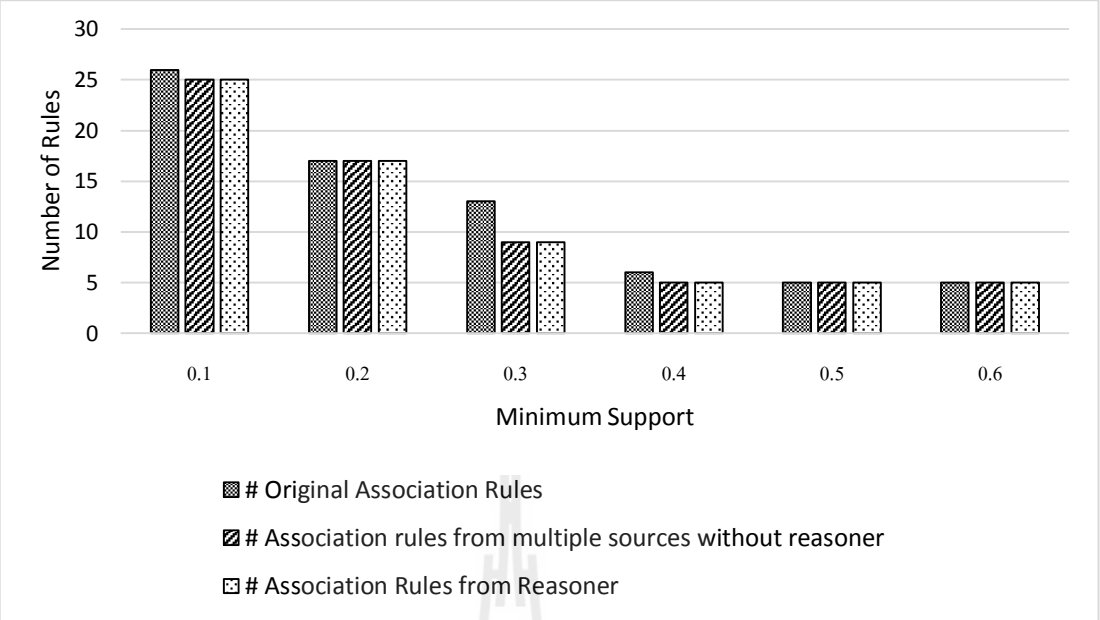
Result / Datasets	Entailment	If-Then Rules
Titanic	-	-
Breast Cancer	Every inv-nodes_equal_0-2 is an irradiat_equal_no that is a node-caps_equal_no .	If inv-nodes=0-2 and irradiat=no Then node-caps=no
	Every irradiat_equal_no is an inv-nodes_equal_0-2 that is a node-caps_equal_no .	If irradiat=no and inv-nodes=0-2 Then node-caps=no
	Every node-caps_equal_no is an inv-nodes_equal_0-2 that is an irradiat_equal_no .	If node-caps=no and inv-nodes=0-2 Then irradiat=no
Heart Disease	-	-

4.3 เปรียบเทียบผลการทดสอบกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งด้วยค่าสนับสนุนที่แตกต่างกัน

การหากฎความสัมพันธ์นั้นจะเลือกใช้ค่าสนับสนุนที่แตกต่างกันออกไป เพื่อต้องการทดสอบว่าเมื่อใช้ค่าสนับสนุนที่แตกต่างกันออกไปจะได้จำนวนกฎความสัมพันธ์ที่เหมือนกันกับการหากฎความสัมพันธ์แบบดั้งเดิมมากหรือน้อย ซึ่งผลการทดสอบจากการหากฎความสัมพันธ์จากข้อมูล 3 ชุด ได้แก่ ข้อมูลผู้รอดชีวิตเรือไททานิค ข้อมูลผู้ป่วยโรคมะเร็งเต้านม และข้อมูลผู้ป่วยโรคหัวใจด้วยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งด้วยค่าสนับสนุนที่แตกต่างกัน ซึ่งจะเห็นได้ว่าผลการทดสอบการหากฎความสัมพันธ์จากข้อมูลผู้รอดชีวิตเรือไททานิคให้จำนวนของกฎความสัมพันธ์ที่ใกล้เคียงกับการหากฎความสัมพันธ์แบบดั้งเดิม แม้จะไม่สามารถอนุมานกฎความสัมพันธ์ใหม่ได้ ดังตารางที่ 4.16 และดังรูปที่ 4.2 ผลการทดสอบการหากฎความสัมพันธ์จากข้อมูลผู้ป่วยโรคมะเร็งเต้านมจะสังเกตได้ว่าเปอร์เซ็นต์จำนวนกฎความสัมพันธ์ที่ถูกปรับปรุงด้วยการอนุมานความรู้ใหม่นั้น สามารถช่วยเพิ่มประสิทธิภาพให้กับการหากฎความสัมพันธ์จากข้อมูลหลายแหล่งได้อย่างชัดเจนเมื่อมีค่าสนับสนุนที่มาก ดังตารางที่ 4.17 และดังรูปที่ 4.3 ผลการทดสอบการหากฎความสัมพันธ์จากข้อมูลผู้ป่วยโรคหัวใจจะสังเกตได้ว่าเปอร์เซ็นต์จำนวนกฎความสัมพันธ์ที่ถูกปรับปรุงด้วยการอนุมานความรู้ใหม่ที่ค่าสนับสนุนที่น้อยนั้น สามารถช่วยเพิ่มประสิทธิภาพได้ในระดับหนึ่ง แต่เมื่อค่าสนับสนุนที่มากจำนวนของกฎความสัมพันธ์นั้นมีจำนวนใกล้เคียงกับการหากฎความสัมพันธ์แบบดั้งเดิม ทำให้เปอร์เซ็นต์จำนวนกฎความสัมพันธ์ที่ถูกปรับปรุงนั้นมีค่าเป็น 0.00 % ดังตารางที่ 4.18 และดังรูปที่ 4.4

ตารางที่ 4.16 เปรียบเทียบจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์แบบดั้งเดิมและการหากฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยข้อมูลผู้รอดชีวิตเรือไททานิค

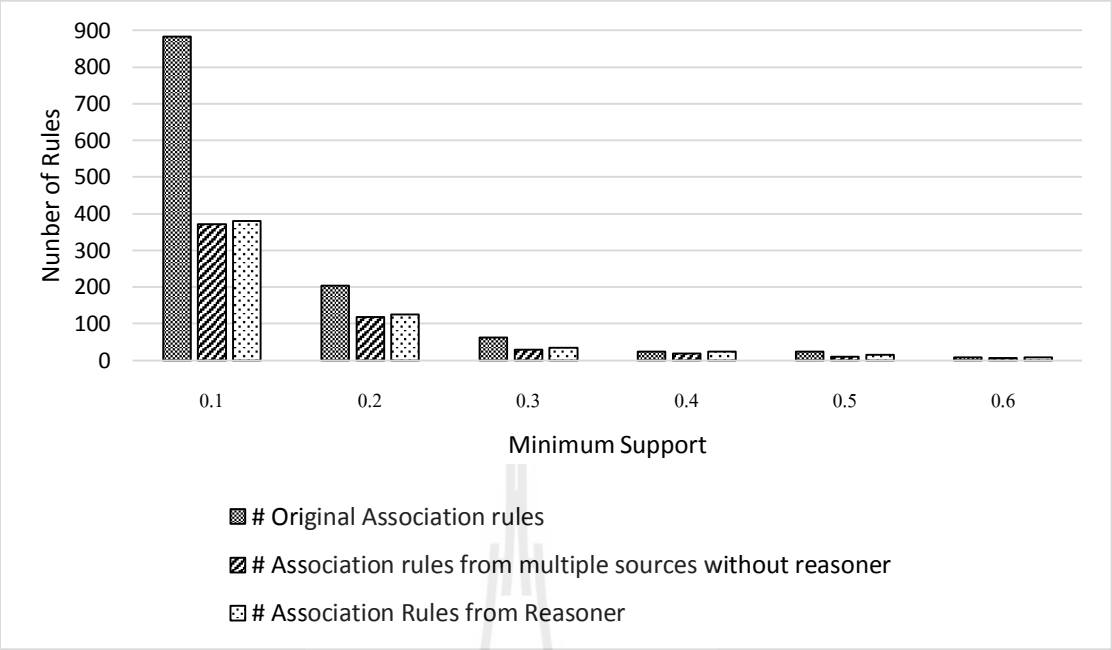
Results / Support	# Original Association rules	# Association rules from multiple sources without reasoner	# Association Rules from Reasoner	% Improvement by Reasoner
0.1	26	25	25	0.00%
0.2	17	17	17	0.00%
0.3	13	9	9	0.00%
0.4	6	5	5	0.00%
0.5	5	5	5	0.00%
0.6	5	5	5	0.00%



รูปที่ 4.2 แผนภูมิแสดงการจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์แบบดั้งเดิมแล การหากฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยข้อมูลผู้รอดชีวิตเรือไททานิค

ตารางที่ 4.17 เปรียบเทียบจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์แบบดั้งเดิมและการหากฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยข้อมูลผู้ป่วยโรคมะเร็งเต้านม

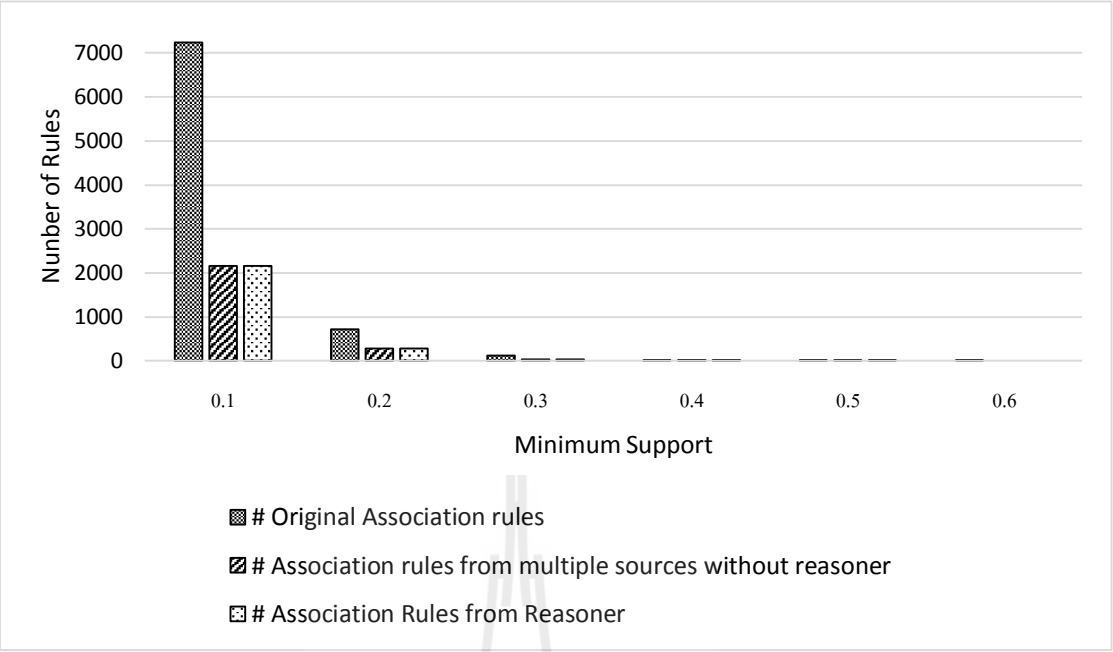
Results / Support	# Original Association rules	# Association rules from multiple sources without reasoner	# Association Rules from Reasoner	% Improvement by Reasoner
0.1	883	372	380	2.15%
0.2	203	119	126	5.88%
0.3	62	30	35	16.67%
0.4	24	19	24	26.32%
0.5	24	10	15	50.00%
0.6	9	6	9	50.00%



รูปที่ 4.3 แผนภูมิแสดงการเปรียบเทียบจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์แบบดั้งเดิมและการหากฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยข้อมูลผู้ป่วยโรคมะเร็งเต้านม

ตารางที่ 4.18 เปรียบเทียบจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์แบบดั้งเดิมและการหากฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยข้อมูลผู้ป่วยโรคหัวใจ

Results / Support	# Original Association rules	# Association rules from multiple sources without reasoner	# Association Rules from Reasoner	% Improvement by Reasoner
0.1	7245	2158	2160	0.09%
0.2	726	282	283	0.35%
0.3	118	42	42	0.00%
0.4	18	10	10	0.00%
0.5	4	4	4	0.00%
0.6	1	0	0	0.00%



รูปที่ 4.4 แผนภูมิแสดงการเปรียบเทียบจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์แบบดั้งเดิมและการหากฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยข้อมูลผู้ป่วยโรคหัวใจ

4.4 เปรียบเทียบผลการทดสอบความถูกต้องของกฎความสัมพันธ์ระหว่างกฎความสัมพันธ์แบบดั้งเดิมและกฎความสัมพันธ์จากข้อมูลหลายแหล่ง

การหากฎความสัมพันธ์นั้นนอกจากจะพิจารณาที่จำนวนของกฎความสัมพันธ์แล้วนั้นจำเป็นต้องพิจารณาในส่วนของความถูกต้องของกฎความสัมพันธ์ระหว่างกฎความสัมพันธ์แบบดั้งเดิมและกฎความสัมพันธ์จากข้อมูลหลายแหล่ง เนื่องจากต้องการทดสอบว่ากฎความสัมพันธ์ที่ได้จากจากข้อมูลหลายแหล่งนั้นเหมือนกับกฎความสัมพันธ์แบบดั้งเดิมหรือไม่ ผลการทดสอบดังตารางที่ 4.19 4.20 และ 4.21 แสดงเปรียบเทียบความถูกต้องของกฎความสัมพันธ์ระหว่างกฎความสัมพันธ์แบบดั้งเดิมและกฎความสัมพันธ์จากข้อมูลหลายแหล่งด้วยข้อมูลผู้รอดชีวิตเรือไททานิค ข้อมูลผู้ป่วยโรคมะเร็งเต้านม และข้อมูลผู้ป่วยโรคหัวใจตามลำดับด้วยค่าสนับสนุนที่ 0.4 ซึ่งจะเห็นได้ว่าข้อมูลผู้รอดชีวิตเรือไททานิคให้ความถูกต้องของกฎความสัมพันธ์คิดเป็น 100% ข้อมูลผู้ป่วยโรคมะเร็งเต้านมให้ความถูกต้องของกฎความสัมพันธ์คิดเป็น 87.50% และข้อมูลผู้ป่วยโรคหัวใจให้ความถูกต้องของกฎความสัมพันธ์คิดเป็น 100%

ตารางที่ 4.19 เปรียบเทียบความถูกต้องของกฎความสัมพันธ์ระหว่างกฎความสัมพันธ์แบบดั้งเดิม และกฎความสัมพันธ์จากข้อมูลหลายแหล่งจากข้อมูลผู้รอดชีวิตเรือไททานิคด้วยค่าสนับสนุนที่ 0.4

กฎความสัมพันธ์แบบดั้งเดิม	กฎความสัมพันธ์จากหลายแหล่ง
{CLASS=crew}=>{AGE=adult}	-
{SURVIVED=no}=>{SEX=male}	{SURVIVED=no}=>{SEX=male}
{SURVIVED=no}=>{AGE=adult}	{SURVIVED=no}=>{AGE=adult}
{SEX=male}=>{AGE=adult}	{SEX=male}=>{AGE=adult}
{SEX=male,SURVIVED=no}=>{AGE=adult}	{SEX=male,SURVIVED=no}=>{AGE=adult}
{AGE=adult,SURVIVED=no}=>{SEX=male}	{AGE=adult,SURVIVED=no}=>{SEX=male}

ตารางที่ 4.20 เปรียบเทียบความถูกต้องของกฎความสัมพันธ์ระหว่างกฎความสัมพันธ์แบบดั้งเดิม และกฎความสัมพันธ์จากข้อมูลหลายแหล่งจากข้อมูลผู้ป่วยโรคมะเร็งเต้านมด้วยค่าสนับสนุนที่ 0.4

กฎความสัมพันธ์แบบดั้งเดิม	กฎความสัมพันธ์จากหลายแหล่ง
{Class=no-recurrence-events}=>{inv-nodes=0-2}	{Class=no-recurrence-events}=>{inv-nodes=0-2}
{Class=no-recurrence-events}=>{irradiat=no}	{Class=no-recurrence-events}=>{irradiat=no}
{Class=no-recurrence-events}=>{node-caps=no}	{Class=no-recurrence-events}=>{node-caps=no}
{inv-nodes=0-2}=>{irradiat=no}	{inv-nodes=0-2}=>{irradiat=no}
{irradiat=no}=>{inv-nodes=0-2}	{irradiat=no}=>{inv-nodes=0-2}
{inv-nodes=0-2}=>{node-caps=no}	{inv-nodes=0-2}=>{node-caps=no}
{node-caps=no}=>{inv-nodes=0-2}	{node-caps=no}=>{inv-nodes=0-2}
{irradiat=no}=>{node-caps=no}	{irradiat=no}=>{node-caps=no}
{node-caps=no}=>{irradiat=no}	{node-caps=no}=>{irradiat=no}
{inv-nodes=0-2,Class=no-recurrence-events}=>{irradiat=no}	{inv-nodes=0-2,Class=no-recurrence-events}=>{irradiat=no}
{irradiat=no,Class=no-recurrence-events}=>{inv-nodes=0-2}	{irradiat=no,Class=no-recurrence-events}=>{inv-nodes=0-2}
{inv-nodes=0-2,irradiat=no}=>{Class=no-recurrence-events}	-

ตารางที่ 4.20 เปรียบเทียบความถูกต้องของกฎความสัมพันธ์ระหว่างกฎความสัมพันธ์แบบดั้งเดิม และกฎความสัมพันธ์จากข้อมูลหลายแหล่งจากข้อมูลผู้ป่วยโรคมะเร็งเต้านมด้วยค่าสนับสนุนที่ 0.4 (ต่อ)

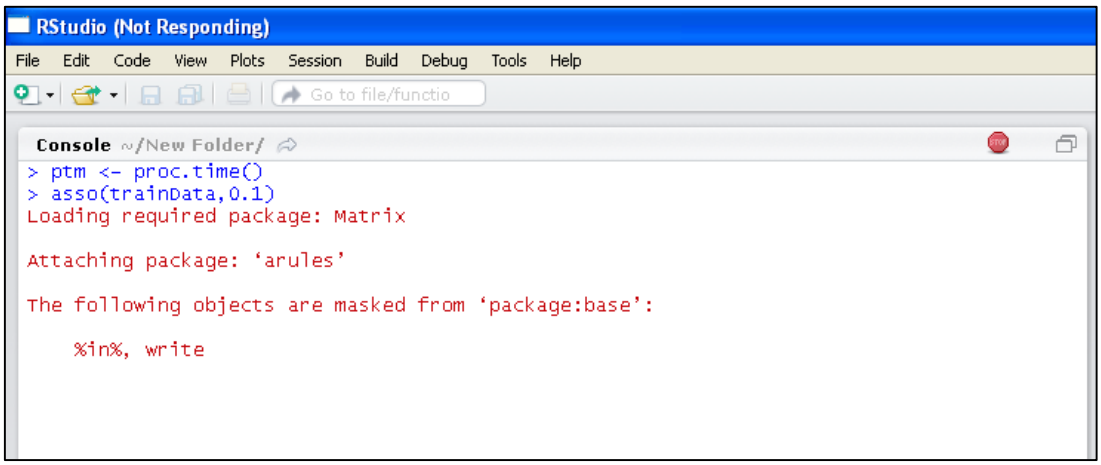
กฎความสัมพันธ์แบบดั้งเดิม	กฎความสัมพันธ์จากหลายแหล่ง
{inv-nodes=0-2,Class=no-recurrence-events}=>{node-caps=no}	{inv-nodes=0-2,Class=no-recurrence-events}=>{node-caps=no}
{node-caps=no,Class=no-recurrence-events}=>{inv-nodes=0-2}	{node-caps=no,Class=no-recurrence-events}=>{inv-nodes=0-2}
{irradiat=no,Class=no-recurrence-events}=>{node-caps=no}	{irradiat=no,Class=no-recurrence-events}=>{node-caps=no}
{node-caps=no,Class=no-recurrence-events}=>{irradiat=no}	{node-caps=no,Class=no-recurrence-events}=>{irradiat=no}
{node-caps=no,irradiat=no}=>{Class=no-recurrence-events}	-
{inv-nodes=0-2,irradiat=no}=>{node-caps=no}	{inv-nodes=0-2,irradiat=no}=>{node-caps=no}
{inv-nodes=0-2,node-caps=no}=>{irradiat=no}	{inv-nodes=0-2,node-caps=no}=>{irradiat=no}
{node-caps=no,irradiat=no}=>{inv-nodes=0-2}	{node-caps=no,irradiat=no}=>{inv-nodes=0-2}
{inv-nodes=0-2,irradiat=no,Class=no-recurrence-events}=>{node-caps=no}	{inv-nodes=0-2,irradiat=no,Class=no-recurrence-events}=>{node-caps=no}
{inv-nodes=0-2,node-caps=no,Class=no-recurrence-events}=>{irradiat=no}	{inv-nodes=0-2,node-caps=no,Class=no-recurrence-events}=>{irradiat=no}
{node-caps=no,irradiat=no,Class=no-recurrence-events}=>{inv-nodes=0-2}	{node-caps=no,irradiat=no,Class=no-recurrence-events}=>{inv-nodes=0-2}
{inv-nodes=0-2,node-caps=no,irradiat=no}=>{Class=no-recurrence-events}	-

ตารางที่ 4.21 เปรียบเทียบความถูกต้องของกฎความสัมพันธ์ระหว่างกฎความสัมพันธ์แบบดั้งเดิม และกฎความสัมพันธ์จากข้อมูลหลายแหล่งจากข้อมูลผู้ป่วยโรคหัวใจด้วยค่าสนับสนุนที่ 0.4

กฎความสัมพันธ์แบบดั้งเดิม	กฎความสัมพันธ์จากหลายแหล่ง
{slope=up}=>{fbs=f}	-
{cp=asympt}=>{fbs=f}	-
{restecg=normal}=>{fbs=f}	-
{thalach=115-158}=>{fbs=f}	-
{trestbps=129-164}=>{fbs=f}	{trestbps=129-164}=>{fbs=f}
{num=<50}=>{exang=no}	{num=<50}=>{exang=no}
{num=<50}=>{fbs=f}	{num=<50}=>{fbs=f}
{oldpeak=0}=>{fbs=f}	{oldpeak=0}=>{fbs=f}
{thal=normal}=>{exang=no}	-
{thal=normal}=>{fbs=f}	{thal=normal}=>{fbs=f}
{age=46-60}=>{fbs=f}	-
{ca=0.0}=>{fbs=f}	{ca=0.0}=>{fbs=f}
{exang=no}=>{fbs=f}	{exang=no}=>{fbs=f}
{sex=male}=>{fbs=f}	{sex=male}=>{fbs=f}
{chol=126-272}=>{fbs=f}	{chol=126-272}=>{fbs=f}
{chol=126-272,ca=0.0}=>{fbs=f}	-
{chol=126-272,exang=no}=>{fbs=f}	{chol=126-272,exang=no}=>{fbs=f}
{sex=male,chol=126-272}=>{fbs=f}	-

4.5 ผลการทดสอบการหากฎความสัมพันธ์แบบดั้งเดิม

จากผลการทดสอบประสิทธิภาพการหากฎความสัมพันธ์แบบดั้งเดิม โดยจะนำข้อมูลผู้ป่วยโรคหัวใจมาใช้ในการทดสอบ ซึ่งจะทำการเพิ่มจำนวนของข้อมูลจากเดิมทั้งหมด 303 แถว เพิ่มเป็น 3,000,000 แถว โดยจะเลือกใช้ค่าสนับสนุนที่ 0.1 สำหรับการหากฎความสัมพันธ์ จากรูปที่ 4.5 แสดงการหากฎความสัมพันธ์แบบดั้งเดิมที่ไม่สามารถประมวลผลได้ จะเห็นได้ว่าโปรแกรมนั้นค้างอยู่ในระหว่างการหากฎความสัมพันธ์ ซึ่งอาจจะต้องใช้เวลานานหรือไม่สามารถประมวลผลได้ อาจเกิดขึ้นเนื่องจากหน่วยความจำเต็ม หรือข้อจำกัดของหน่วยประมวลผลกลาง



รูปที่ 4.5 การหากฎความสัมพันธ์แบบดั้งเดิมที่ไม่สามารถประมวลผลได้

4.6 อภิปรายผล

จากผลการทดสอบประสิทธิภาพการหากฎความสัมพันธ์จากข้อมูล 3 ชุด ได้แก่ ข้อมูลผู้รอดชีวิตเรือไททานิค ข้อมูลผู้ป่วยโรคมะเร็งเต้านม และข้อมูลผู้ป่วยโรคหัวใจด้วยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งด้วยค่าสนับสนุนที่แตกต่างกัน ได้แก่ ค่าสนับสนุนที่ 0.1 0.2 0.3 0.4 0.5 และ 0.6 สามารถสรุปผลการทดสอบเปรียบเทียบได้ดังนี้

1) การเปรียบเทียบจำนวนกฎความสัมพันธ์จากค่าสนับสนุนที่แตกต่างกันจากข้อมูลในแต่ละชุด จากตารางที่ 4.16 4.17 และ 4.18 ตามลำดับ จะเห็นได้ว่าเมื่อหากฎความสัมพันธ์ด้วยค่าสนับสนุนที่น้อย จะให้กฎความสัมพันธ์จากกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งมีจำนวนที่น้อยเมื่อเทียบกับการหากฎความสัมพันธ์แบบดั้งเดิม แต่เมื่อหากฎความสัมพันธ์ด้วยค่าสนับสนุนที่มากจะให้จำนวนกฎความสัมพันธ์ที่ใกล้เคียงกับการหากฎความสัมพันธ์แบบดั้งเดิม ดังรูปที่ 4.2 4.3 และ 4.4 ตามลำดับ

2) กฎความสัมพันธ์ใหม่ที่ได้จากกฎความสัมพันธ์เดิมเมื่อใช้ค่าสนับสนุนที่แตกต่างกันจะเห็นได้ว่าข้อมูลผู้รอดชีวิตเรือไททานิคไม่ปรากฏกฎความสัมพันธ์ใหม่ เนื่องกฎความสัมพันธ์เดิมเมื่อนำไปทำการอนุมานความรู้ใหม่ แต่กฎความสัมพันธ์ที่ได้จากการอนุมานความรู้ใหม่นั้นปรากฏอยู่ในกฎความสัมพันธ์เดิมอยู่แล้ว ดังนั้นถือได้ว่ากฎความสัมพันธ์เหล่านั้นไม่ใช่กฎความสัมพันธ์ใหม่ ทำให้ไม่ปรากฏกฎความพันธ์ใหม่จากข้อมูลชุดนี้ ส่วนข้อมูลผู้ป่วยโรคมะเร็งเต้านมและข้อมูลผู้ป่วยโรคหัวใจจะเห็นได้ว่าสามารถอนุมานกฎความสัมพันธ์ใหม่จากกฎความสัมพันธ์เดิมนั้นได้ ซึ่งเมื่อพิจารณาจากค่าสนับสนุนที่แตกต่างกันจะเห็นได้ว่าจำนวนกฎความสัมพันธ์ใหม่ที่ได้ลดลงเมื่อใช้ค่าสนับสนุนที่มากขึ้น

3) จากผลการทดสอบที่ได้กฎความสัมพันธ์ใหม่ที่ได้จากกฎความสัมพันธ์เดิม สามารถนำกฎความสัมพันธ์ใหม่ที่ได้นั้นไปเพิ่มเติมในส่วนของกฎความสัมพันธ์ที่ขาดหายไปได้เมื่อเทียบกับกฎความสัมพันธ์ที่ได้จากการหากฎความสัมพันธ์แบบดั้งเดิม ดังนั้นในส่วนนี้สามารถช่วยเพิ่มประสิทธิภาพให้กับการหากฎความสัมพันธ์ด้วยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งให้มีจำนวนที่เหมือนกันใกล้เคียงกับการการหากฎความสัมพันธ์แบบดั้งเดิมมากที่สุด

4) การเปรียบเทียบความถูกต้องของกฎความสัมพันธ์ระหว่างกฎความสัมพันธ์แบบดั้งเดิมและกฎความสัมพันธ์จากข้อมูลหลายแหล่ง จะเห็นได้ว่ากฎความสัมพันธ์จากข้อมูลหลายแหล่งให้ความถูกต้องของกฎความสัมพันธ์ที่สูง แต่ในบางส่วนที่ไม่ได้ถึง 100% เนื่องจากเป็นกฎความสัมพันธ์ที่ได้จากการอนุมานความรู้ใหม่ ซึ่งมีบางกฎที่ไม่เหมือนกับกฎความสัมพันธ์แบบดั้งเดิม

จากผลการทดสอบประสิทธิภาพกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง ผลการทดสอบสรุปได้ว่าเมื่อหากฎความสัมพันธ์จากจากข้อมูล 3 ชุด ได้แก่ ข้อมูลผู้รอดชีวิตเรือไททานิค ข้อมูลผู้ป่วยโรคมะเร็งเต้านม และข้อมูลผู้ป่วยโรคหัวใจด้วยค่าสนับสนุนที่มากจะให้จำนวนกฎความสัมพันธ์ที่ใกล้เคียงกับการหากฎความสัมพันธ์แบบดั้งเดิมมากกว่าค่าสนับสนุนที่น้อย กฎความสัมพันธ์ใหม่ที่ได้จากกฎความสัมพันธ์เดิมจากการหากฎความสัมพันธ์จากข้อมูลผู้ป่วยโรคมะเร็งเต้านม และข้อมูลผู้ป่วยโรคหัวใจสามารถช่วยเพิ่มจำนวนกฎความสัมพันธ์ที่ขาดหายไปได้เมื่อเทียบกับจำนวนกฎความสัมพันธ์ที่ได้จากการหากฎความสัมพันธ์แบบดั้งเดิม

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

ปัจจุบันเทคโนโลยีได้เข้ามามีบทบาทเป็นอย่างมากในการช่วยจัดเก็บข้อมูลในหน่วยงานหรือองค์กรต่าง ๆ แต่สิ่งที่ตามมาคือข้อมูลมีขนาดใหญ่หรือข้อมูลมีการกระจายตัวกันอยู่ตามแหล่งต่าง ๆ เนื่องจากการที่สามารถจัดเก็บข้อมูลได้อย่างง่าย ทำให้การหากฎความสัมพันธ์ซึ่งเป็นกระบวนการหนึ่งทางด้านการทำเหมืองข้อมูลเพื่อให้ได้ความรู้ใหม่จากข้อมูลเหล่านี้นั้นทำได้ยาก ดังนั้นได้มีงานวิจัยที่จะนำเสนอการหากฎความสัมพันธ์จากข้อมูลที่มีขนาดใหญ่หรือข้อมูลมีการกระจายตัวกันอยู่ตามแหล่งต่าง ๆ แต่งานวิจัยส่วนมากจำเป็นต้องใช้คอมพิวเตอร์ที่มีประสิทธิภาพสูงในการประมวลผล และสิ่งที่ตามมาคือค่าใช้จ่ายที่เพิ่มขึ้นตามมานั้นเอง ดังนั้นงานวิจัยนี้ได้เสนอกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง โดยไม่จำเป็นต้องจัดหาคอมพิวเตอร์ที่มีประสิทธิภาพสูงในการประมวลผล แต่ให้ประสิทธิภาพใกล้เคียงกับการหากฎความสัมพันธ์แบบดั้งเดิมมากที่สุด

ในงานวิจัยนี้มุ่งเน้นในกระบวนการออกแบบอัลกอริทึมและพัฒนาโปรแกรมเพื่อค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง ซึ่งจะทำให้ได้กฎความสัมพันธ์ที่มีประสิทธิภาพใกล้เคียงกับการหากฎความสัมพันธ์แบบดั้งเดิม โดยพิจารณาจากจำนวนกฎความสัมพันธ์ที่เหมือนกันกับกฎความสัมพันธ์ที่ได้จากการหากฎความสัมพันธ์แบบดั้งเดิม ดังนั้นงานวิจัยนี้ได้พัฒนาและออกแบบอัลกอริทึมใหม่ในส่วนต่าง ๆ ดังนี้

1) การรวมกฎความสัมพันธ์โดยจะค้นหากฎความสัมพันธ์ที่ปรากฏทุก ๆ แหล่งความรู้เพื่อนำมาเก็บไว้ในแหล่งความรู้เพียงแหล่งเดียว

2) แปลงกฎความสัมพันธ์ที่อยู่ในรูปแบบทั่วไปให้อยู่ในรูปแบบของภาษาธรรมชาติ เพื่อนำกฎความสัมพันธ์ไปแทนความรู้สำหรับการนำไปสร้างเป็นออนโทโลยี

3) การนำกฎความสัมพันธ์ที่ถูกแทนความรู้แล้วนำไปสร้างเป็นออนโทโลยีไปตรวจสอบความขัดแย้ง และอนุมานกฎความสัมพันธ์ใหม่จากกฎความสัมพันธ์เดิม โดยความรู้ใหม่ที่ได้สามารถนำไปเพิ่มเติมในส่วนของกฎความสัมพันธ์ที่ขาดหายไปได้

5.1 สรุปผลการวิจัย

ผลการทดสอบประสิทธิภาพกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งด้วยข้อมูลผู้รอดชีวิตจากเรือไททานิค ข้อมูลผู้ป่วยโรคมะเร็งเต้านม และข้อมูลผู้ป่วยโรคหัวใจด้วยค่าสนับสนุนที่แตกต่างกัน โดยเมื่อใช้ค่าสนับสนุนที่น้อยข้อมูลทั้ง 3 ชุด จะให้จำนวนกฎความสัมพันธ์ที่แตกต่างจากการหากฎความสัมพันธ์แบบดั้งเดิม แต่เมื่อใช้ค่าสนับสนุนที่มากข้อมูลทั้ง 3 ชุด จะให้จำนวนกฎความสัมพันธ์ที่ใกล้เคียงหรือเทียบเท่ากับการหากฎความสัมพันธ์แบบดั้งเดิม และในขั้นตอนของการตรวจสอบความขัดแย้งและอนุมานความรู้ใหม่ที่ได้จากความรู้เดิมนั้น การหากฎความสัมพันธ์ด้วยกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งด้วยจากข้อมูลผู้ป่วยโรคมะเร็งเต้านม และข้อมูลผู้ป่วยโรคหัวใจ สามารถนำกฎความสัมพันธ์ใหม่ที่ได้ไปเพิ่มเติมในส่วนของกฎความสัมพันธ์ที่ขาดหายไปได้ ส่วนข้อมูลผู้รอดชีวิตจากเรือไททานิคที่ไม่ปรากฏกฎความสัมพันธ์ใหม่นั้น เนื่องจากจำนวนกฎ จำนวนคอลัมน์ที่น้อยเกิดไป แต่เมื่อดูจากจำนวนกฎความสัมพันธ์ที่ได้แล้วนั้นก็มีจำนวนใกล้เคียงกับการหากฎความสัมพันธ์แบบดั้งเดิม ดังนั้นงานวิจัยที่ได้เสนอกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งสามารถช่วยค้นหาและรวมกฎความสัมพันธ์ที่ได้จากข้อมูลที่กระจายตัวกันอยู่หรือข้อมูลที่มีขนาดใหญ่ได้ ซึ่งมีประสิทธิภาพใกล้เคียงกับการหากฎความสัมพันธ์แบบดั้งเดิมในค่าสนับสนุนที่มาก โดยกระบวนการหากฎความสัมพันธ์นั้นไม่จำเป็นต้องใช้คอมพิวเตอร์ที่มีประสิทธิภาพสูงในการประมวลผล

5.2 ปัญหาและข้อเสนอแนะ

กลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งนั้นเป็นการทำงานแบบกึ่งอัตโนมัติ ซึ่งในขั้นตอนของการแทนความรู้ด้วยภาษาธรรมชาติและการตรวจสอบความขัดแย้งของกฎความสัมพันธ์นั้นผู้ใช้จะต้องเป็นคนทำเอง และเมื่อใช้ค่าสนับสนุนที่น้อยในการหากฎความสัมพันธ์กลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งยังให้จำนวนของกฎความสัมพันธ์ที่น้อยเมื่อเทียบกับการหากฎความสัมพันธ์แบบดั้งเดิม

กลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง สามารถรวมกฎความสัมพันธ์ที่ได้จากการหากฎความสัมพันธ์ในแต่ละแหล่งข้อมูล โดยไม่จำเป็นต้องรวมข้อมูลจากแหล่งต่าง ๆ ให้อยู่ในแหล่งข้อมูลเพียงแหล่งเดียวเพื่อนำไปหากฎความสัมพันธ์ ซึ่งสามารถช่วยลดภาระในการจัดซื้อคอมพิวเตอร์ที่มีประสิทธิภาพสูงสำหรับการนำมาประมวลผล แต่ยังให้ประสิทธิภาพที่ใกล้เคียงกับการหากฎความสัมพันธ์จากข้อมูลเพียงแหล่งเดียว โดยสามารถนำไปประยุกต์ใช้กับงานหรือข้อมูลได้หลากหลายด้าน เช่น การหากฎความสัมพันธ์จากข้อมูลทางการแพทย์ที่กระจายตัวกันอยู่ตามคลินิก หรือโรงพยาบาลต่าง ๆ เป็นต้น

สามารถนำอัลกอริทึมกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่งไปพัฒนาต่อยอดให้อักอริทึมมีการทำงานแบบอัตโนมัติเพื่อลดความผิดพลาดของผู้ใช้และช่วยลดเวลาการทำงาน ในอนาคตถ้านำจุดเด่นในส่วนของการรวมกฎความสัมพันธ์ไปปรับปรุงเพื่อให้ได้จำนวนกฎความสัมพันธ์ที่ใกล้เคียงหรือเท่ากับการหากฎความสัมพันธ์เดิมมากที่สุด โดยไม่จำเป็นต้องใช้คอมพิวเตอร์ที่มีประสิทธิภาพสูง ตัวอย่างเช่น ปรับปรุงเทคนิคในการค้นหาและรวมกฎความสัมพันธ์เพื่อให้ได้จำนวนกฎมายิ่งขึ้น ใช้เครื่องมือหรืออัลกอริทึมอื่นที่สามารถอนุมานความรู้ได้ดีกว่าเครื่องมือที่ใช้ในงานวิจัยนี้ เป็นต้น

รายการอ้างอิง

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

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In **Proceedings of the 20th international conference on very large databases** (pp. 487–499).

Agrawal, R., & Shafer, J. C. (1996). Parallel mining of association rules. **IEEE Transactions on Knowledge and Data Engineering**. 8(6):962–969.

Angluin, D., & Smith, C. H. (1983). Inductive inference: Theory and methods. **ACM Computing Surveys**, 15(3):237-269.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., & Zaharia, M. (2010). A view of cloud computing. **Communications of the ACM**. 53(4):50-58.

Clark, P., Harrison, P., Jenkins, T., Thompson, J. A., & Wojcik, R. H. (2005). Acquiring and Using World Knowledge Using a Restricted Subset of English. In **Proceeding of FLAIRS Conference** (pp. 506-511).

Chattratichat, J., Darlington, J., Guo, Y., Hedvall, S., Köhler, M., & Syed, J. (1999). An architecture for distributed enterprise data mining. In **High-Performance Computing and Networking** (pp. 573-582). Springer Berlin Heidelberg.

Cheung, D. W., Han, J., Ng, V. T., Fu, A. W., & Fu, Y. (1996). A fast distributed algorithm for mining association rules. In **Proceeding of the Fourth International Conference on Parallel and Distributed Information Systems** (pp. 31–42).

Cheung, D. W., Lee, S. D., & Xiao, Y. (2002). Effect of data skewness and workload balance in parallel data mining. **IEEE Transactions on Knowledge and Data Engineering**. 14(3):498–514.

Cheung, D. W., Ng, V. T., & Fu, A. W. (1996). Efficient mining of association rules in distributed databases. **IEEE Transactions on Knowledge and Data Engineering**. 8(6):911–922.

Elayyadi, I., Benbernou, S., Ouziri, M., & Younas, M. (2014). A tensor-based distributed discovery of missing association rules on the cloud**. Future Generation Computer Systems**.35:49-56.

Fuchs, N. E., Kaljurand, K., & Schneider, G. (2006). Attempto Controlled English Meets the Challenges of Knowledge Representation, Reasoning, Interoperability and User Interfaces. In **Proceeding of FLAIRS Conference** (Vol. 12, pp. 664-669).

Fuchs, N. E., Schwertel, U., & Schwitter, R. (1999). Attempto Controlled English—not just another logic specification language. In **Logic-based Program Synthesis and Transformation** (pp. 1-20). Springer Berlin Heidelberg.

Grossman, R., & Gu, Y. (2008). Data mining using high performance data clouds: experimental studies using sector and sphere. In **Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining** (pp. 920-927).

Gruber, T. (2014). **Ontology** [On-line]. Available: http://tomgruber.org/writing/ontology-definition- 2007. htm

Gu, Y., & Grossman, R. L. (2009). Sector and Sphere: the design and implementation of a high-performance data cloud. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, 367(1897):2429-2445.

Kaljurand, K. (2007). **Attempto controlled English as a Semantic Web language**. Tartu University Press.

Kuhn, T. (2008). Acewiki: A natural and expressive semantic wiki. **arXiv preprint arXiv**:0807.4618.

Li, T., Zhu, S., & Ogihara, M. (2003). A new distributed data mining model based on similarity. In **Proceedings of the 2003 ACM Symposium on Applied Computing** (pp. 432-436).

Leighton, F. T. (1992). **Introduction to Parallel Algorithms and Architectures** (pp. 222-232). San Francisco: Morgan Kaufmann.

MacGregor, R. M. (1991). Using a description classifier to enhance deductive inference. In **Proceedings of the Seventh IEEE Conference on Artificial Intelligence Applications** (Vol. 1, pp. 141-147). IEEE.

Schwitter, R. (2002). English as a formal specification language. In **Proceedings of the 13th International Workshop on Database and Expert Systems Applications** (pp. 228-232). IEEE.

Sowa, J. F. (2004). Common logic controlled english. **Technical report**, 2004. Draft, 24 February 2004, http://www. jfsowa. com/clce/specs. htm

Tsarkov, D., & Horrocks, I. (2006). FaCT++ description logic reasoner: System description. In **Automated Reasoning** (pp. 292-297). Springer Berlin Heidelberg.

Tseng, F. S., Kuo, Y. H., & Huang, Y. M. (2010). Toward boosting distributed association rule mining by data de-clustering. **Information Sciences**, 180(22):4263-4289.

Tsoumakas, G., & Vlahavas, I. P. (2009). **Distributed Data Mining**.157-164

Yu, K. M., Zhou, J., Hong, T. P., & Zhou, J. L. (2010). A load-balanced distributed parallel mining algorithm. **Expert Systems with Applications**, 37(3):2459-2464.

Wikipedia, The Free Encyclopedia (2014b). **Inference** [On-line]. Available: http://en.wikipedia.org/wiki/Inference

ภาคผนวก ก

การใช้งานโปรแกรม

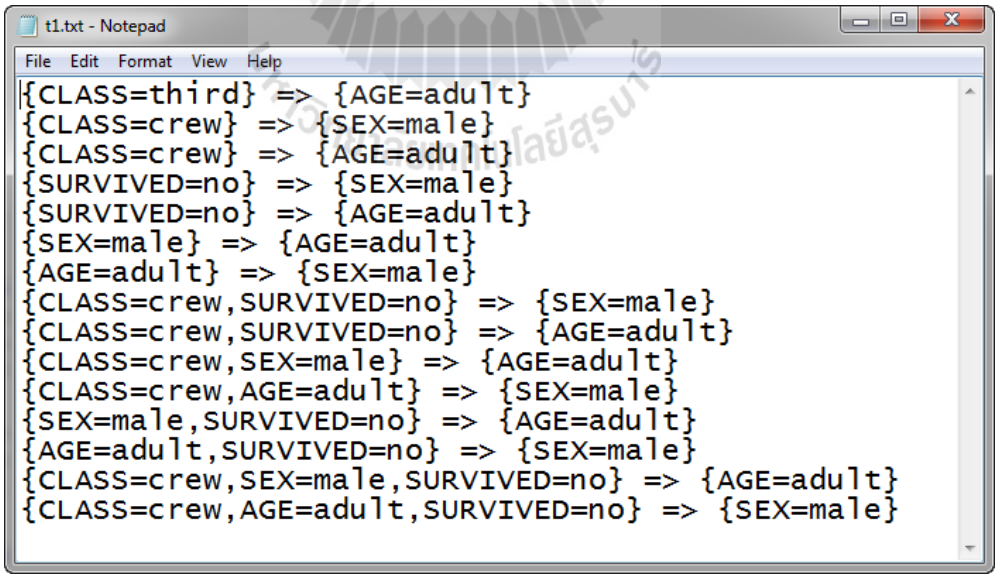
การใช้งานโปรแกรม

เนื้อหาในส่วนนี้จะอธิบายการใช้งานของโปรแกรมกลไกการค้นหาและรวมกฎความสัมพันธ์จากหลายแหล่ง โดยจะแบ่งการทำงานของโปรแกรมออกเป็นขั้นตอนดังนี้

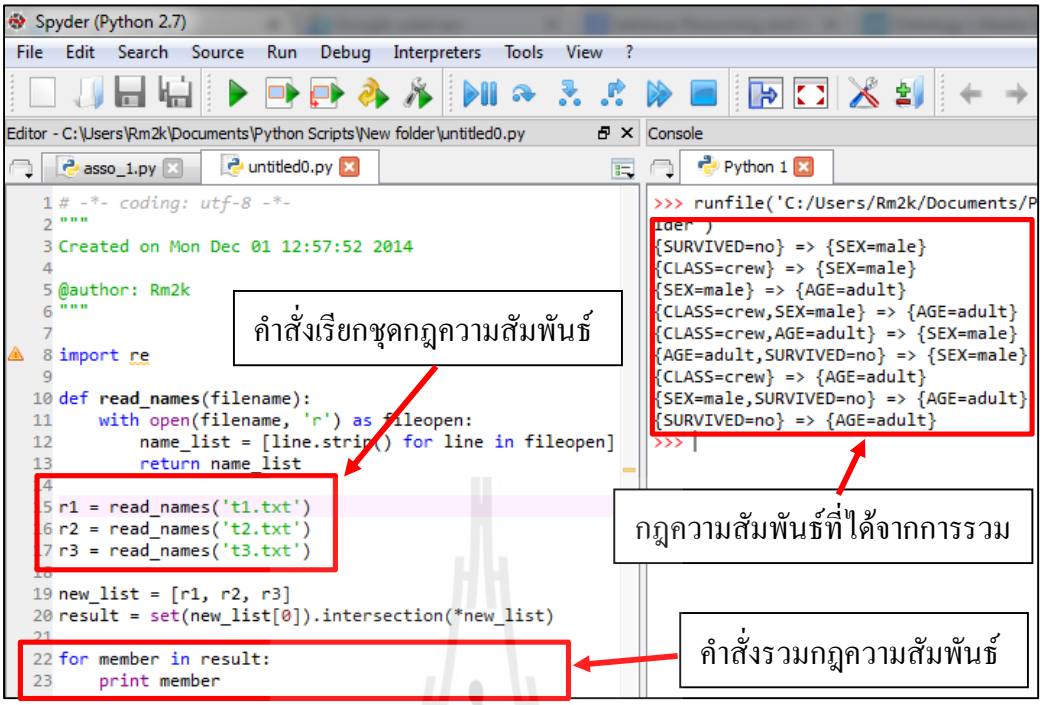
ก.1 การใช้งานโปรแกรมรวมกฎความสัมพันธ์จากแหล่งความรู้ต่าง ๆ

การเรียกใช้โปรแกรมรวมกฎความสัมพันธ์จากแหล่งความรู้ต่าง ๆ นั้น จำเป็นต้องมีแหล่งความรู้กฎความสัมพันธ์อยู่ก่อนแล้ว ซึ่งอาจจะถูกเก็บบนฐานข้อมูลที่อยู่ไกลออกไป หรือถูกจัดเก็บอยู่บนคอมพิวเตอร์เครื่องหลัก ในงานวิจัยนี้แหล่งความรู้กฎความสัมพันธ์ถูกจัดเก็บอยู่บนคอมพิวเตอร์เครื่องหลัก โดยข้อมูลจะอยู่ในรูปแบบของไฟล์นามสกุล .txt ดังรูปที่ ก.1 แสดงตัวอย่างกฎความสัมพันธ์ที่ถูกจัดเก็บอยู่ในรูปแบบของไฟล์นามสกุล .txt

การรวมกฎความสัมพันธ์จะเรียกไฟล์นามสกุล .txt ขึ้นมาด้วยฟังก์ชัน read_names() โดย 1 ไฟล์หมายถึงแหล่งความรู้กฎความสัมพันธ์ 1 แหล่ง ดังรูปที่ ก.2 แสดงตัวอย่างการเรียกใช้โปรแกรมรวมกฎความสัมพันธ์จากแหล่งความรู้ต่าง ๆ ซึ่งเป็นการรวมกฎความสัมพันธ์จากแหล่งความรู้กฎความสัมพันธ์ทั้งหมด 3 แหล่ง โดยการรวมกฎความสัมพันธ์จะใช้ฟังก์ชัน intersection() เพื่อดึงเอาเฉพาะกฎความสัมพันธ์ที่ปรากฏทุกแหล่งความรู้ สุดท้ายผลลัพธ์ที่ได้คือกฎความสัมพันธ์เพียงชุดเดียวที่ได้จากการดึงเอาเฉพาะกฎความสัมพันธ์ที่ปรากฏทุกแหล่งความรู้เพื่อนำไปแปลงรูปแบบให้อยู่รูปแบบของภาษาธรรมชาติ



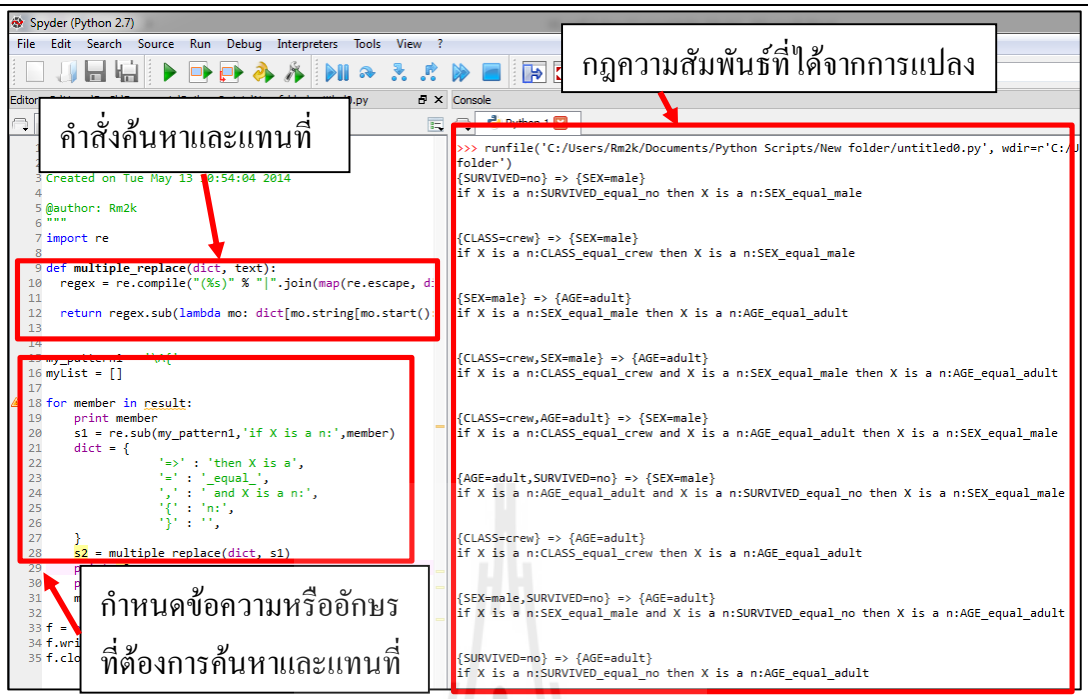
รูปที่ ก.1 ตัวอย่างกฎความสัมพันธ์ที่ถูกจัดเก็บอยู่ในรูปแบบของไฟล์นามสกุล .txt



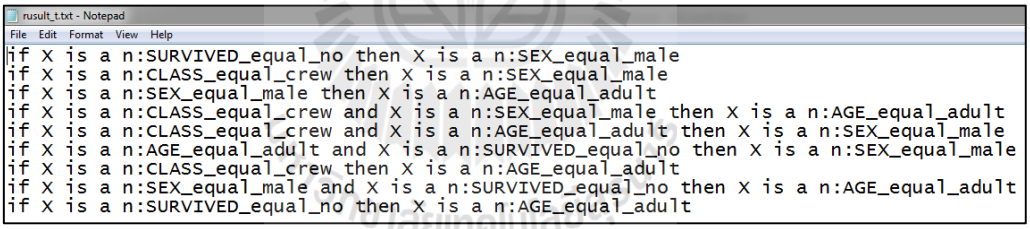
รูปที่ ก.2 ตัวอย่างการเรียกใช้โปรแกรมรวมกฎความสัมพันธ์จากแหล่งความรู้ต่าง ๆ

ก.2 การใช้งานโปรแกรมแปลงรูปแบบกฎความสัมพันธ์ทั่วไปให้อยู่ในรูปแบบภาษาธรรมชาติ

การเรียกใช้งานโปรแกรมแปลงรูปแบบกฎความสัมพันธ์ทั่วไปให้อยู่ในรูปแบบภาษาธรรมชาตินั้น จะใช้เทคนิคการค้นหาและแทนที่ข้อความหรือตัวอักษรที่ต้องเพื่อทำให้กฎความสัมพันธ์ในรูปแบบทั่วไป จากรูปที่ ก.3 แสดงตัวอย่างการใช้งานโปรแกรมแปลงรูปแบบกฎความสัมพันธ์ทั่วไปให้อยู่ในรูปแบบภาษาธรรมชาติ โดยเริ่มต้นจะรับกฎความสัมพันธ์ที่ได้จากการรวมกฎความสัมพันธ์จากแหล่งต่าง ๆ แล้วทำการกำหนดข้อความหรือตัวอักษรที่ต้องการค้นหาและแทนที่ตามที่กล่าวในข้างต้น และนำข้อความหรือตัวอักษรที่ต้องการค้นหาพร้อมกับกฎความสัมพันธ์ไปเข้าฟังก์ชัน multiple_replace() เพื่อแปลงกฎความสัมพันธ์ให้อยู่ในรูปแบบของภาษาธรรมชาติ ตัวอย่างผลลัพธ์ที่ได้ เช่น {SURVIVED=no} => {SEX=male} จะได้กฎความสัมพันธ์ที่เขียนในรูปแบบภาษาธรรมชาติคือ ‘if X is a n:SURVIVED_equal_no then X is a n:SEX_equal_male’ เป็นต้น ผลลัพธ์สุดท้ายคือกฎความสัมพันธ์ที่อยู่ในรูปแบบของภาษาธรรมชาติเพื่อนำไปใช้ในขั้นตอนของการสร้างเป็นออนโทโลยีและตรวจสอบความขัดแย้ง ดังรูปที่ ก.4 แสดงตัวอย่างกฎความสัมพันธ์ที่อยู่ในรูปแบบภาษาธรรมชาติ



รูปที่ ก.3 ตัวอย่างการใช้งานโปรแกรมแปลงรูปแบบกฎความสัมพันธ์ทั่วไปให้อยู่ในรูปแบบภาษาธรรมชาติ

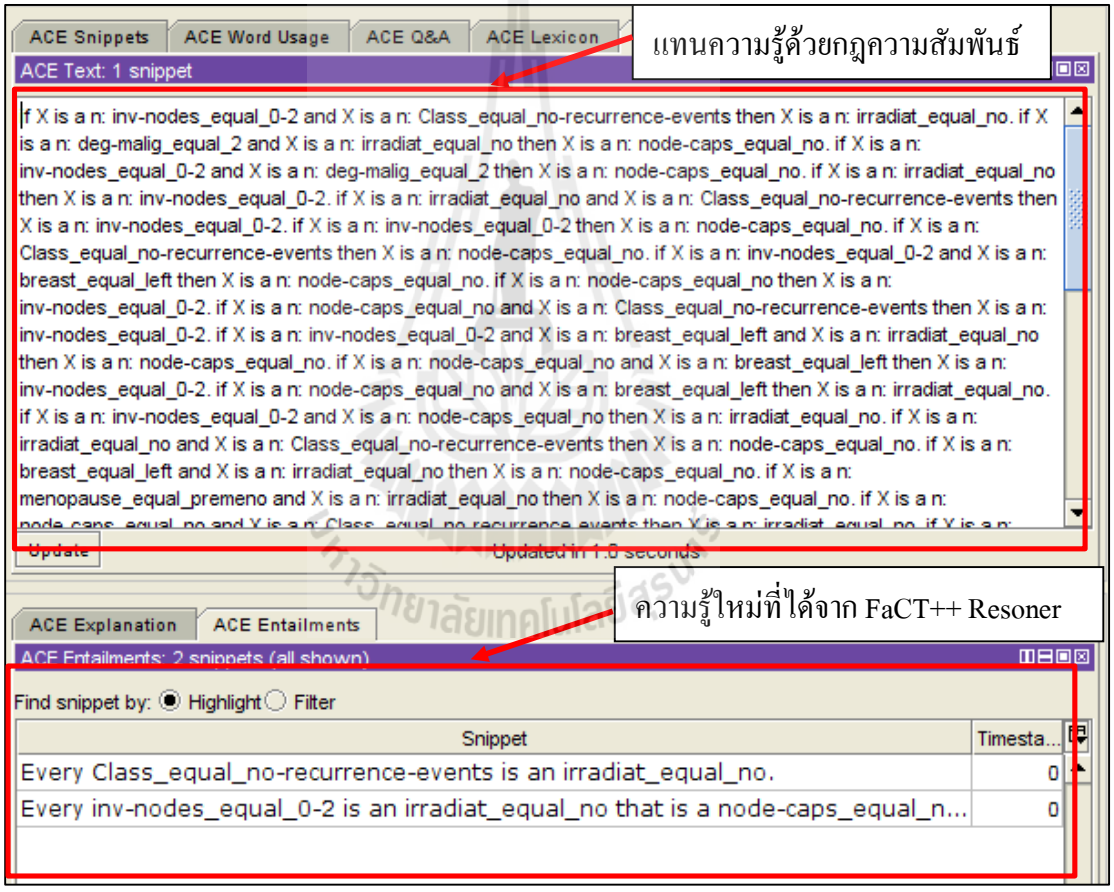


รูปที่ ก.4 ตัวอย่างกฎความสัมพันธ์ที่อยู่ในรูปแบบภาษาธรรมชาติ

ก.3 การใช้งานโปรแกรมตรวจสอบความขัดแย้งของกฎความสัมพันธ์ที่ได้จากแหล่งความรู้ต่าง ๆ

การเรียกใช้งานโปรแกรมตรวจสอบความขัดแย้งของกฎความสัมพันธ์ที่ได้จากแหล่งความรู้ต่าง ๆ นั้น จะนำกฎความสัมพันธ์ที่ได้จากโปรแกรมแปลงรูปแบบกฎความสัมพันธ์ทั่วไปให้อยู่ในรูปแบบภาษาธรรมชาติไปใช้บน ACE View ซึ่งเป็นปลั๊กอินบนโปรแกรม Protégé Editor ซึ่งกฎความสัมพันธ์ที่จะนำมาใช้ได้ต้องถูกต้องตามกฎของภาษาและไวยากรณ์ของ ACE ที่ได้ทำการกำหนดไว้ ไม่เช่นนั้นจะไม่สามารถนำกฎความสัมพันธ์เหล่านั้นไปสร้างเป็นออนโทโลยีได้

จากรูปที่ ก.5 แสดงตัวอย่างการแทนความรู้และตรวจสอบความขัดแย้งของกฎความสัมพันธ์ โดยเริ่มต้นจะนำกฎความสัมพันธ์ที่อยู่ในรูปแบบของภาษาธรรมชาติไปแทนความรู้ในส่วนของ ACE Text เพื่อนำไปสร้างเป็นออนโทโลยี แต่ถ้ามีข้อผิดพลาดเกิดขึ้นจะมีการแจ้งเตือนให้กับผู้ใช้รับทราบ ถ้าไม่มีข้อผิดพลาดเกิดขึ้นสามารถเรียกใช้ FaCT++ Reasoner เพื่อตรวจสอบความขัดแย้งของกฎความสัมพันธ์ โดยถ้าเกิดความขัดแย้งเกิดขึ้นจะมีการแจ้งเตือนให้กับผู้ใช้รับทราบ ถ้าไม่มีความขัดแย้งเกิดขึ้นจะได้ความรู้ใหม่ที่ได้จากการตรวจสอบความขึ้นแย้งปรากฏขึ้นในส่วนของ ACE Entailment และความรู้ที่ได้จากส่วนนี้สามารถนำไปแปลงให้อยู่ในรูปแบบของกฎความสัมพันธ์ได้เพื่อนำไปเพิ่มในชุดกฎความสัมพันธ์เดิม



รูปที่ ก.5 แสดงตัวอย่างการแทนความรู้และตรวจสอบความขัดแย้งของกฎความสัมพันธ์

ภาคผนวก ข

รหัสต้นฉบับของโปรแกรม

โปรแกรมการรวมกฎความสัมพันธ์แบบกระจาย

```
import re

def read_names(filename):
    with open(filename, 'r') as fileopen:
        name_list = [line.strip() for line in fileopen]
        return name_list

def multiple_replace(dict, text):
  regex = re.compile("(%s)" % "|".join(map(re.escape, dict.keys())))

  return regex.sub(lambda mo: dict[mo.string[mo.start():mo.end()]], text)

r1 = read_names('t1.txt')
r2 = read_names('t2.txt')
r3 = read_names('t3.txt')

new_list = [r1, r2, r3]
result = set(new_list[0]).intersection(*new_list)

my_pattern1 = '\A{'
myList = []

for member in result:
    print member
    s1 = re.sub(my_pattern1,'if X is a n:',member)
    dict = {
            '=>' : 'then X is a',
            '=' : '_equal_',
```

```
            ',' : ' and X is a n:',
            '{' : 'n:',
            '}' : '',
    }
    s2 = multiple_replace(dict, s1)
    print s2
    myList.append(s2)

f = open("rusult_t.txt", "w")
f.write("\n".join(map(lambda x: str(x), myList)))
f.close()
```

ภาคผนวก ค

บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างการศึกษา

รายชื่อบทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างการศึกษา

นันทวุฒิ คะอังกุ, กิตติศักดิ์ เกิดประสพ, นิตยา เกิดประสพ. 2557. **วิธีหากฎความสัมพันธ์จากข้อมูลหลายชุด.** ในงานประชุมวิชาการระดับชาติด้านเทคโนโลยีสารสนเทศ ครั้งที่ 6. 27 - 28 กุมภาพันธ์ 2557

นันทวุฒิ คะอังกุ, กิตติศักดิ์ เกิดประสพ, นิตยา เกิดประสพ. 2557. **การหากฎความสัมพันธ์จากข้อมูลแบบกระจาย.** ในการประชุมวิชาการระดับชาติมหาวิทยาลัยเทคโนโลยีราชมงคล ครั้งที่ 6. มหาวิทยาลัยเทคโนโลยีราชมงคลสุวรรณภูมิ ศูนย์หันตรา. 23 - 25 กรกฎาคม 2557

Nuntawut Kaoungku, Kittisak Kerdprasop and Nittaya Kerdprasop. 2014. **A Technique to Association Rule Mining on Multiple Datasets**. Journal of Advances in Information Technology, 5.2 (2014): 53-57.

การประชุมวิชาการระดับประเทศด้านเทคโนโลยีสารสนเทศ (National Conference on Information Technology: NCIT) ครั้งที่ 6

วิธีการหากฎความสัมพันธ์จากข้อมูลหลายชุด

นันทวุฒิ คะอังกุ, กิตติศักดิ์ เกิดประสพ, นิตยา เกิดประสพ

สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
Emails: b5111299@gmail.com

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาและพัฒนาวิธีการหากฎความสัมพันธ์จากข้อมูลหลายชุด ซึ่งในปัจจุบันด้วยเทคโนโลยีและระบบสารสนเทศต่าง ๆ ทำให้หน่วยงานหรือองค์กรมีการจัดเก็บข้อมูลอย่างมีระบบแต่ปัญหาที่เพิ่มขึ้นมาคือข้อมูลเหล่านั้นมีขนาดที่ใหญ่ขึ้นซึ่งเป็นเรื่องยากในการหากฎความสัมพันธ์ เนื่องจากจำเป็นต้องใช้คอมพิวเตอร์ที่มีประสิทธิภาพสูงในการประมวลผลซึ่งตามมาด้วยค่าใช้จ่ายที่เพิ่มมากขึ้นนั้นเอง แต่มีวิธีที่จะสามารถช่วยแก้ไขปัญหาตรงนี้ได้คือการกระจายข้อมูลไปประมวลผลตามเครื่องคอมพิวเตอร์หลาย ๆ เครื่อง แล้วนำกฎความสัมพันธ์ที่ได้จากแต่ละเครื่องมารวมกันเพื่อให้ได้กฎความสัมพันธ์ที่มีประสิทธิภาพใกล้เคียงกับวิธีการหากฎความสัมพันธ์จากข้อมูลเพียงชุดเดียว ดังนั้นผู้วิจัยจึงทำการศึกษาและพัฒนาวิธีการหากฎความสัมพันธ์จากข้อมูลหลายชุด

คำสำคัญ-- การหากฎความสัมพันธ์; ข้อมูลหลายชุด; ภาษาควบคุม

1. บทนำ

ปัจจุบันด้วยเทคโนโลยีที่พัฒนาอย่างรวดเร็วทำให้หน่วยงานหรือองค์การต่าง ๆ ได้นำเทคโนโลยีมาประยุกต์ใช้กับงานของหน่วยงานหรือองค์การนั้น ๆ มากยิ่งขึ้น ด้วยเทคโนโลยีเหล่านี้ทำให้การเก็บข้อมูลนั้นเป็นไปได้ง่ายและเป็นระบบ แต่สิ่งที่ตามมาคือข้อมูลที่ถูกจัดเก็บไว้นั้นมีขนาดใหญ่ทำให้การค้นหากฎความสัมพันธ์จากข้อมูลเหล่านั้นเป็นไปได้ยาก เนื่องจากต้องใช้คอมพิวเตอร์ที่มีประสิทธิภาพสูงในการประมวลผลและมีค่าใช้จ่ายที่สูง ซึ่งทำให้มีเฉพาะหน่วยงานหรือองค์กรที่มีขนาดใหญ่เท่านั้นที่จะมีศักยภาพด้านการเงินในการค้นหากฎความสัมพันธ์จากข้อมูลที่มีขนาดใหญ่ได้ แต่ก็มีเทคนิคที่จะเข้ามาช่วยแก้ไขปัญหาตรงนี้ลงได้นั้นก็คือการกระจายข้อมูลไปประมวลผลตามเครื่องคอมพิวเตอร์หลาย ๆ เครื่อง โดยเครื่องคอมพิวเตอร์เหล่านั้นไม่จำเป็นต้องมีประสิทธิภาพสูงในการประมวลผลก็สามารถหากฎความสัมพันธ์ได้ แต่ในขั้นตอนของการรวมกฎความสัมพันธ์ที่ได้ในแต่ละเครื่องนั้นอาจมีโอกาสที่กฎความสัมพันธ์เหล่านั้นจะขัดแย้งกันเอง และกฎความสัมพันธ์ที่ได้อาจจะไม่มีประสิทธิภาพเมื่อเปรียบเทียบกับการหากฎความสัมพันธ์จากข้อมูลเพียงชุดเดียว ดังนั้นในขั้นตอนของการรวมกฎความสัมพันธ์นั้นจำเป็นต้องมีเทคนิคมาช่วยจัดการในปัญหาที่กล่าวมาข้างต้น หากต้องการที่จะหากฎความสัมพันธ์จากข้อมูลหลายชุด

ขั้นตอนในการรวมกฎความสัมพันธ์จากแหล่งต่าง ๆ นั้นมีความสำคัญเป็นอย่างมาก เนื่องจากจะต้องให้ได้กฎความสัมพันธ์ที่มีประสิทธิภาพใกล้เคียงกับการหากฎความสัมพันธ์จากข้อมูลเพียงชุดเดียวมากที่สุด และกฎความสัมพันธ์ที่ได้นั้นจะต้องไม่ขัดแย้งกันเอง ซึ่งการตรวจความขัดแย้งของกฎความสัมพันธ์นั้นจะใช้ Fact++ Reasoner ซึ่งอยู่ในโปรแกรม Protégé และจำเป็นต้องเขียนกฎความสัมพันธ์ให้อยู่ในรูปแบบของ Attempto Controlled English (ACE) [1] ซึ่งเป็น Controlled Natural Language (CNL) อยู่บนโปรแกรม Protégé

งานวิจัยทางด้านการหากฎความสัมพันธ์จากข้อมูลหลายชุดยังปรากฏอยู่น้อยมาก ซึ่งส่วนใหญ่จะเป็นการนำไปใช้กับอัลกอริทึมอื่น ๆ ทางด้านการทำเหมืองข้อมูล อาจเนื่องจากการหากฎความสัมพันธ์จากข้อมูลหลายชุดนั้นมีความยุ่งยากในขั้นตอนของการรวมกฎความสัมพันธ์จากแหล่งต่าง ๆ

เพื่อให้ได้กฎความสัมพันธ์ที่มีประสิทธิภาพใกล้เคียงกับการหากฎความสัมพันธ์จากข้อมูลเพียงชุดเดียวมากที่สุด โดยงานวิจัยที่ปรากฏอยู่นั้นก็ไม่ได้มีการเปรียบเทียบประสิทธิภาพให้เห็นอย่างชัดเจน [2, 3, 4]

จากที่กล่าวมาข้างต้นจะเห็นได้ว่าการที่จะหากฎความสัมพันธ์จากข้อมูลที่มีขนาดใหญ่นั้นทำได้ยาก ทำให้มีความจำเป็นต้องกระจายข้อมูลไปประมวลผลตามเครื่องคอมพิวเตอร์หลาย ๆ เครื่อง แต่การที่จะนำกฎความสัมพันธ์ที่ได้จากแต่ละเครื่องนั้นอาจจะเกิดปัญหาในส่วนของความขัดแย้งกันเองของกฎความสัมพันธ์และประสิทธิภาพของกฎความสัมพันธ์ ดังนั้นงานวิจัยนี้จึงได้เสนอวิธีการหากฎความสัมพันธ์จากข้อมูลหลายชุด

2. การหากฎความสัมพันธ์

การหากฎความสัมพันธ์ (Association Rule Mining) เป็นกระบวนการหนึ่งที่ได้รับความนิยมในการหาความสัมพันธ์ระหว่างข้อมูล ซึ่งมีวิธีในการหากฎความสัมพันธ์ด้วยกันหลากหลายวิธี ในงานวิจัยนี้ใช้อัลกอริทึม Apriori [5] ในการหากฎความสัมพันธ์ วิธีการแบ่งช่วงข้อมูลและการหากฎความสัมพันธ์อธิบายด้วยข้อมูลตัวอย่างตามตารางที่ 1

ตารางที่ 1. รายการซื้อสินค้าของลูกค้าทั้งหมด

รายการสินค้า	นม	น้ำ	ขนม	ไส้กรอก
1	1	1	0	0
2	0	1	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0

ตารางที่ 2. ความถี่ของการซื้อสินค้าของลูกค้า เพื่อหาความสัมพันธ์ของสินค้าแต่ละอย่าง

	นม	น้ำ	ขนม	ไส้กรอก
นม	2*	2	1	0
น้ำ	2	4*	2	0
ขนม	1	1	2*	0
ไส้กรอก	0	0	0	1*

จากตารางที่ 1 เป็นข้อมูลรายการซื้อสินค้าของลูกค้า แล้วนำไปผ่านการหาความถี่ของการซื้อสินค้าของลูกค้าในแต่ละชิ้นของสินค้า เพื่อหาความสัมพันธ์ของสินค้าแต่ละอย่างซึ่งจะแสดงได้ดังตารางที่ 2 หลังจากนั้นก็จะนำสินค้าที่มีความถี่สูงไปสร้างกฎความสัมพันธ์ ซึ่งอยู่ในรูปแบบของ IF condition Then result โดยเกณฑ์ที่ใช้ในการหากฎนั้น มีดังนี้

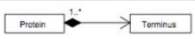
- Support เป็นค่าที่บอกว่าความถี่ที่เกิดขึ้นบ่อยมากน้อยแค่ไหน
- Confidence เป็นค่าที่บอกว่าโอกาสที่จะเกิดขึ้น เช่น ถ้ามี condition เกิดขึ้น โอกาสที่จะเกิด result มีมากน้อยแค่ไหน

3. Attempto Controlled English

ACE เป็นภาษาควบคุม (controlled natural language) ที่มีพื้นฐานมาจาก First-order logic language ซึ่งเป็นการรวมเอาข้อดีของ Formal language และ Natural languages เพื่อต้องการที่จะให้ภาษาที่ใช้เขียนอยู่ในรูปแบบที่มนุษย์และคอมพิวเตอร์เข้าใจได้ โดยสามารถเขียนอยู่ในรูปของประโยคภาษาอังกฤษอย่างง่ายได้ [2] ดังแสดงตามรูปที่ 1 เป็นตัวอย่างการเปรียบเทียบระหว่าง FOL, DL, OWL, UML และ ACE ทำให้ผู้ใช้ไม่จำเป็นต้องมีความรู้ทางด้านคอมพิวเตอร์ก็สามารถที่จะใช้ประโยคภาษาอังกฤษอย่างง่ายในการนำไปประยุกต์ใช้กับงานด้านต่าง ๆ ได้ เช่น Semantic web, Expert system เป็นต้น ACE จะเป็น Plugin ของโปรแกรม Protégé [6] ซึ่งในงานวิจัยนี้จะเขียนกฎความสัมพันธ์ให้อยู่ในรูปแบบของ ACE เนื่องจากจะนำกฎความสัมพันธ์ไปตรวจสอบความขัดแย้งด้วย Reasoner และต้องการเปรียบเทียบประสิทธิภาพระหว่างกฎความสัมพันธ์จากข้อมูลหลายชุดและการหากฎความสัมพันธ์จากข้อมูลเพียงชุดเดียว ดังตารางที่ 3 แสดงตัวอย่างการแปลงกฎความสัมพันธ์ให้อยู่ในรูปแบบของ ACE

first-order logic	$\forall X(protein(X) \rightarrow \exists Y(terminus(Y) \wedge has(X,Y)))$
DL	$Protein \sqsubseteq \exists has.Terminus$

OWL (RDF/XML)	`<owl:Class rdf:ID="Protein">`
	`<rdfs:subClassOf>`
	`<owl:Restriction>`
	`<owl:onProperty rdf:resource="#has"/>`
	`<owl:someValuesFrom rdf:resource="#Terminus"/>`
	`</owl:Restriction>`
	`</rdfs:subClassOf>`
	`</owl:Class>`

UML	
ACE	Every protein has a terminus.

รูปที่ 1. ตัวอย่างการเปรียบเทียบระหว่าง FOL, DL, OWL, UML และ ACE [7]

ตารางที่ 3. ตัวอย่างการแปลงกฎความสัมพันธ์ให้อยู่ในรูปแบบของ ACE

Association rules	*Association rules in ACE*
{CLASS=crew} => {SEX=male}	If X is a ***crew*** then X is a ***male***.
{CLASS=crew, AGE=adult} => {SEX=male}	If X is a ***crew*** and X is an ***adult*** then X is a ***male***.
{CLASS=crew, AGE=adult, SURVIVED=no} => {SEX=male}	If X is a ***crew*** and X is an ***adult*** and X is a n: ***not-survivor*** then X is a ***male***.

4. FaCT++ Reasoner

FaCT++ เป็น Reasoner ถูกพัฒนามาจากอัลกอริทึม FaCT โดยใช้ภาษา C++ ในการพัฒนา ซึ่งมีพื้นฐานมาจาก Description Logics (DL) เพื่อใช้สำหรับการตรวจสอบความไม่สอดคล้องกันจาก Ontology ที่ถูกสร้างขึ้นมา [8] ตัวอย่างเช่น

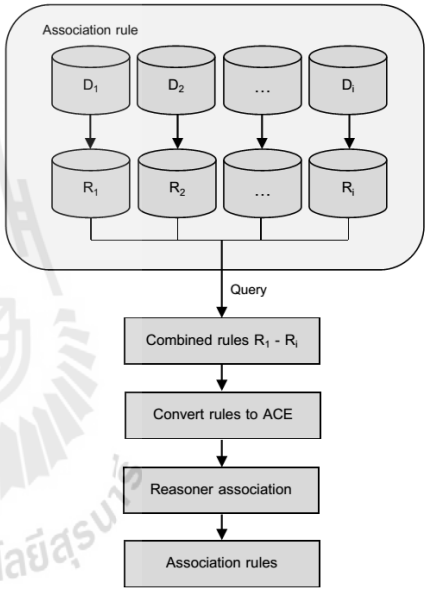
Every man is a human.
John is a man.
John is not a human.

จากตัวอย่างจะเห็นได้ว่าเกิดความขัดแย้งกันขึ้นในประโยคที่ John is not a human เนื่องจาก 2 ประโยคก่อนหน้านี้ได้บอกไปแล้วว่า John is a human ทำให้ตัวอย่างนี้เกิดความขัดแย้งกันเอง ทำให้ไม่สามารถนำไปสร้างเป็น Ontology ได้ ซึ่งในงานวิจัยนี้เนื่องจากการหากฎความสัมพันธ์จะหาจากแหล่งที่แตกต่างกันออกไป อาจทำให้กฎความสัมพันธ์นั้นเกิดความขัดแย้งกันเอง ดังนั้นจึงจำเป็นต้องใช้ Reasoner มาช่วยจัดการตรวจสอบความขัดแย้งกันเองของการหากฎความสัมพันธ์จากข้อมูลหลายชุด

5. การหากฎความสัมพันธ์จากข้อมูลหลายชุด

ผู้วิจัยมีกรอบแนวคิดในการหากฎความสัมพันธ์จากข้อมูลหลายชุด โดยข้อมูลจะถูกแบ่งออกไปตามแหล่งต่าง ๆ เพื่อช่วยในการประมวลผลหากฎความสัมพันธ์ แทนที่การหากฎความสัมพันธ์จากข้อมูลเพียงชุดเดียวที่มีขนาดใหญ่ซึ่งจำเป็นต้องใช้คอมพิวเตอร์ที่มีประสิทธิภาพสูงในการประมวลผล แต่การหากฎความสัมพันธ์จากข้อมูลหลายชุดนั้นอาจทำให้กฎความสัมพันธ์ที่ได้เกิดความขัดแย้งกัน ดังนั้นงานวิจัยนี้จึงมีกรอบแนวคิดในการหากฎความสัมพันธ์จากข้อมูลหลายชุด

จากรูปที่ 2 แสดงขั้นตอนการหากฎความสัมพันธ์จากข้อมูลหลายชุด ซึ่งขั้นตอนแรกจะหากฎความสัมพันธ์จากข้อมูล $D_1, D_2, \ldots, D_i$ โดย $i = 1, 2, \ldots, n$ ขั้นที่สองจะรวมกฎความสัมพันธ์ที่ได้จากขั้นตอนแรก $R = R_1 \cap R_2 \cap \ldots \cap R_i$ โดย $i = 1, 2, \ldots, n$ ดังรูปที่ 3 ขั้นตอนที่สามจะแปลงกฎความสัมพันธ์ให้อยู่ในรูปแบบ ACE ขั้นตอนที่สี่จะตรวจสอบความขัดแย้งของกฎความสัมพันธ์ด้วย FaCT++ Reasoner และขั้นตอนสุดท้ายจะได้กฎความสัมพันธ์จากข้อมูลหลายชุดที่มีประสิทธิภาพใกล้เคียงกับกฎความสัมพันธ์จากข้อมูลชุดเดียว ซึ่งสามารถตรวจสอบได้จาก Ontology ที่สร้างจากโปรแกรม Protégé



รูปที่ 2. แสดงขั้นตอนการหากฎความสัมพันธ์จากข้อมูลหลายชุด

R_1	R_2	R_3
{CLASS=third} => {AGE=adult}		
{CLASS=crew} => {SEX=male}	{CLASS=crew} =>{SEX=male}	{CLASS=crew} => {SEX=male}
{CLASS=crew} => {AGE=adult}	{CLASS=crew} =>{AGE=adult}	{CLASS=crew} => {AGE=adult}

R
{CLASS=crew} => {SEX=male}
{CLASS=crew} => {AGE=adult}

รูปที่ 3. ตัวอย่างการรวมกฎความสัมพันธ์จากข้อมูลหลายชุด

ตารางที่ 4. ตัวอย่างข้อมูลผู้ป่วยมะเร็งเต้านม

age	*menopause*	*tumor-size*	*inv-nodes*	*node-caps*	*deg-malig*	*breast*	*breast-quad*	*irradiat*	*Class*
40-49	premeno	15-19	0-2	yes	3	right	left_up	no	recurrence-events
50-59	ge40	15-19	0-2	no	1	right	central	no	no-recurrence-events
50-59	ge40	35-39	0-2	no	2	left	left_low	no	recurrence-events
40-49	premeno	35-39	0-2	yes	3	right	left_low	yes	no-recurrence-events
40-49	premeno	30-34	3-5	yes	2	left	right_up	no	recurrence-events

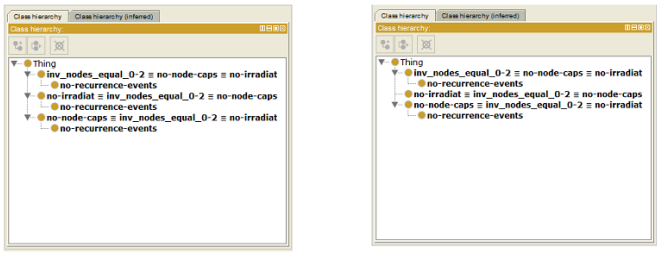
ตารางที่ 5. Entailment จากการหากฎความสัมพันธ์ด้วย Minimum support 0.3

Entailment	*ACE If-then*
Every inv_nodes_equal_0-2 is a no-irradiat that is a no-node-caps .	If X is a n:inv_nodes_equal_0-2 and X is a n:no-irradiat then X is a n:no-node-caps
Every no-irradiat is an inv_nodes_equal_0-2 that is a no-node-caps .	If X is a n:no-irradiat and X is a n:inv_nodes_equal_0-2 then X is a n:no-node-caps
Every no-node-caps is an inv_nodes_equal_0-2 that is a no-irradiat .	If X is a n:no-node-caps and X is a n:inv_nodes_equal_0-2 then X is a n:no-irradiat

ตารางที่ 6. Entailment จากการหากฎความสัมพันธ์ด้วย Minimum support 0.5

Entailment	*ACE If-then*
Every no-recurrence-events is a no-irradiat .	If X is a n:no-recurrence-events then X is a n:no-irradiat .
Every inv_nodes_equal_0-2 is a no-irradiat that is a no-node-caps .	If X is a n: inv_nodes_equal_0-2 and X is a n: no-irradiat then X is a n:no-node-caps .
Every no-irradiat is an inv_nodes_equal_0-2 that is a no-node-caps .	If X is a n:no-irradiat and X is a n:inv_nodes_equal_0-2 then X is a n:no-node-caps
Every no-node-caps is an inv_nodes_equal_0-2 that is a no-irradiat .	If X is a n:no-node-caps and X is a n:inv_nodes_equal_0-2 then X is a n:no-irradiat

รูปที่ 4. ตัวอย่าง Ontology การหากฎความสัมพันธ์จากข้อมูลชุดเดียว (a) และจากข้อมูลหลายชุด (b) Minimum support 0.3



รูปที่ 5. แสดง Ontology การหากฎความสัมพันธ์จากข้อมูลชุดเดียว (a) และจากข้อมูลหลายชุด (b) Minimum support 0.5

6. ผลการวิจัย

งานวิจัยนี้ได้ทำการทดลองเปรียบเทียบผลลัพธ์ที่ได้จากการหากฎความสัมพันธ์จากข้อมูลชุดเดียวกับข้อมูลหลายชุด โดยชุดข้อมูลที่ใช้เป็นชุดข้อมูลของผู้ป่วยมะเร็งเต้านม (Breast cancer) จาก UCI Machine Learning Repository ข้อมูลนี้มี 286 เรคคอร์ด 10 แอททริบิวต์ จากตารางที่ 4 เป็นตัวอย่างข้อมูลของผู้ป่วยโรงมะเร็งจำนวน 5 เรคคอร์ด

การทดลองจะแบ่งข้อมูลผู้ป่วยมะเร็งเต้านมสำหรับหากฎความสัมพันธ์ออกเป็น 3 ชุด โดยใช้ Minimum support ที่ 0.3 และ 0.5 จากตารางที่ 7 แสดงการเปรียบเทียบจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์จากข้อมูลชุดเดียวกับข้อมูลหลายชุดด้วย Minimum support 0.3 จะเห็นได้ว่ากฎความสัมพันธ์จากข้อมูลหลายชุดยังขาดอยู่มาก และจากรูปที่ 4 เมื่อพิจารณา Ontology จะเห็นได้ว่ากฎความสัมพันธ์จากข้อมูลหลายชุดกับการหากฎความสัมพันธ์จากข้อมูลชุดเดียวมีลักษณะที่แตกต่างกันอย่างชัดเจน และจำนวนกฎความสัมพันธ์ระหว่างการหากฎความสัมพันธ์จากข้อมูลชุดเดียวกับข้อมูลหลายชุดด้วย Minimum support 0.5 จะเห็นได้ว่ามีเพียงบางกฎความสัมพันธ์เท่านั้นที่ยังขาดอยู่ และจากรูปที่ 5 เมื่อพิจารณา Ontology จะเห็นว่ากฎความสัมพันธ์จากข้อมูลหลายชุดมีประสิทธิภาพใกล้เคียงกับการหากฎความสัมพันธ์จากข้อมูลชุดเดียว

จากการหากฎความสัมพันธ์ด้วย Minimum support 0.3 และ 0.5 ตามลำดับ ซึ่งกฎความสัมพันธ์จากข้อมูลหลายชุดที่ได้มานั้นยังมีส่วนที่ขาดอยู่เมื่อเทียบกับกฎความสัมพันธ์จากข้อมูลเพียงชุดเดียว จากตารางที่ 7 จะเห็นได้ว่ากฎความสัมพันธ์จากข้อมูลหลายชุดที่มีค่า Minimum support 0.5 จะให้จำนวนกฎความสัมพันธ์ใกล้เคียงกับจำนวนกฎความสัมพันธ์จากข้อมูลเพียงชุดเดียวมากกว่ากฎความสัมพันธ์จากข้อมูลหลายชุดที่มีค่า Minimum support 0.3 และในส่วนกฎความสัมพันธ์จากข้อมูลหลายชุดที่ได้มานั้นยังมีส่วนที่ขาดอยู่นั้นสามารถนำบางกฎความสัมพันธ์จากตารางที่ 5 และ 6 ซึ่งเป็นกฎความสัมพันธ์ที่ได้จากการ Entailment ในขั้นตอนการ Reasoner สามารถนำไปแทนในบางกฎความสัมพันธ์ที่ขาดหายไปได้

ตารางที่ 7. แสดงการเปรียบเทียบจำนวนกฎความสัมพันธ์จากข้อมูลหลายชุดและจำนวนกฎความสัมพันธ์จากข้อมูลชุดเดียว

Minimum support	จำนวนกฎจากข้อมูลชุดเดียว	จำนวนกฎจากข้อมูลหลายชุด
0.3	62	37
0.5	24	19

7. สรุปและอภิปรายผล

การหากฎความสัมพันธ์จากข้อมูลที่มีขนาดใหญ่นั้นจำเป็นต้องใช้คอมพิวเตอร์ที่มีประสิทธิภาพสูงในการประมวลผลเพื่อหากฎความสัมพันธ์ ซึ่งต้องใช้ค่าใช้จ่ายที่สูง การกระจายข้อมูลออกไปประมวลตามเครื่องคอมพิวเตอร์ทั่วไปเป็นทางเลือกหนึ่งที่จะช่วยแก้ไขปัญหาตรงนี้ได้ โดยขั้นตอนการรวมกฎความสัมพันธ์ในแต่ละแหล่งนั้นจะนำมาเฉพาะกฎความสัมพันธ์ที่เหมือนเท่านั้น และจะนำตรวจสอบความขัดแย้งของกฎความสัมพันธ์ เนื่องจากกฎความสัมพันธ์มาจากแหล่งที่แตกต่างกันอาจทำให้กฎความสัมพันธ์ที่ได้เมื่อนำมารวมกันเกิดความขัดแย้งกันเอง

จากการทดลองดังกล่าวจะเห็นได้ว่าการหากฎความสัมพันธ์ด้วยค่า Minimum support ที่มาจากข้อมูลหลายชุดนั้นมีประสิทธิภาพใกล้เคียงกับการหากฎความสัมพันธ์จากข้อมูลเพียงชุดเดียว ซึ่งพิจารณาได้จากการนำกฎความสัมพันธ์ไปสร้างเป็น Ontology และกฎความสัมพันธ์ที่ได้นั้นไม่ขัดแย้งกันเอง แต่ยังมีบางกฎความสัมพันธ์ที่ยังขาดหายไปเมื่อเปรียบเทียบกับกฎความสัมพันธ์จากข้อมูลเพียงชุดเดียว ซึ่งสามารถนำผลที่ได้จาก Reasoner มาเติมในส่วนของบางกฎความสัมพันธ์ที่ยังขาดหายไปได้

เอกสารอ้างอิง

[1] Fuchs, Norbert E., Kaarel Kaljurand, and Tobias Kuhn (2008). Attempto controlled english for knowledge representation. Reasoning Web. Springer Berlin Heidelberg, pp.104-124.

[2] Domingos, Pedro (2003). Prospects and challenges for multi-relational data mining. ACM SIGKDD explorations newsletter, vol.5, no.1, pp.80-83.

[3] Han, Jiawei, and Yongjian Fu (1999). Mining multiple-level association rules in large databases. Knowledge and Data Engineering, IEEE Transactions, vol.11, no.5, pp.798-805.

[4] Zhao Yanchang, et al (2007). Mining for combined association rules on multiple datasets. Proceedings of the 2007 international workshop on Domain driven data mining, pp.18-23.

[5] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami (1993). Mining Association Rules between Sets of Items in Large Database. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207-216

[6] Kaljurand, Kaarel (2008). ACE View-An Ontology and Rule Editor based on Controlled English. International Semantic Web Conference (Posters & Demos). vol. 401.

[7] Norbert E. Fuchs and Kaarel Kaljurand (2006). Attempto Controlled English: Language, Tools and Applications [Online]. Available URL: http://attempto.ifi.uzh.ch/site/courses/files/ACE.Course.UniZH.1206.Getting_Started.pdf

[8] Tsarkov, Dmitry, and Ian Horrocks (2006). FaCT++ description logic reasoner: System description. Automated reasoning. Springer Berlin Heidelberg, pp.292-297.

การหากฎความสัมพันธ์จากข้อมูลแบบกระจาย
DISTRIBUTED ASSOCIATION RULE MINING

นันทวุฒิ คะอังกุ*, กิตติศักดิ์ เกิดประสพ, นิตยา เกิดประสพ
Nuntawut Kaoungku*, Kittisak Kerdprasop, Nittaya Kerdprasop

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อเสนอผลการศึกษาเบื้องต้นและเสนอกรอบแนวคิดเกี่ยวกับการหากฎความสัมพันธ์จากแหล่งข้อมูลที่กระจายกัน เนื่องจากปัจจุบันเทคโนโลยีได้เข้ามามีบทบาทในการจัดเก็บข้อมูลตามหน่วยงานหรือองค์กรต่าง ๆ มากขึ้น ทำให้การจัดเก็บข้อมูลนั้นสามารถทำได้ง่ายและเป็นระบบ แต่ด้วยเทคโนโลยีที่เข้ามามีบทบาทนี้ทำให้ข้อมูลถูกกระจายกันอยู่ตามแหล่งต่าง ๆ ไม่ได้ถูกจัดเก็บเข้ามาไว้ในที่เดียว เช่น ข้อมูลทางการแพทย์ของโรงพยาบาลแต่ละแห่ง เป็นต้น ทำให้การหากฎความสัมพันธ์เพื่อให้ได้ฐานความรู้เพียงหนึ่งเดียวของข้อมูลเหล่านั้นทำได้ยาก เนื่องจากการหากฎความสัมพันธ์โดยทั่วไปแล้วเป็นการหากฎความสัมพันธ์จากข้อมูลเพียงชุดเดียวในลักษณะรวมศูนย์ ดังนั้นงานวิจัยนี้ต้องการศึกษาและเสนอกรอบแนวคิดเกี่ยวกับการหากฎความสัมพันธ์จากข้อมูลที่กระจายกัน

คำสำคัญ การหากฎความสัมพันธ์, การทำเหมืองข้อมูลแบบกระจาย

Abstract

The aims of this paper are to present the preliminary study results and to propose a conceptual framework for distributed association rule mining. Current technology plays an important role in data collection among modern agencies and organizations. With the advancement of computer and internet technologies, data are stored in several places such as the medical information of each hospital. This situation has made association rule mining a difficult task because traditional mining method has been designed for centralized analysis. Therefore, this paper intends to solve the distributed association rule mining problem by proposing a framework for this distributive situation.

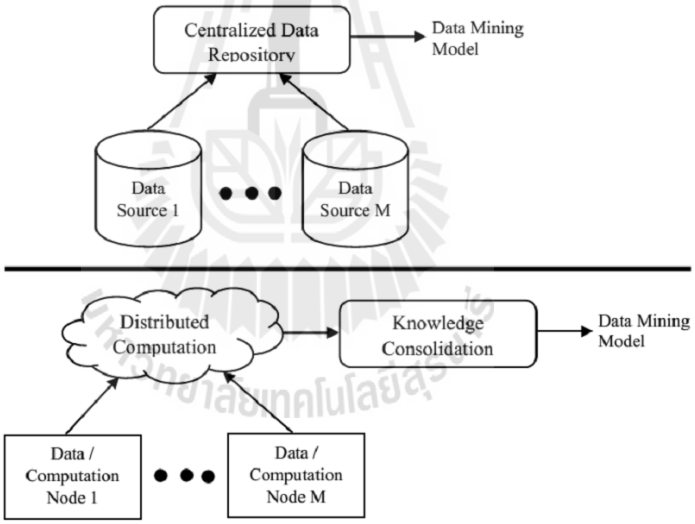
Keywords: Association rule mining, Distributed data mining

สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
School of Computer Engineering, Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima, 30000.
*Corresponding author. E-mail: b5111299@gmail.com

1. บทนำ

ปัจจุบันด้วยเทคโนโลยีที่พัฒนาอย่างรวดเร็วทำให้การจัดเก็บข้อมูลนั้นเป็นเรื่องที่สามารถทำได้ง่าย ซึ่งจะเห็นได้ตามหน่วยงานหรือองค์กรต่าง ๆ ส่วนใหญ่จะใช้คอมพิวเตอร์สำหรับการจัดเก็บข้อมูล แต่ด้วยการจัดเก็บข้อมูลที่สามารถทำได้ง่ายนี้เองเป็นสาเหตุทำให้ข้อมูลที่ถูกจัดเก็บนั้นมีขนาดใหญ่หรือข้อมูลกระจายอยู่ตามแหล่งต่าง ๆ เช่น ข้อมูลทางการแพทย์ที่กระจายอยู่ตามโรงพยาบาลต่าง ๆ เป็นต้น ซึ่งทำให้เป็นเรื่องที่ยากในการสกัดความรู้จากแหล่งข้อมูลเหล่านั้น เนื่องจากต้องใช้คอมพิวเตอร์ที่มีประสิทธิภาพสูงในการประมวลผลข้อมูลที่มีขนาดใหญ่ หรือการสกัดความรู้จากข้อมูลที่กระจายกันอยู่เพื่อให้ได้ฐานความรู้เพียงหนึ่งเดียวนั้นทำได้ยาก [7]

ด้วยเหตุผลที่กล่าวมาข้างต้นทำให้การทำเหมืองข้อมูล (Data Mining) นั้นไม่ได้อยู่ในขอบเขตเดิมที่เป็นการสกัดความรู้จากข้อมูลเพียงชุดเดียวและมีขนาดของข้อมูลที่ไม่ใหญ่มากนัก แต่จะเป็นการหาความรู้จากข้อมูลที่กระจายอยู่ตามแหล่งต่าง ๆ หรือข้อมูลที่มีด้วยกันหลาย ๆ ชุด ตามรูปที่ 1 จะเห็นได้ว่าการการทำเหมืองข้อมูลจากข้อมูลที่กระจายกันอยู่นั้นจะแตกต่างจากการทำเหมืองข้อมูลแบบเดิมที่จำเป็นต้องรวมข้อมูลให้อยู่ที่เดียวกันก่อนถึงจะสามารถทำได้



รูปที่ 1 ตัวอย่างการทำเหมืองข้อมูลแบบเดิมและการทำเหมืองข้อมูลที่กระจายกันอยู่

2. วิธีดำเนินการวิจัย

งานวิจัยนี้เป็นการเสนอผลการศึกษาเบื้องต้นและเสนอกรอบแนวคิดเกี่ยวกับการหากฎความสัมพันธ์จากข้อมูลที่กระจายกัน ซึ่งจะแบ่งส่วนของการดำเนินงานออกเป็น 2 ส่วนคือ ศึกษาการหากฎความสัมพันธ์และศึกษาการหากฎความสัมพันธ์แบบกระจาย

2.1 การหากฎความสัมพันธ์

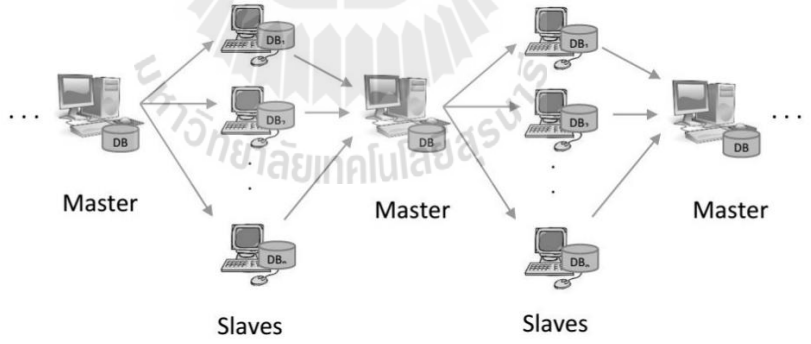
การหากฎความสัมพันธ์ (Association rule mining) คือ ความสัมพันธ์ของเหตุการณ์หรือวัตถุที่เกิดขึ้นร่วมกันหรือพร้อมกัน หรือเพื่อหารูปแบบที่เกิดขึ้นบ่อย (Frequent Pattern) เพื่อนำไปใช้ในการวิเคราะห์ หรือทำนายปรากฏการณ์ต่าง ๆ ซึ่งการหากฎความสัมพันธ์นั้นสามารถนำไปใช้งานได้หลายรูปแบบ เช่น การวิเคราะห์พฤติกรรมซื้อสิ้นค้า ถ้าลูกค้าซื้อสินค้า A แล้วมักจะซื้อ B ตามไปด้วย เป็นต้น โดยจะอยู่ในรูปแบบ **IF** condition **Then** result [5] เกณฑ์ในการคัดเลือกความสัมพันธ์จะพิจารณาจาก

- ค่าสนับสนุน (Support) คือ ค่าที่บอกว่าความถี่ของความสัมพันธ์ของเหตุการณ์ A และ B ที่เกิดขึ้นบ่อยมากน้อยแค่ไหน สามารถหาได้จาก

  $Support\{A \rightarrow B\} = P(A \wedge B) =$ ความน่าจะเป็นที่เกิดเหตุการณ์ A และ B

- ค่าความเชื่อมั่น (Confidence) คือ ค่าที่บอกว่าโอกาสที่จะเกิดขึ้นมีมากน้อยเพียงใด เช่น ถ้ามี condition A เกิดขึ้น โอกาสที่จะเกิด result B มีมากน้อยแค่ไหน สามารถหาได้จาก

$$Confidence\{A \rightarrow B\} = \frac{\text{ความถี่ของ } A \text{ และ } B}{\text{ความถี่ของ } A}$$



รูปที่ 2 ตัวอย่างการหากฎความสัมพันธ์ด้วยอัลกอริทึม Apriori แบบขนาน

2.2 การหากฎความสัมพันธ์แบบกระจาย

การหาความสัมพันธ์ของเหตุการณ์หรือวัตถุที่เกิดขึ้นร่วมกันหรือพร้อมกันจากข้อมูลที่มีขนาดใหญ่หรือข้อมูลที่กระจายกันอยู่นั้นเป็นไปได้ค่อนข้างยาก เนื่องจากการหาความสัมพันธ์แบบเดิมนั้นถูกพัฒนามาไว้สำหรับการหาความสัมพันธ์จากข้อมูลเพียงชุดเดียว ทำให้จำเป็นต้องรวมข้อมูลจากหลาย ๆ แหล่งมารวมเป็นข้อมูลเพียงชุดเดียว แต่ข้อมูลที่ได้นั้นอาจมีขนาดใหญ่เกินไปสำหรับการนำไปประมวลผลหากฎความสัมพันธ์ ดังนั้นจึงทำให้มีเทคนิค 2 เทคนิคที่จะนำมาช่วยแก้ไขปัญหาในส่วนนี้

2.2.1 การหากฎความสัมพันธ์แบบขนาน (Parallel Association Rule Mining)

การประมวลผลทางด้านคอมพิวเตอร์ในสมัยก่อนจะเป็นในรูปแบบการทำงานแบบ Serial ที่มีเพียงเหตุการณ์เดียวเกิดขึ้นในเวลาเดียว ซึ่งทำให้ไม่เหมาะสำหรับการนำไปประมวลผลข้อมูลที่มีขนาดใหญ่ได้ แต่ด้วยปัจจุบันเทคโนโลยีทางด้านคอมพิวเตอร์ได้ถูกพัฒนาขึ้นมาเป็นอย่างมากที่จะสามารถแก้ไขปัญหาในจุดนี้ได้ และระบบคอมพิวเตอร์ดังกล่าวก็มีความสามารถในการประมวลผลในลักษณะที่เราเรียกว่า Parallel Processing คือเหตุการณ์มากกว่า 1 เหตุการณ์ เกิดขึ้นพร้อมกันในเวลาเดียว [3] *"ตัวอย่างเปรียบการทำงานของการทำงานแบบ Serial และ Parallel ถ้าต้องการสร้างบ้าน 1 หลังต้องใช้เวลา 1 เดือนและคนงานทั้งหมด 10 คนในการสร้าง แต่ถ้าต้องการสร้างบ้าน 10 หลังให้เสร็จภายใน 1 เดือนจะทำอย่างไร?* การทำงานแบบ *Parallel จะต้องจ้างคนงานเพิ่มมาเป็น 100 คน โดยให้คนงาน 10 คน สร้างบ้าน 1 หลัง"*

จากรูปที่ 2 แสดงตัวอย่างการการหากฎความสัมพันธ์ด้วยอัลกอริทึม Apriori แบบขนาน จะเห็นได้ว่าเริ่มต้น Master จะแบ่งข้อมูลออกเป็นชุด ๆ DB_1, DB_2, ... , DB_n ส่งไปให้ Slaves แต่ละตัวสำหรับการนำไปหาความถี่ หลังจากนั้นจะรวมความถี่ที่ได้จากแต่ละชุด Slaves มาไว้ที่ Master แล้วนำไปตัด item set ที่มีค่าน้อยกว่า minimum support ที่กำหนดไว้ แล้วทำการแบ่งข้อมูลออกเป็นชุด ๆ DB_1, DB_2, ... , DB_n ส่งไปให้ Slaves สำหรับการจับคู่ของแต่ละ Item set หลังจากนั้นจะรวมการจับคู่ของแต่ละ Item set ที่ได้จากแต่ละชุด Slaves มาไว้ที่ Master และจะกลับไปที่ขั้นตอนที่ 1 ทำแบบนี้ไปเรื่อย ๆ จนกว่า Item set ไม่เปลี่ยนแปลง [1]

2.2.2 การหากฎความสัมพันธ์แบบกระจายจากความคล้ายคลึง (Similarity Based Distributed Association Rule Mining)

การหากฎความสัมพันธ์โดยทั่วไปสามารถหาได้จากข้อมูลเพียงชุดเดียว แต่ด้วยข้อมูลบางประเภทที่ไม่ได้มีการจัดเก็บข้อมูลไว้ในแหล่งข้อมูลเพียงแหล่งเดียว ทำให้การนำข้อมูลเหล่านี้มาหากฎความสัมพันธ์นั้นจำเป็นต้องรวมข้อมูลที่ถูกกระจายกันอยู่ให้เป็นข้อมูลเพียงชุดเดียว เพื่อสามารถนำไปหากฎความสัมพันธ์ได้ แต่เนื่องจากข้อมูลที่กระจายกันอยู่นั้นมีการจัดเก็บข้อมูลที่มีลักษณะแตกต่างกันออกไป ซึ่งอาจทำให้การนำข้อมูลเหล่านี้มารวมกันแล้วนำไปหากฎความสัมพันธ์นั้นได้ประสิทธิภาพที่ไม่ดีพอสำหรับการนำไปทำนายผลในอนาคต ดังนั้นข้อมูลที่กระจายกันอยู่ตามแหล่งต่าง ๆ นั้นจะต้องมีลักษณะข้อมูลที่มีความ

คล้ายคลึงกัน โดยสามารถวัดความคล้ายคลึงกันของข้อมูลได้จากมาตรวัดความคล้ายคลึง (Similarity Measure) [2] ซึ่งสามารถหาได้จากสมการดังต่อไปนี้

$$Sim(A,B) = \frac{2I_3}{I_1 + I_2}$$

โดยกำหนดให้

$$I_1 = \sum_{i,j} \frac{|A_i \cap A_j|}{|A_i \cup A_j|} \log\left(1 + \frac{|A_i \cap A_j|}{|A_i \cup A_j|}\right) \min\{C_{Ai}, C_{Aj}\},$$

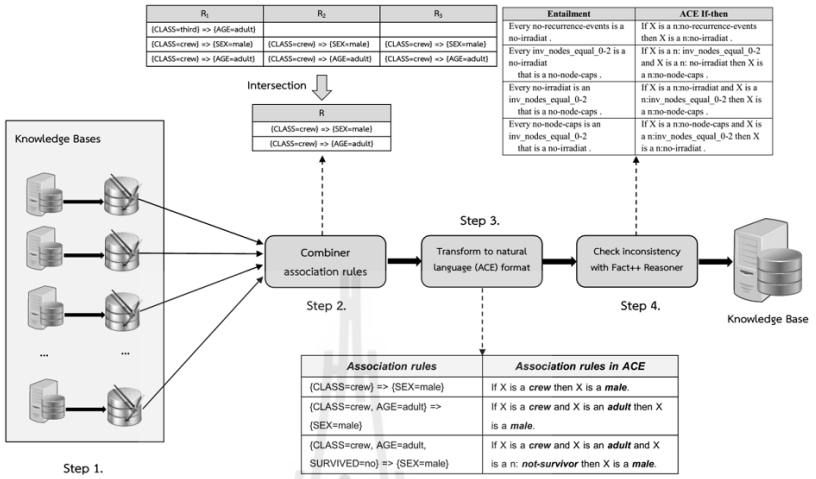
$$I_2 = \sum_{i,j} \frac{|B_i \cap B_j|}{|B_i \cup B_j|} \log\left(1 + \frac{|B_i \cap B_j|}{|B_i \cup B_j|}\right) \min\{C_{Bi}, C_{Bj}\},$$

$$I_3 = \sum_{i,j} \frac{|A_i \cap B_j|}{|A_i \cup B_j|} \log\left(1 + \frac{|A_i \cap B_j|}{|A_i \cup B_j|}\right) \min\{C_{Ai}, C_{Bj}\}$$

จากสมการจะเป็นการวัดความคล้ายคลึงของข้อมูล 2 ชุด คือ ข้อมูล A และ B ซึ่งค่าที่ได้ออกมาจะอยู่ระหว่าง 0 ถึง 1 โดยถ้ามีค่ามากแสดงว่าข้อมูล A และ B มีความคล้ายคลึงกันมาก แต่ถ้าค่าที่ได้ออกมามีค่าน้อยแสดงว่าข้อมูล A และ B มีความคล้ายคลึงกันน้อย

3. กรอบแนวคิด

ผู้วิจัยมีกรอบแนวคิดในการหากฎความสัมพันธ์แบบกระจาย โดยจากรูปที่ 3 มีข้อมูลที่อยู่ตามแหล่งข้อมูลต่าง ๆ หรือข้อมูลที่ถูกแบ่งออกไปหากฎความสัมพันธ์ตามแหล่งข้อมูลต่าง ๆ ซึ่งการหากฎความสัมพันธ์ของข้อมูลแต่ละชุดจะไม่ขึ้นต่อกัน เมื่อได้กฎความสัมพันธ์ของข้อมูลแต่ละชุดแล้วในขั้นตอนของการรวมกฎความสัมพันธ์นั้น จะรวมกฎความสัมพันธ์โดยนำมาเฉพาะกฎความสัมพันธ์ที่เหมือนกันของแต่ละชุดข้อมูล แต่เนื่องจากกฎความสัมพันธ์ที่ได้มาจากข้อมูลหลายชุดอาจทำให้กฎความสัมพันธ์ที่ได้ออกมาเกิดความขัดแย้งกันเองและมีบางกฎความสัมพันธ์ที่ยังขาดหายไปเมื่อเปรียบเทียบกับการหากฎความสัมพันธ์จากข้อมูลเพียงชุดเดียว ดังนั้นจะนำกฎความสัมพันธ์ที่ได้ไปแปลงรูปให้อยู่ในรูปแบบของภาษาควบคุม (Controlled Natural Language) [4] เพื่อที่จะสามารถนำกฎความสัมพันธ์ที่ได้เข้าไปใช้ใน FaCT++ Reasoner [6] ซึ่งเป็นเครื่องมือสำหรับการตรวจสอบความขัดแย้งและสามารถเรียนรู้จากข้อมูลที่ตรวจสอบเพื่อให้ได้ความรู้ใหม่ออกมาได้ โดยความรู้ใหม่ที่ได้ออกมานั้นสามารถนำไปเพิ่มเติมในส่วนของกฎความสัมพันธ์ที่ยังขาดหายไปได้ สุดท้ายจะได้กฎความสัมพันธ์เพียงชุดเดียวที่มาจากข้อมูลหลายชุดที่มีประสิทธิภาพใกล้เคียงกับการหากฎความสัมพันธ์แบบดั้งเดิม



รูปที่ 3 กรอบแนวคิดการหากฎความสัมพันธ์แบบกระจาย

4. สรุปและอภิปรายผล

การจัดเก็บข้อมูลในปัจจุบันสามารถทำได้ง่าย เนื่องจากเทคโนโลยีที่ทันสมัยที่ถูกพัฒนามาอย่างต่อเนื่อง ด้วยเหตุผลนี้เองข้อมูลที่ถูกจัดเก็บนั้นมีขนาดที่ใหญ่ หรือถูกกระจายไปอยู่ตามแหล่งข้อมูลต่าง ๆ ทำให้การทำเหมืองข้อมูลด้วยวิธีหากฎความสัมพันธ์นั้นทำได้ยาก ซึ่งงานวิจัยที่ปรากฏอยู่ยังไม่สามารถตอบโจทย์ของการหากฎความสัมพันธ์แบบกระจายได้เท่าที่ควร ซึ่งจะต้องสามารถหากฎความสัมพันธ์ได้มากกว่า 1 ชุด โดยการหากฎความสัมพันธ์แต่ละชุดข้อมูลจะไม่ขึ้นต่อกัน และสามารถช่วยลดค่าใช้จ่ายในการซื้อคอมพิวเตอร์ที่มีประสิทธิภาพสูงในการประมวลผล

จากกรอบแนวคิดที่ได้นำเสนอไปสามารถหากฎความสัมพันธ์แบบกระจายได้โดยแต่ละชุดข้อมูลจะไม่ขึ้นต่อกัน และสามารถหากฎความสัมพันธ์จากคอมพิวเตอร์หลาย ๆ เครื่องได้พร้อม ๆ กัน ซึ่งไม่จำเป็นต้องใช้เครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูงก็สามารถจะหากฎความสัมพันธ์ได้ แทนที่จะใช้คอมพิวเตอร์ที่มีประสิทธิภาพสูงที่มีราคาแพงในการประมวลผล หรือช่วยแก้ปัญหาในส่วนของแหล่งความรู้ที่กระจายกันอยู่ตามแหล่งต่าง ๆ ที่เป็นเรื่องยากสำหรับการนำกฎความสัมพันธ์เหล่านั้นมาให้เป็นฐานความรู้เพียงหนึ่งเดียว

5. เอกสารอ้างอิง

[1] Huy, Pham Nguyen Anh, and Ho Tu Bao (2005). "A Distributed Algorithm for Mining Association rules." , pp.190-195

[2] LI, Tao; ZHU, Shenghuo; OGIHARA, Mitsunori (2003). A new distributed data mining model based on similarity. In: *Proceedings of the 2003 ACM symposium on Applied computing*. ACM, pp. 432-436.

[3] Leighton, Frank Thomson (1992). Introduction to parallel algorithms and architectures, pp. 439

[4] Norbert E. Fuchs and Kaarel Kaljurand (2006). Attempto Controlled English: Language, Tools and Applications [Online]. Available URL: http://attempto.ifi.uzh.ch/site/courses/files/ACE.Course.UniZH.1206.Getting_Started.pdf

[5] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami (1993). Mining Association Rules between Sets of Items in Large Database. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207-216

[6] Tsarkov, Dmitry, and Ian Horrocks (2006). FaCT++ description logic reasoner: System description. Automated reasoning. Springer Berlin Heidelberg, pp.292-297.

[7] Tsoumakas, Grigorios, and Ioannis P. Vlahavas (2009). "Distributed Data Mining.", pp.157-164

A Technique to Association Rule Mining on Multiple Datasets

Nuntawut Kaoungku
School of Computer Engineering, Institute of Engineering Suranaree University of Technology, Thailand.
Email: b5111299@gmail.com

Kittisak Kerdprasop and Nittaya Kerdprasop
School of Computer Engineering, Institute of Engineering Suranaree University of Technology, Thailand.
Email: KittisakThailand@gmail.com and nittaya@sut.ac.th

***Abstract*— This research aims at studying the method for association rule mining on multiple datasets. Current with technology and information systems enabling agencies or organization has a data-storage system, but the problem is that those with a larger data set, which is difficult in the association rule mining, because it requires a computer with a high-performance to process, which was followed by a cost increase. How it can help solve this problem is to distribute data process according to multiple computers, then combined rules of each machine using Fact + + Reasoner for check conflicts of rules, and will therefore have powerful association rules similar to the method for association rule mining on one dataset. We thus propose a technique for association rule mining on multiple datasets.**

***Index Terms*—Association rule mining, Controlled language, Attempto Controlled English**

I. Introduction

Current, with the rapid development of technology allows agencies and organizations have adopted various technologies applied to the agency or the organization even more. These technologies make it possible to easily and systematically, but what follows is data that is stored large, which is difficult in association rule mining. As it required a computer with high-performance to processing and high cost, which the large organizations to have the financial resources to association rule mining from large data set. There is a technique to help fix this problem is to distribute the data set to be processed by multiple computers, by the computer, it does not require a high-performance to processing in association rule mining. However, it may have a conflict with the association rules in the process of combining association rules from each machine, and association rules from multiple datasets may be inefficient compared to the association rules from only data set. So in the process of combining association rules requires a technique to help fix the problems mentioned above.

Manuscript received November 30, 2013; revised December 15, 2013; accepted December 31, 2013.
Journal of Computers (JCP, ISSN 1796-203X), corresponding author.

Step in the combine association rules from distributed data is essential as well, as association rules from multiple datasets must be close to the most powerful association rules from one datasets and association rules must be inconsistency. Examination of conflict in association rules is used Fact + + Reasoner [7] and need to write rules in the form of Attempto Controlled English (ACE) [2], which is a Controlled Natural Language (CNL) on the Protégé.

Researches related to association rule mining on multiple datasets have to appear very little. Probably, due to the association rule mining on multiple dataset that is difficult process of combined association rules from distributed data, association rules with efficient close to that of association rule mining from one datasets. The researchers appeared, there was an inefficient comparison clearly. [1, 3, 8]

From the above it can be seen that association rule mining relations from large dataset it is difficult, there is a need to distribute data processing according to multiple computers. Combining association rule from each of computers may be a problem in the conflict of association rules and the efficiency of association rules. We thus propose a technique to association rule mining on multiple datasets.

II. Background

A. Association rule mining

Association Rule Mining is a process that has been popular in the relationship between the data that is how most association rule mining in a variety of ways. In this paper, the algorithm Apriori [6] of the association rule mining.

TABLE 1
PURCHASE TRANSACTIONS OF ALL CUSTOMERS

Order	Milk	Water	Candy	Sausage
1	1	1	0	0
2	0	1	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0

Table 1 is show a purchase transaction of all customers and then fined the frequency pattern of purchases of customers in each piece, to find the relationship of each product which is shown in Table 2, after which it will be used with high frequency items set to generate association rules, which is in the form of IF condition Then result by the criteria used in the present are the following:

- Support is the frequency of the event occurring
- Confidence is the frequency of the incident with other events occurring together.

TABLE 2
THE FREQUENCY OF CUSTOMER PURCHASES.

	Milk	Water	Candy	Sausage
Milk	2*	2	1	0
Water	2	4*	2	0
Candy	1	1	2*	0
Sausage	0	0	0	1*

B. Attempto Controlled English

ACE is a controlled natural language that is based on first-order logic language, which combines the advantages of natural languages and formal language to want to make the writing language in the form of human and machine can understand, can be written in the form of simple English sentences [2], as shown by Figure 1 is an example of comparison between FOL, DL, OWL, UML, and ACE. ACE is a plugin of Protégé editor, in this research association rules are written in the form of ACE because will lead association rules to check conflicts with Fact++ Reasoner in Protégé editor. Table 3 shows an example of converting association rules in the form of ACE.

first-order logic	$\forall X(protein(X) \rightarrow \exists Y(terminus(Y) \wedge has(X,Y)))$
DL	$Protein \sqsubseteq \exists has.Terminus$
OWL (RDF/XML)	`<owl:Class rdf:ID="Protein"> <rdfs:subClassOf> <owl:Restriction> <owl:onProperty rdf:resource="#has"/> <owl:someValuesFrom rdf:resource="#Terminus"/> </owl:Restriction> </rdfs:subClassOf> </owl:Class>`
UML	Protein — 1..* — Terminus
ACE	Every protein has a terminus.

Figure 1 Example of comparison between FOL, DL, OWL, UML, and ACE

TABLE 3
EXAMPLE OF CONVERTING ASSOCIATION RULES IN THE FORM OF ACE

Original association rules	Association rules in ACE
{CLASS=crew} => {SEX=male}	If X is a **crew** then X is a **male**.
{CLASS=crew, AGE=adult} => {SEX=male}	If X is a **crew** and X is an **adult** then X is a **male**.
{CLASS=crew, AGE=adult, SURVIVED=no} => {SEX=male}	If X is a **crew** and X is an **adult** and X is a n: **not-survivor** then X is a **male**.

C. FaCT++ Reasoner

FaCT + + is reasoner was developed from FaCT algorithm using C + + language development, which is based on Description Logics (DL), to be used for checking the inconsistency of Ontology [7], for example following:

> Every man is a human.
> John is a man.
> John is not a human.

For example, it can be seen that the conflict in the sentence "John is not a human", because two sentences have previously said that "John is a human" and could not use sentences in an example to created ontology. In this research, association rule mining from distribute data, may be association rules is a conflict, so it need FaCT++ Reasoner to checking the conflict of the association rules from multiple datasets

III. METHODOLOGY

This research proposed a technique for association rule mining on multiple datasets, the data is divided according to multiple computers to help in the association rule mining, replace association rule mining from large dataset, which require a computer with high-performance to process. But in the process of combined association rules from multiple datasets is difficult, because to the association rules with performance close to the association rule mining from large dataset and association rules that may conflict.

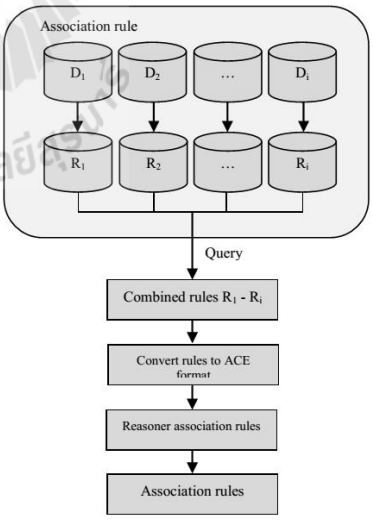


Figure 2 Conceptual framework of the research

R_1	R_2	R_3
{CLASS=third} => {AGE=adult}		
{CLASS=crew} => {SEX=male}	{CLASS=crew} =>{SEX=male}	{CLASS=crew} => {SEX=male}
{CLASS=crew} => {AGE=adult}	{CLASS=crew} =>{AGE=adult}	{CLASS=crew} => {AGE=adult}

R
{CLASS=crew} => {SEX=male}
{CLASS=crew} => {AGE=adult}

Figure 3 Example combined association rules

Figure 1 shows conceptual framework of the research. First, association rule mining from multiple datasets by $D_1, D_2, \dots, D_i$ with $i = 1, 2, \dots, n$. Second, combined association rules from first step by $R = R_1 \cap R_2 \cap \dots \cap R_i$ with $i = 1, 2, \dots, n$ which is shown in figure 3. Third, converting association rules in the form of ACE. Forth, checking the conflict of the association rules with FaCT++ Reasoner. Finally, the association rules from multiple datasets with similar efficient to the association rules from one dataset, this can be checked from the ontology created from Protégé editor.

IV. Experiment Result

This research experimented to compare the results from the association rule mining on multiple datasets and the association rule mining on one dataset used Breast-cancer dataset from the UCI Machine Learning Repository. Breast-cancer dataset has 10 attributes and 286 data instances, figure 4 is an example of Breast-cancer dataset are 5 instants.

The experiment will divided breast-cancer dataset for association rule mining to three datasets, which use minimum support for association rule mining at 0.3 and 0.5. Table 4 show comparative results of association rule mining on multiple dataset and association rule mining on one dataset with minimum support 0.3, it can be seen that the association rules from multiple dataset missing a lot, and when considering ontology Figure 5 shows that association rules from multiple dataset and association rules from one dataset are clearly different. Table 6 show comparative results of association rule mining on multiple dataset and association rule mining on one dataset with minimum support 0.5, it can be seen that there are some association rules from multiple dataset still missing, and when considering Ontology of Figure 6 shows that association rules from multiple dataset effectively close association rules from one dataset.

Association rule mining with minimum support 0.3 and 0.5, table 8 show association rule mining from multiple dataset with minimum support 0.5 it provides the number of rules closely to association rule mining from one dataset more than minimum support 0.3. Association rule mining from multiple dataset there is a missing, can be certain association rules of table 5 and table 7 to fill in some missing association rules.

age	menopause	tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad	irradiat	Class
40-49	premeno	15-19	0-2	yes	3	right	left_up	no	recurrence-events
50-59	ge40	15-19	0-2	no	1	right	central	no	no-recurrence-events
50-59	ge40	35-39	0-2	no	2	left	left_low	no	recurrence-events
40-49	premeno	35-39	0-2	yes	3	right	left_low	yes	no-recurrence-events
40-49	premeno	30-34	3-5	yes	2	left	right_up	no	recurrence-events

Figure 4 Example of Breast-cancer dataset

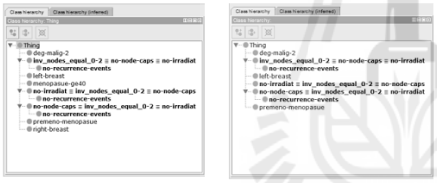
TABLE 4
COMPARATIVE RESULTS OF ASSOCIATION RULE MINING ON MULTIPLE DATASET AND ASSOCIATION RULE MINING ON ONE DATASET WITH MINIMUM SUPPORT 0.3

Original association rules	Combined association rules
if X is a n:inv_nodes_equal_0-2 and X is a n:left-breast then X is a n:no-node-caps.	if X is a n:inv_nodes_equal_0-2 and X is a n:left-breast then X is a n:no-node-caps.
if X is a n:no-node-caps and X is a n:left-breast then X is a n:inv_nodes_equal_0-2.	if X is a n:no-node-caps and X is a n:left-breast then X is a n:inv_nodes_equal_0-2.
if X is a n:premeno-menopasue and X is a n:inv_nodes_equal_0-2 then X is a n:no-node-caps.	if X is a n:premeno-menopasue and X is a n:inv_nodes_equal_0-2 then X is a n:no-node-caps.
if X is a n:premeno-menopasue and X is a n:no-node-caps then X is a n:inv_nodes_equal_0-2.	if X is a n:premeno-menopasue and X is a n:no-node-caps then X is a n:inv_nodes_equal_0-2.
if X is a n:left-breast and X is a n:no-irradiat then X is a n:no-node-caps.	if X is a n:left-breast and X is a n:no-irradiat then X is a n:no-node-caps.
if X is a n:no-node-caps and X is a n:left-breast then X is a n:no-irradiat.	if X is a n:no-node-caps and X is a n:left-breast then X is a n:no-irradiat.
if X is a n:inv_nodes_equal_0-2 and X is a n:left-breast then X is a n:no-irradiat.	if X is a n:inv_nodes_equal_0-2 and X is a n:left-breast then X is a n:no-irradiat.
if X is a n:left-breast and X is a n:no-irradiat then X is a n:inv_nodes_equal_0-2.	if X is a n:left-breast and X is a n:no-irradiat then X is a n:inv_nodes_equal_0-2.
if X is a n:inv_nodes_equal_0-2 and X is a n:left-breast and X is a n:no-irradiat then X is a n:no-node-caps.	if X is a n:inv_nodes_equal_0-2 and X is a n:left-breast and X is a n:no-irradiat then X is a n:no-node-caps.
if X is a n:inv_nodes_equal_0-2 and X is a n:no-node-caps and X is a n:left-breast then X is a n:no-irradiat.	if X is a n:inv_nodes_equal_0-2 and X is a n:no-node-caps and X is a n:left-breast then X is a n:no-irradiat.
if X is a n:no-node-caps and X is a n:left-breast and X is a n:no-irradiat then X is a n:inv_nodes_equal_0-2.	if X is a n:no-node-caps and X is a n:left-breast and X is a n:no-irradiat then X is a n:inv_nodes_equal_0-2.
if X is a n:premeno-menopasue and X is a n:no-irradiat then X is a n:no-node-caps.	if X is a n:premeno-menopasue and X is a n:no-irradiat then X is a n:no-node-caps.
if X is a n:premeno-menopasue and X is a n:no-node-caps then X is a n:no-irradiat.	if X is a n:premeno-menopasue and X is a n:no-node-caps then X is a n:no-irradiat.
if X is a n:premeno-menopasue and X is a n:inv_nodes_equal_0-2 then X is a n:no-irradiat.	
if X is a n:premeno-menopasue and X is a n:no-irradiat then X is a n:inv_nodes_equal_0-2.	
if X is a n:inv_nodes_equal_0-2 and X is a n:deg-malig-2 then X is a n:no-node-caps.	if X is a n:inv_nodes_equal_0-2 and X is a n:deg-malig-2 then X is a n:no-node-caps.
if X is a n:no-node-caps and X is a n:deg-malig-2 then X is a n:inv_nodes_equal_0-2.	if X is a n:no-node-caps and X is a n:deg-malig-2 then X is a n:inv_nodes_equal_0-2.
if X is a n:left-breast and X is a n:no-recurrence-events then X is a n:inv_nodes_equal_0-2.	if X is a n:left-breast and X is a n:no-recurrence-events then X is a n:inv_nodes_equal_0-2.
if X is a n:inv_nodes_equal_0-2 and X is a n:left-breast then X is a n:no-recurrence-events.	
if X is a n:menopasue-ge40 and X is a n:inv_nodes_equal_0-2 then X is a n:no-node-caps.	
if X is a n:menopasue-ge40 and X is a n:no-node-caps then X is a n:inv_nodes_equal_0-2.	
if X is a n:inv_nodes_equal_0-2 and X is a n:right-breast then X is a n:no-node-caps.	
if X is a n:no-node-caps and X is a n:right-breast then X is a n:inv_nodes_equal_0-2.	
if X is a n:left-breast and X is a n:no-recurrence-events then X is a n:no-node-caps.	
if X is a n:premeno-menopasue and X is a n:inv_nodes_equal_0-2 and X is a n:no-irradiat then X is a n:no-node-caps.	
if X is a n:premeno-menopasue and X is a n:inv_nodes_equal_0-2 and X is a n:no-node-caps then X is a n:no-irradiat.	
if X is a n:premeno-menopasue and X is a n:no-node-caps and X is a n:no-irradiat then X is a n:inv_nodes_equal_0-2.	
if X is a n:left-breast and X is a n:no-recurrence-events then X is a n:no-irradiat.	
if X is a n:inv_nodes_equal_0-2 and X is a n:left-breast and X is a n:no-recurrence-events then X is a n:no-node-caps.	
if X is a n:no-node-caps and X is a n:left-breast and X is a n:no-recurrence-events then X is a n:inv_nodes_equal_0-2.	

if X is a n:inv_nodes_equal_0-2 and X is a n:no-node-caps and X is a n:left-breast then X is a n:no-recurrence-events.	
if X is a n:menopasue-ge40 and X is a n:no-irradiat then X is a n:no-node-caps.	
if X is a n:menopasue-ge40 and X is a n:no-node-caps then X is a n:no-irradiat.	
if X is a n:premeno-menopasue and X is a n:no-recurrence-events then X is a n:no-node-caps.	
if X is a n:deg-malig-2 and X is a n:no-irradiat then X is a n:no-node-caps.	if X is a n:deg-malig-2 and X is a n:no-irradiat then X is a n:no-node-caps.
if X is a n:no-node-caps and X is a n:deg-malig-2 then X is a n:no-irradiat.	if X is a n:no-node-caps and X is a n:deg-malig-2 then X is a n:no-irradiat.
if X is a n:right-breast and X is a n:no-irradiat then X is a n:no-node-caps.	
if X is a n:no-node-caps and X is a n:right-breast then X is a n:no-irradiat.	

TABLE 5
ENTAILMENT FROM REASONER ASSOCIATION RULES WITH MINIMUM SUPPORT 0.3

Entailment	ACE If-then
Every inv_nodes_equal_0-2 is a no-irradiat that is a no-node-caps .	If X is a n:inv_nodes_equal_0-2 and X is a n:no-irradiat then X is a n:no-node-caps .
Every no-irradiat is an inv_nodes_equal_0-2 that is a no-node-caps .	If X is a n:no-irradiat and X is a n:inv_nodes_equal_0-2 then X is a n:no-node-caps .
Every no-node-caps is an inv_nodes_equal_0-2 that is a no-irradiat .	If X is a n:no-node-caps and X is a n:inv_nodes_equal_0-2 then X is a n:no-irradiat .



(a) (b)

Figure 5 Ontology from association rule mining on one dataset (a) and association rule mining on multiple dataset (b) with minimum support 0.3

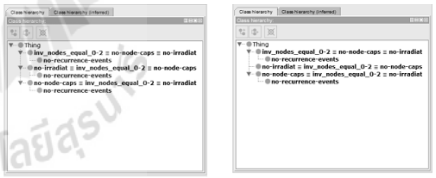
TABLE 6
COMPARATIVE RESULTS OF ASSOCIATION RULE MINING ON MULTIPLE DATASET AND ASSOCIATION RULE MINING ON ONE DATASET WITH MINIMUM SUPPORT 0.5

Original association rules	Combined association rules
if X is a n:inv_nodes_equal_0-2 then X is a n:no-node-caps.	if X is a n:inv_nodes_equal_0-2 then X is a n:no-node-caps.
if X is a n:no-node-caps then X is a n:inv_nodes_equal_0-2.	if X is a n:no-node-caps then X is a n:inv_nodes_equal_0-2.
if X is a n:no-irradiat then X is a n:no-node-caps.	if X is a n:no-irradiat then X is a n:no-node-caps.
if X is a n:no-node-caps then X is a n:no-irradiat.	if X is a n:no-node-caps then X is a n:no-irradiat.
if X is a n:inv_nodes_equal_0-2 then X is a n:no-irradiat.	
if X is a n:no-irradiat then X is a n:inv_nodes_equal_0-2.	if X is a n:no-irradiat then X is a n:inv_nodes_equal_0-2.
if X is a n:inv_nodes_equal_0-2 and X is a n:no-irradiat then X is a n:no-node-caps.	if X is a n:inv_nodes_equal_0-2 and X is a n:no-irradiat then X is a n:no-node-caps.
if X is a n:inv_nodes_equal_0-2 and X is a n:no-node-caps then X is a n:no-irradiat.	if X is a n:inv_nodes_equal_0-2 and X is a n:no-node-caps then X is a n:no-irradiat.
if X is a n:no-node-caps and X is a n:no-irradiat then X is a n:inv_nodes_equal_0-2.	if X is a n:no-node-caps and X is a n:inv_nodes_equal_0-2.
if X is a n:no-recurrence-events then X is a n:no-node-caps.	if X is a n:no-recurrence-events then X is a n:no-node-caps.
if X is a n:no-recurrence-events then X is a n:inv_nodes_equal_0-2.	if X is a n:no-recurrence-events then X is a n:inv_nodes_equal_0-2.
if X is a n:no-recurrence-events then X is a n:no-irradiat.	
if X is a n:inv_nodes_equal_0-2 and X is a n:no-recurrence-events then X is a n:no-node-caps.	if X is a n:inv_nodes_equal_0-2 and X is a n:no-recurrence-events then X is a n:no-node-caps.

if X is a n:no-node-caps and X is a n:no-recurrence-events then X is a n:inv_nodes_equal_0-2.	if X is a n:no-node-caps and X is a n:no-recurrence-events then X is a n:inv_nodes_equal_0-2.
if X is a n:no-irradiat and X is a n:no-recurrence-events then X is a n:no-node-caps.	if X is a n:no-irradiat and X is a n:no-recurrence-events then X is a n:no-node-caps.
if X is a n:no-node-caps and X is a n:no-recurrence-events then X is a n:no-irradiat.	if X is a n:no-node-caps and X is a n:no-recurrence-events then X is a n:no-irradiat.
if X is a n:no-node-caps and X is a n:no-irradiat then X is a n:no-recurrence-events.	
if X is a n:inv_nodes_equal_0-2 and X is a n:no-recurrence-events then X is a n:no-node-caps.	if X is a n:inv_nodes_equal_0-2 and X is a n:no-irradiat.
if X is a n:no-irradiat and X is a n:no-recurrence-events then X is a n:inv_nodes_equal_0-2.	if X is a n:no-irradiat and X is a n:no-recurrence-events then X is a n:inv_nodes_equal_0-2.
if X is a n:inv_nodes_equal_0-2 and X is a n:no-irradiat then X is a n:no-recurrence-events.	
if X is a n:inv_nodes_equal_0-2 and X is a n:no-irradiat and X is a n:no-recurrence-events then X is a n:no-node-caps.	if X is a n:inv_nodes_equal_0-2 and X is a n:no-irradiat and X is a n:no-recurrence-events then X is a n:no-node-caps.
if X is a n:inv_nodes_equal_0-2 and X is a n:no-node-caps and X is a n:no-recurrence-events then X is a n:no-irradiat.	if X is a n:inv_nodes_equal_0-2 and X is a n:no-node-caps and X is a n:no-recurrence-events then X is a n:no-irradiat.
if X is a n:no-node-caps and X is a n:no-irradiat and X is a n:no-recurrence-events then X is a n:inv_nodes_equal_0-2.	if X is a n:no-node-caps and X is a n:no-irradiat and X is a n:no-recurrence-events then X is a n:inv_nodes_equal_0-2.
if X is a n:inv_nodes_equal_0-2 and X is a n:no-node-caps and X is a n:no-irradiat then X is a n:no-recurrence-events.	

TABLE 7
ENTAILMENT FROM REASONER ASSOCIATION RULES WITH MINIMUM SUPPORT 0.5

Entailment	ACE If-then
Every no-recurrence-events is a no-irradiat .	If X is a n:no-recurrence-events then X is a n:no-irradiat .
Every inv_nodes_equal_0-2 is a no-irradiat that is a no-node-caps .	If X is a n: inv_nodes_equal_0-2 and X is a n: no-irradiat then X is a n:no-node-caps .
Every no-irradiat is an inv_nodes_equal_0-2 that is a no-node-caps .	If X is a n:no-irradiat and X is a n:inv_nodes_equal_0-2 then X is a n:no-node-caps .
Every no-node-caps is an inv_nodes_equal_0-2 that is a no-irradiat .	If X is a n:no-node-caps and X is a n:inv_nodes_equal_0-2 then X is a n:no-irradiat .



(a) (b)

Figure 6 Ontology from association rule mining on one dataset (a) and association rule mining on multiple dataset (b) with minimum support 0.5

TABLE 8
COMPERATIVE RESULTS OF NUMBER OF RULES FROM ONE DATASET AND NUMBER OF RULES FROM MULTIPLE DATASET

Minimum support	Number of rules from one dataset	Number of rules from multiple dataset
0.3	65	37
0.5	24	19

V. CONCLUSION

Association rule mining from large dataset, need a computer with high-performance to process and high cost. There is a technique to help fix this problem is to distribute datasets to be processed by multiple computers. The process combined association rules from distribute datasets take the same association rules and checking the conflict of the association rules.

From such experiments can be seen that association rule mining from multiple dataset with minimum support that many have a closely efficient the association rule mining from one dataset. This consider from ontology and inconsistency association rules, but association rule mining from multiple dataset there is a missing, can be result of reasoned process to fill missing association rules.

REFERENCES

[1] Domingos, Pedro. Prospects and challenges for multi-relational data mining. ACM SIGKDD explorations newsletter, vol.5, no.1, pp.80-83.

[2] Fuchs, Norbert E., Kaarel Kaljurand, and Tobias Kuhn (2008). Attempto controlled english for knowledge representation. Reasoning Web. Springer Berlin Heidelberg, pp.104-124.

[3] Han, Jiawei, and Yongjian Fu (1999). Mining multiple-level association rules in large databases. Knowledge and Data Engineering, IEEE Transactions, vol.11, no.5, pp.798-805.

[4] Kaljurand, Kaarel (2008). ACE View-An Ontology and Rule Editor based on Controlled English. International Semantic Web Conference (Posters & Demos). vol. 401.

[5] Norbert E. Fuchs and Kaarel Kaljurand (2006). Attempto Controlled English: Language, Tools and Applications [Online]. Available URL: http://attempto.ifi.uzh.ch/site/courses/files/ACE.Course.UniZH.1206.Getting_Started.pdf

[6] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami (1993). Mining Association Rules between Sets of Items in Large Database. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207-216

[7] Tsarkov, Dmitry, and Ian Horrocks (2006). FaCT++ description logic reasoner: System description. Automated reasoning. Springer Berlin Heidelberg, pp.292-297.

[8] Zhao Yanchang, et al (2007). Mining for combined association rules on multiple datasets. Proceedings of the 2007 international workshop on Domain driven data mining, pp.18-23.

Nuntawut Kaoungku is currently a doctoral student with the School of Computer Engineering, Suranaree University of Technology, Thailand. He received his bachelor degree in Computer Engineering from Suranaree University of Technology, Thailand, in 2012, and master degree in Computer Engineering from Suranaree University of Technology, Thailand, in 2013. He current research includes semantic web and association.

Nittaya Kerdprasop is an associate professor at the School of Computer Engineering, Suranaree University of Technology, Thailand. She received her bachelor degree in Radiation Techniques from Mahidol University, Thailand, in 1985, master degree in Computer Science from the Prince of Songkla University, Thailand, in 1991 and doctoral degree in Computer Science from Nova Southeastern University, U.S.A, in 1999. She is a member of ACM and IEEE Computer Society. Her research of interest includes Knowledge Discovery in Databases, Artificial Intelligence, Logic Programming, and Intelligent Databases.

Kittisak Kerdprasop is an associate professor and chair of the School of Computer Engineering, Suranaree University of Technology, Thailand. He received his bachelor degree in Mathematics from Srinakarinwirot University, Thailand, in 1986, mas ter degree in Computer Science from the Prince of Songkla University, Thailand, in 1991 and doctoral degree in Computer Science from Nova Southeastern University, U.S.A., in 1999. His current research includes Data mining, Artificial Intelligence, Functional and Logic Programming Languages, Computational Statistics.

ประวัติผู้เขียน

นายนันทวุฒิ คะอังกุ เกิดเมื่อวันที่ 24 มีนาคม พ.ศ. 2532 ที่ อำเภอเมือง จังหวัดสกลนคร เริ่มเข้าศึกษาระดับชั้นอนุบาล 1 ถึงชั้นประถมศึกษาปีที่ 6 ที่โรงเรียนบ้านโพนงามคุรุราษฎร์วิทยา อำเภอกุดบาก จังหวัดสกลนคร จากนั้นได้เข้าศึกษาต่อในระดับมัธยมศึกษาตอนต้นและตอนปลาย ที่โรงเรียนกุดบากพัฒนาศึกษา อำเภอกุดบาก จังหวัดสกลนคร ปีการศึกษา 2551 ได้เข้าศึกษาต่อระดับปริญญาตรีในสาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี และสำเร็จการศึกษาเมื่อปี พ.ศ. 2554 ภายหลังสำเร็จการศึกษาในระดับปริญญาตรี ได้เข้าศึกษาในระดับปริญญาโท สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ในปี 2555 และภายหลังสำเร็จการศึกษาในระดับปริญญาโท ได้เข้าศึกษาในระดับปริญญาเอก สาขาวิชาวิศวกรรมคอมพิวเตอร์ สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ในปี 2556

ในระหว่างการศึกษาได้รับความอนุเคราะห์อย่างดียิ่งจากอาจารย์ประจำวิชา Database System ได้รับความไว้วางใจให้เป็นผู้ช่วยสอนปฏิบัติการ และได้รับการตีพิมพ์เผยแพร่บทความวิชาการซึ่งรายละเอียดสามารถดูได้ที่ภาคผนวก ค