การทำเหมืองรูปแบบ
: ทฤษฎีและการฝึกปฏิบัติ

PATTERN MINING
: THEORY AND PRACTICE

ผศ.ดร. พนิดา ทรงรัมย์

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ
มหาวิทยาลัยมหาสารคาม

ชื่อหนังสือ การทำเหมืองรูปแบบ : ทฤษฎีและการฝึกปฏิบัติ

ผู้แต่ง พนิดา ทรงรัมย์

พิมพ์ครั้งที่ 1 เมษายน 2563 จำนวนพิมพ์ 300 เล่ม

จัดทำโดย พนิดา ทรงรัมย์

พิมพ์ที่ โรงพิมพ์มหาวิทยาลัยราชภัฏมหาสารคาม
เลขที่ 80 ถนนนครสวรรค์ ตำบลตลาด อำเภอเมือง
จังหวัดมหาสารคาม 44000
โทร. 0-4372-2118-9 ต่อ 141 โทรสาร 0-4374-2618
rmupress@gmail.com

ข้อมูลทางบรรณานุกรมของหอสมุดแห่งชาติ
National Library of Thailand Cataloging in Publication Data

พนิดา ทรงรัมย์.
การทำเหมืองรูปแบบ : ทฤษฎีและการฝึกปฏิบัติ, มหาสารคาม : โรงพิมพ์มหาวิทยาลัยราชภัฏมหาสารคาม, 2563.
199 หน้า
ISBN 978-616-568-369-2

คำนำ

หนังสือ การทำเหมืองรูปแบบ: ทฤษฎีและการฝึกปฏิบัติ เป็นหนังสือที่ผู้เขียนตั้งใจเขียนขึ้นเพื่อให้ผู้อ่านเข้าใจทฤษฎีและขั้นตอนการทำเหมืองรูปแบบในแบบต่างๆ สามารถฝึกปฏิบัติจริง และสามารถนำไปประยุกต์ใช้ในด้านต่างๆ รวมถึงให้นิสิต นักศึกษา นักวิจัย สามารถนำความรู้ที่ได้จากหนังสือเล่มนี้ประกอบการค้นคว้า การวิจัย และการทดลองได้

การทำเหมืองรูปแบบในหนังสือเล่มนี้ ประกอบไปด้วย การทำเหมืองเซตรายการความถี่ การทำเหมืองรูปแบบลำดับเหตุการณ์ การทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติ การทำเหมืองกฎความสัมพันธ์ การจำแนกเชิงความสัมพันธ์ และการประยุกต์ใช้การทำเหมืองรูปแบบ ในแต่ละเรื่องจะอธิบายถึงทฤษฎีที่เกี่ยวข้องและขั้นตอนวิธีพร้อมกับยกตัวอย่าง เพื่อให้ผู้อ่านสามารถเข้าใจทฤษฎีที่เกี่ยวข้องและขั้นตอนวิธีได้ง่าย นอกจากนี้ ในแต่ละเรื่องยังประกอบไปด้วยตัวอย่างคำสั่งพร้อมกับคำอธิบาย เพื่อให้ผู้อ่านสามารถเข้าใจคำสั่งสำหรับการทำเหมืองรูปแบบและสามารถฝึกปฏิบัติตามได้

ผู้เขียนหวังเป็นอย่างยิ่งว่า หนังสือเล่มนี้จะเป็นประโยชน์แก่นิสิต นักศึกษา นักวิจัย รวมถึงบุคคลทั่วไปที่สนใจการทำเหมืองรูปแบบ หากหนังสือเล่มนี้มีข้อบกพร่องประการใด ผู้เขียนขอน้อมรับไว้ทั้งหมด

ผู้ช่วยศาสตราจารย์ ดร. พนิดา ทรงรัมย์
Email: panida.s@msu.ac.th

สารบัญ

หน้า

บทที่ 1
บทนำ
(Introduction)

1.1 การค้นหาองค์ความรู้ในฐานข้อมูล

การค้นหาองค์ความรู้ในฐานข้อมูล (Knowledge discovery in databases) คือ กระบวนการค้นหาข้อมูลสารสนเทศหรือองค์ความรู้จากฐานข้อมูลขนาดใหญ่ เพื่อนำองค์ความรู้ที่ได้ไปใช้ประโยชน์ในด้านต่างๆ เช่น การตัดสินใจ การพยากรณ์ การแนะนำ การจำแนก เป็นต้น การค้นหาองค์ความรู้ในฐานข้อมูลขนาดใหญ่เป็นการจัดการข้อมูลให้อยู่ในรูปแบบที่สามารถนำไปใช้ประโยชน์ได้ ซึ่งสามารถใช้กระบวนการการทำเหมืองข้อมูล (Data mining) เพื่อสกัดรูปแบบที่น่าสนใจหรือรูปแบบที่สามารถนำไปใช้ประโยชน์ได้

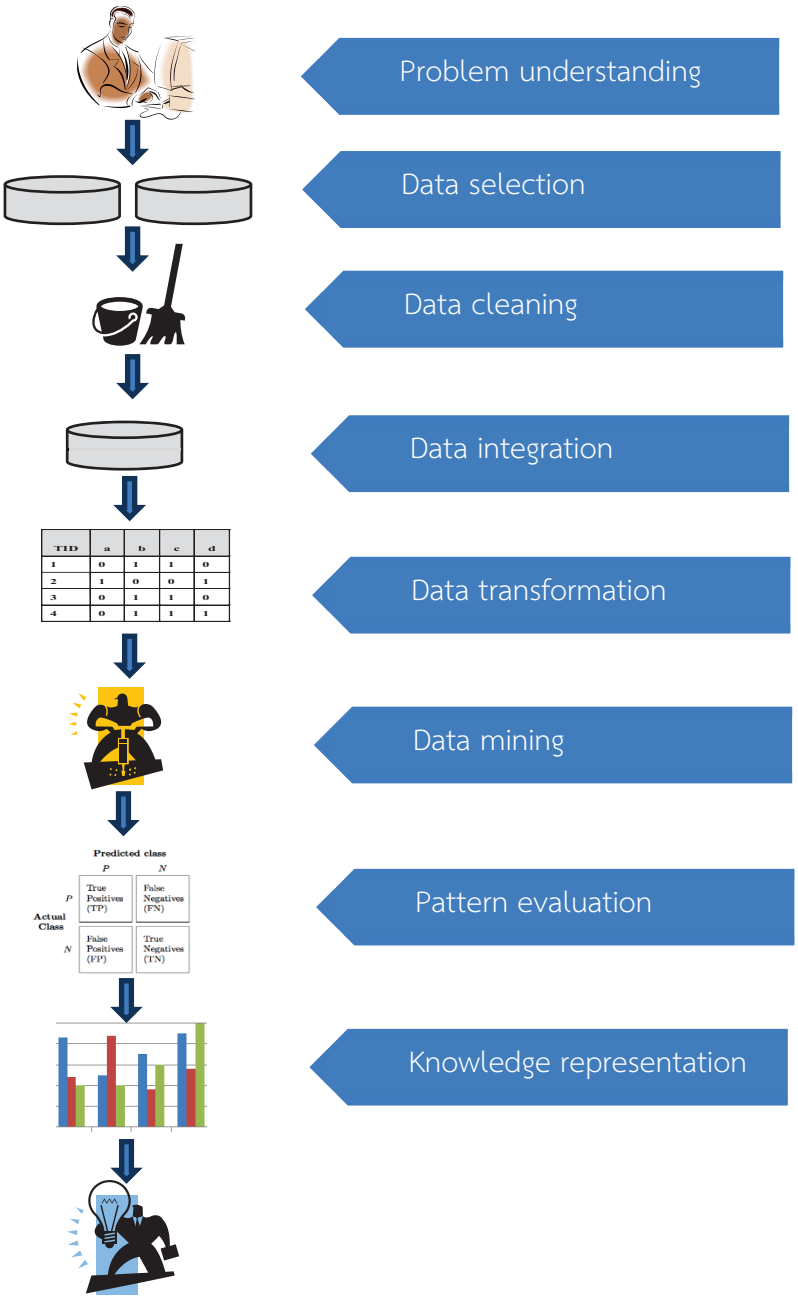
การทำเหมืองข้อมูลเป็นการนำเอาความรู้ในหลากหลายสาขามารวมกัน เช่น เทคโนโลยีฐานข้อมูล (Database technology) ปัญญาประดิษฐ์ (Artificial intelligence) การเรียนรู้ของเครื่อง (Machine learning) ข่ายงานประสาทเทียม (Neural network) สถิติ (Stat) การรู้จำรูปแบบ (Pattern recognition) เป็นต้น โดยกระบวนการในการทำเหมืองข้อมูลประกอบไปด้วย 8 กระบวนการหลัก ดังรูปที่ 1.1 ซึ่งมีรายละเอียดดังต่อไปนี้

- Problem understanding เป็นการทำความเข้าใจปัญหาในสิ่งที่ต้องการ โดยมีการตั้งเป้าหมายที่ต้องการแก้ไขปัญหา และระบุผลลัพธ์ที่ต้องการ เช่น ต้องการหารูปแบบความสัมพันธ์ของการซื้อสินค้า เพื่อนำรูปแบบความสัมพันธ์ดังกล่าวไปใช้ในการเพิ่มยอดการขายสินค้า เป็นต้น

- Data selection เป็นการเลือกข้อมูลที่จะนำมาใช้ในการค้นหาองค์ความรู้ หรือนำมาใช้ในการแก้ปัญหา เช่น ข้อมูลที่ใช้ในการหาความสัมพันธ์ของการซื้อสินค้า ก็คือ รายการซื้อสินค้าของลูกค้า เป็นต้น

- Data cleaning เป็นการทำความสะอาดข้อมูล โดยกำจัดสิ่งรบกวนออกไป เช่น กำจัดข้อมูลที่ไม่ครบถ้วนออกไป หรือทำให้ข้อมูลมีความถูกต้องสมบูรณ์ เป็นต้น

- Data integration เป็นการรวบรวมข้อมูลเข้าด้วยกัน โดยข้อมูลที่ต้องการรวบรวมอาจจะมาจากหลายแหล่ง เช่น ต้องการรวบรวมทั้งรายการสินค้าและข้อมูลพื้นฐานของลูกค้า เพื่อใช้ในการวิเคราะห์พฤติกรรมการซื้อสินค้าของลูกค้า เป็นต้น



รูปที่ 1.1 กระบวนการในการทำเหมืองข้อมูล

- Data transformation เป็นการแปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสม เพื่อให้สามารถทำเหมืองข้อมูลได้ เช่น ต้องการวิเคราะห์ความคิดเห็นต่อสินค้าจากข้อความคิดเห็นบนเฟสบุ๊ก จำเป็นต้องแปลงข้อความคิดเห็นให้อยู่ในรูปแบบเวกเตอร์ เพื่อให้สามารถนำไปประมวลผลได้ หรือการแปลงข้อมูลให้อยู่ในรูปแบบตัวเลข เพื่อให้ง่ายต่อการประมวลผล เป็นต้น

- Data mining คือ กระบวนการที่ใช้ในการสกัดข้อมูลสารสนเทศหรือองค์ความรู้ที่ต้องการ โดยจะต้องเลือกเทคนิคที่เหมาะสมที่จะใช้ในการแก้ปัญหา เช่น ต้องการหาความสัมพันธ์ของการซื้อสินค้าโดยใช้เทคนิคการทำเหมืองกฎความสัมพันธ์ (Association rule mining) เป็นต้น

- Pattern evaluation เป็นการประเมินรูปแบบที่ได้ว่ามีประสิทธิภาพหรือประสิทธิผลเพียงใด การประเมินสามารถทำได้โดยการทดสอบใช้กับสถานการณ์จริงหรือเหตุการณ์จำลอง เพื่อดูประสิทธิภาพหรือประสิทธิผลที่ได้ นอกจากนี้ยังต้องพิจารณาว่ามีข้อผิดพลาดเกิดขึ้นหรือไม่ เมื่อเกิดข้อผิดพลาดจำเป็นต้องดำเนินการแก้ไขก่อนนำมาใช้งานจริง

- Knowledge representation เป็นการนำเสนอองค์ความรู้ที่ได้จากการค้นพบ โดยใช้เทคนิคการนำเสนอที่ให้ผู้ใช้สามารถเข้าใจและนำไปประยุกต์ใช้ได้

1.2 เทคนิคการหาองค์ความรู้ในฐานข้อมูล

ปัจจุบันมีการนำเสนอเทคนิคที่หลากหลายในการค้นหาองค์ความรู้ในฐานข้อมูลขนาดใหญ่ สามารถแบ่งเทคนิคดังกล่าวออกเป็น 2 กลุ่ม ได้ดังนี้

1. การเรียนรู้แบบมีผู้สอน (Supervised learning) หรือเรียกว่า การสร้างตัวแบบในการทำนาย (Predictive modeling) เป็นเทคนิคที่เน้นการเรียนรู้จากข้อมูลที่มีอยู่ในอดีต เพื่อนำมาสร้างสมการหรือรูปแบบสำหรับหาคำตอบ วิธีการเรียนรู้แบบมีผู้สอนประกอบด้วยดังนี้

- การจำแนกข้อมูล (Classification) เป็นการเรียนรู้จากข้อมูลในอดีตเพื่อสร้างตัวจำแนกข้อมูล (Classifier) และใช้ตัวจำแนกดังกล่าวในการทำนายประเภทของข้อมูล ตัวจำแนกถูกสร้างขึ้นจากชุดข้อมูลเรียนรู้ (Training set) ซึ่งเป็นข้อมูลที่จำแนกประเภทไว้แล้ว จากนั้นทำการวัดประสิทธิภาพตัวจำแนกก่อนที่จะนำตัวจำแนกไปใช้จริง โดยใช้ชุดข้อมูลทดสอบ (Testing set) ซึ่งเป็นข้อมูลที่ไม่มีการระบุประเภทของข้อมูล ปัจจุบันมีการพัฒนาและนำเสนอวิธีการที่หลากหลายเพื่อสร้างตัวจำแนกที่มีประสิทธิภาพ เช่น ซัพพอร์ตเวกเตอร์แมชชีน (Support vector machine) นาอีฟเบส์ (Naïve bayes) ต้นไม้ตัดสินใจ (Decision tree) การค้นหาเพื่อนบ้านที่ใกล้ที่สุด k ตัว (K-nearest neighbor) เป็นต้น

- การวิเคราะห์การถดถอย (Regression analysis) เป็นการศึกษาเกี่ยวกับความสัมพันธ์ของตัวแปร เพื่อใช้ในการประมาณค่าของตัวแปรตัวหนึ่ง ซึ่งเรียกว่า ตัวแปรตาม (Dependent variable) โดยอาศัยความรู้จากตัวแปรอื่น ซึ่งเรียกว่า ตัวแปรอิสระ (Independent

variable) วิธีการในการวิเคราะห์การถดถอยมีหลายวิธี เช่น การถดถอยเชิงเส้นอย่างง่าย (Simple linear regression) การถดถอยพหุคูณ (Multiple linear regression) เป็นต้น

2. การเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning) หรือเรียกว่า การสร้างตัวแบบเชิงบรรยาย (Descriptive model) เป็นเทคนิคที่เน้นการนำข้อมูลที่มีอยู่มาพิจารณาหาความสัมพันธ์ของข้อมูลเป็นหลัก วิธีการเรียนรู้แบบไม่มีผู้สอนประกอบด้วยดังนี้

- การแบ่งกลุ่มข้อมูล (Clustering) เป็นการจัดกลุ่มข้อมูลโดยอาศัยความคล้ายคลึงของข้อมูล โดยจัดข้อมูลที่มีความคล้ายคลึงกันไว้ในกลุ่มเดียวกัน ซึ่งความคล้ายคลึงของข้อมูลอาจจะพิจารณาจากความเหมือน (Similarity) หรือความใกล้ชิด (Proximity) โดยสามารถคำนวณจากการวัดระยะห่างระหว่างข้อมูล ซึ่งมีหลายวิธี เช่น การวัดระยะแบบยุคลิด (Euclidean distance) การวัดระยะแบบแมนฮัตตัน (Manhattan distance) เป็นต้น การแบ่งข้อมูลนอกจากจะสามารถสรุปได้ว่าแต่ละข้อมูลควรอยู่กลุ่มใด การแบ่งข้อมูลยังช่วยในเรื่องการเลือกตัวแทนในกลุ่มข้อมูล ทำให้จำนวนข้อมูลที่จะนำไปใช้ในการวิเคราะห์ต่อมีจำนวนลดลง นอกจากนี้การแบ่งกลุ่มข้อมูลสามารถแสดงให้เห็นถึงข้อมูลที่มีความผิดปกติหรือข้อมูลที่อยู่นอกกลุ่มได้
- การทำเหมืองกฎความสัมพันธ์ (Association rule mining) เป็นการค้นหารูปแบบความสัมพันธ์ของข้อมูลที่น่าสนใจจากฐานข้อมูลขนาดใหญ่ โดยแสดงออกมาอยู่ในรูปของกฎ ที่แสดงความสัมพันธ์ในรูปแบบ If then (ถ้า แล้ว) ทำให้ผู้ใช้กฎสามารถเข้าใจความสัมพันธ์ได้ง่าย จึงทำให้การทำเหมืองกฎความสัมพันธ์เป็นวิธีการหนึ่งที่ได้รับความนิยมในการนำไปประยุกต์ในด้านต่างๆ การทำเหมืองกฎความสัมพันธ์ประกอบด้วย 2 ขั้นตอน คือ การขุดค้นรูปแบบความถี่ และการสร้างกฎความสัมพันธ์ โดยการขุดค้นรูปแบบความถี่ถือเป็นขั้นตอนหลักในการหากฎความสัมพันธ์ ซึ่งในหนังสือเล่มนี้จะกล่าวถึงกระบวนการขุดค้นรูปแบบความถี่ในรูปแบบต่างๆ และการหากฎความสัมพันธ์ รวมถึงการนำกฎความสัมพันธ์ไปประยุกต์ใช้ในการจำแนกข้อมูล

1.3 ที่มาของการทำเหมืองรูปแบบ

การทำเหมืองรูปแบบเริ่มต้นจากความต้องการในการวิเคราะห์รูปแบบการซื้อสินค้าของลูกค้า จากตารางที่ 1.1 เป็นรายการการซื้อสินค้าของลูกค้าในร้านค้าแห่งหนึ่ง แต่ละรายการเปลี่ยนแปลง (Transaction) จะประกอบไปด้วยรายการสินค้าที่ซื้อในหนึ่งครั้งหรือในหนึ่งใบเสร็จ ซึ่งจะเห็นได้ว่า มีรูปแบบการซื้อสินค้าที่เกิดซ้ำๆ คือ มีการซื้อไข่ไก่กับนมบ่อย และเมื่อไรก็ตามที่ซื้อไข่ไก่จะมีการซื้อนม

ด้วยเสมอ รูปแบบดังกล่าวแสดงให้เห็นถึงพฤติกรรมการซื้อสินค้าของลูกค้า ดังนั้นทางร้านสามารถจัดสินค้าที่มีความสัมพันธ์กันให้อยู่ใกล้กัน เพื่อให้ลูกค้าเกิดความสะดวกสบายในการซื้อสินค้าและเพิ่มยอดการขายสินค้าให้กับทางร้านได้ การหารูปแบบความสัมพันธ์ของการซื้อสินค้าจึงเป็นเรื่องที่น่าสนใจและเป็นประโยชน์อย่างมากกับองค์กร

ตารางที่ 1.1 รายการซื้อสินค้า

รายการเปลี่ยนแปลง	รายการที่ซื้อ
1	**ไข่ไก่** น้ำมันพืช **นม**
2	**นม** วุ้นเส้น **ไข่ไก่** กะทิ
3	**ไข่ไก่** เนย **นม** กะทิ
4	กะทิ **นม ไข่ไก่**
5	เนย นม น้ำตาล น้ำมันพืช แป้ง

เนื่องจากการหารูปแบบความสัมพันธ์ของการซื้อสินค้าในฐานข้อมูลขนาดใหญ่ เป็นเรื่องยากและใช้เวลายาวนานถ้าใช้มนุษย์ในการวิเคราะห์ ดังนั้นจึงได้มีการคิดค้นวิธีการในการขุดค้นรูปแบบความสัมพันธ์ของการซื้อสินค้าขึ้น โดย Agrawal และคณะ ได้นำเสนอการทำเหมืองกฎความสัมพันธ์เพื่อขุดค้นรูปแบบที่แสดงให้เห็นถึงพฤติกรรมการซื้อสินค้าของลูกค้า โดยแบ่งขั้นตอนการทำเหมืองกฎความสัมพันธ์ออกเป็น 2 ขั้นตอน คือ การหาเซตรายการสินค้าที่มีการซื้อบ่อยและการสร้างกฎเพื่อแสดงให้เห็นถึงรูปแบบความสัมพันธ์ของการซื้อสินค้า ขั้นตอนวิธี Apriori ถูกนำเสนอขึ้นครั้งแรกเพื่อค้นหาเซตรายการความถี่ (Frequent itemset) หรือเซตรายการสินค้าที่ปรากฏร่วมกันบ่อยในฐานข้อมูลการซื้อขายสินค้า แล้วสร้างกฎความสัมพันธ์จากเซตรายการความถี่ เพื่อใช้ในการวิเคราะห์พฤติกรรมการซื้อสินค้าของลูกค้า จากการนำเสนอการทำเหมืองกฎความสัมพันธ์เพื่อค้นหารูปแบบการซื้อสินค้า ต่อมาได้มีการนำเสนอการทำเหมืองรูปแบบในแบบต่างๆ ขึ้น เช่น การทำเหมืองรูปแบบลำดับเหตุการณ์ การทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติ การสร้างกฎความสัมพันธ์เชิงลำดับ เป็นต้น

1.4 ความหมายของการทำเหมืองรูปแบบ

การทำเหมืองรูปแบบในหนังสือเล่มนี้ หมายถึง การค้นหารูปแบบที่น่าสนใจ มีประโยชน์ หรือซ่อนอยู่ในฐานข้อมูลขนาดใหญ่ ซึ่งอาจจะเป็นรูปแบบที่เกิดร่วมกันบ่อยหรือรูปแบบที่แสดงอยู่ในรูปของกฎความสัมพันธ์ $X \longrightarrow Y$ โดย X คือ เหตุ ส่วน Y คือ ผล เช่น ไข่ไก่ $\longrightarrow$ นม แสดงถึงความสัมพันธ์ว่า

ลูกค้าที่ซื้อไข่ไก่มักจะซื้อนมด้วย เป็นต้น โดยรูปแบบในหนังสือเล่มนี้ หมายถึง รูปแบบของข้อมูลที่ไม่พิจารณาลำดับการเกิด หรือรูปแบบของข้อมูลที่พิจารณาลำดับการเกิด หรือรูปแบบของข้อมูลที่มีทั้งส่วนที่พิจารณาลำดับการเกิดและไม่พิจารณาลำดับการเกิด โดยรูปแบบดังกล่าวเป็นรูปแบบที่มีความถี่เพียงพอที่จะนำไปใช้ประโยชน์ในแต่ละด้านได้ โดยจะพิจารณาจากความถี่ของรูปแบบว่ามากกว่าหรือเท่ากับความถี่ขั้นต่ำที่กำหนดไว้หรือไม่ เมื่อได้รูปแบบที่มีความถี่เพียงพอแล้ว สามารถนำรูปแบบดังกล่าวไปสร้างกฎความสัมพันธ์ เพื่อแสดงให้เห็นถึงความสัมพันธ์ภายในรูปแบบที่ได้

1.5 ทำไมต้องใช้การทำเหมืองรูปแบบ

ปัจจุบันข้อมูลได้เพิ่มขึ้นมหาศาลโดยเฉพาะข้อมูลที่อยู่บนเครือข่ายสังคมออนไลน์ ข้อมูลดังกล่าวสามารถนำมาใช้ประโยชน์เพื่อค้นหาองค์ความรู้ที่ซ่อนอยู่ แล้วนำองค์ความรู้ดังกล่าวมาใช้พัฒนาองค์กร การทำเหมืองรูปแบบเป็นวิธีการหนึ่งที่นิยมใช้เพื่อค้นหาองค์ความรู้หรือรูปแบบที่ซ่อนอยู่ในฐานข้อมูลขนาดใหญ่ เนื่องจากผลลัพธ์ที่ได้แสดงอยู่ในรูปแบบที่มนุษย์สามารถเข้าใจง่าย และสามารถใช้ในการตัดสินใจได้อย่างมีประสิทธิภาพ ถ้าพูดถึงการทำเหมืองรูปแบบคนมักจะนึกถึงเรื่องการสร้างกฎความสัมพันธ์ที่ใช้ในการวิเคราะห์รูปแบบการซื้อสินค้าของลูกค้า ซึ่งแสดงให้ถึงเห็นความสัมพันธ์ของการซื้อสินค้าของลูกค้า เช่น ลูกค้ามักจะซื้อผ้าอ้อมพร้อมกับเบียร์ ซื้อขนมปังกับนม เป็นต้น ซึ่งความสัมพันธ์ดังกล่าวกล่าวถูกซ่อนอยู่ในรายการการซื้อสินค้าของลูกค้า ทำให้เข้าใจถึงพฤติกรรมการซื้อสินค้าของลูกค้า และช่วยสนับสนุนเพิ่มยอดการขายสินค้า นอกจากนี้การทำเหมืองรูปแบบยังถูกนำไปใช้อย่างแพร่หลายในด้านต่างๆ ดังตัวอย่างต่อไปนี้

- การวิเคราะห์การเข้าถึงเว็บเพจ ซึ่งสามารถใช้การทำเหมืองรูปแบบเพื่อค้นหารูปแบบการเข้าถึงเว็บเพจที่เกิดร่วมกันบ่อย เพื่อช่วยให้ผู้พัฒนาเว็บไซต์สามารถเข้าใจพฤติกรรมการเข้าถึงเว็บเพจของผู้ใช้งาน สามารถกำหนดเว็บเพจที่ตรงกับความต้องการของผู้ใช้งานได้ ทำให้เว็บไซต์มีความน่าสนใจหรือใช้งานง่ายมากขึ้น และมีจำนวนผู้ใช้เพิ่มขึ้น ถ้าพิจารณาลำดับการเข้าถึงหน้าเว็บเพจร่วมด้วย จะทำให้ทราบว่าผู้ใช้งานนิยมเข้าหน้าเว็บเพจไหนก่อนหลัง เช่น ผู้ใช้มักจะเข้าไปที่เมนูค้นหาสินค้า แล้วคลิกไปที่เมนูโปรโมชั่น เป็นต้น
- การทำนายปรากฏการณ์ธรรมชาติ ซึ่งสามารถใช้การทำเหมืองรูปแบบเพื่อค้นหารูปแบบของการเกิดปรากฏการณ์ตามธรรมชาติที่เกิดร่วมกันบ่อยหรือรูปแบบของลำดับการเกิดปรากฏการณ์ธรรมชาติ เช่น เมื่อเกิดแผ่นดินไหว มักจะมีเหตุการณ์สึนามิตามมา เป็นต้น การทำนายปรากฏการณ์ธรรมชาติที่จะเกิดขึ้น ทำให้ผู้ที่เกี่ยวข้องสามารถหาทางป้องกัน

หรือเตรียมการสำหรับเหตุการณ์ที่จะเกิดขึ้นได้ เพื่อไม่ให้เกิดความเสียหายหรือเกิดความเสียหายน้อยที่สุด

- การแพทย์สามารถใช้การทำเหมืองรูปแบบเพื่อค้นหารูปแบบที่น่าสนใจ เช่น การค้นหารูปแบบการจ่ายยาที่ให้ประสิทธิภาพในการรักษา เพื่อให้แพทย์สามารถจ่ายยารักษาผู้ป่วยได้อย่างมีประสิทธิภาพ และลดค่าใช้จ่ายในการรักษา เป็นต้น
- การทำเหมืองรูปแบบสามารถสร้างรูปแบบที่ใช้ในการจำแนกข้อมูลต่างๆ ได้อย่างมีประสิทธิภาพ เช่น การจำแนกข้อคิดเห็น การจำแนกเว็บไซต์หลอกลวง การจำแนกรูปภาพ เป็นต้น รูปแบบที่ได้อยู่ในรูปแบบของกฎที่สามารถเข้าใจได้ง่ายโดยมนุษย์ และมีประสิทธิภาพในการจำแนกสูง

1.6 ค่าสนับสนุน (Support) และความสำคัญของค่าสนับสนุน

ค่าสนับสนุนเป็นค่าที่บ่งบอกถึงความถี่ของการเกิดข้อมูลร่วมกัน ซึ่งสามารถพิจารณาได้ 2 แบบ คือ ค่าสนับสนุนแบบสัมบูรณ์ (Absolute support) และค่าสนับสนุนแบบสัมพัทธ์ (Relative support) โดยค่าสนับสนุนแบบสัมบูรณ์เป็นค่าความถี่ของการเกิดข้อมูล X หรือจำนวนรายการเปลี่ยนแปลงที่มีข้อมูล X (แทนด้วย $supp(X)$) ส่วนค่าสนับสนุนแบบสัมพัทธ์เป็นค่าที่บ่งบอกเปอร์เซ็นต์ของการเกิดข้อมูล X เมื่อเทียบกับข้อมูลทั้งหมด (แทนด้วย $rel\_supp(X)$) ซึ่งสามารถคำนวณได้จากจำนวนรายการเปลี่ยนแปลงที่ปรากฏ X หารด้วยจำนวนรายการเปลี่ยนแปลงทั้งหมดคูณด้วย 100 นอกจากนี้ค่าสนับสนุนยังแบ่งออกเป็น 2 กรณี คือ กรณีที่พิจารณาการเกิดร่วมกันของข้อมูล และกรณีที่พิจารณาลำดับการเกิดของข้อมูล

ตารางที่ 1.2 รายการการซื้อสินค้า

รายการเปลี่ยนแปลง	รายการ
1	กระเป๋า **รองเท้า เสื้อผ้า**
2	**เสื้อผ้า** ครีม **รองเท้า** เฟอร์นิเจอร์
3	**รองเท้า** ครีม **เสื้อผ้า** หนังสือ
4	กระเป๋า เสื้อผ้า

ในกรณีที่พิจารณาการเกิดร่วมกัน เช่น การซื้อรองเท้าและเสื้อผ้าร่วมกัน มีรายการรองเท้าและเสื้อผ้าเกิดขึ้นร่วมกันใน 3 รายการเปลี่ยนแปลง (ดังตารางที่ 1.2) คือ รายการเปลี่ยนแปลงที่ 1 2

และ 3 ดังนั้นค่าสนับสนุนแบบสัมบูรณ์มีค่าเท่ากับ *supp*(รองเท้า∪เสื้อผ้า) = 3 และค่าสนับสนุนแบบสัมพัทธ์ของรายการรองเท้าและเสื้อผ้ามีค่าเท่ากับ *rel_supp*(รองเท้า∪เสื้อผ้า) = 3/4 x 100 = 75% จะเห็นได้ว่าค่าสนับสนุนแบบสัมพัทธ์แสดงให้เห็นว่า รายการรองเท้าและเสื้อผ้าเกิดขึ้นร่วมกัน 75% จากจำนวนข้อมูลทั้งหมด

ในกรณีที่พิจารณาลำดับการเกิดของข้อมูล เช่น จากตารางที่ 1.2 การซื้อรองเท้าแล้วตามด้วยซื้อเสื้อผ้าเกิดขึ้นใน 2 รายการเปลี่ยนแปลง คือ รายการเปลี่ยนแปลงที่ 1 และ 3 ดังนั้นค่าสนับสนุนแบบสัมบูรณ์ของรายการรองเท้าตามด้วยเสื้อผ้ามีค่าเท่ากับ *supp*(รองเท้า→เสื้อผ้า) = 2 และค่าสนับสนุนแบบสัมพัทธ์เท่ากับ *rel_supp*(รองเท้า→เสื้อผ้า) = 2/4 x 100 = 50% แสดงให้เห็นว่า มีการซื้อรองเท้าแล้วซื้อเสื้อผ้าตามหลัง 50% จากจำนวนข้อมูลทั้งหมด

ค่าสนับสนุนแบบสัมพัทธ์แสดงให้เห็นถึงเปอร์เซ็นต์ของการเกิดข้อมูล *X* เมื่อเทียบกับข้อมูลทั้งหมด ดังนั้นค่าสนับสนุนแบบสัมพัทธ์จึงนิยมใช้กับงานที่ต้องการนำการทำเหมืองรูปแบบไปประยุกต์ใช้ เช่น การแพทย์ การศึกษา การตลาด เป็นต้น ส่วนค่าสนับสนุนแบบสัมบูรณ์ถูกใช้เพื่อให้ง่ายต่อการแสดงขั้นตอนวิธีการทำเหมืองรูปแบบ ดังนั้นในหนังสือเล่มนี้จะใช้ค่าสนับสนุนแบบสัมบูรณ์เพื่อแสดงให้เห็นถึงกระบวนการการทำเหมืองรูปแบบในแบบต่างๆ แต่อย่างไรก็ตาม ถ้าผู้อ่านต้องการนำการทำเหมืองรูปแบบไปประยุกต์ใช้กับงานด้านต่างๆ ผู้อ่านก็สามารถกำหนดค่าสนับสนุนแบบสัมพัทธ์ได้

ค่าสนับสนุนเป็นตัววัดที่มีความสำคัญ เนื่องจากแสดงให้เห็นถึงจำนวนการเกิดของรูปแบบข้อมูล รูปแบบที่มีค่าสนับสนุนน้อยแสดงว่ามีโอกาสเกิดรูปแบบดังกล่าวน้อยในชุดข้อมูลนั้น จึงถือว่าเป็นรูปแบบที่ไม่น่าสนใจ เช่น ถ้าสินค้าใดมีคนซื้อน้อย แสดงว่าสินค้าดังกล่าวไม่ได้รับความสนใจ ดังนั้นสินค้าดังกล่าวจะไม่ถูกนำเสนอให้กับลูกค้า เป็นต้น ค่าสนับสนุนขั้นต่ำ (Minimum support threshold) จึงถูกกำหนดขึ้น เพื่อเป็นตัวคัดกรองเอาเฉพาะรูปแบบที่สนใจเท่านั้น งานในแต่ละด้านอาจจะกำหนดค่าสนับสนุนขั้นต่ำแตกต่างกัน

1.7 ค่าความเชื่อมั่น (Confidence) และความสำคัญของค่าความเชื่อมั่น

ค่าความเชื่อมั่นเป็นค่าหนึ่งที่มีความสำคัญในการทำเหมืองรูปแบบ เป็นค่าที่บ่งบอกถึงโอกาสการเกิดของข้อมูลในกฎความสัมพันธ์

ในกรณีที่ต้องการพิจารณาการเกิดร่วมกันของข้อมูล ค่าความเชื่อมั่นของกฎ $X \longrightarrow Y$ แสดงให้เห็นถึงโอกาสการเกิด *Y* เมื่อมี *X* โดยพิจารณาเป็นเปอร์เซ็นต์ของการเกิด และสามารถคำนวณได้จาก $supp(X \cup Y)/supp(X) \times 100$ เช่น ต้องการหากฎความสัมพันธ์ที่แสดงพฤติกรรมการซื้อสินค้าร่วมกัน

ของลูกค้าและไม่พิจารณาลำดับการซื้อ กฎความสัมพันธ์ รองเท้า $\longrightarrow$ เสื้อผ้า เป็นความสัมพันธ์ของการซื้อรองเท้าและเสื้อผ้าร่วมกัน ค่าความเชื่อมั่นของกฎดังกล่าวสามารถหาได้จาก

$$supp(\text{รองเท้า} \cup \text{เสื้อผ้า})/supp(\text{รองเท้า}) \times 100 = 3/3 = 100\%$$

แสดงให้เห็นว่าเมื่อซื้อรองเท้าแล้วจะมีโอกาสซื้อเสื้อผ้าร่วมด้วยถึง 100% เป็นต้น

ในกรณีที่พิจารณาลำดับการเกิดของข้อมูล ค่าความเชื่อมั่นของกฎ $X \longrightarrow Y$ หมายถึง โอกาสของการเกิด Y ตามหลัง X เช่น ต้องการหากฎความสัมพันธ์ที่แสดงพฤติกรรมการซื้อสินค้าร่วมกันของลูกค้าและพิจารณาลำดับการซื้อด้วย กฎความสัมพันธ์ รองเท้า $\longrightarrow$ เสื้อผ้า เป็นความสัมพันธ์ของการซื้อรองเท้าแล้วตามด้วยซื้อเสื้อผ้า ค่าความเชื่อมั่นของกฎสามารถหาได้จาก

$$supp(\text{รองเท้า} \rightarrow \text{เสื้อผ้า})/supp(\text{รองเท้า}) \times 100 = 2/3 = 67\%$$

แสดงให้เห็นว่าเมื่อซื้อรองเท้าแล้วมีโอกาสที่จะซื้อเสื้อผ้าตามหลังเท่ากับ 67%

ค่าความเชื่อมั่นเป็นตัวที่บ่งบอกถึงโอกาสของการเกิดข้อมูลที่สัมพันธ์กัน ถ้าค่าความเชื่อมั่นสูงแสดงว่ามีโอกาสเกิดข้อมูลที่สัมพันธ์กันสูง แต่ถ้าค่าความเชื่อมั่นต่ำแสดงว่ามีโอกาสเกิดข้อมูลที่สัมพันธ์กันน้อย เช่น ลูกค้ามักซื้อเสื้อผ้าหลังจากซื้อรองเท้าเสมอ แสดงว่าค่าความเชื่อมั่นสูง ดังนั้นพนักงานขายสินค้าควรจะแนะนำเสื้อผ้าหลังจากที่ลูกค้าซื้อรองเท้า เพื่อเพิ่มโอกาสในการขายสินค้า เป็นต้น ค่าความเชื่อมั่นขั้นต่ำ (Minimum confidence threshold) จึงเป็นตัวคัดกรองตัวหนึ่งที่ถูกนำมาใช้ในการกรองเอาเฉพาะรูปแบบความสัมพันธ์ที่น่าสนใจเท่านั้น งานในแต่ละด้านอาจจะกำหนดค่าความเชื่อมั่นขั้นต่ำที่แตกต่างกัน

1.8 ข้อมูลสำหรับการทำเหมืองรูปแบบ

ปัจจุบันข้อมูลถูกเก็บอยู่ในรูปแบบของฐานข้อมูลที่หลากหลาย ข้อมูลจากฐานข้อมูลดังกล่าวสามารถดึงหรือรวบรวม แล้วนำมาจัดอยู่ในรูปแบบของข้อมูลรายการเปลี่ยนแปลง (Transactional data) เพื่อนำไปใช้ในการวิเคราะห์ โดยแต่ละรายการเปลี่ยนแปลงจะประกอบไปด้วยหนึ่งเหตุการณ์ เช่น รายการสินค้าที่ซื้อในหนึ่งครั้ง การกดถูกใจ (Like) เพจของผู้ใช้หนึ่งคน ข้อความหนึ่งข้อความ ข้อมูลของผู้ป่วยหนึ่งคน เป็นต้น การทำเหมืองรูปแบบสามารถทำได้กับข้อมูลที่หลากหลาย เช่น ข้อความ มัลติมีเดีย กราฟ เป็นต้น โดยข้อมูลจะต้องเป็นข้อมูลแบบนามบัญญัติ (Nominal) หรือแบบอันดับ (Ordinal) ซึ่งข้อมูลแบบนามบัญญัติ หมายถึง ข้อมูลที่แบ่งเป็นกลุ่มเป็นพวก เช่น เพศ อาชีพ ศาสนา สีผิว เป็นต้น ส่วนข้อมูลแบบอันดับ หมายถึง ข้อมูลที่สามารถแบ่งเป็นกลุ่มได้และยังสามารถบอกอันดับที่ของความแตกต่างได้ เช่น เกรด อันดับของการแข่งขัน เป็นต้น

ในหนังสือเล่มนี้แยกข้อมูลสำหรับการทำเหมืองรูปแบบออกเป็น 2 ชนิด คือ ข้อมูลที่ไม่พิจารณาลำดับการเกิดของข้อมูลและข้อมูลที่พิจารณาลำดับการเกิดของข้อมูล โดยมีรายละเอียดดังนี้

1. ข้อมูลที่ไม่พิจารณาลำดับการเกิด หมายความว่า ข้อมูลใดเกิดก่อนหลังไม่มีความสำคัญ เรียกข้อมูลลักษณะนี้ว่า เซตรายการ เช่น รายการซื้อสินค้าในตารางที่ 1.3 ซึ่งแต่ละรายการเปลี่ยนแปลงประกอบไปด้วยรายการสินค้าที่ซื้อในหนึ่งครั้ง เช่น รายการเปลี่ยนแปลงที่ 1 เป็นข้อมูลการซื้อ สบู่ ยาสีฟัน และแชมพูพร้อมกันโดยไม่สนใจลำดับการซื้อ เป็นต้น

จากตัวอย่างข้อมูลในตารางที่ 1.3 ข้อมูลที่ใช้สำหรับการทำเหมืองรูปแบบเป็นข้อมูลชนิดเดียวกัน คือ สินค้า การทำเหมืองรูปแบบยังใช้กับข้อมูลที่ไม่ใช่ข้อมูลชนิดเดียวกันได้ เช่น จากตารางที่ 1.4 เป็นข้อมูลปัจจัยที่ใช้ในการวิเคราะห์โรคหัวใจ โดยปัจจัยประกอบไปด้วย อาชีพ วัย เพศ การออกกำลังกาย การสูบบุหรี่ เป็นต้น

ตารางที่ 1.3 การซื้อสินค้าของลูกค้า

รายการเปลี่ยนแปลง	รายการ
1	สบู่ ยาสีฟัน แชมพู
2	ผงซักฟอก ยาสีฟัน สบู่
3	แชมพู แปรงสีฟัน ยาสีฟัน
4	สบู่ แชมพู แปรงสีฟัน

ตารางที่ 1.4 ปัจจัยในการวิเคราะห์โรคหัวใจ

คนไข้	อาชีพ	วัย	เพศ	ออกกำลังกาย	การสูบบุหรี่	โรคหัวใจ
1	เกษตรกร	กลางคน	ชาย	ไม่เคย	ไม่สูบ	เป็น
2	อาจารย์	กลางคน	หญิง	ประจำ	สูบ	ไม่เป็น
3	พนักงานบริษัท	ชรา	ชาย	ไม่เคย	สูบ	เป็น
4	เกษตรกร	ชรา	หญิง	ประจำ	ไม่สูบ	ไม่เป็น

2. ข้อมูลที่พิจารณาลำดับการเกิดของข้อมูล แต่ละรายการจะต้องเรียงลำดับตามการเกิดของข้อมูล เรียกข้อมูลลักษณะนี้ว่า ลำดับเหตุการณ์ (Sequence) เช่น การเกิดปรากฏการณ์ธรรมชาติ การคลิกลิงก์ในเว็บไซต์ ลำดับการเกิดโรค ลำดับของคำ เป็นต้น ข้อมูลที่พิจารณาลำดับยังแบ่งออกเป็น 2 กลุ่ม คือ ข้อมูลเดี่ยวที่เกิดตามลำดับ กับ ข้อมูลกลุ่มที่เกิดตามลำดับ

ตารางที่ 1.5 แสดงตัวอย่างข้อมูลเดี่ยวที่เกิดตามลำดับ ซึ่งเป็นข้อมูลการคลิกลิงก์ในเว็บไซต์ แต่ละรายการเปลี่ยนแปลง หมายถึง การเข้าไปยังเว็บไซต์แต่ละครั้ง โดยในครั้งที่ 1 มีการคลิกลิงก์ A แล้วคลิกที่ลิงก์ B แล้วตามด้วยลิงก์ C ซึ่งเป็นการคลิกลิงก์ตามลำดับ

ส่วนตารางที่ 1.6 แสดงตัวอย่างข้อมูลกลุ่มที่เกิดตามลำดับ เช่น ข้อมูลการจ่ายยาให้ผู้ป่วยเพื่อรักษาโรคชนิดหนึ่ง โดยแต่ละรายการเปลี่ยนแปลง หมายถึง การจ่ายยาให้ผู้ป่วยแต่ละคน รายการที่อยู่ในวงเล็บเดียวกัน หมายถึง ยาที่จ่ายไปพร้อมกันในหนึ่งครั้งการรักษา (กลุ่มเดียวกัน) เช่น ผู้ป่วยคนที่หนึ่ง ครั้งแรกที่มารักษาได้รับยา A และ B พร้อมกัน ครั้งที่ 2 ได้รับยา A และ C พร้อมกัน แล้วต่อมาได้รับยาเฉพาะ D ในการรักษาครั้งที่ 3 เป็นต้น

นอกจากนี้แล้วการทำเหมืองรูปแบบสามารถขุดค้นรูปแบบที่น่าสนใจจากข้อมูลที่ประกอบไปด้วยข้อมูลทั้ง 2 ชนิด คือ ประกอบไปด้วยข้อมูลที่ไม่พิจารณาลำดับการเกิดและข้อมูลที่พิจารณาลำดับการเกิด เช่น ตัวอย่างในตารางที่ 1.7 เป็นประวัติของผู้ป่วยและประวัติการจ่ายยาให้ผู้ป่วย ซึ่งส่วนแรกเป็นข้อมูลพื้นฐานของผู้ป่วย ประกอบไปด้วย ภูมิลำเนา เพศ ระดับการศึกษา สถานภาพ อาชีพ ในส่วนนี้ไม่จำเป็นต้องพิจารณาลำดับการเกิด ส่วนที่สอง คือ ประวัติการจ่ายยา ซึ่งจะต้องพิจารณาลำดับการจ่ายยา เป็นต้น

ตารางที่ 1.5 การคลิกลิงก์ในเว็บไซต์

รายการเปลี่ยนแปลง	ลิงก์
1	(A B C)
2	(B C D F)
3	(A C D G)
4	(C D F G)

ตารางที่ 1.6 ประวัติการจ่ายยาให้ผู้ป่วย

รายการเปลี่ยนแปลง	ยา
1	< (A B) (A C) (D) >
2	< (A) (C D) (E) >
3	< (A) (C D) (F) >
4	< (C D) (E F) >

ตารางที่ 1.7 ประวัติการรักษาผู้ป่วย

รายการเปลี่ยนแปลง	ภูมิลำเนา	เพศ	ระดับการศึกษา	สถานภาพ	อาชีพ	ประวัติการจ่ายยา
1	มหาสารคาม	หญิง	ตรี	แต่งงาน	อาจารย์	< (A B) (A C) (D) >
2	มหาสารคาม	ชาย	ตรี	โสด	อาจารย์	< (A) (C D) (E) >
3	มหาสารคาม	ชาย	โท	แต่งงาน	อาจารย์	< (A) (C D) (F) >
4	ร้อยเอ็ด	ชาย	โท	โสด	เกษตรกร	< (C D) (E F) >

บทสรุป

การทำเหมืองรูปแบบเป็นการค้นหารูปแบบที่น่าสนใจ มีประโยชน์ หรือซ่อนอยู่ในฐานข้อมูลขนาดใหญ่ เป็นเทคนิคหนึ่งในการค้นหาองค์ความรู้จากฐานข้อมูล โดยผลลัพธ์ที่ได้จะอยู่ในรูปแบบที่ผู้ใช้สามารถเข้าใจได้ง่าย การทำเหมืองรูปแบบถูกนำเสนอครั้งแรกเพื่อค้นหากฎความสัมพันธ์ที่แสดงให้เห็นถึงรูปแบบการซื้อสินค้า ซึ่งประกอบไปด้วย 2 ขั้นตอนหลัก คือ การค้นหารูปแบบที่เกิดร่วมกันบ่อยและการสร้างกฎความสัมพันธ์

การค้นหารูปแบบที่เกิดร่วมกันบ่อยจำเป็นต้องทราบค่าสนับสนุนของแต่ละรูปแบบ เพื่อบ่งบอกถึงความถี่ของการเกิดรูปแบบ รูปแบบใดมีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ แสดงว่ารูปแบบดังกล่าวมีความถี่เพียงพอที่จะนำไปใช้ ส่วนการสร้างกฎความสัมพันธ์จำเป็นต้องทราบค่าความเชื่อมั่น เพื่อบ่งบอกถึงโอกาสการเกิดของข้อมูลในกฎความสัมพันธ์ กฎความสัมพันธ์ที่มีค่าความเชื่อมั่นมากกว่าหรือเท่ากับค่าความเชื่อมั่นขั้นต่ำ จะถือว่าเป็นกฎที่ยอมรับได้

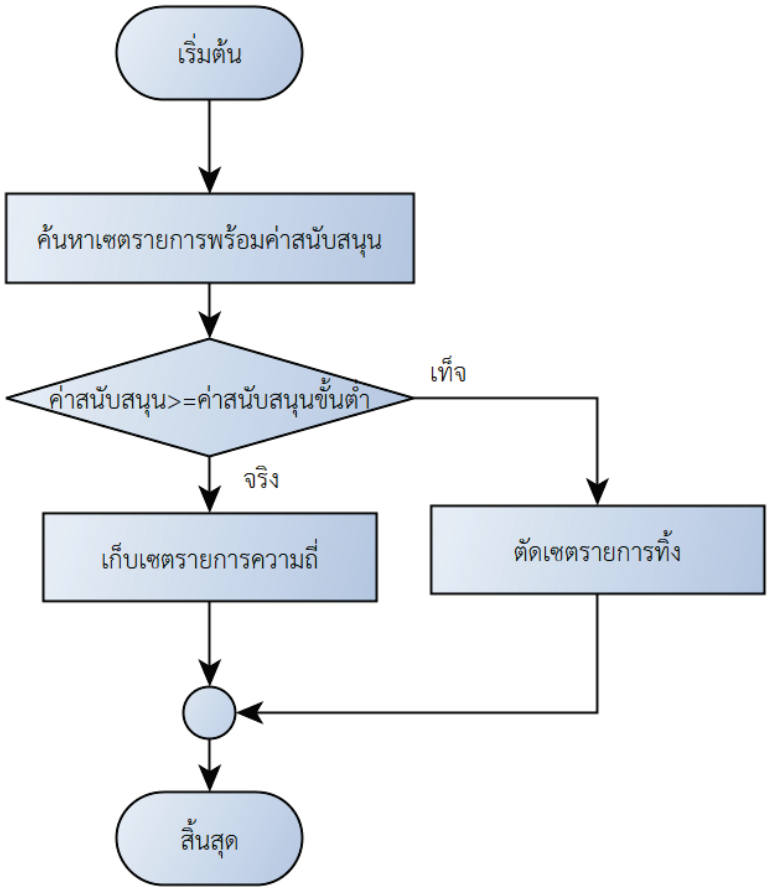
ข้อมูลที่ใช้ในการทำเหมืองรูปแบบ จะต้องเป็นข้อมูลแบบนามบัญญัติ ที่สามารถแบ่งเป็นกลุ่มเป็นพวกได้ หรือข้อมูลแบบอันดับที่สามารถแบ่งเป็นกลุ่มและยังสามารถบอกอันดับที่ของความแตกต่างได้ โดยสามารถค้นหารูปแบบที่น่าสนใจบนข้อมูลที่ไม่พิจารณาลำดับการเกิด และข้อมูลที่พิจารณาลำดับการเกิด รวมถึงข้อมูลที่มีส่วนที่ไม่พิจารณาลำดับการเกิดและส่วนที่พิจารณาลำดับการเกิด

แบบฝึกหัดท้ายบท

1. จงอธิบายความหมายของการค้นหาองค์ความรู้ในฐานข้อมูล
2. จงอธิบายกระบวนการหลักทั้งหมดในการทำเหมืองข้อมูล
3. จงอธิบายเทคนิคในการค้นหาองค์ความรู้ในฐานข้อมูลขนาดใหญ่แต่ละเทคนิค
4. จงอธิบายความหมายของการทำเหมืองรูปแบบ
5. จงยกตัวอย่างการประยุกต์ใช้กฎความสัมพันธ์
6. จงอธิบายความหมายค่าสนับสนุนและความสำคัญของค่าสนับสนุน
7. จงอธิบายความหมายค่าความเชื่อมั่นและความสำคัญของค่าความเชื่อมั่น
8. จงยกตัวอย่างข้อมูลที่ไม่ต้องพิจารณาลำดับการเกิด พร้อมกับอธิบายลักษณะของข้อมูล
9. จงยกตัวอย่างข้อมูลที่ต้องพิจารณาลำดับการเกิด พร้อมอธิบายลักษณะของข้อมูล
10. จงยกตัวอย่างข้อมูลที่ประกอบไปด้วย 2 ส่วน คือ ส่วนที่พิจารณาลำดับการเกิดของข้อมูลและส่วนที่ไม่ต้องพิจารณาลำดับการเกิดของข้อมูล พร้อมอธิบายลักษณะของข้อมูล

บทที่ 2
การทำเหมืองเซตรายการความถี่
(Frequent Itemset Mining)

การทำเหมืองเซตรายการความถี่เป็นขั้นตอนหนึ่งที่สำคัญในการสร้างกฎความสัมพันธ์ การทำเหมืองเซตรายการความถี่เป็นการค้นหาเซตรายการที่ปรากฏร่วมกันบ่อย กระบวนการโดยทั่วไปของการค้นหาเซตรายการความถี่แสดงดังรูปที่ 2.1 เริ่มจากการค้นหาเซตรายการและค่าสนับสนุน จากนั้นตรวจสอบเซตรายการว่ามีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำหรือไม่ ถ้าเซตรายการใดมีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ จะถือว่าเซตรายการดังกล่าวเป็นเซตรายการความถี่



รูปที่ 2.1 กระบวนการการทำเหมืองเซตรายการความถี่

การทำเหมืองเซตรายการความถี่ถูกนำเสนอครั้งแรกโดย Agrawal และ คณะ เพื่อใช้ในการค้นหาสินค้าที่มีการซื้อร่วมกันบ่อย ปัจจุบันการทำเหมืองเซตรายการความถี่ถูกนำมาประยุกต์ใช้ในด้านต่างๆ มากมาย เช่น การจำแนกรูปภาพ การวิเคราะห์การจราจร การวิเคราะห์ความคิดเห็นของลูกค้า

เป็นต้น มีการนำเสนอขั้นตอนวิธีต่างๆ เพื่อใช้ในการทำเหมืองเซตรายการความถี่ เช่น Apriori, FP-Growth, Eclat และ LCM เป็นต้น ทุกขั้นตอนวิธีให้ผลลัพธ์เดียวกันแตกต่างกันตรงวิธีการในการค้นหาเซตรายการความถี่และโครงสร้างที่ใช้ในการเก็บข้อมูลเพื่อค้นหาเซตรายการความถี่

ขั้นตอนวิธี FP-Growth เป็นขั้นตอนวิธีหนึ่งที่ได้รับความนิยมอย่างแพร่หลายและถูกนำไปประยุกต์ใช้ในด้านต่างๆ เช่น ด้านการรักษาพยาบาล การวิเคราะห์ข้อความ การวิเคราะห์การตลาด เป็นต้น เนื่องจากขั้นตอนวิธี FP-Growth สามารถลดปัญหาเรื่องการสร้างเซตรายการคู่แข่ง (Candidate) และลดปัญหาการอ่านข้อมูลจากฐานข้อมูลหลายครั้ง ดังนั้นในหนังสือเล่มนี้จะกล่าวถึงการทำเหมืองเซตรายการความถี่โดยใช้ขั้นตอนวิธี FP-Growth

2.1 ลักษณะข้อมูลที่ใช้ในการทำเหมืองเซตรายการความถี่

การทำเหมืองเซตรายการความถี่เป็นการขุดค้นเซตรายการที่เกิดร่วมกันบ่อยๆ โดยไม่สนใจลำดับของการเกิด สนใจแค่การเกิดร่วมกัน เช่น ลูกค้าซื้อเบียร์พร้อมกับผ้าอ้อม ไม่สนใจว่าลูกค้าจะซื้อเบียร์หรือผ้าอ้อมก่อน พิจารณาแค่ซื้อร่วมกัน เป็นต้น โดยทั่วไปข้อมูลที่ใช้ในการทำเหมืองเซตรายการความถี่ จะเป็นข้อมูลที่อยู่ในรูปแบบรายการเปลี่ยนแปลง ถ้าข้อมูลไม่อยู่ในรูปแบบรายการเปลี่ยนแปลงจำเป็นต้องเตรียมข้อมูลให้อยู่ในรูปแบบรายการเปลี่ยนแปลงก่อน การกำหนดความหมายของแต่ละรายการเปลี่ยนแปลงในแต่ละงานอาจจะแตกต่างกัน เช่น หนึ่งรายการเปลี่ยนแปลงสำหรับการซื้อสินค้าหมายถึง การซื้อสินค้าในหนึ่งใบเสร็จ หนึ่งรายการเปลี่ยนแปลงสำหรับการกดถูกใจเพจ หมายถึง การกดถูกใจเพจของผู้ใช้หนึ่งคน เป็นต้น โดยตัวอย่างลักษณะข้อมูลที่อยู่ในรูปแบบรายการเปลี่ยนแปลงแสดงได้ดังตารางที่ 2.1 เป็นตัวอย่างการซื้อสินค้า แต่ละรายการเปลี่ยนแปลง หมายถึง การซื้อสินค้าในหนึ่งใบเสร็จ

ในรายการเปลี่ยนแปลงที่ 1 หมายถึง ลูกค้าซื้อสินค้า A C และ F

ในรายการเปลี่ยนแปลงที่ 2 หมายถึง ลูกค้าซื้อสินค้า A B และ C

ในรายการเปลี่ยนแปลงที่ 3 หมายถึง ลูกค้าซื้อสินค้า A C D และ F

ในรายการเปลี่ยนแปลง 4 หมายถึง ลูกค้าซื้อสินค้า B D E และ F

ตารางที่ 2.1 ตัวอย่างชุดข้อมูลการซื้อสินค้า

รายการเปลี่ยนแปลง	เซตรายการ
1	(A C F)
2	(A B C)
3	(A C D F)
4	(B D E F)

2.2 นิยามที่เกี่ยวข้อง

กำหนดให้ $I = \{i_1 , i_2 , \dots , i_m\}$ คือ เซตของรายการ (Item) ทั้งหมดในชุดข้อมูล และกำหนดให้ $T = \{t_1 , t_2 , \dots t_m\}$ คือ เซตของรายการเปลี่ยนแปลงทั้งหมดในชุดข้อมูล แต่ละรายการเปลี่ยนแปลง t_i ประกอบไปด้วยเซตย่อยของ I

กำหนดให้ X คือ เซตรายการ (Itemset) โดยที่ $X \subseteq I$ และ $g(X)$ แทน รายการเปลี่ยนแปลงที่มีเซตรายการ X จำนวนรายการเปลี่ยนแปลงที่มีเซตรายการ X แทนด้วย $|g(X)|$

จากชุดข้อมูลในตารางที่ 2.1 จะเห็นได้ว่า I = {A, B, C, D, E, F} และ T= {1, 2, 3, 4} ถ้าสนใจรายการ A และ C จะได้เซตรายการ X= (A C) ซึ่งปรากฏในรายการเปลี่ยนแปลงที่ 1 2 และ 3 ดังนั้น $|g(X)| = |\{1,2,3\}| = 3$

นิยามที่ 2.1 ความยาวของเซตรายการ X คือ จำนวนรายการที่ปรากฏอยู่ในเซตรายการ X

ตัวอย่างที่ 2.1 เซตรายการ (A C) มีความยาวเท่ากับ 2 เนื่องจากเซตรายการ (A C) ประกอบไปด้วยรายการ A และ C

นิยามที่ 2.2 ค่าสนับสนุนของเซตรายการ X คือ ความถี่ของการเกิด X หรือจำนวนรายการเปลี่ยนแปลงที่พบ X ($|g(X)|$)

ตัวอย่างที่ 2.2 จากตารางที่ 2.1 เซตรายการ (A C) ปรากฏอยู่ในรายการเปลี่ยนแปลงที่ 1 2 และ 3 ดังนั้นค่าสนับสนุนมีค่าเท่ากับ 3

นิยามที่ 2.3 เซตรายการความถี่ คือ เซตรายการที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ

ตัวอย่างที่ 2.3 ถ้ากำหนดค่าสนับสนุนขั้นต่ำให้มีค่าเท่ากับ 2 เซตรายการ (A C) เป็นเซตรายการความถี่ เนื่องจากค่าสนับสนุนของเซตรายการ (A C) มีค่าเท่ากับ 3 ซึ่งมีค่ามากกว่าค่าสนับสนุนขั้นต่ำ

2.3 ขั้นตอนวิธีสำหรับการทำเหมืองเซตรายการความถี่

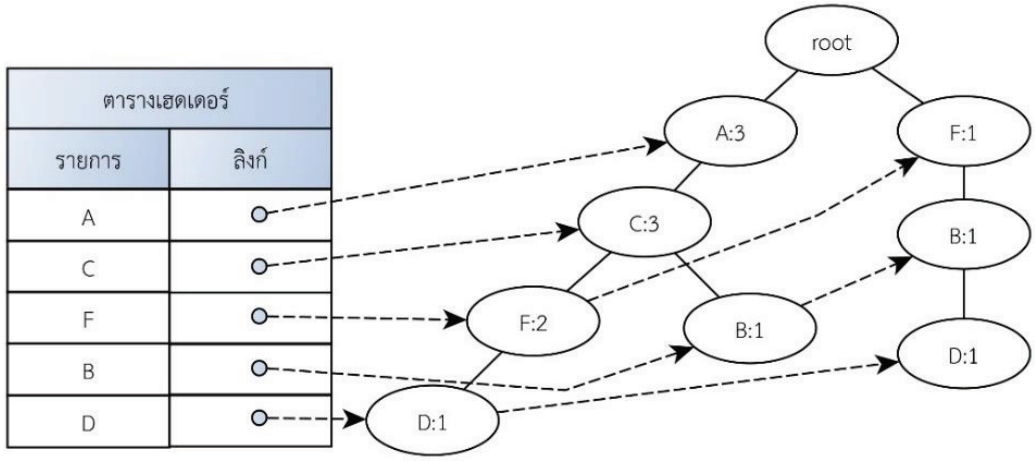
ปัจจุบันมีการนำเสนอขั้นตอนวิธีต่างๆ เพื่อขุดค้นเซตรายการความถี่ ขั้นตอนวิธีแรกที่นำเสนอการทำเหมืองเซตรายการความถี่ คือ Apriori ซึ่งทำการสร้างเซตรายการความถี่ที่มีความยาว k จากการขยายเซตรายการความถี่ที่มีความยาว k-1 จากนั้นทำการตรวจสอบว่าเซตรายการที่ขยายมีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำหรือไม่ เซตรายการที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำจะถูกเก็บไว้เพื่อทำการขยายต่อไป ส่วนเซตรายการที่มีค่าสนับสนุนน้อยกว่าค่าสนับสนุนขั้นต่ำจะถูกตัดทิ้งไป จะเห็นได้ว่าขั้นตอนวิธี Apriori ทำการสร้างเซตรายการคู่แข่งก่อน แล้วค่อยตรวจสอบว่าผ่านค่าสนับสนุนขั้นต่ำหรือไม่ ซึ่งทำให้ต้องอ่านฐานข้อมูลหลายครั้งเพื่อขยายเซตรายการและคำนวณค่าสนับสนุน เพื่อแก้ปัญหาดังกล่าว ได้มีการนำเสนอขั้นตอนวิธี FP-Growth ซึ่งใช้โครงสร้างข้อมูลที่เรียกว่า FP-tree (Frequent pattern tree) ในการเก็บเซตรายการความถี่ ขั้นตอนการสร้างเซตรายการความถี่ของขั้นตอนวิธี FP-Growth แบ่งออกเป็น 2 ขั้นตอนหลักๆ คือ การสร้าง FP-tree และการค้นหาเซตรายการความถี่จาก FP-tree โดยมีรายละเอียดดังต่อไปนี้

ขั้นตอนที่ 1 การสร้าง FP-tree

โครงสร้าง FP-tree เป็นโครงสร้างต้นไม้ ซึ่งแต่ละโหนดประกอบไป 4 ฟิลด์ คือ

- item-name: รายการ
- count: ความถี่ของรายการ
- parent-link: ลิงก์ที่เชื่อมไปยังโหนดแม่
- node-link: ลิงก์ที่เชื่อมไปยังโหนดอื่นใน FP-tree ที่มีรายการเหมือนกัน

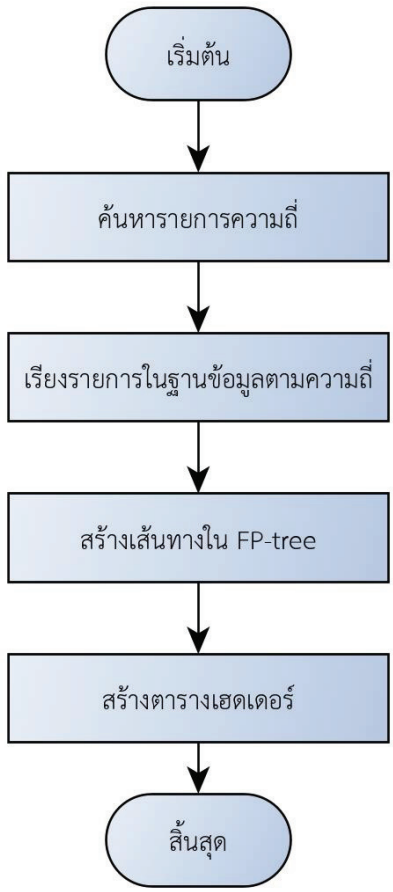
นอกจากนี้ยังมีตารางเฮดเดอร์ (Header table) ซึ่งประกอบไปด้วย รายการความถี่ และลิงก์ที่เชื่อมไปยังโหนดแรกที่มีรายการความถี่นั้นใน FP-tree เช่น รายการความถี่ F ในตารางเฮดเดอร์ลิงก์ไปยังโหนด F โหนดแรกใน FP-tree เป็นต้น โดยตัวอย่างโครงสร้าง FP-tree และตารางเฮดเดอร์แสดงได้ดังรูปที่ 2.2 ขั้นตอนในการสร้าง FP-tree แสดงได้ดังรูปที่ 2.3 และมีรายละเอียดดังนี้



หมายถึง ลิงก์ที่เชื่อมโหนดที่มีรายการเหมือนกัน

หมายถึง ลิงก์ที่เชื่อมโหนดแม่

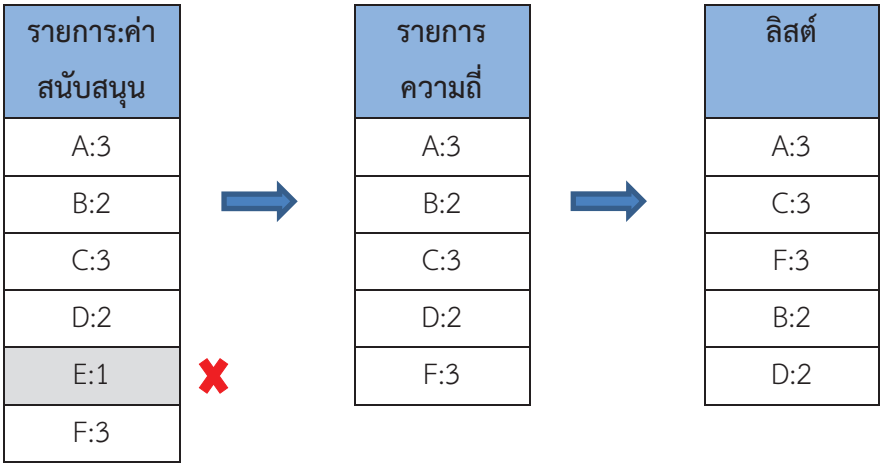
รูปที่ 2.2 ตัวอย่าง FP-tree และตารางเฮดเดอร์



รูปที่ 2.3 ขั้นตอนการสร้าง FP-tree

1. ทำการอ่านฐานข้อมูลครั้งแรกเพื่อค้นหารายการความถี่ (เซตรายการที่มีความยาว 1) โดยพิจารณารายการที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ จากนั้นเรียงลำดับรายการตามค่าสนับสนุนจากมากไปหาน้อยใส่ไว้ในลิสต์ (List)

ตัวอย่างที่ 2.4 ถ้ากำหนดค่าสนับสนุนขั้นต่ำให้มีค่าเท่ากับ 2 จากตารางที่ 2.1 จะได้รายการความถี่ทั้งหมด 5 รายการ คือ A:3, B:2, C:3, D:2 และ F:3 เมื่อเรียงลำดับตามค่าสนับสนุนจากมากไปน้อยจะได้รายการในลิสต์ดังรูปที่ 2.4 ค. (ในหนังสือเล่มนี้เซตรายการและค่าสนับสนุนของเซตรายการเขียนอยู่ในรูป เซตรายการ: ค่าสนับสนุน)



รูปที่ 2.4 รายการความถี่

2. อ่านฐานข้อมูลครั้งที่ 2 เพื่อสร้าง FP-tree โดยกำหนดให้โหนดราก (Root node) คือ Null อ่านข้อมูลทีละรายการเปลี่ยนแปลง เรียงเซตรายการตามรายการความถี่ที่อยู่ในลิสต์ (ดังตัวอย่างในตารางที่ 2.2) แล้วนำเซตรายการที่เรียงแล้วไปสร้าง FP-tree

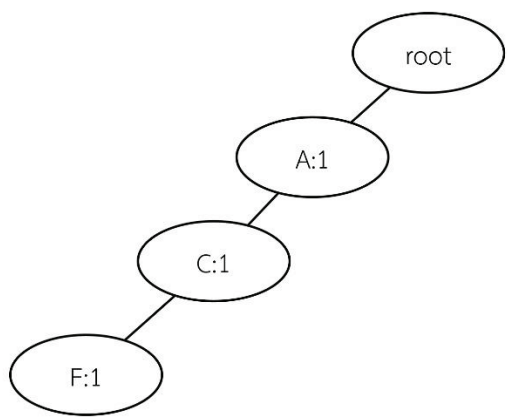
ตารางที่ 2.2 ตัวอย่างเซตรายการที่เรียงแล้ว

รายการเปลี่ยนแปลง	เซตรายการ
1	(A C F)
2	(A C B)
3	(A C F D)
4	(F B D)

3. ทำการสร้าง FP-tree โดยมีเงื่อนไขว่า ถ้าไม่มีเส้นทางของเซตรายการนั้นใน FP-tree ให้เพิ่มโหนดใหม่เข้าไป (โดยพิจารณาเส้นทางตั้งแต่โหนดราก) แล้วกำหนดค่าความถี่เริ่มต้นของโหนดเป็น 1 แต่ถ้ามีเส้นทางของเซตรายการใน FP-tree อยู่แล้ว ให้เพิ่มค่าความถี่ของโหนดในเส้นทางนั้นอีก 1 และถ้ามีโหนดที่มีรายการเหมือนกันแต่อยู่คนละเส้นทาง จะต้องทำการเชื่อมโหนดที่มีรายการเหมือนกัน

ตัวอย่างที่ 2.5 จากตารางที่ 2.2 เมื่ออ่านเซตรายการที่อยู่ในรายการเปลี่ยนแปลงที่ 1 จะได้ FP-tree ดังรูปที่ 2.5 โดยแต่ละรายการจะถูกนำไปสร้างโหนดใหม่ ทุกโหนดจะมีค่าความถี่เท่ากับ 1 เนื่องจากเป็นการเพิ่มเส้นทางใหม่เข้าไป

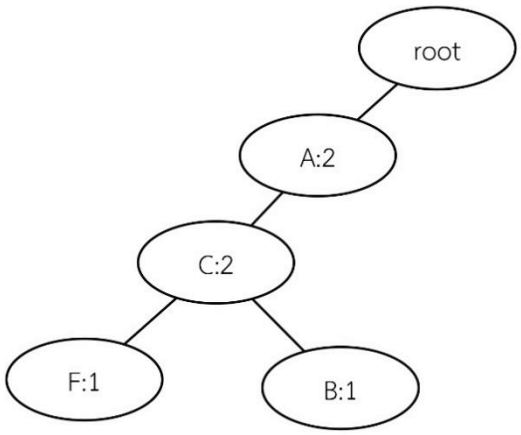
รายการเปลี่ยนแปลง	เซตรายการที่เรียงแล้ว
1	(A C F)



รูปที่ 2.5 โครงสร้าง FP-tree เมื่ออ่านรายการเปลี่ยนแปลงที่ 1

เมื่ออ่านเซตรายการในรายการเปลี่ยนแปลงที่ 2 แล้ว พบว่ามีเส้นทางของ A และ C ใน FP-tree แล้ว ดังนั้นจึงเพิ่มค่าความถี่ของโหนด A และ C อีก 1 จากนั้นจึงเพิ่มโหนด B เข้าไปเป็นโหนดใหม่ เนื่องจากไม่มีโหนด B ในเส้นทางดังกล่าว จะได้ FP-tree ดังรูปที่ 2.6

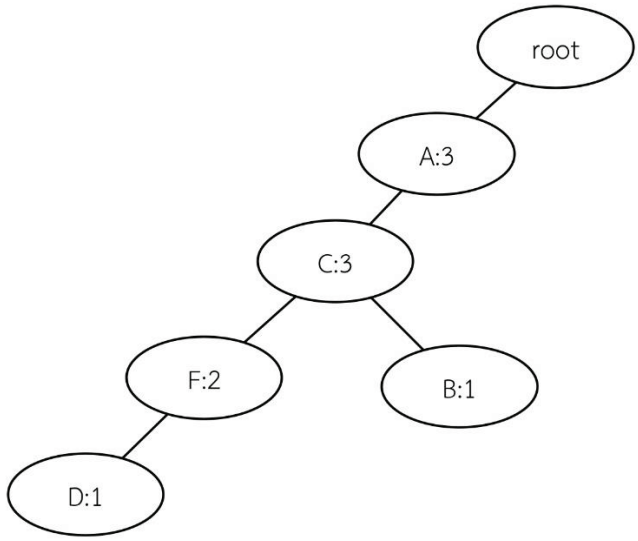
รายการเปลี่ยนแปลง	เซตรายการที่เรียงแล้ว
2	(A C B)



รูปที่ 2.6 โครงสร้าง FP-tree เมื่ออ่านรายการเปลี่ยนแปลงที่ 2

เมื่ออ่านเซตรายการในรายการเปลี่ยนแปลงที่ 3 แล้วพบว่ามีเส้นทางของ A C F ใน FP-tree แล้ว ดังนั้นจึงเพิ่มค่าความถี่ของโหนด A C และ F อีก 1 จากนั้นจึงเพิ่มโหนด D เข้าไปเป็นโหนดใหม่ ซึ่งจะได้ FP-tree ดังรูปที่ 2.7

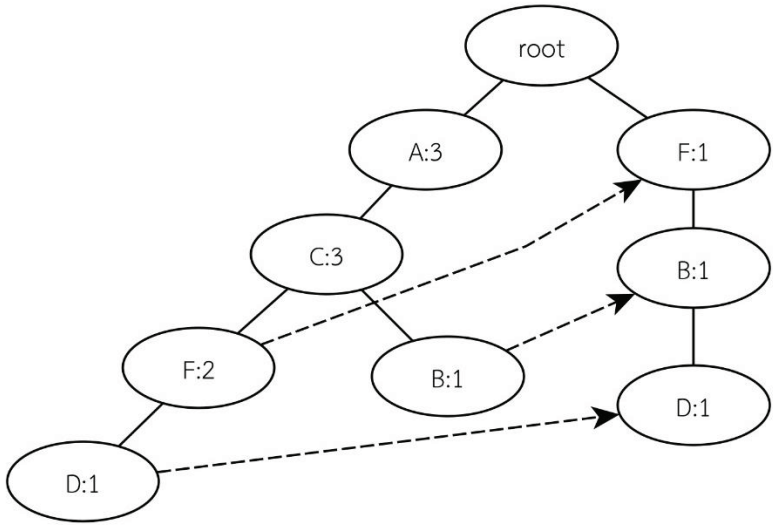
รายการเปลี่ยนแปลง	เซตรายการที่เรียงแล้ว
3	(A C F D)



รูปที่ 2.7 โครงสร้าง FP-tree เมื่ออ่านรายการเปลี่ยนแปลงที่ 3

เมื่ออ่านเซตรายการในรายการเปลี่ยนแปลงที่ 4 แล้วพบว่าไม่มีเส้นทางที่เริ่มจากโหนด F ดังนั้น จึงเพิ่มโหนด F ต่อจากรากโหนด แล้วเพิ่มโหนด B และ D ต่อจากโหนด F ดังรูปที่ 2.8 จากนั้นทำการเชื่อมโหนดที่มีรายการเดียวกัน เช่น เชื่อมโหนด F แรกกับโหนด F ตัวที่ 2 เข้าด้วยกัน และเชื่อมโหนด D แรกกับโหนด D ตัวที่ 2 เป็นต้น

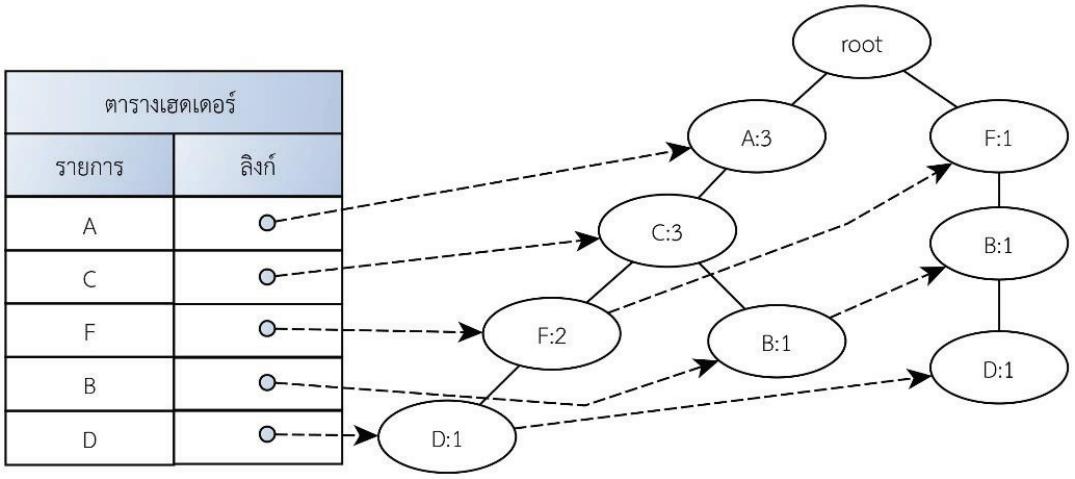
รายการเปลี่ยนแปลง	เซตรายการที่เรียงแล้ว
4	(F B D)



รูปที่ 2.8 โครงสร้าง FP-tree เมื่ออ่านรายการเปลี่ยนแปลงที่ 4

4. สร้างตารางเฮดเดอร์เพื่อให้ง่ายในการท่อง FP-tree โดยในตารางเฮดเดอร์ ประกอบไปด้วย 2 ฟิลด์ คือ

- item: รายการความถี่
- header of node-links: ลิงก์ที่เชื่อมไปยังโหนดแรกใน FP-tree ที่มีรายการเหมือนกันในตารางเฮดเดอร์ เช่น ในรูปที่ 2.9 รายการ A ในตารางเฮดเดอร์เชื่อมไปยังโหนด A ใน FP-tree รายการ F ในตารางเฮดเดอร์เชื่อมไปยังโหนด F แรกใน FP-tree เป็นต้น



รูปที่ 2.9 โครงสร้าง FP-tree ที่สมบูรณ์

ขั้นตอนที่ 2 การค้นหาเซตรายการความถี่

เมื่อสร้าง FP-tree เสร็จแล้ว ทำการขุดค้นเซตรายการความถี่จาก FP-tree โดยไม่ต้องอ่านฐานข้อมูลอีก การขุดค้นเซตรายการความถี่จาก FP-tree มีหลักการทำงานดังนี้

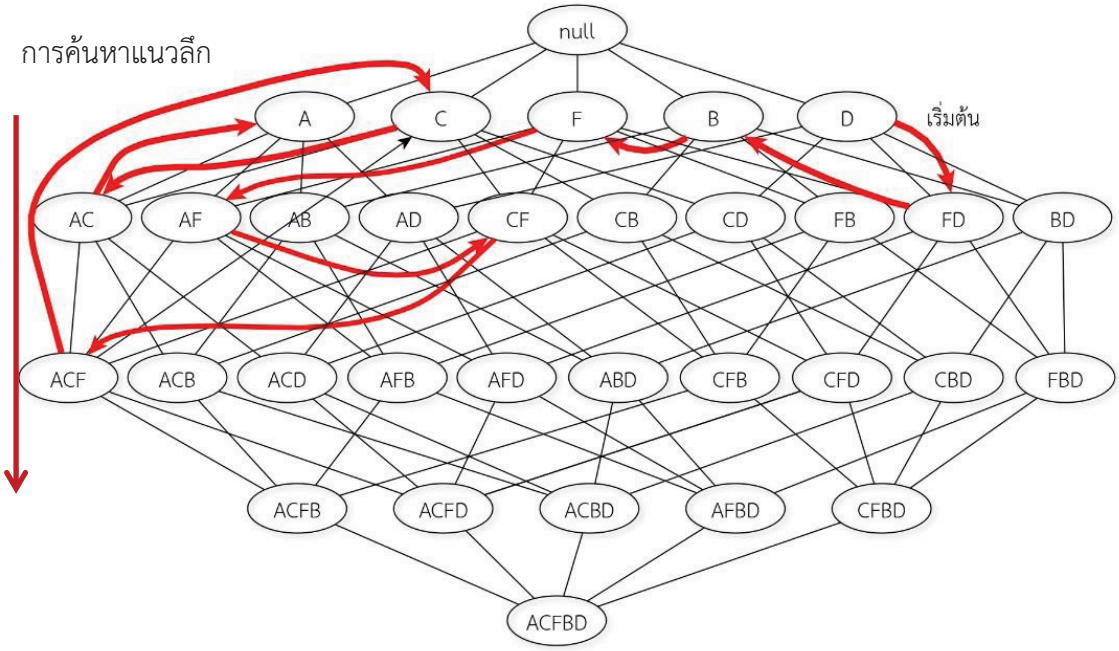
- พิจารณาขยายรายการความถี่ที่อยู่ในตารางเฮดเดอร์จากล่างขึ้นบน เช่น จากตารางเฮดเดอร์ในรูปที่ 2.9 รายการแรกที่พิจารณา คือ D และตามด้วย B, F, C และ A ตามลำดับ

- ทำการขยายรายการความถี่ไปเรื่อยๆ จนกว่าจะขยายรายการความถี่ในตารางเฮดเดอร์หมดทุกรายการ

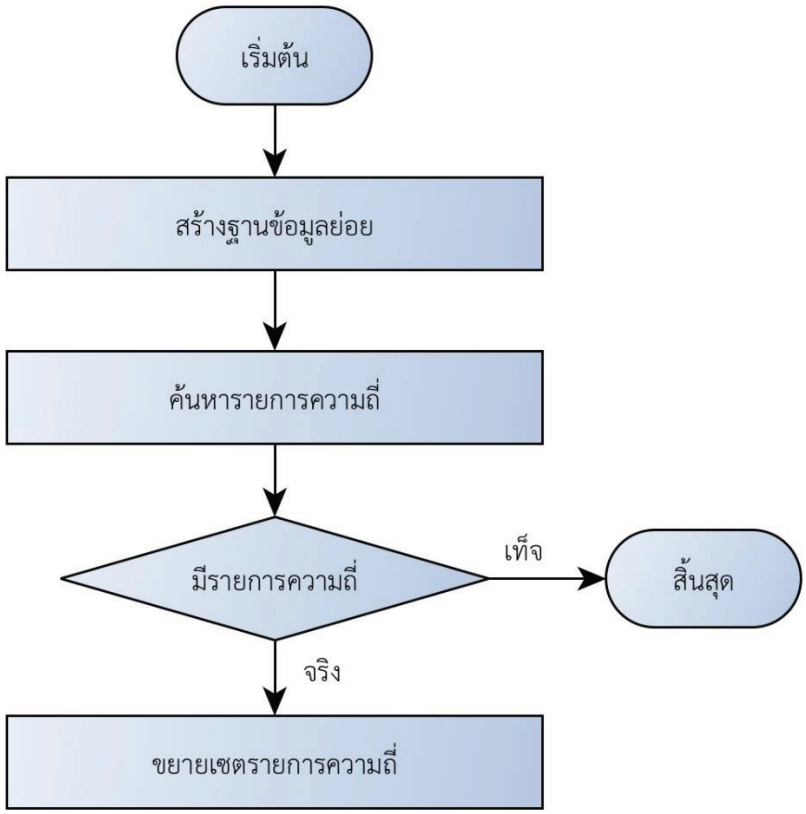
- ค้นหาเซตรายการความถี่โดยใช้หลักการค้นหาแนวลึก (Depth first search) รายการความถี่ความยาว k ได้จากการขยายเซตรายการความถี่ความยาว k-1 เช่น ขยายรายการ D ไปเรื่อยๆ จนกว่าจะไม่สามารถขยายได้ แล้วย้อนกลับมาขยาย B ดังรูปที่ 2.10 แสดงลำดับการค้นหาเซตรายการความถี่ (เส้นสีแดง)

- ทำการสร้างฐานข้อมูลย่อยเพื่อค้นหารายการความถี่ที่จะนำมาขยายเซตรายการความถี่

แต่ละเซตรายการความถี่มีขั้นตอนการขยายดังรูปที่ 2.11 ซึ่งสามารถอธิบายรายละเอียดได้ดังต่อไปนี้



รูปที่ 2.10 เซตรายการที่เป็นไปได้ทั้งหมด

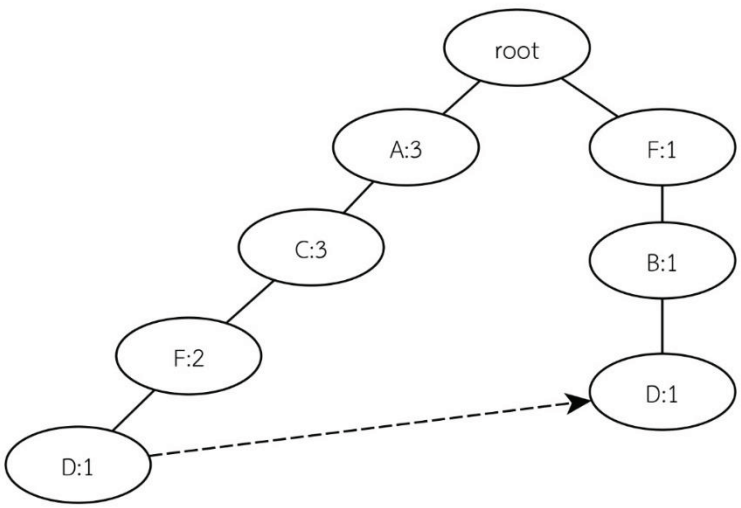


รูปที่ 2.11 ขั้นตอนการขยายเซตรายการความถี่

<u>ขั้นตอนที่ 2.1</u> สร้างฐานข้อมูลย่อย (Conditional sub-database) ของเซตรายการความถี่ *X* โดยทำการพิจารณาเส้นทางที่อยู่ก่อนเซตรายการ *X* (Prefix path sub-tree)

ตัวอย่างที่ 2.6 จากรูปที่ 2.9 รายการแรกที่พิจารณา คือ D ทำการพิจารณาเส้นทางที่มีรายการ D เป็นโหนดสุดท้าย ซึ่งมี 2 เส้นทาง คือ (A, C, F, D) และ (F, B, D) (ดังรูปที่ 2.12) ในแต่ละเส้นทางจะพิจารณาเส้นทางที่เกิดก่อนโหนด D เพื่อสร้างฐานข้อมูลย่อย ซึ่งจะเห็นได้ว่า จากเส้นทางแรกได้รูปแบบ (A C F) มีความถี่เท่ากับ 1 (นับเฉพาะความถี่ที่เกิดร่วมกับรายการ D ในเส้นทางนี้) และเส้นทางที่ 2 ได้รูปแบบ (F B) และมีความถี่เท่ากับ 1

ดังนั้นในฐานข้อมูลย่อยของรายการ D ประกอบไปด้วย {(A C F):1, (F B):1}



รูปที่ 2.12 เส้นทางที่มีโหนด D ต่อท้าย

<u>ขั้นตอนที่ 2.2</u> ทำการค้นหารายการความถี่ โดยการพิจารณารายการในฐานข้อมูลย่อยว่า มีรายการใดที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำหรือไม่ รายการที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ จะถือเป็นรายการความถี่ ถ้าไม่มีรายการความถี่จะหยุดการขยายเซตรายการความถี่ *X*

ตัวอย่างที่ 2.7 ฐานข้อมูลย่อยของรายการ D ประกอบไปด้วย {(A C F):1, (F B):1} พบว่ามีรายการทั้งหมด 4 รายการ และมีค่าสนับสนุนดังนี้ (A):1 (B):1 (C):1 และ (F):2 ซึ่งมีรายการเดียวที่ผ่านค่าสนับสนุนขั้นต่ำ คือ (F):2 ดังนั้นรายการความถี่มีรายการเดียว คือ F

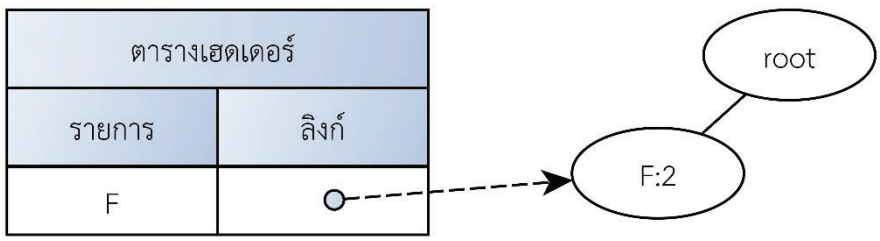
<u>ขั้นตอนที่ 2.3</u> ถ้ามีรายการความถี่ ให้ขยายเซตรายการความถี่ *X* กับรายการความถี่ *i* ทั้งหมด และกำหนดค่าสนับสนุนให้มีค่าเท่ากับค่าสนับสนุนของรายการความถี่ *i*

ตัวอย่างที่ 2.8 รายการความถี่ที่ได้จากขั้นตอนที่ 2 มีรายการเดียว คือ (F): 2 ดังนั้น ทำการขยายเซตรายการความถี่ โดยการรวมกันระหว่าง F กับ D เป็น (FD):2 (ค่าสนับสนุนของเซตรายการความถี่ (FD) พิจารณาจากค่าสนับสนุนของรายการ F)

จากนั้นวนกลับไปสร้าง FP-tree ตามขั้นตอนการสร้าง FP-tree โดยใช้ฐานข้อมูลย่อยในการสร้าง FP-tree และทำการขุดค้นเซตรายการความถี่ตามขั้นตอนที่ 1-3 วนทำซ้ำลักษณะนี้ไปเรื่อยๆ จนกว่าจะขยายรายการที่อยู่ในตารางเฮดเดอร์ครบทุกรายการ

ตัวอย่างที่ 2.9 จากฐานข้อมูลย่อยของ D คือ {(A C F):1, (F B):1} มีรายการความถี่ คือ F ดังนั้น ข้อมูลที่จะนำไปสร้าง FP-tree มีแค่ {(F):2} เมื่อสร้าง FP-tree จะได้ดังรูปที่ 2.13 ซึ่งจะเห็นได้ว่า ไม่มีเส้นทางที่เกิดก่อน F ดังนั้นไม่มีฐานข้อมูลย่อย ไม่มีรายการความถี่ จึงหยุดการค้นหา

สรุปได้ว่ารายการ D สามารถสร้างเซตรายการความถี่ได้ทั้งหมด 2 เซตรายการ คือ (D):2 และ (FD):2



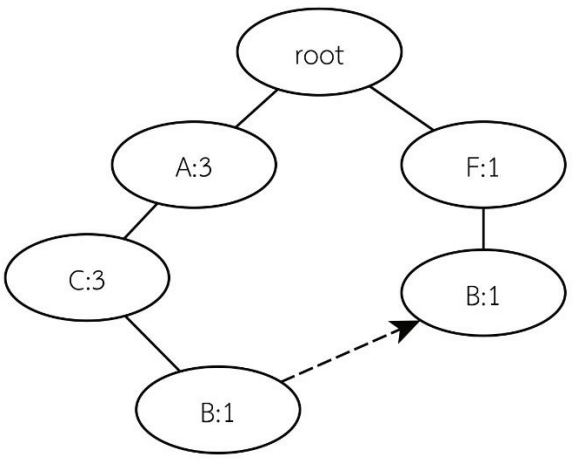
รูปที่ 2.13 FP-tree ย่อยของ D

เมื่อรายการ D ไม่สามารถขยายต่อได้ จะกลับไปพิจารณารายการที่อยู่ในตารางเฮดเดอร์ตัวถัดไป (ในรูปที่ 2.9)

<u>รายการที่ 2</u> ที่พิจารณา คือ B (พิจารณารายการที่อยู่ในตารางเฮดเดอร์จากล่างขึ้นข้างบน) โดยทำการพิจารณาเส้นที่มีโหนด B เป็นโหนดสุดท้าย ซึ่งมี 2 เส้นทาง คือ (A, C, B) และ (F, B) (ดังรูปที่ 2.14)

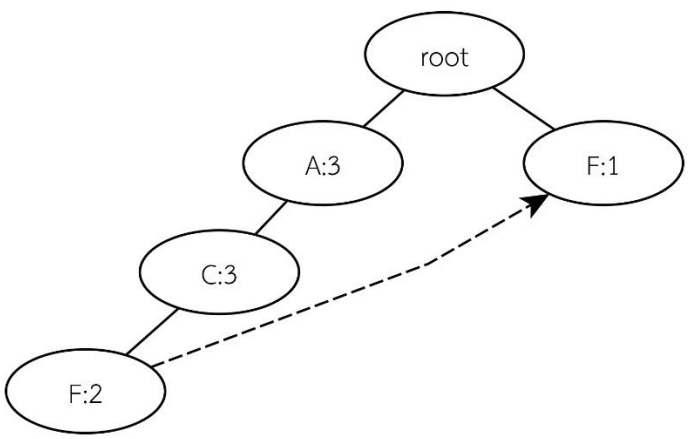
ในแต่ละเส้นทางจะพิจารณาเส้นทางที่เกิดก่อนโหนด B เพื่อสร้างฐานข้อมูลย่อย ซึ่งจะเห็นได้ว่าจากเส้นทางแรกได้รูปแบบ (A C) มีความถี่เท่ากับ 1 (นับเฉพาะความถี่ที่เกิดร่วมกับรายการ B ในเส้นทางนี้) และเส้นทางที่ 2 ได้รูปแบบ (F) และมีความถี่เท่ากับ 1 ดังนั้นในฐานข้อมูลย่อยของรายการ B ประกอบไปด้วย {(A C):1, (F):1} จากนั้นทำการค้นหารายการความถี่ ซึ่งพบว่าไม่มีรายการที่ผ่านค่าสนับสนุนขั้นต่ำ ดังนั้นจึงหยุดการค้นหา

สรุปได้ว่ารายการ B สามารถสร้างเซตรายการความถี่ได้ทั้งหมด 1 เซตรายการ คือ (B):2



รูปที่ 2.14 เส้นทางที่มีโหนด B ต่อท้าย

<u>รายการที่ 3</u> ที่พิจารณา คือ F โดยทำการพิจารณาเส้นทางที่มีโหนด F ต่อท้าย ซึ่งมี 2 เส้นทาง คือ (A, C, F) และ (F) (ดังรูปที่ 2.15) และได้ 1 รูปแบบจากเส้นทางแรก คือ (A C) มีความถี่เท่ากับ 2 (นับเฉพาะความถี่ที่เกิดร่วมกับรายการ F ในเส้นทางนี้) ส่วนเส้นทางที่ 2 ไม่มีรูปแบบที่อยู่ก่อน F ดังนั้นในฐานข้อมูลย่อยของรายการ F ประกอบไปด้วย {(A C):2}

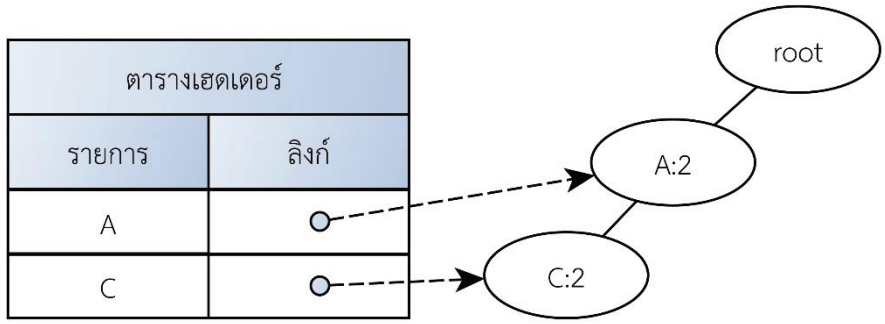


รูปที่ 2.15 เส้นทางที่มีโหนด F ต่อท้าย

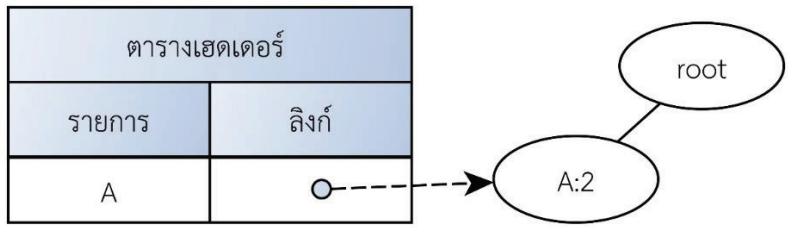
จากนั้นทำการหารายการความถี่ ซึ่งพบว่ามี 2 รายการที่ผ่านค่าสนับสนุนขั้นต่ำ คือ (A):2, (C):2 ทำการสร้างเซตรายการความถี่ โดยการรวมกันระหว่าง F กับ A และ F กับ C จะได้เซตรายการความถี่ (AF):2 และ (CF):2 (ค่าสนับสนุนของเซตรายการ (AF) จะเท่ากับค่าสนับสนุนของ A และค่าสนับสนุนของเซตรายการ (CF) จะเท่ากับค่าสนับสนุนของ C)

ทำการสร้าง FP-tree จากฐานข้อมูลย่อย ซึ่งมีแค่เส้นทางเดียว คือ (A, C) (ดังรูปที่ 2.16) แล้วทำการพิจารณารายการที่อยู่ตารางเฮดเดอร์จากล่างขึ้นบน ซึ่งรายการแรกที่พิจารณา คือ รายการ C ซึ่งเส้นทางที่มีโหนด C ต่อท้าย คือ (A, C)

ดังนั้นฐานข้อมูลย่อยของ CF คือ {(A):2} และรายการความถี่ที่ได้ คือ A ดังนั้นนำ A ไปรวมกับ CF จะได้เซตรายการความถี่ (ACF):2 และเมื่อสร้าง FP-tree ย่อยจากฐานข้อมูลย่อยของ ACF ปรากฏว่าไม่มีเส้นทางที่อยู่ก่อนดังรูปที่ 2.17 ดังนั้นจึงหยุดการค้นหาเพราะไม่มีรายการความถี่ จากนั้นย้อนกลับไปพิจารณารายการ A ในตารางเฮดเดอร์ (ในรูปที่ 2.16) ไม่มีเส้นทางที่อยู่ก่อน A ดังนั้นจึงหยุดการค้นหา



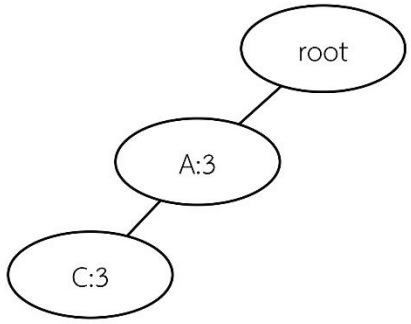
รูปที่ 2.16 FP-tree ย่อยของ F



รูปที่ 2.17 FP-tree ย่อยของ CF

สรุปได้ว่ารายการ F สามารถสร้างเซตรายการความถี่ได้ทั้งหมด 4 เซตรายการ คือ (F):3, (AF):2, (CF):2, (ACF):2

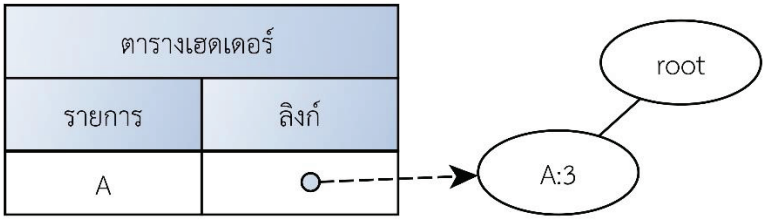
<u>รายการที่ 4</u> ที่พิจารณา คือ C โดยทำการพิจารณาเส้นทางที่โหนด C ต่อท้าย ซึ่งมี 1 เส้นทาง คือ (A, C) ดังรูปที่ 2.18 และรูปแบบที่ได้ คือ (A):3 ดังนั้นในฐานข้อมูลย่อยของรายการ C ประกอบไปด้วย {(A):3} และรายการความถี่คือ A สามารถสร้างเซตรายการความถี่โดยการรวมกันระหว่าง A กับ C เป็น (AC):3



รูปที่ 2.18 เส้นทางที่มีโหนด C ต่อท้าย

จากนั้นทำการสร้าง FP-tree จากฐานข้อมูลย่อย {(A):3} ดังรูปที่ 2.19 ซึ่งพบว่ามีรายการเดียวที่ผ่านค่าสนับสนุนขั้นต่ำ คือ (A):3 ดังนั้น FP-tree ย่อยมีแค่เส้นทางเดียวคือ (A) ไม่มีเส้นทางที่อยู่ก่อน ดังนั้นจึงหยุดการค้นหา

สรุปได้ว่ารายการ C สามารถสร้างเซตรายการความถี่ได้ทั้งหมด 2 เซตรายการ คือ (C):3, (AC):3



รูปที่ 2.19 FP-tree ย่อยของ C

<u>รายการที่ 5</u> ที่พิจารณา คือ A ซึ่งมีเส้นทางเดียวและไม่มีเส้นทางที่อยู่ก่อน (ดังรูปที่ 2.9) ดังนั้นจึงหยุดการค้นหา

สรุปได้ว่ารายการ A สามารถสร้างเซตรายการความถี่ได้ทั้งหมด 1 เซตรายการ คือ (A):3

เซตรายการความถี่ที่ได้จากการขยายรายการความถี่แสดงได้ดังตารางที่ 2.3 ซึ่งประกอบไปด้วยเซตรายการความถี่ทั้งหมด 10 เซตรายการ คือ (D):2, (FD):2, (B):2, (F):3, (AF):2, (CF):2, (ACF):2, (C):3, (AC):3, (A):3

ตารางที่ 2.3 ผลการทำเหมืองเซตรายการความถี่

รายการความถี่	ฐานข้อมูลย่อย	เซตรายการความถี่
D	{(A C F) :1, (F B) :1}	(D):2, (FD):2
B	{(A C):1, (F):1}	(B):2
F	{(A C):2}	(F):3, (AF):2, (CF):2, (ACF):2
C	{(A):3}	(C):3, (AC):3
A	∅	(A):3

2.4 การทำเหมืองเซตรายการแบบอื่น

ขั้นตอนวิธีสำหรับการทำเหมืองเซตรายการความถี่ส่วนใหญ่ ทำสร้างเซตรายการความยาว k จากเซตรายการความยาว k-1 ซึ่งการขยายลักษณะแบบนี้ สามารถสร้างเซตรายการที่เป็นไปได้จำนวนมากดังแสดงในรูปที่ 2.10 โดยจะเห็นได้ว่า สามารถสร้างเซตรายการที่เป็นไปได้ถึง 31 เซตรายการ จากการขยายรายการ A C F B และ D ถึงแม้ค่าสนับสนุนขั้นต่ำจะถูกนำมาใช้เป็นตัวกรองเพื่อคัดเลือกเอาเฉพาะเซตรายการความถี่ที่ต้องการเท่านั้น แต่ว่าเซตรายการความถี่จำนวนมาก ก็อาจจะถูกสร้างขึ้นเมื่อกำหนดให้ค่าสนันสนุนขั้นต่ำมีค่าต่ำ ซึ่งการสร้างเซตรายการความถี่จำนวนมาก ทำให้ยากต่อการวิเคราะห์ เสียเวลาในการสร้างเซตรายการความถี่จำนวนมาก ทำให้มีการนำเสนอการทำเหมืองเซตรายการแบบปิด (Closed itemset mining) และการทำเหมืองเซตรายการความยาวสูงสุด (Maximal itemset mining) เพื่อลดจำนวนรูปแบบ โดยนิยามที่เกี่ยวข้องมีดังนี้

กำหนดให้ $I = \{i_1, i_2, \ldots, i_m\}$ คือ เซตของรายการทั้งหมดในชุดข้อมูล, X และ Y คือ เซตรายการ โดยที่ $X,Y \subseteq I$ และกำหนดให้ FI คือ เซตของเซตรายการความถี่

นิยามที่ 2.4 เซตรายการ Y เป็นซูเปอร์เซตของเซตรายการ X (แทนด้วย $X \subset Y$) ก็ต่อเมื่อ รายการทั้งหมดที่อยู่ในเซตรายการ X ปรากฏอยู่ในเซตรายการ Y และเซตรายการ X ถือว่าเป็นเซตย่อยของเซตรายการ Y

ตัวอย่างที่ 2.10 เซตรายการ (ACF) เป็นซูเปอร์เซตของเซตรายการ (CF) เนื่องจากรายการ C และ F ปรากฏในเซตรายการ (ACF)

นิยามที่ 2.5 เซตรายการ *X* เป็นเซตรายการแบบปิด ก็ต่อเมื่อ ค่าสนับสนุนของเซตรายการ *X* มากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ และไม่มีซูเปอร์เซตที่มีค่าสนับสนุนเท่ากัน ดังนั้นเซตรายการแบบปิดทั้งหมดสามารถเขียนแทนด้วยสมการที่ 2.1

$$CI = \{X \mid X \in FI \wedge (\nexists Y \in FI \mid X \subset Y \wedge supp(X) = supp(Y))\} \quad (2.1)$$

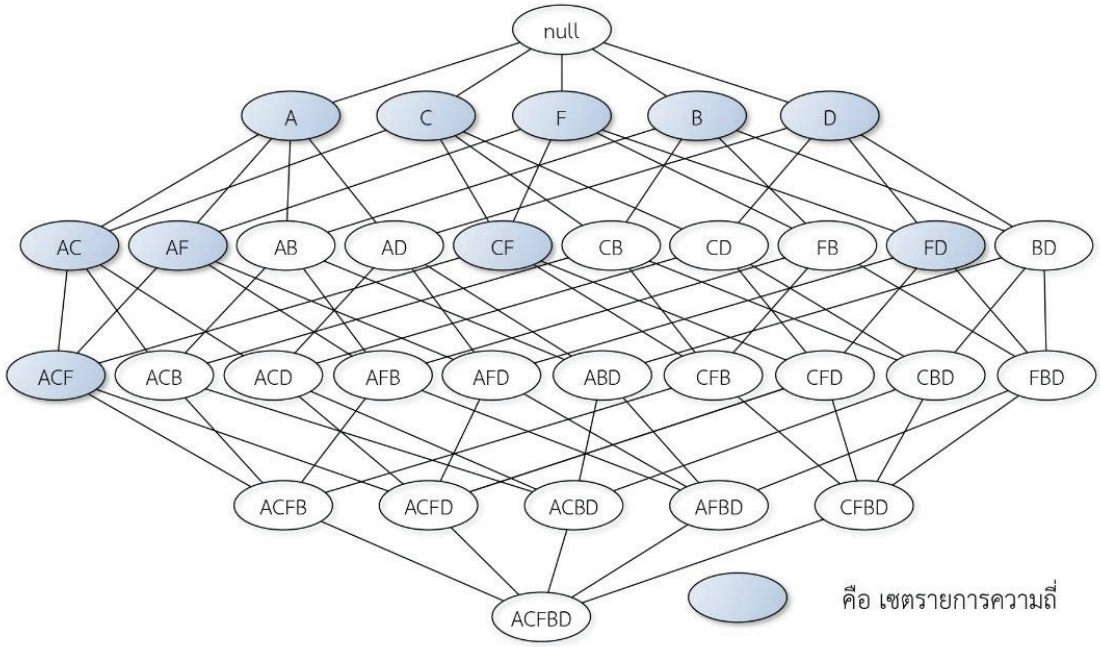
ตัวอย่างที่ 2.11 จากตัวอย่างในตารางที่ 2.1 ถ้ากำหนดค่าสนับสนุนขั้นต่ำเท่ากับ 2 จะได้เซตรายการความถี่ทั้งหมดเท่ากับ *FI* = {(D):2, (FD):2, (B):2, (F):3, (AF):2, (CF):2, (ACF):2, (C):3, (AC):3, (A):3} (ดังรูปที่ 2.20)

พิจารณาเซตรายการความถี่ (D):2 และ (FD):2 ใน *FI* มีค่าสนับสนุนเท่ากัน และ (D):2 เป็นเซตย่อยของ (FD):2 ดังนั้นเซตรายการ (D):2 ไม่ใช่เซตรายการแบบปิด ส่วนเซตรายการ (FD):2 เป็นเซตรายการแบบปิด เนื่องจากไม่มีเซตรายการใดที่มีค่าสนับสนุนเท่ากับ (FD):2 และเป็นซูเปอร์เซตของ (FD):2

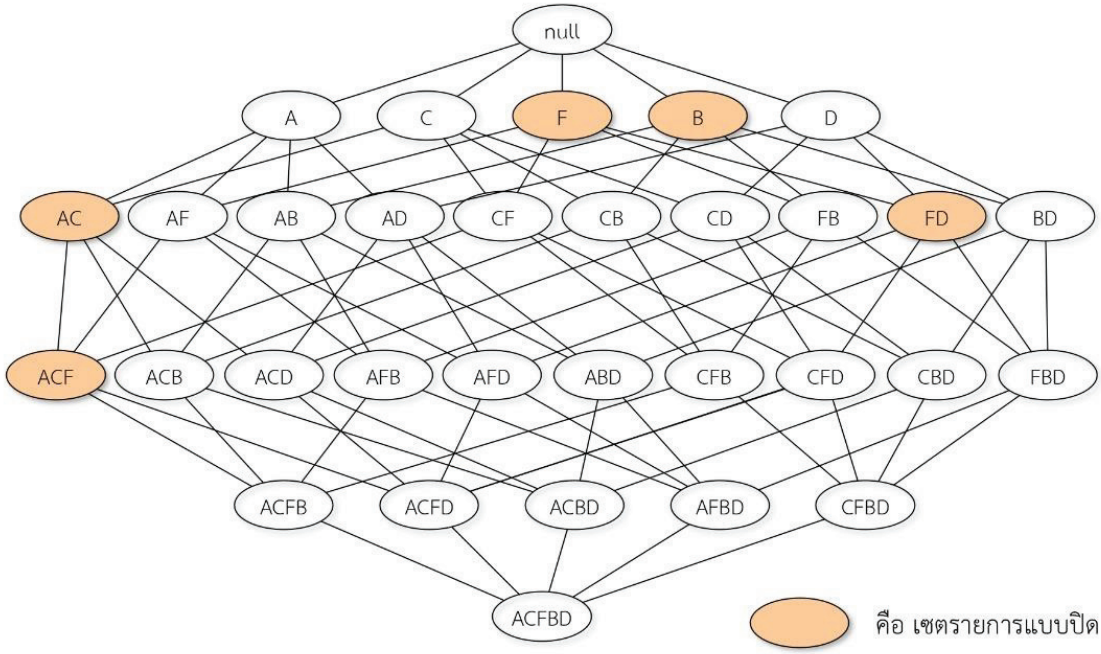
พิจารณาเซตรายการความถี่ (AF):2 และ (CF):2 ใน *FI* พบว่า เป็นเซตย่อยของ (ACF):2 ดังนั้นเซตรายการความถี่ดังกล่าวไม่ใช่เซตรายการแบบปิด ส่วน (ACF):2 ถือว่าเป็นเซตรายการแบบปิด เนื่องจากไม่มีเซตรายการใดที่มีค่าสนับสนุนเท่ากับ (ACF):2 และเป็นซูเปอร์เซตของ (ACF):2

พิจารณาเซตรายการความถี่ (C):3 และ (A):3 ใน *FI* พบว่าเป็นเซตย่อยของ (AC):3 ดังนั้นเซตรายการความถี่ดังกล่าวไม่ใช่เซตรายการแบบปิด ส่วน (AC):3 ถือว่าเป็นเซตรายการแบบปิด เนื่องจากไม่มีเซตรายการใดที่มีค่าสนับสนุนเท่ากับ (AC):3 และเป็นซูเปอร์เซตของ (AC):3

ดังนั้นเซตรายการแบบปิดทั้งหมดจะประกอบไปด้วย *CI* = {(FD):2, (B):2, (F):3, (ACF):2, (AC):3} (ดังรูปที่ 2.21)



รูปที่ 2.20 เซตรายการความถี่



รูปที่ 2.21 เซตรายการแบบปิด

นิยามที่ 2.6 เซตรายการ X เป็นเซตรายการความยาวสูงสุด ก็ต่อเมื่อ ค่าสนับสนุนของเซตรายการ X มากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำและไม่มีซูเปอร์เซต เซตรายการความยาวสูงสุดทั้งหมดสามารถเขียนแทนด้วยสมการที่ 2.2

$$MI = \{X \mid X \in FI \wedge (\nexists Y \in FI \mid X \subset Y)\} \quad (2.2)$$

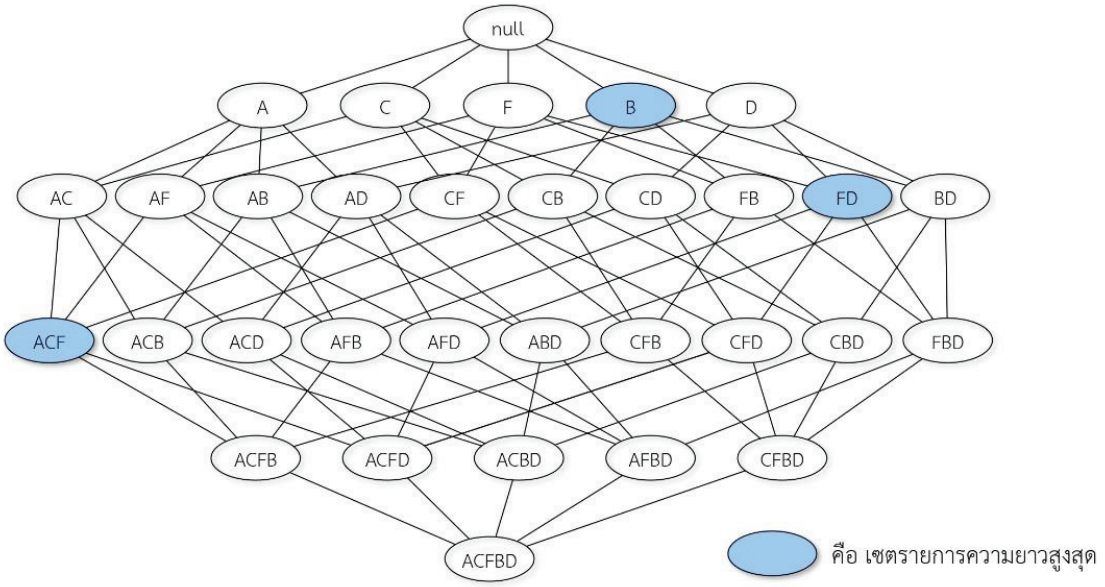
ตัวอย่างที่ 2.12 จากตัวอย่างข้อมูลในตารางที่ 2.1 ถ้ากำหนดค่าสนับสนุนขั้นต่ำเท่ากับ 2 จะได้เซตรายการความถี่ทั้งหมดเท่ากับ FI = {(D):2, (FD):2, (B):2, (F):3, (AF):2, (CF):2, (ACF):2, (C):3, (AC):3, (A):3} (ดังรูปที่ 2.20)

พิจารณาเซตรายการความถี่ (D):2 และ (FD):2 ใน FI พบว่า (D):2 เป็นเซตย่อยของ (FD):2 ดังนั้น (D):2 ไม่ใช่เซตรายการความยาวสูงสุด และเมื่อพิจารณา (FD):2 ปรากฏว่าไม่มีซูเปอร์เซต ดังนั้น (FD):2 เป็นเซตรายการความยาวสูงสุด

พิจารณาเซตรายการความถี่ (B):2 ใน FI ปรากฏว่าไม่มีซูเปอร์เซต ดังนั้น (B):2 เป็นเซตรายการความยาวสูงสุด

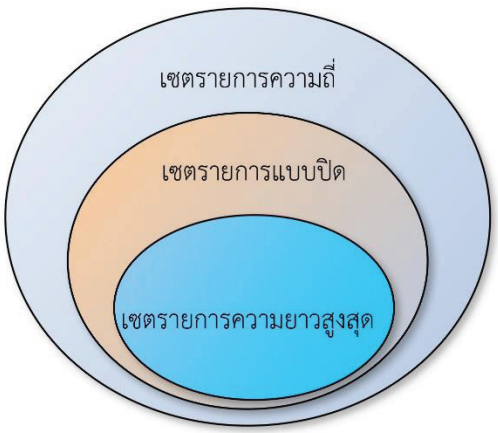
พิจารณาเซตรายการความถี่ (F):3, (AF):2, (CF):2, (C):3, (AC):3 และ (A):3 ใน FI ปรากฏว่าทุกเซตรายการเป็นเซตย่อยของ (ACF):2 ดังนั้นเซตรายการดังกล่าวไม่ใช่เซตรายการความยาวสูงสุด ส่วน (ACF):2 ไม่มีซูเปอร์เซต ดังนั้น (ACF):2 เป็นเซตรายการความยาวสูงสุด

สามารถสรุปได้ว่า เซตรายการความยาวสูงสุดประกอบไปด้วย MI = {(FD):2, (B):3, (ACF):2} (ดังรูปที่ 2.22)



รูปที่ 2.22 เซตรายการความยาวสูงสุด

จะเห็นได้ว่าจำนวนรูปแบบที่ได้จากการทำเหมืองเซตรายการแบบปิด จะมีจำนวนน้อยกว่าการทำเหมืองเซตรายการความถี่ ปัจจุบันมีหลายขั้นตอนวิธีถูกพัฒนาขึ้นสำหรับการทำเหมืองเซตรายการแบบปิด เช่น A-CLOSE, CHARM, CLOSET, CLOSET+, DCI_CLOSED และ LCM เป็นต้น ส่วนการทำเหมืองเซตรายการความยาวสูงสุดจะให้จำนวนรูปแบบที่น้อยที่สุด (ดังรูปที่ 2.23) ขั้นตอนวิธีที่ถูกนำเสนอเพื่อขุดค้นเซตรายการความยาวสูงสุด เช่น LCM, FPMax และ Charm_MFI เป็นต้น



รูปที่ 2.23 เปรียบเทียบจำนวนเซตรายการแบบต่างๆ

2.5 ตัวอย่างการทำเหมืองเซตรายการความถี่โดยใช้ SPMF

SPMF เป็นคลังโปรแกรมที่พัฒนาขึ้นเพื่อทำการขุดค้นรูปแบบและกฎความสัมพันธ์โดยเฉพาะ ประกอบไปด้วยขั้นตอนวิธีต่างๆ มากมาย เช่น ขั้นตอนวิธีสำหรับการสร้างกฎความสัมพันธ์ ขั้นตอนวิธีสำหรับการทำเหมืองเซตรายการความถี่ ขั้นตอนวิธีสำหรับการทำเหมืองรูปแบบลำดับเหตุการณ์ ขั้นตอนวิธีสำหรับการสร้างกฎความสัมพันธ์เชิงลำดับ เป็นต้น การติดตั้ง SPMF (ดังภาคผนวก ก) และการใช้งานคำสั่งใน SPMF สามารถทำได้ง่าย ทำให้ SPMF เป็นเครื่องมือที่ได้รับความนิยมในการทำเหมืองรูปแบบ

ใน SPMF มีขั้นตอนวิธีสำหรับขุดค้นเซตรายการที่น่าสนใจหลายขั้นตอนวิธี และการเตรียมข้อมูลเพื่อใช้ใน SPMF สามารถทำได้ง่ายโดยแทนรายการเป็นตัวเลข โดยรายละเอียดการเตรียมข้อมูลสำหรับการทำเหมืองเซตรายการความถี่โดยใช้ SPMF มีดังต่อไปนี้

2.5.1 การเตรียมชุดข้อมูลเซตรายการ

ชุดข้อมูลเซตรายการเป็นชุดข้อมูลที่ไม่สนใจลำดับการเกิดของข้อมูล เช่น การซื้อสินค้าที่ไม่ต้องการพิจารณาว่าซื้อสินค้าไหนก่อนหลัง รู้แค่ว่าซื้อพร้อมกัน เป็นต้น ดังนั้นขั้นตอนวิธีที่นำมาใช้กับชุด

ข้อมูลเหล่านี้ จะทำการประมวลผลหารูปแบบที่น่าสนใจโดยไม่พิจารณาการเกิดของข้อมูล เช่น การทำเหมืองเซตรายการความถี่ การทำเหมืองเซตรายการแบบปิด การทำเหมืองเซตรายการความยาวสูงสุด การสร้างกฎความสัมพันธ์ เป็นต้น การเตรียมชุดข้อมูลเซตรายการประกอบด้วยเงื่อนไขดังต่อไปนี้

- ไฟล์ชุดข้อมูลนำเข้า SPMF อยู่ในรูปของไฟล์ข้อความ (Text file) ซึ่งสามารถใช้ Editor ต่างๆ สร้างได้ เช่น Notepad, Editplus เป็นต้น
- รายการแต่ละรายการจะต้องแทนด้วยตัวเลขจำนวนเต็มบวก
- แต่ละรายการคั่นด้วยเว้นวรรค 1 เว้นวรรค
- เซตรายการใน 1 รายการเปลี่ยนแปลงแทนด้วยข้อมูล 1 บรรทัดในไฟล์
- ในแต่ละบรรทัดจะต้องเรียงตัวเลขจากน้อยไปมาก
- ห้ามมีบรรทัดที่มีข้อมูล (เซตรายการ) เหมือนกัน

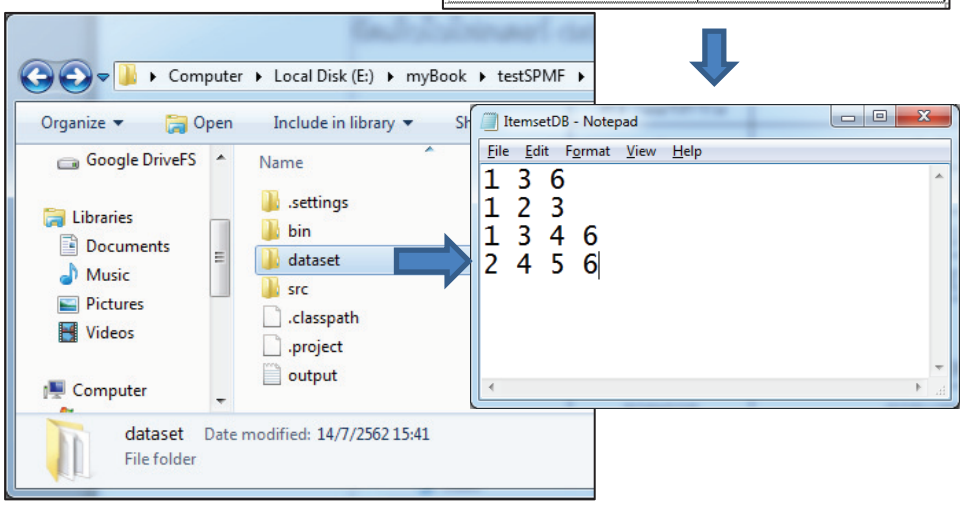
กำหนดให้ชุดข้อมูลที่ต้องการทำเหมืองเซตรายการความถี่เป็นดัง

ตารางที่ 2.1 ซึ่งประกอบไปด้วย 4 รายการเปลี่ยนแปลง และรายการในชุดข้อมูลประกอบไปด้วย {A, B, C, D, E, F} สามารถนำชุดข้อมูลไปสร้างไฟล์ชุดข้อมูลได้โดยทำการแทนรายการทั้งหมดด้วยตัวเลขจำนวนเต็มบวก สามารถแทนรายการด้วยตัวเลขจำนวนเต็มบวกตามตารางที่ 2.4 จะได้ไฟล์ชุดข้อมูลนำเข้าดังรูปที่ 2.24 ซึ่งเป็นไฟล์ที่สร้างขึ้นโดยใช้โปรแกรม Notepad จากนั้นทำการบันทึกไฟล์ชื่อ ItemsetDB.txt และจัดเก็บในโฟลเดอร์ dataset

ตารางที่ 2.4 การแทนค่าด้วยตัวเลขจำนวนเต็มในชุดข้อมูลเซตรายการ

รายการ	การแทนค่า
A	1
B	2
C	3
D	4
E	5
F	6

รายการเปลี่ยนแปลง	เซตรายการ
1	(A C F)
2	(A B C)
3	(A C D F)
4	(B D E F)



รูปที่ 2.24 ตัวอย่างไฟล์นำเข้าและการจัดเก็บ

2.5.2 ตัวอย่างคำสั่งสำหรับการทำเหมืองเซตรายการความถี่

คำสั่งสำหรับสร้างเซตรายการความถี่ด้วยขั้นตอนวิธี FP-Growth แสดงได้ดังตัวอย่างคำสั่งที่ 2.1 โดยแต่ละคำสั่งสามารถอธิบายได้ดังนี้

ตัวอย่างคำสั่งที่ 2.1

```
1.  package mySpmfProject;
2.
3.  import java.io.IOException;
4.  import ca.pfv.spmf.algorithms.frequentpatterns.fpgrowth.AlgoFPGrowth;
5.
6.  public class FPGrowthTest {
7.          public static void main(String [] arg) throws IOException{
8.                  String input = ".//dataset/ItemsetDB.txt";
9.                  String output = ".//output.txt";
10
11.                 double minsupp = 0.5;
12.                 AlgoFPGrowth algo = new AlgoFPGrowth();
13.                 algo.runAlgorithm(input, output, minsupp);
14.                 algo.printStats();
15.         }
16. }
```

บรรทัดที่ 3 เป็นการ import คลาส IOException สำหรับจัดการข้อผิดพลาดเกี่ยวกับไฟล์

บรรทัดที่ 4 เป็นการ import คลาส AlgoFPGrowth เพื่อเรียกใช้ขั้นตอนวิธี FP-Growth

บรรทัดที่ 8 เป็นการกำหนดตำแหน่งชุดข้อมูลนำเข้า ซึ่งในตัวอย่างคือไฟล์ ItemsetDB.txt

บรรทัดที่ 9 เป็นการกำหนดตำแหน่งไฟล์สำหรับเก็บผลลัพธ์ ซึ่งในไฟล์แสดงเซตรายการความถี่และค่าสนับสนุนแบบสัมบูรณ์ดังรูปที่ 2.26 เช่น 1 6 #SUP: 2 หมายถึง เซตรายการความถี่ 1 6 มีค่าสนับสนุนเท่ากับ 2

บรรทัดที่ 11 เป็นการกำหนดค่าสนับสนุนขั้นต่ำแบบสัมพัทธ์ ซึ่งในตัวอย่างกำหนดค่าสนับสนุนขั้นต่ำแบบสัมพัทธ์ให้มีค่าเท่ากับ 0.5 หรือ 50% เมื่อเทียบกับค่าสนับสนุนขั้นต่ำแบบสัมบูรณ์จะมีค่าเท่ากับ 2

บรรทัดที่ 12 เป็นการสร้างอ๊อบเจ็กต์ของคลาส AlgoFPGrowth

บรรทัดที่ 13 เป็นการเรียกใช้เมทธอด runAlgorithm เพื่อสั่งให้ขั้นตอนวิธี FP-Growth ทำการประมวลผล โดยมีพารามิเตอร์ 3 ตัว คือ

- ไฟล์นำเข้า (input)
- ไฟล์ผลลัพธ์ (output)
- ค่าสนับสนุนขั้นต่ำแบบสัมพัทธ์ (minsupp)

บรรทัดที่ 14 เป็นการแสดงค่าสถิติต่างๆ ที่ได้จากการประมวลผล ผลลัพธ์จะได้ดังรูปที่ 2.25 โดยแสดงค่าต่างๆ ดังนี้

- จำนวนรายการเปลี่ยนแปลง (Transactions count from database)
- หน่วยความจำที่ใช้ (Max memory usage)
- จำนวนเซตรายการความถี่ (Frequent itemsets count)
- เวลาในการประมวลผล (Total time)

```
=============  FP-GROWTH 0.96r19 - STATS =============
 Transactions count from database : 4
 Max memory usage: 11.481971740722656 mb
 Frequent itemsets count : 10
 Total time ~ 8 ms
===================================================
```

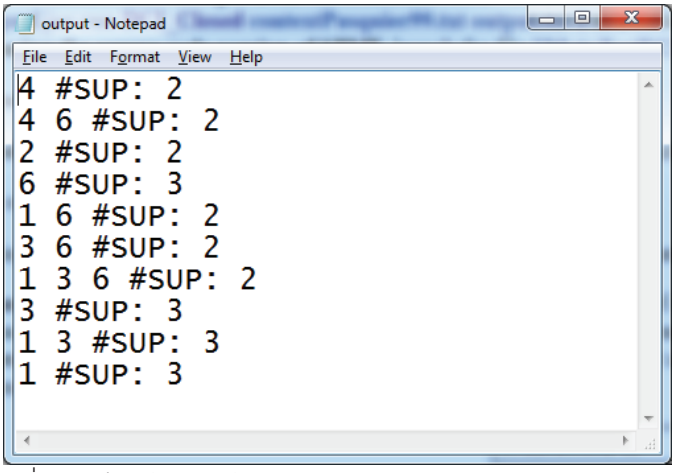
รูปที่ 2.25 แสดงค่าทางสถิติจากการประมวลผล FPGrowthTes.java

เมื่อแปลงข้อมูลในไฟล์ผลลัพธ์กลับคืนตามตารางที่ 2.4 จะได้ดังรูปที่ 2.27 โดยเซตรายการความถี่ที่ได้ สามารถแปลความหมายได้ดังตัวอย่างต่อไปนี้

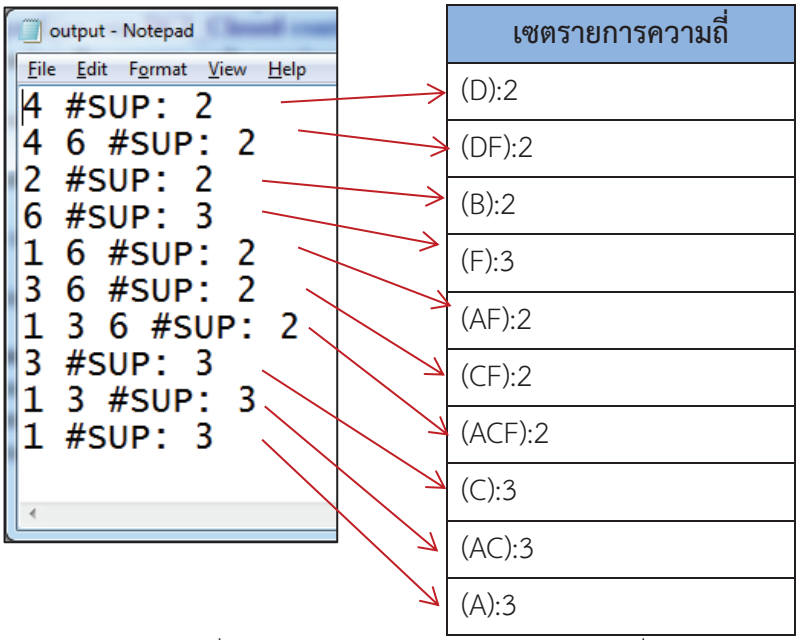
เซตรายการความถี่ที่ 1 ความถี่ในการซื้อสินค้า D คือ 2

เซตรายการความถี่ที่ 2 ความถี่ในการซื้อสินค้า D ร่วมกับสินค้า F คือ 2

เซตรายการความถี่ที่ 3 ความถี่ในการซื้อสินค้า B คือ 2



รูปที่ 2.26 ไฟล์ผลลัพธ์จากการประมวลผล FPGrowthTes.java



รูปที่ 2.27 แปลงผลลัพธ์เซตรายการความถี่

2.5.3 ตัวอย่างคำสั่งสำหรับการทำเหมืองเซตรายการแบบปิด

ถ้าต้องการขุดค้นเซตรายการแบบปิดหรือเซตรายการที่ยาวที่สุด สามารถปรับเปลี่ยนคำสั่งจากคำสั่งในตัวอย่างคำสั่งที่ 2.1 โดยปรับเปลี่ยนคำสั่งบางคำสั่งดังตัวอย่างในตัวอย่างคำสั่งที่ 2.2 ซึ่งเป็นตัวอย่างคำสั่งการเรียกใช้ขั้นตอนวิธี DCI_CLOSED สำหรับการทำเหมืองเซตรายการแบบปิด

ตัวอย่างคำสั่งที่ 2.2

```
1.  package mySpmfProject;
2.
3.  import java.io.IOException;
4.  import ca.pfv.spmf.algorithms.frequentpatterns.dci_closed.AlgoDCI_Closed;
5.
6.  public class Dci_closedTest {
7.          public static void main(String [] arg) throws IOException{
8.                  String input = ".//dataset/ItemsetDB.txt";
9.                  String output = ".//output.txt";
10
11.                 int minsup = 2;  // means 2 transactions
12.                 AlgoDCI_Closed algorithm = new AlgoDCI_Closed();
13.                 algorithm.runAlgorithm(input, output, minsup);
14.         }
15. }
```

บรรทัดที่ 4 เป็นการ import คลาส AlgoDCI_Closed เพื่อเรียกใช้ขั้นตอนวิธี DCI_Closed

บรรทัดที่ 8 เป็นการกำหนดตำแหน่งชุดข้อมูลนำเข้า ซึ่งในตัวอย่างคือไฟล์ ItemsetDB.txt

บรรทัดที่ 9 เป็นการกำหนดตำแหน่งไฟล์สำหรับเก็บผลลัพธ์

บรรทัดที่ 11 เป็นการกำหนดค่าสนับสนุนขั้นต่ำแบบสัมบูรณ์ ซึ่งในตัวอย่างกำหนดค่าสนับสนุนขั้นต่ำแบบสัมบูรณ์ให้มีค่าเท่ากับ 2

บรรทัดที่ 12 เป็นการสร้างอ๊อบเจ็กต์ของคลาส AlgoDCI_Closed

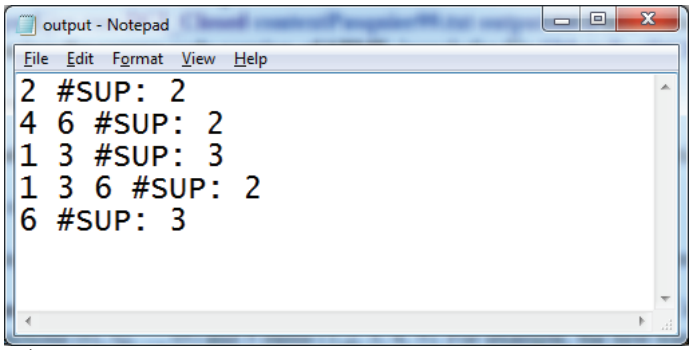
บรรทัดที่ 13 เรียกใช้เมทธอด AlgoDCI_Closed เพื่อสั่งให้ขั้นตอนวิธี AlgoDCI_Closed ทำการประมวลผลและแสดงค่าสถิติต่างๆ ดังรูปที่ 2.28 โดยมีพารามิเตอร์ 3 ตัว คือ

- ไฟล์นำเข้า
- ไฟล์ผลลัพธ์
- ค่าสนับสนุนขั้นต่ำแบบสัมบูรณ์

ไฟล์ผลลัพธ์ที่ได้จากการประมวลผลแสดงได้รูปที่ 2.29 ซึ่งจะเห็นได้ว่าการทำเหมืองเซตรายการแบบปิดให้จำนวนรูปแบบที่น้อยกว่าการทำเหมืองเซตรายการความถี่

```
Running the DCI-Closed algorithm
========== DCI_CLOSED - STATS ============
 Number of transactions: 4
 Number of frequent closed itemsets: 5
 Total time ~: 3 ms
```

รูปที่ 2.28 แสดงค่าทางสถิติจากการประมวลผล Dci_closedTest.java



รูปที่ 2.29 ไฟล์ผลลัพธ์จากการประมวลผล Dci_closedTest.java

บทสรุป

การทำเหมืองเซตรายการความถี่ เป็นการค้นหาเซตรายการที่ปรากฏร่วมกันบ่อย โดยพิจารณาจากเซตรายการที่มีค่าสนับสนุนมากว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ การทำเหมืองเซตรายการความถี่เป็นการขุดค้นรูปแบบที่น่าสนใจบนข้อมูลที่ไม่พิจารณาลำดับการเกิด มีการนำเสนอขั้นตอนวิธีที่หลากหลายสำหรับการทำเหมืองเซตรายการความถี่ ขั้นตอนวิธี FP-Growth เป็นขั้นตอนหนึ่งที่มีประสิทธิภาพในการค้นหาเซตรายการความถี่ โดยขั้นตอน FP-Growth สร้างโครงสร้างข้อมูลที่เรียกว่า FP-tree สำหรับเก็บข้อมูล เพื่อลดปัญหาการอ่านฐานข้อมูลหลายครั้ง จากนั้นทำการขุดค้นเซตรายการความถี่บนโครงสร้าง FP-tree โดยใช้การค้นหาแนวลึก

นอกจากนี้ยังมีการนำเสนอการทำเหมืองเซตรายการแบบปิด และการทำเหมืองเซตรายการความยาวสูงสุด เพื่อลดจำนวนรูปแบบที่สร้างขึ้นจากการทำเหมืองเซตรายการความถี่

แบบฝึกหัดท้ายบท

1. จงยกตัวอย่างข้อมูลที่เหมาะสำหรับการทำเหมืองเซตรายการความถี่
2. จงอธิบายความหมายของการทำเหมืองเซตรายการความถี่
3. ความยาวของเซตรายการ (B A C D) เท่ากับเท่าไร
4. จงอธิบายหลักการค้นหาเซตรายการความถี่ด้วยขั้นตอนวิธี FP-Growth
5. การทำเหมืองเซตรายการความถี่ด้วยขั้นตอนวิธี FP-Growth มีข้อดีอย่างไร
6. จากตารางชุดข้อมูลข้างล่าง จงแสดงวิธีการสร้าง FP-Tree เมื่อกำหนดค่าสนับสนุนขั้นต่ำแบบสัมบูรณ์เท่ากับ 2

รายการเปลี่ยนแปลง	เซตรายการ
1	(A C)
2	(A B C)
3	(A D)
4	(B C)

7. จากข้อ 6 จงแสดงวิธีการค้นหาเซตรายการความถี่ด้วยขั้นตอนวิธี FP-Growth
8. จากตารางชุดข้อมูลในข้อ 6 จงสร้างไฟล์ชุดข้อมูลเซตรายการสำหรับการทำเหมืองเซตรายการความถี่ด้วย SPMF
9. จงเขียนโปรแกรมโดยใช้ SPMF เพื่อขุดค้นเซตรายการความถี่ด้วยขั้นตอนวิธี FP-Growth โดยใช้ไฟล์ที่ได้จากข้อ 8
10. การขุดค้นเซตรายการแบบปิดแตกต่างจากการทำเหมืองเซตรายการความถี่อย่างไร

บทที่ 3
การทำเหมืองรูปแบบลำดับเหตุการณ์
(Sequential Pattern Mining)

ข้อมูลลำดับเหตุการณ์เป็นข้อมูลที่เก็บรวบรวมตามระยะเวลาอย่างต่อเนื่องกัน เช่น ข้อมูลปรากฏการณ์ธรรมชาติที่ต้องพิจารณาว่าเหตุการณ์ใดเกิดก่อนหลัง ข้อมูลลำดับการคลิกลิงก์ในเว็บไซต์ที่ต้องการพิจารณาว่าคลิกลิงก์ไหนก่อนหลัง ข้อมูลลำดับการเกิดโรคที่ต้องพิจารณาว่าเกิดโรคใดก่อนหลัง เป็นต้น การขุดค้นรูปแบบที่ซ่อนอยู่หรือรูปแบบที่น่าสนใจจากข้อมูลลำดับเหตุการณ์ จำเป็นจะต้องพิจารณาลำดับการเกิดของข้อมูล ถ้าลำดับการเกิดของข้อมูลไม่ได้ถูกพิจารณา จะทำให้สูญเสียสารสนเทศที่สำคัญไป ทำให้รูปแบบที่ได้ไม่สามารถนำไปใช้ประโยชน์ได้อย่างแท้จริง เช่น การคลิกลิงก์บนเว็บไซต์ ถ้าไม่พิจารณาลำดับการคลิกลิงก์ ก็จะทำให้ทราบแค่เพียงว่ามีการคลิกลิงก์ไหนบ้าง แต่ไม่ทราบว่ามีการคลิกลิงก์ไหนก่อนหลัง ปรากฏการณ์ธรรมชาติ ถ้าไม่พิจารณาลำดับการเกิด ก็จะไม่ทราบว่าเหตุการณ์ไหนเกิดก่อนและเหตุการณ์ไหนเกิดตามหลัง เป็นต้น การขุดค้นรูปแบบที่ต้องพิจารณาลำดับการเกิด สามารถทำได้โดยใช้การทำเหมืองรูปแบบลำดับเหตุการณ์ ซึ่งในปัจจุบันมีหลายงานที่ประยุกต์ใช้การทำเหมืองรูปแบบลำดับเหตุการณ์ เช่น การวิเคราะห์เหตุการณ์ทางธรรมชาติ การวิเคราะห์การกดลิงก์ในเว็บไซต์ ชีวสารสนเทศ การจำแนกข้อความ เป็นต้น

3.1 ลักษณะข้อมูลที่ใช้ในการทำเหมืองรูปแบบลำดับเหตุการณ์

การทำเหมืองรูปแบบลำดับเหตุการณ์ เป็นการค้นหารูปแบบของข้อมูลที่เกิดขึ้นบ่อย โดยจะต้องพิจารณาลำดับการเกิดของข้อมูลด้วย เช่น ถ้ามีแผ่นดินไหวแถวชายหาดมักจะเกิดสึนามิตามมา เป็นต้น ข้อมูลที่ใช้ในการขุดค้นลำดับเหตุการณ์จะต้องเรียงตามลำดับการเกิด ในส่วนนี้จะขอยกตัวอย่างข้อมูลการจ่ายยาผู้ป่วย ที่กินยาแล้วหายจากโรคหัวใจดังตัวอย่างในตารางที่ 3.1 ซึ่งจะเห็นได้ว่า ในแต่ละครั้งหมอจะจ่ายยาแตกต่างกันไป เช่น ผู้ป่วยรหัส 1001 ครั้งแรกได้รับยา A และ B พร้อมกัน ครั้งที่ 2 ได้รับยา A และ C พร้อมกัน ครั้งที่ 3 ได้รับยาแค่ตัวเดียว คือ D เป็นต้น

จากข้อมูลในตารางที่ 3.1 จะต้องจัดเตรียมข้อมูลให้อยู่ในรูปแบบรายการเปลี่ยนแปลง ซึ่งกำหนดให้หนึ่งรายการเปลี่ยนแปลง คือ ลำดับการจ่ายยาของผู้ป่วยหนึ่งคน ผลการจัดเตรียมข้อมูลแสดงได้ดังตารางที่ 3.2 โดยกำหนดให้ยาที่อยู่ในวงเล็บเดียวกัน หมายถึง ยาที่จ่ายให้ผู้ป่วยพร้อมกัน และข้อมูลจะต้องถูกเรียงตามลำดับการให้ยา ตัวอย่างข้อมูลในตารางที่ 3.2 ถือว่าเป็นชุดข้อมูลลำดับเหตุการณ์

ตารางที่ 3.1 ประวัติการจ่ายยาให้ผู้ป่วย

รหัสคนไข้	ครั้งที่	การจ่ายยา
1001	1	(A B)
1001	2	(A C)
1001	3	(D)
1002	1	(A)
1002	2	(C D)
1002	3	(E)
1003	1	(A)
1003	2	(C D)
1003	3	(F)
1004	1	(C D)
1004	2	(E F)

ตารางที่ 3.2 ตัวอย่างชุดข้อมูลลำดับเหตุการณ์การจ่ายยา

รายการเปลี่ยนแปลง	ประวัติการจ่ายยา
1	< (A B) (A C) (D) >
2	< (A) (C D) (E) >
3	< (A) (C D) (F) >
4	< (C D) (E F) >

3.2 นิยามที่เกี่ยวข้อง

กำหนดให้ $I = \{i_1, i_2, \ldots, i_m\}$ คือ เซตของรายการทั้งหมดในชุดข้อมูล, $T = \{t_1, t_2, \ldots, t_k\}$ คือ เซตของรายการเปลี่ยนแปลงทั้งหมดในชุดข้อมูล และ X คือ เซตรายการหรือเรียกว่าเหตุการณ์ โดยที่ $X \subseteq I$

นิยามที่ 3.1 ลำดับเหตุการณ์ คือ เหตุการณ์หนึ่งเหตุการณ์หรือมากกว่าหนึ่งเหตุการณ์ที่เกิดขึ้นตามลำดับ แทนด้วย $S = \langle X_1, X_2, \ldots, X_n \rangle$ โดยที่ $X_i \subseteq I \ (1 \le i \le n)$ โดยแต่ละรายการเปลี่ยนแปลง t_j ประกอบไปด้วย S_j ซึ่งหมายความว่า 1 รายการเปลี่ยนแปลงประกอบไปด้วย 1 ลำดับเหตุการณ์

ตัวอย่างที่ 3.1 จากตารางที่ 3.2 ในแต่ละรายการเปลี่ยนแปลง คือ ประวัติการจ่ายยาของผู้ป่วยแต่ละคน ดังนั้นประวัติการจ่ายยาของผู้ป่วย 1 คน คือ 1 ลำดับเหตุการณ์ โดยการจ่ายยาในแต่ละครั้ง คือ แต่ละเหตุการณ์ เช่น ในรายการเปลี่ยนแปลงที่ 1 มีลำดับเหตุการณ์ คือ (A B) (A C) (D) ซึ่งประกอบไปด้วย 3 เหตุการณ์ที่เกิดตามลำดับ คือ (A B), (A C) และ (D) ซึ่งหมายความว่า ผู้ป่วยคนที่หนึ่งได้รับยา A และ B พร้อมกัน ต่อมาได้รับยา A และ C พร้อมกัน และต่อมาได้รับยา D

นิยามที่ 3.2 ความยาวของลำดับเหตุการณ์ คือ จำนวนรายการที่ปรากฏในลำดับเหตุการณ์

ตัวอย่างที่ 3.2 ลำดับเหตุการณ์ (A B) (A C) (D) มีความยาวเท่ากับ 5 เนื่องจากมีรายการปรากฏทั้งหมด 5 รายการ (นับตัวซ้ำด้วย)

นิยามที่ 3.3 S_a =<A_1, A_2, ..., A_n> เป็นลำดับเหตุการณ์ย่อยของลำดับเหตุการณ์ S_b =<B_1, B_2, ..., B_m> ก็ต่อเมื่อมีจำนวนเต็ม $1 \leq i_1 < i_2 < ... < i_n \leq m$ ซึ่ง $A_1 \subseteq B_{i1}$, $A_2 \subseteq B_{i2}$, ..., $A_n \subseteq B_{in}$ ซึ่งสามารถแทนด้วย $S_a \sqsubseteq S_b$

ตัวอย่างที่ 3.3 ลำดับเหตุการณ์ <(A) (D)> เป็นลำดับเหตุการณ์ย่อยของ <(A B) (A C) (D)> เพราะลำดับเหตุการณ์ <(A) (D)> มีเหตุการณ์ (A) แล้วเกิดเหตุการณ์ (D) ตามหลังเหมือนกับลำดับเหตุการณ์ <(A B) (A C) (D)> ซึ่ง (A) $\subseteq$ (A B) และ (D) $\subseteq$ (D)

นิยามที่ 3.4 ค่าสนับสนุนของลำดับเหตุการณ์ S =<A_1, A_2, ..., A_n> คือ จำนวนรายการเปลี่ยนแปลงที่ปรากฏ A_i ตามลำดับ

ตัวอย่างที่ 3.4 จากตารางที่ 3.2 ค่าสนับสนุนของลำดับเหตุการณ์ <(A) (D)> คือ รายการเปลี่ยนแปลงที่เกิด (A) แล้วตามหลังด้วย (D) ซึ่งปรากฏในรายการเปลี่ยนแปลงที่ 1 2 และ 3 ดังนั้นค่าสนับสนุนของลำดับเหตุการณ์ <(A) (D)> เท่ากับ 3

นิยามที่ 3.5 รูปแบบลำดับเหตุการณ์ (Sequential pattern) คือ ลำดับเหตุการณ์ที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ

ตัวอย่างที่ 3.5 ถ้ากำหนดค่าสนับสนุนขั้นต่ำให้มีค่าเท่ากับ 2 ลำดับเหตุการณ์ <(A) (D)> ถือว่าเป็นรูปแบบลำดับเหตุการณ์หรือลำดับเหตุการณ์ความถี่ เนื่องจากค่าสนับสนุนของลำดับเหตุการณ์ <(A) (D)> คือ 3 ซึ่งมีค่ามากกว่าค่าสนับสนุนขั้นต่ำ

นิยามที่ 3.6 กำหนดให้ S_a =<A_1, A_2, ..., A_n> คือ ลำดับเหตุการณ์ และ S_b =<B_1, B_2, ..., B_m> คือลำดับเหตุการณ์นำหน้า (Prefix) โดยที่ ($m \le n$) ลำดับเหตุการณ์ตามหลัง (Postfix) S_b ในลำดับเหตุการณ์ S_a คือ <A_m, A_{m+1}, ..., A_n> โดยที่ $A_m = (A_m - B_m)$

ตัวอย่างที่ 3.6 ถ้าลำดับเหตุการณ์นำหน้า คือ <(A) (C)> เมื่อพิจารณาลำดับเหตุการณ์ <(A) (C D) (E)> แล้วพบว่า ลำดับเหตุการณ์ตามหลัง <(A) (C)> ในลำดับเหตุการณ์ <(A) (C D) (E)> คือ (_D)(E) (เครื่องหมาย _ข้างหน้า D หมายความว่า D เกิดพร้อมกับ C ซึ่ง C เป็นเหตุการณ์สุดท้ายในลำดับเหตุการณ์นำหน้า <(A) (C)>)

นิยามที่ 3.7 ฐานข้อมูลโปรเจค (Project database) ของลำดับเหตุการณ์นำหน้า *S* เป็นฐานข้อมูลที่ประกอบไปด้วยลำดับเหตุการณ์ตามหลังลำดับเหตุการณ์ *S*

ตัวอย่างที่ 3.7 สมมติลำดับเหตุการณ์นำหน้า คือ <(A) (C)> และจากตารางที่ 3.2 พบว่าลำดับเหตุการณ์ <(A) (C)> ปรากฏในรายการเปลี่ยนแปลงที่ 1 2 และ 3 เท่านั้น

ดังนั้นฐานข้อมูลโปรเจคของลำดับเหตุการณ์นำหน้า <(A) (C)> จะพิจารณาจากลำดับเหตุการณ์ในรายการเปลี่ยนแปลงที่ 1 2 และ 3 ซึ่งมีลำดับเหตุการณ์ตามหลัง คือ <(D)>, <(_ D) (E)> และ <(_ D) (F)>

ดังนั้นฐานข้อมูลโปรเจคของลำดับเหตุการณ์นำหน้า <(A) (C)> คือ {<(D)>, <(_ D) (E)>, <(_ D) (F)>}

นิยามที่ 3.8 การขยายแบบรายการ (i-extension) คือ การขยายรูปแบบลำดับเหตุการณ์กับรายการ สามารถขยายรูปแบบลำดับเหตุการณ์ *S*=<A_1, A_2, ..., A_n> กับรายการ *i* โดยพิจารณา *i* เป็นรายการที่อยู่ใน A_n

ตัวอย่างที่ 3.8 สมมติรูปแบบลำดับเหตุการณ์ คือ <(A) (D)> ทำการขยายแบบรายการกับ E จะได้รูปแบบลำดับเหตุการณ์ <(A) (D E)> โดยที่ E อยู่ในเหตุการณ์เดียวกันกับ D

นิยามที่ 3.9 การขยายแบบลำดับเหตุการณ์ (s-extension) เป็นการขยายรูปแบบลำดับเหตุการณ์ $S = \langle A_1, A_2, ..., A_n \rangle$ กับเหตุการณ์ i โดยพิจารณา i เป็นลำดับเหตุการณ์ที่เกิดหลังจาก A_n

ตัวอย่างที่ 3.9 สมมติรูปแบบลำดับเหตุการณ์ คือ <(A) (D)> ทำการขยายแบบลำดับเหตุการณ์กับเหตุการณ์ (E) จะได้รูปแบบลำดับเหตุการณ์ <(A) (D) (E)>

3.3 ขั้นตอนวิธีสำหรับการทำเหมืองรูปแบบลำดับเหตุการณ์

ปัจจุบันมีหลายขั้นตอนวิธีถูกนำเสนอเพื่อใช้ในการทำเหมืองรูปแบบลำดับเหตุการณ์ เช่น GSP, SPADE, PrefixSpan, SPAM, LAPIN-SPAM, CM-SPAM และ CM-SPADE เป็นต้น ขั้นตอนวิธี PrefixSpan เป็นขั้นตอนวิธีหนึ่งที่มีประสิทธิภาพ โดยทำการแบ่งข้อมูลตามลำดับเหตุการณ์นำหน้า แล้วทำการค้นหารูปแบบลำดับเหตุการณ์แนวลึก (Depth first search) แบบเรียกซ้ำ (Recursive) ทำให้ขนาดของข้อมูลที่ใช้ในการค้นหารูปแบบลำดับเหตุการณ์มีขนาดลดลงเรื่อยๆ นอกจากนี้ยังขจัดปัญหาการสร้างรูปแบบลำดับเหตุการณ์คู่แข่ง ขั้นตอนการทำงานของ PrefixSpan และมีรายละเอียดได้ดังนี้

ขั้นตอนที่ 1 ค้นหารูปแบบลำดับเหตุการณ์ความยาว 1 หรือรายการความถี่ โดยทำการค้นหารายการทั้งหมดในฐานข้อมูล และคำนวณหาค่าสนับสนุนของแต่ละรายการ จากนั้นทำการตรวจสอบค่าสนับสนุนของแต่ละรายการว่ามีค่ามากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำหรือไม่ ถ้ารายการใดมีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ รายการดังกล่าวถือว่าเป็นรายการความถี่ ซึ่งรายการความถี่ดังกล่าวถูกพิจารณาเป็นรูปแบบลำดับเหตุการณ์ความยาว 1

ตัวอย่างที่ 3.10 ถ้ากำหนดค่าสนับสนุนขั้นต่ำเท่ากับ 2 จากตารางที่ 3.2 รายการและค่าสนับสนุนของแต่ละเซตรายการแสดงได้ดังตารางที่ 3.3

ตารางที่ 3.3 รายการความถี่

รายการ	ค่าสนับสนุน
A	3
B	1
C	4
D	4
E	2
F	2

รายการ A มีค่าสนับสนุนหรือความถี่เท่ากับ 3 เนื่องจากปรากฏในรายการเปลี่ยนแปลงที่ 1 2 และ 3 และมีค่าสนับสนุนมากกว่าค่าสนับสนุนขั้นต่ำ ดังนั้นรายการ A เป็นรายการความถี่

ส่วนรายการ B ปรากฏในรายการเปลี่ยนแปลงที่ 1 ดังนั้นค่าสนับสนุนของรายการ B มีค่าเท่ากับ 1 ซึ่งน้อยกว่าค่าสนับสนุนขั้นต่ำที่กำหนดไว้ ดังนั้นรายการ B จะถูกตัดทิ้ง

ส่วนรายการ C และ D ปรากฏในรายการเปลี่ยนแปลงที่ 1 2 3 และ 4 ดังนั้นค่าสนับสนุนของรายการ C และ D คือ 4 ซึ่งมีค่ามากกว่าค่าสนับสนุนขั้นต่ำ ดังนั้น รายการ C และ D เป็นรายการความถี่

ส่วนรายการ E ปรากฏในรายการเปลี่ยนแปลงที่ 2 และ 4 มีค่าสนับสนุนเท่ากับ 2 ซึ่งมีค่าเท่ากับค่าสนับสนุนขั้นต่ำ ดังนั้นรายการ E เป็นรายการความถี่

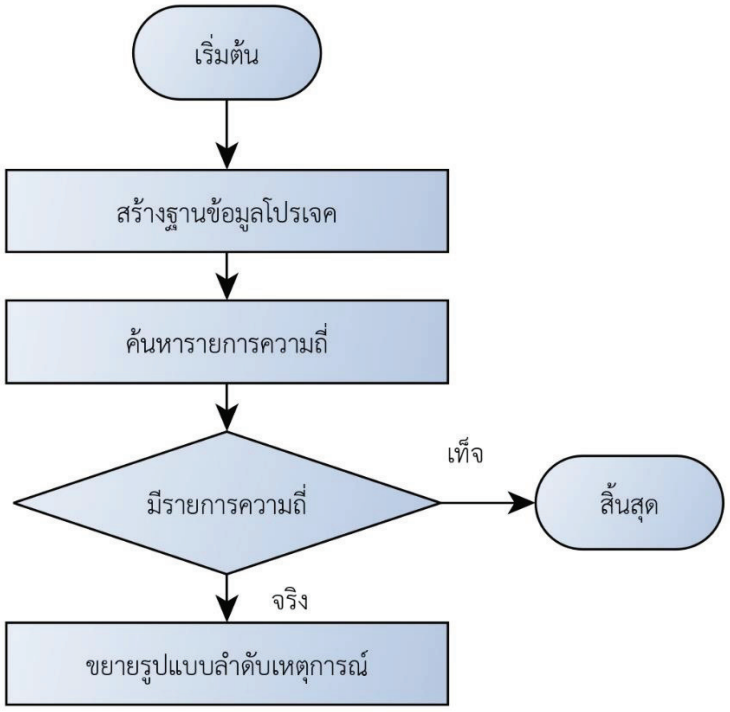
ส่วนรายการ F ปรากฏในรายการเปลี่ยนแปลงที่ 3 และ 4 มีค่าสนับสนุนเท่ากับ 2 ซึ่งมีค่าเท่ากับค่าสนับสนุนขั้นต่ำ ดังนั้นรายการ F เป็นรายการความถี่

สรุปได้ว่ารายการความถี่มีทั้งหมด 5 รายการ คือ A, C, D, E และ F ซึ่งทั้ง 5 รายการถือว่าเป็นลำดับเหตุการณ์ความถี่หรือรูปแบบลำดับเหตุการณ์ความยาว 1 แทนด้วย <(A)>, <(C)>, <(D)>, <(E)> และ <(F)>

ขั้นตอนที่ 2 ขยายรูปแบบลำดับเหตุการณ์เพื่อค้นหารูปแบบลำดับเหตุการณ์อื่น ซึ่งมีหลักการทำงานดังนี้

- รูปแบบลำดับเหตุการณ์ความยาว *k* ได้จากการขยายรูปแบบลำดับเหตุการณ์ความยาว *k*-1
- ทำการขยายรูปแบบลำดับเหตุการณ์ความยาว 1 ไปเรื่อยๆ จนกว่าจะขยายรูปแบบลำดับเหตุการณ์ความยาว 1 ครบทุกรูปแบบ
- ค้นหารูปแบบลำดับเหตุการณ์โดยใช้หลักการค้นหาแนวลึก เช่น ขยายรูปแบบลำดับเหตุการณ์ <(A)> ไปเรื่อยๆ จนกว่าจะไม่สามารถขยายได้ แล้วย้อนกลับมาขยายรูปแบบลำดับเหตุการณ์ <(C)>, <(D)>, <(E)> และ <(F)> ตามลำดับ
- ทำการสร้างฐานข้อมูลโปรเจคเพื่อค้นหารายการความถี่ที่จะนำมาขยายรูปแบบลำดับเหตุการณ์

แต่ละรูปแบบลำดับเหตุการณ์มีขั้นตอนการขยายดังรูปที่ 3.1 และมีรายละเอียดได้ดังนี้



รูปที่ 3.1 ขั้นตอนการขยายรูปแบบลำดับเหตุการณ์

ขั้นตอนที่ 2.1 สร้างฐานข้อมูลโปรเจค โดยจะพิจารณารูปแบบลำดับเหตุการณ์ *S* (รูปแบบที่ต้องการขยาย) เป็นลำดับเหตุการณ์นำหน้า

ตัวอย่างที่ 3.11 พิจารณาขยายรูปแบบลำดับเหตุการณ์ <(A)> จากชุดข้อมูลในตารางที่ 3.2 ฐานข้อมูลโปรเจคของรูปแบบลำดับเหตุการณ์ <(A)> พิจารณาจากข้อมูลในรายการเปลี่ยนแปลงที่ประกอบไปด้วย A ซึ่งก็คือรายการเปลี่ยนแปลงที่ 1 2 และ 3 (ดังตารางที่ 3.4)

ตารางที่ 3.4 ฐานข้อมูลโปรเจคของ <(A)>

รายการเปลี่ยนแปลง	ประวัติการกินยา		ฐานข้อมูลโปรเจคของ <(A)>
1	< (**A** B) (A C) (D)>	→	< (_ B) (A C) (D) >
2	< (**A**) (C D) (E) >		< (C D) (E) >
3	< (**A**) (C D) (F) >		< (C D) (F) >

ในรายการเปลี่ยนแปลงที่ 1 มีรายการ A เกิดพร้อมกับ B ในเหตุการณ์แรก ดังนั้นลำดับเหตุการณ์ตามหลัง <(A)> ในรายการเปลี่ยนแปลงที่ 1 คือ <(_ B) (A C) (D)>

ลำดับเหตุการณ์ตามหลังของรูปแบบลำดับเหตุการณ์ <(A)> ในรายการเปลี่ยนแปลงที่ 2 คือ <(C D) (E)>

ลำดับเหตุการณ์ตามหลังของรูปแบบลำดับเหตุการณ์ <(A)> ในรายการเปลี่ยนแปลงที่ 3 คือ <(C D) (F)>

ดังนั้นฐานข้อมูลโปรเจคของรูปแบบลำดับเหตุการณ์ <(A)> ประกอบไปด้วย 3 ลำดับเหตุการณ์ คือ <(_ B) (A C) (D)>, <(C D) (E)> และ <(C D) (F)>

จากชุดข้อมูลในตารางที่ 3.2 ฐานข้อมูลโปรเจคของรูปแบบลำดับเหตุการณ์ความยาว 1 ทั้งหมด แสดงได้ตารางที่ 3.5 ซึ่งมีฐานข้อมูลโปรเจคทั้งหมด 5 ฐานข้อมูล และเมื่อพิจารณารูปแบบลำดับเหตุการณ์ F แล้วปรากฏว่าไม่มีเหตุการณ์ตามหลังรูปแบบลำดับเหตุการณ์ F ดังนั้นจึงไม่มีฐานข้อมูลโปรเจคของ F

ตารางที่ 3.5 ฐานข้อมูลโปรเจคทั้งหมด

รูปแบบลำดับเหตุการณ์	**ฐานข้อมูลโปรเจค**
<(A)>	< (_ B) (A C) (D) > < (C D) (E) > < (C D) (F) >
<(C)>	< (D) > < (_D) (E) > < (_ D) (F) > < (_ D) (E F) >
<(D)>	< (E) > < (F) > < (E F) >
<(E)>	< (_ F) >
<(F)>	∅

<u>ขั้นตอนที่ 2.2</u> ค้นหารายการความถี่จากฐานข้อมูลโปรเจคของรูปแบบลำดับเหตุการณ์ *S* ซึ่งการค้นหารายการความถี่จะต้องพิจารณารายการที่เกิดในเหตุการณ์เดียวกันกับเหตุการณ์สุดท้ายของ *S* และรายการที่เกิดหลังเหตุการณ์สุดท้ายของ S ถ้าไม่พบรายการความถี่จะหยุดการขยาย

ตัวอย่างที่ 3.12 ค้นหาเซตรายการความถี่จากฐานข้อมูลโปรเจคของรูปแบบลำดับเหตุการณ์ <(A)> จากฐานข้อมูลโปรเจคของรูปแบบลำดับเหตุการณ์ <(A)> มีรายการและค่าสนับสนุนดังตารางที่ 3.6 ซึ่งจะเห็นได้ว่ารายการเปลี่ยนแปลงที่ 1 ในฐานข้อมูลโปรเจคของ <(A)> สามารถพิจารณารายการ C ได้ 2 รายการ คือ รายการ C ที่เกิดหลัง <(A) > และรายการ C ที่เกิดในเหตุการณ์เดียวกันกับ <(A)> แทนด้วย (_C) ซึ่งรายการ C ที่เกิดในเหตุการณ์เดียวกันกับ <(A)> ปรากฏในรายการเปลี่ยนแปลงที่ 1 เท่านั้น ดังนั้นค่าสนับสนุนเท่ากับ 1

ส่วนรายการ C ที่เกิดหลัง <(A)> ปรากฏในทุกลำดับเหตุการณ์ในฐานข้อมูลโปรเจค ดังนั้นค่าสนับสนุนจึงเท่ากับ 3 เมื่อพิจารณารายการทั้งหมดปรากฏว่ารายการความถี่ประกอบไปด้วย C และ D

ตารางที่ 3.6 รายการความถี่ในฐานข้อมูลโปรเจคของ <(A)>

ฐานข้อมูลโปรเจคของ <(A)>
< (_ B) **(A C)** (D) >
< (C D) (E) >
< (C D) (F) >

เกิดในเหตุการณ์เดียวกับ A

รายการ	ค่าสนับสนุน	
A	1	✘
B	1	✘
C	3	
D	3	
E	1	✘
F	1	✘
(_C)	1	✘

<u>ขั้นตอนที่ 2.3</u> ทำการขยายรูปแบบลำดับเหตุการณ์กับรายการความถี่ รูปแบบลำดับเหตุการณ์ที่มีความยาว *k-1* ขยายเป็นรูปแบบลำดับเหตุการณ์ที่มีความยาว *k* โดยการขยายมี 2 แบบ คือ แบบรายการ และแบบลำดับเหตุการณ์ ดังที่นิยามไว้ในนิยามที่ 3.8 และ 3.9 ส่วนค่าสนับสนุนของรูปแบบลำดับเหตุการณ์ที่ขยายเสร็จแล้ว จะมีค่าเท่ากับค่าสนับสนุนของรายการความถี่

ตัวอย่างที่ 3.13 จากฐานข้อมูลโปรเจคของรูปแบบลำดับเหตุการณ์ <(A)> มีรายการความถี่ 2 รายการ คือ C กับ D ซึ่งทั้ง 2 รายการเป็นเหตุการณ์ที่เกิดหลังรูปแบบลำดับเหตุการณ์ <(A)> ดังนั้นทำการขยาย <(A)> แบบลำดับเหตุการณ์กับรายการความถี่ทั้งสอง จะได้รูปแบบลำดับเหตุการณ์ทั้งหมด 2 รูปแบบ ลำดับเหตุการณ์ คือ <(A) (C)>:3 และ <(A) (D)>:3 โดยค่าสนับสนุนของรูปแบบลำดับเหตุการณ์ทั้ง 2 รูปแบบ จะเท่ากับค่าสนับสนุนของรายการ C และ D ตามลำดับ

เมื่อได้รูปแบบลำดับเหตุการณ์ <(A) (C)> และ <(A) (D)> แล้วทำการขยายรูปแบบลำดับเหตุการณ์ <(A) (C)> ก่อนตามขั้นตอนที่ 2 จากนั้นทำการขยายรูปแบบลำดับเหตุการณ์ <(A) (D)> โดยเริ่มจากการสร้างฐานข้อมูลโปรเจคของ <(A) (C)> จะได้ผลลัพธ์ดังตารางที่ 3.7 (แถวแรก) พิจารณาฐานข้อมูลโปรเจคของรูปแบบลำดับเหตุการณ์ <(A) (C)> เพื่อหาค่าสนับสนุนของรายการในฐานข้อมูล โดยรายการและค่าสนับสนุนที่ได้แสดงได้ดังตารางที่ 3.8 พบว่ารายการความถี่มีรายการเดียว คือ (_D)

ตารางที่ 3.7 ฐานข้อมูลโปรเจค

รูปแบบลำดับเหตุการณ์	ฐานข้อมูลโปรเจค
<(A) (C)>	< (_ D) (E) > < (_ D) (F) >
<(A) (D)>	< (E) > < (F) >

ตารางที่ 3.8 รายการความถี่ในฐานข้อมูลโปรเจคของ <(A) (C)>

ฐานข้อมูลโปรเจคของ <(A) (C)>
(_ D) (E)
< (_ D) (F) >

รายการ	ค่าสนับสนุน	
_D	2	
E	1	✘
F	1	✘

จากนั้นทำการขยายรูปแบบลำดับเหตุการณ์ <(A) (C)> แบบรายการกับ (_D) จะได้รูปแบบลำดับเหตุการณ์ <(A) (CD)>:2

ทำการขยายรูปแบบลำดับเหตุการณ์ <(A) (CD)> ต่อตามขั้นตอนที่ 2 โดยเริ่มจากสร้างฐานข้อมูลโปรเจคของรูปแบบลำดับเหตุการณ์ <(A) (CD)> แล้วพบว่ามีลำดับเหตุการณ์ในฐานข้อมูลโปรเจค 2 ลำดับเหตุการณ์ คือ <(E)> และ <(F)> และไม่พบรายการความถี่ดังแสดงในตารางที่ 3.9 ดังนั้นจึงหยุดขยายรูปแบบลำดับเหตุการณ์ <(A) (CD)>

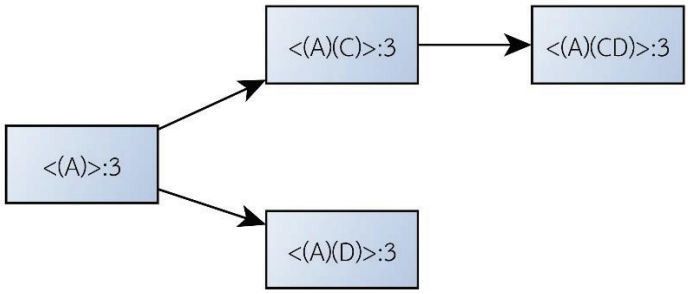
ตารางที่ 3.9 รายการความถี่

ฐานข้อมูลโปรเจคของ <(A) (CD)>
< (E) >
< (F) >

รายการ	ค่าสนับสนุน	
E	1	✖
F	1	✖

จากนั้นย้อนกลับไปขยายรูปแบบลำดับเหตุการณ์ <(A) (D)> สร้างฐานข้อมูลโปรเจคของรูปแบบลำดับเหตุการณ์ <(A) (D)> (แสดงในตารางที่ 3.7 แถวที่ 2) แล้วพบว่าไม่มีรายการความถี่ดังนั้นจึงหยุดขยายรูปแบบลำดับเหตุการณ์ <(A) (D)>

สรุปได้ว่ารูปแบบลำดับเหตุการณ์ <(A)> สามารถขยายรูปแบบลำดับเหตุการณ์ได้ทั้งหมด 4 รูปแบบลำดับเหตุการณ์ดังรูปที่ 3.2



รูปที่ 3.2 รูปแบบลำดับเหตุการณ์ที่ได้จากการขยาย <(A)>

เมื่อไม่สามารถขยายรูปแบบลำดับเหตุการณ์ <(A)> ได้แล้ว ย้อนกลับไปขยายรูปแบบลำดับเหตุการณ์ <(C)> โดยเริ่มจากสร้างฐานข้อมูลโปรเจคของรูปแบบลำดับเหตุการณ์ <(C)> (ดังตารางที่ 3.5 แถวที่ 2) และทำการขยายรูปแบบลำดับเหตุการณ์จนกว่าจะไม่สามารถขยายลำดับเหตุการณ์ได้

ต่อมาพิจารณาขยายรูปแบบลำดับเหตุการณ์ <(D)> และ <(E)> ตามลำดับ

จากชุดตัวอย่างข้อมูลในตารางที่ 3.2 สามารถขุดค้นรูปแบบลำดับเหตุการณ์ได้ทั้งหมด 15 รูปแบบลำดับเหตุการณ์ ดังแสดงในตารางที่ 3.10

ตารางที่ 3.10 รูปแบบลำดับเหตุการณ์ทั้งหมด

รูปแบบลำดับเหตุการณ์ความยาว 1	รูปแบบลำดับเหตุการณ์ความยาว k+1
<(A)>:3	<(A) (C)>:3, <(A) (CD)>:2, <(A) (D)>:3
<(C)>:4	<(CD) >:3, <(CD)(E) >:2, <(CD)(F) >:2 <(C) (E)>:2, <(C) (F)>:2
<(D)>:4	<(D)(E)>:2, <(D)(F)>:2
<(E)>:2	∅
<(F)>:2	∅

3.4 การทำเหมืองรูปแบบลำดับเหตุการณ์แบบอื่น

นอกจากจะมีการทำเหมืองรูปแบบลำดับเหตุการณ์แล้ว ยังมีการทำเหมืองรูปแบบลำดับเหตุการณ์แบบอื่น เช่น การทำเหมืองรูปแบบลำดับเหตุการณ์แบบปิด (Closed sequential pattern mining) การทำเหมืองรูปแบบลำดับเหตุการณ์ *k* (Top-k sequential pattern mining) และการทำเหมืองรูปแบบลำดับเหตุการณ์ความยาวสูงสุด (Maximal sequential pattern mining) เป็นต้น ซึ่งการทำเหมืองรูปแบบลำดับเหตุการณ์ดังกล่าวถูกนำเสนอเพื่อแก้ปัญหาการสร้างรูปแบบลำดับเหตุการณ์จำนวนมาก ซึ่งทำให้ยากต่อการนำไปใช้ในการวิเคราะห์ การทำเหมืองรูปแบบลำดับเหตุการณ์แบบปิด การทำเหมืองรูปแบบลำดับเหตุการณ์ *k* และการทำเหมืองรูปแบบลำดับเหตุการณ์ความยาวสูงสุด สามารถนิยามได้ดังต่อไปนี้

กำหนดให้ S_a และ S_b คือ ลำดับเหตุการณ์ และกำหนดให้ *FS* คือ เซตของรูปแบบลำดับเหตุการณ์

นิยามที่ 3.10 ลำดับเหตุการณ์ S_a เป็นรูปแบบลำดับเหตุการณ์แบบปิด ก็ต่อเมื่อ ค่าสนับสนุนของลำดับเหตุการณ์ S_a มากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำและลำดับเหตุการณ์ S_a ไม่ได้เป็นลำดับเหตุการณ์ย่อยของลำดับเหตุการณ์อื่นที่มีค่าสนับสนุนเท่ากัน เซตของรูปแบบลำดับเหตุการณ์แบบปิด สามารถเขียนแทนด้วยสมการที่ 3.1

$$CS = \{ S_a \mid S_a \in FS \wedge (\nexists S_b \in FS \mid S_a \sqsubseteq S_b \wedge supp(S_a) = supp(S_b))\} \quad (3.1)$$

ตัวอย่างที่ 3.14 จากตารางที่ 3.10 จะได้รูปแบบลำดับเหตุการณ์ทั้งหมด 10 รูปแบบ ดังนี้

FS = {<(A) (C)>:3, <(A) (CD)>:2, <(A) (D)>:3,< (CD)>:3, <(CD) (E)>:2, <(CD)(F)>:2, <(C) (E)>:2, <(C) (F)>:2, <(D) (E)>:2, <(D) (F)>:2}

พิจารณารูปแบบลำดับเหตุการณ์ <(D)(F)>:2 และ <(C)(F)>:2 ใน *FS* ปรากฏว่าเป็นลำดับเหตุการณ์ย่อยของ <(CD)(F) >:2 และมีค่าสนับสนุนเท่ากัน ดังนั้นรูปแบบลำดับเหตุการณ์ <(D)(F)>:2 และ <(C)(F)>:2 ไม่ใช่รูปแบบลำดับเหตุการณ์แบบปิด

พิจารณารูปแบบลำดับเหตุการณ์ <(C)(E)>:2 และ <(D)(E)>:2 ใน *FS* ปรากฏว่าเป็นลำดับเหตุการณ์ย่อยของ <(CD)(E) >:2 และมีค่าสนับสนุนเท่ากัน ดังนั้นรูปแบบลำดับเหตุการณ์ <(C)(E)>:2 และ <(D)(E)>:2 ไม่ใช่รูปแบบลำดับเหตุการณ์แบบปิด

ส่วนรูปแบบลำดับเหตุการณ์ที่เหลือใน *FS* ถือว่าเป็นรูปแบบลำดับเหตุการณ์แบบปิด เนื่องจากไม่ได้เป็นลำดับเหตุการณ์ย่อยของรูปแบบลำดับเหตุการณ์อื่นที่มีค่าสนับสนุนเท่ากัน

สรุปรูปแบบลำดับเหตุการณ์แบบปิดจะประกอบไปด้วย 6 รูปแบบดังนี้

CS = {<(A) (C)>:3, <(A) (CD)>:2, <(A) (D)>:3, <(CD) >:3, <(CD) (E) >:2, <(CD) (F) >:2}

ซึ่งจะเห็นได้ว่าจำนวนรูปแบบที่ได้จากการทำเหมืองรูปแบบลำดับเหตุการณ์แบบปิดมีจำนวนน้อยกว่าการทำเหมืองรูปแบบลำดับเหตุการณ์ ปัจจุบันมีหลายขั้นตอนวิธีถูกพัฒนาขึ้นเพื่อขุดค้นรูปแบบลำดับเหตุการณ์แบบปิด เช่น CloSpan, BIDE, ClaSP, CM-ClaSP และ CloFAST เป็นต้น

นิยามที่ 3.11 ลำดับเหตุการณ์ S_a เป็นรูปแบบลำดับเหตุการณ์ความยาวสูงสุด ก็ต่อเมื่อ ค่าสนับสนุนของลำดับเหตุการณ์ S_a มากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำและลำดับเหตุการณ์ S_a ไม่ได้เป็นลำดับเหตุการณ์ย่อยของลำดับเหตุการณ์อื่น เซตของรูปแบบลำดับเหตุการณ์ความยาวสูงสุด สามารถเขียนแทนด้วยสมการที่ 3.2

$$MS = \{ S_a \mid S_a \in FS \wedge (\nexists S_b \in FS \mid S_a \sqsubseteq S_b) \} \quad (3.2)$$

ตัวอย่างที่ 3.15 จากตัวอย่างที่ 3.14 พิจารณารูปแบบลำดับเหตุการณ์ <(A) (C)>:3, <(A) (D)>:3 และ <(CD)>:3 ใน *FS* ปรากฏว่าเป็นรูปแบบลำดับเหตุการณ์ย่อยของรูปแบบลำดับเหตุการณ์ <(A) (CD)>:2 ดังนั้นรูปแบบลำดับเหตุการณ์ <(A) (C)>:3, <(A) (D)>:3 และ <(CD)>:3 ไม่ใช่รูปแบบลำดับเหตุการณ์ความยาวสูงสุด

พิจารณารูปแบบลำดับเหตุการณ์ <(C) (E)>:2 และ <(D) (E)>:2 เป็นลำดับเหตุการณ์ย่อยของ <(CD) (E) >:2 ดังนั้น <(C) (E)>:2 และ <(D) (E)>:2 ไม่ใช่รูปแบบลำดับเหตุการณ์ความยาวสูงสุด

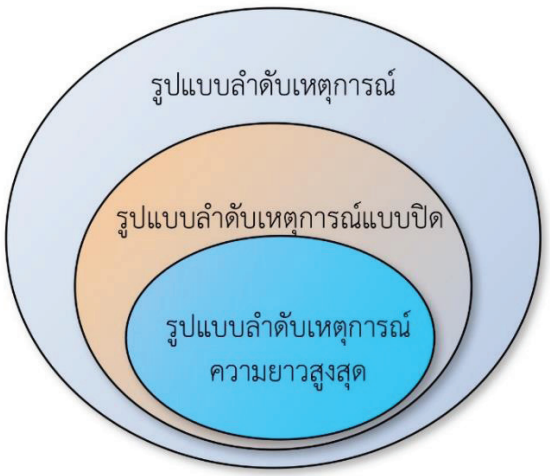
พิจารณารูปแบบ <(C) (F)>:2 และ <(D) (F)>:2 เป็นลำดับเหตุการณ์ย่อยของ <(CD) (F)>:2 จึงถือว่าไม่ใช่รูปแบบลำดับเหตุการณ์ความยาวสูงสุด

พิจารณารูปแบบลำดับเหตุการณ์ <(A) (CD)>:2, <(CD) (E)>:2 และ <(CD) (F)>:2 ไม่ใช่รูปแบบลำดับเหตุการณ์ย่อยของรูปแบบลำดับเหตุการณ์ใด จึงถือว่าเป็นรูปแบบลำดับเหตุการณ์ความยาวสูงสุด

สรุปรูปแบบลำดับเหตุการณ์สูงสุดทั้งหมดจะประกอบไปด้วย 3 รูปแบบ ดังนี้

MS = {<(A) (CD)>:2, <(CD) (E) >:2, <(CD) (F) >:2}

จะเห็นได้ว่าการทำเหมืองรูปแบบลำดับเหตุการณ์ความยาวสูงสุดจะให้จำนวนรูปแบบที่น้อยที่สุด (ดังรูปที่ 3.3) ขั้นตอนวิธีที่ถูกนำเสนอเพื่อขุดค้นรูปแบบลำดับเหตุการณ์ความยาวสูงสุด เช่น VMSP และ MaxSP เป็นต้น



รูปที่ 3.3 เปรียบเทียบจำนวนรูปแบบลำดับเหตุการณ์แบบต่างๆ

การทำเหมืองรูปแบบลำดับเหตุการณ์ *k* เป็นการทำเหมืองรูปแบบลำดับเหตุการณ์ที่มีค่าสนับสนุนสูงสุด *k* รูปแบบ การขุดค้นลักษณะแบบนี้ถูกนำเสนอขึ้นมาเพื่อให้ผู้ที่นำไปประยุกต์ใช้มีความสะดวก และง่ายต่อการวิเคราะห์ เนื่องจากผู้ใช้สามารถระบุจำนวนรูปแบบที่ต้องการได้ สมมติ *k* = 5 รูปแบบลำดับเหตุการณ์ที่มีค่าสนับสนุนสูงสุด 5 รูปแบบจะถูกขุดค้น แต่ถ้ามีรูปแบบลำดับเหตุการณ์ที่มีค่าสนับสนุนเท่ากับรูปแบบลำดับเหตุการณ์ตัวที่ *k* รูปแบบดังกล่าวจะถูกนำมาเป็นผลลัพธ์ด้วย เช่น เซต

ของรูปแบบลำดับเหตุการณ์ k = 2 คือ {<(A) (C)>:3, <(A) (D)>:3, <(CD) >:3} ขั้นตอนวิธีที่ใช้สำหรับขุดค้นรูปแบบลำดับเหตุการณ์ k เช่น TSP , TKS และ Skopus เป็นต้น

3.5 ตัวอย่างการทำเหมืองรูปแบบลำดับเหตุการณ์โดยใช้ SPMF

ใน SPMF มีขั้นตอนวิธีสำหรับการทำเหมืองรูปแบบลำดับเหตุการณ์หลายขั้นตอนวิธี นอกจากนี้ยังมีขั้นตอนวิธีสำหรับการทำเหมืองรูปแบบลำดับเหตุการณ์แบบปิด การทำเหมืองรูปแบบลำดับเหตุการณ์ความยาวสูงสุด ซึ่งแต่ละขั้นตอนวิธีสามารถเรียกใช้ได้ง่าย และการเตรียมชุดข้อมูลลำดับเหตุการณ์เพื่อใช้ใน SPMF สามารถทำได้ง่ายโดยแทนรายการเป็นตัวเลข รายละเอียดการเตรียมข้อมูลสำหรับการทำเหมืองรูปแบบลำดับเหตุการณ์มีดังต่อไปนี้

3.5.1 การเตรียมชุดข้อมูลลำดับเหตุการณ์

ชุดข้อมูลลำดับเหตุการณ์เป็นชุดข้อมูลที่พิจารณาลำดับการเกิดของข้อมูล ขั้นตอนวิธีที่นำมาใช้กับข้อมูลเหล่านี้ จะเป็นขั้นตอนวิธีที่ทำการประมวลผลหารูปแบบโดยพิจารณาลำดับการเกิดของข้อมูล เงื่อนไขการแปลงชุดข้อมูลลำดับเหตุการณ์สำหรับนำเข้าโปรแกรม SPMF มีดังนี้

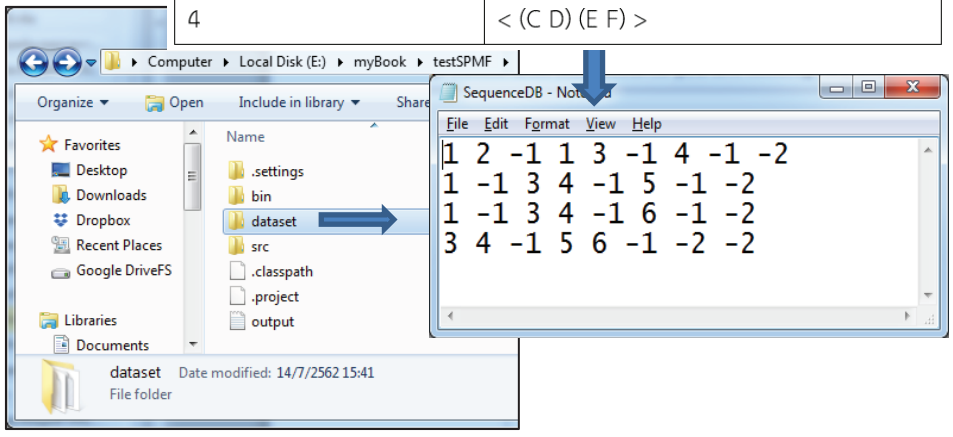
- ไฟล์นำเข้าอยู่ในรูปของไฟล์ข้อความ (Text file)
- รายการแต่ละรายการจะต้องแทนด้วยตัวเลขจำนวนเต็มบวก
- ใช้ 1 เว้นวรรค สำหรับคั่นระหว่างตัวเลข
- ในเหตุการณ์เดียวกัน ตัวเลขจะต้องเรียงลำดับจากน้อยไปมาก
- ลำดับเหตุการณ์ใน 1 รายการเปลี่ยนแปลงแทนด้วยข้อมูล 1 บรรทัดในไฟล์
- ใช้ -1 คั่นระหว่างเหตุการณ์
- ใช้ -2 เพื่อบ่งบอกการสิ้นสุดของลำดับเหตุการณ์
- ห้ามมีบรรทัดที่มีข้อมูล (ลำดับเหตุการณ์) เหมือนกัน

สมมติชุดข้อมูลที่ต้องการทำเหมืองรูปแบบลำดับเหตุการณ์เป็นดังตารางที่ 3.2 ซึ่งประกอบไปด้วย 4 รายการเปลี่ยนแปลง ทำการแทนรายการด้วยตัวเลขจำนวนเต็มบวกตามตารางที่ 3.11 เมื่อแทนค่าชุดข้อมูลด้วยตัวเลขแล้ว ทำการบันทึกไฟล์ชื่อ SequenceDB.txt และจัดเก็บในโฟลเดอร์ dataset ดังรูปที่ 3.4

ตารางที่ 3.11 การแทนค่าด้วยตัวเลขจำนวนเต็มในชุดข้อมูลลำดับเหตุการณ์

รายการ	การแทนค่า
A	1
B	2
C	3
D	4
E	5
F	6
G	7

รายการเปลี่ยนแปลง	ประวัติการจ่ายยา
1	< (A B) (A C) (D) >
2	< (A) (C D) (E) >
3	< (A) (C D) (F) >
4	< (C D) (E F) >



รูปที่ 3.4 ตัวอย่างไฟล์นำเข้าและการจัดเก็บ

จากรูปที่ 3.4 ข้อมูลในหนึ่งบรรทัดแทนด้วยลำดับเหตุการณ์ 1 ลำดับเหตุการณ์ เช่น ลำดับเหตุการณ์แรก คือ < (A B) (A C) (D) > ซึ่งมี (A B) เป็นเหตุการณ์แรก แล้วตามด้วยเหตุการณ์ (A C) แล้วตามด้วยเหตุการณ์ (D) เมื่อแทนด้วยตัวเลขตามตารางที่ 3.11 เหตุการณ์แรก จะได้ 1 2 เหตุการณ์ที่สองจะได้ 1 3 และเหตุการณ์ที่สามจะได้ 4 ใช้ -1 เป็นตัวคั่นระหว่างเหตุการณ์ และใช้ -2 เป็นตัวบอกการสิ้นสุดของลำดับเหตุการณ์ จะได้รูปแบบข้อมูลนำเข้า คือ 1 2 -1 1 3 -1 4 -1 -2 เป็นต้น

3.5.2 ตัวอย่างคำสั่งสำหรับการทำเหมืองรูปแบบลำดับเหตุการณ์

คำสั่งสำหรับการทำเหมืองรูปแบบลำดับเหตุการณ์ด้วยขั้นตอนวิธี PrefixSpan แสดงได้ดังตัวอย่างคำสั่งที่ 3.1 โดยแต่ละคำสั่งสามารถอธิบายได้ดังนี้

ตัวอย่างคำสั่งที่ 3.1

```
1.  package mySpmfProject;
2.
3.  import java.io.IOException;
4.  import ca.pfv.spmf.algorithms.sequentialpatterns.prefixspan.AlgoPrefixSpan;
5.
6.  public class PrefixSpanTest {
7.          public static void main(String [] arg) throws IOException{
8.
9.                  String input = ".//dataset/SequenceDB.txt";
10                 String output = ".//output.txt";
11.
12.                 double minsup = 0.5;
13.                 AlgoPrefixSpan algo = new AlgoPrefixSpan();
14.
15.                 algo.runAlgorithm(input, minsup, output);
16.                 algo.printStatistics();
17.         }
18  }
```

บรรทัดที่ 4 เป็นการ import คลาส AlgoPrefixSpan เพื่อเรียกใช้ขั้นตอนวิธี PrefixSpan

บรรทัดที่ 9 เป็นการกำหนดตำแหน่งชุดข้อมูลนำเข้า ซึ่งในตัวอย่างคือไฟล์ SequenceDB.txt

บรรทัดที่ 10 เป็นการกำหนดตำแหน่งไฟล์สำหรับเก็บผลลัพธ์ ซึ่งในไฟล์แสดงรูปแบบลำดับเหตุการณ์และค่าสนับสนุนแบบสัมบูรณ์ดังรูปที่ 3.6

บรรทัดที่ 12 เป็นการกำหนดค่าสนับสนุนขั้นต่ำแบบสัมพัทธ์ ซึ่งในตัวอย่างกำหนดค่าสนับสนุนขั้นต่ำให้มีค่าเท่ากับ 0.5 หรือ 50% เมื่อเทียบกับค่าสนับสนุนขั้นต่ำแบบสัมบูรณ์จะมีค่าเท่ากับ 2

บรรทัดที่ 13 เป็นการสร้างอ๊อบเจ็กต์ของคลาส AlgoPrefixSpan

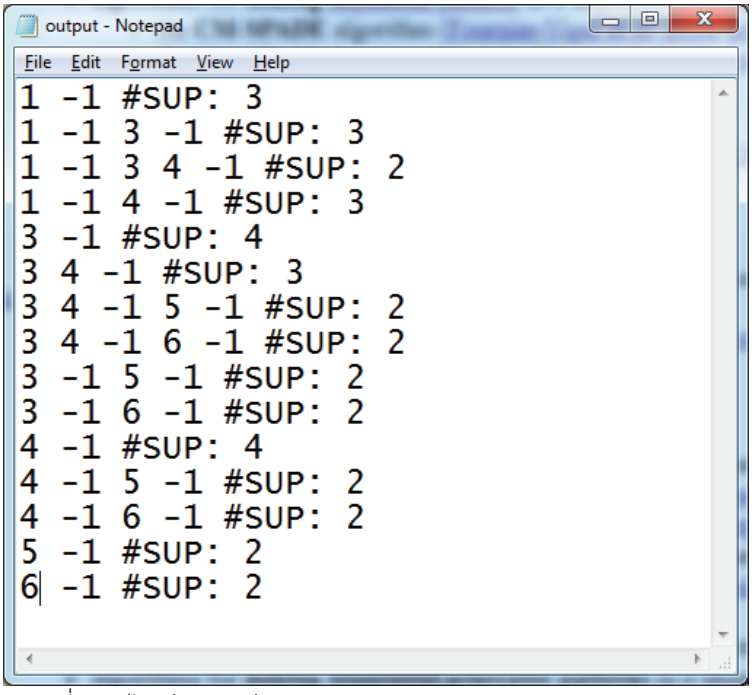
บรรทัดที่ 15 เป็นการเรียกใช้เมทธอด runAlgorithm เพื่อสั่งให้ขั้นตอนวิธี FP-Growth ประมวลผล โดยมีพารามิเตอร์ 3 ตัว คือ ไฟล์นำเข้า (input) ไฟล์ผลลัพธ์ (output) และค่าสนับสนุนขั้นต่ำ (minsup)

บรรทัดที่ 16 เป็นการแสดงค่าสถิติต่างๆ ที่ได้จากการประมวลผลดังรูปที่ 3.5 แสดงข้อมูลดังต่อไปนี้

- เวลาในการประมวลผล (Total time)
- จำนวนรูปแบบลำดับเหตุการณ์ (Frequent sequences count)
- หน่วยความจำที่ใช้ (Max memory)
- ค่าสนับสนุนขั้นต่ำแบบสัมบูรณ์ (minsup)
- จำนวนรูปแบบ (Pattern count)

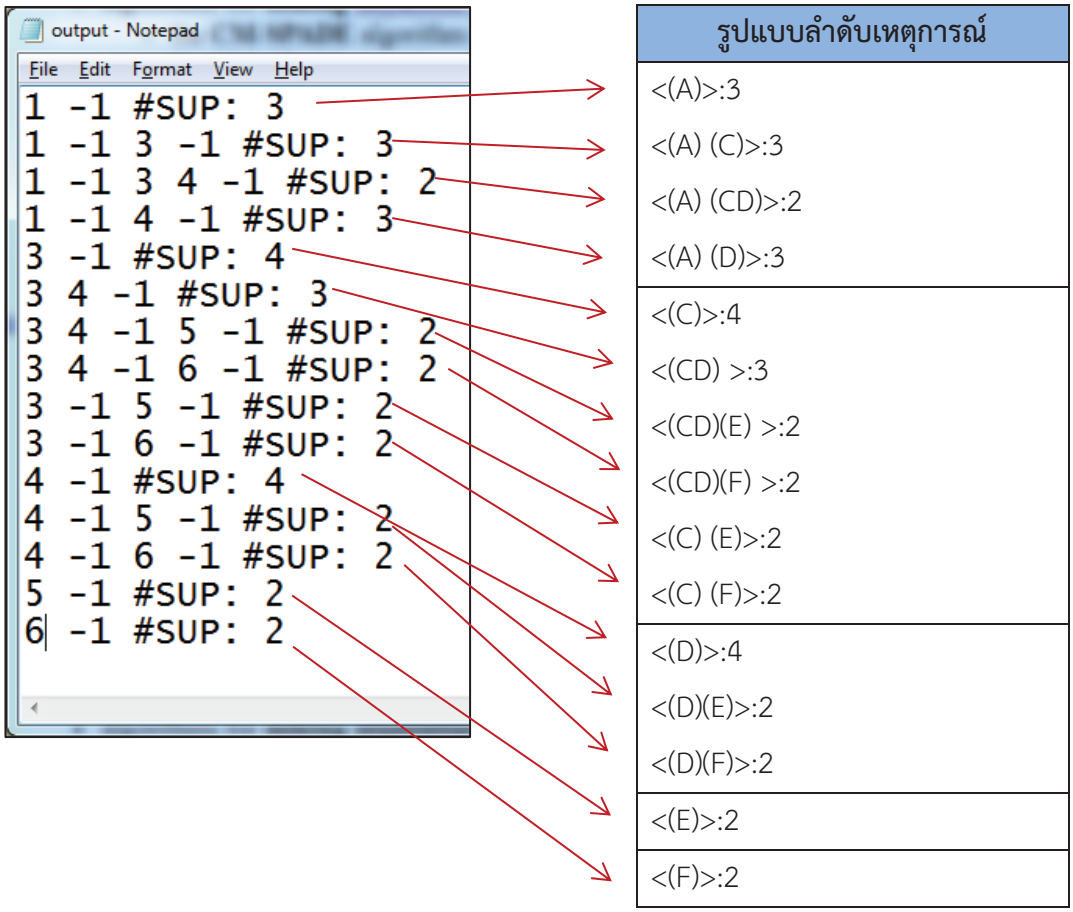
```
=============  PREFIXSPAN 0.99-2016 - STATISTICS =============
 Total time ~ 6 ms
 Frequent sequences count : 15
 Max memory (mb) : 10.969802856445312
 minsup = 2 sequences.
 Pattern count : 15
       ==================================================
```

รูปที่ 3.5 แสดงค่าทางสถิติจากการประมวลผล PrefixSpanTest.java



รูปที่ 3.6 ไฟล์ผลลัพธ์จากการประมวลผล PrefixSpanTest.java

เมื่อแปลงผลลัพธ์กลับคืนตามตารางที่ 3.11 จะได้ดังรูปที่ 3.7



รูปที่ 3.7 แปลงผลลัพธ์รูปแบบลำดับเหตุการณ์

โดยรูปแบบลำดับเหตุการณ์ที่ได้ สามารถแปลความหมายได้ดังตัวอย่างต่อไปนี้

รูปแบบลำดับเหตุการณ์ที่ 1 มีการจ่ายยา A ให้กับผู้ป่วยจำนวน 3 คน

รูปแบบลำดับเหตุการณ์ที่ 2 มีการจ่ายยา A แล้วตามด้วยจ่ายยา C ให้กับผู้ป่วยจำนวน 3 คน

รูปแบบลำดับเหตุการณ์ที่ 3 มีการจ่ายยา A แล้วตามด้วยจ่ายยา C พร้อมกับยา D ให้กับผู้ป่วยจำนวน 2 คน

บทสรุป

การทำเหมืองรูปแบบลำดับเหตุการณ์เป็นการค้นหารูปแบบของข้อมูลที่เกิดขึ้นบ่อย ซึ่งพิจารณาจากรูปแบบที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ โดยรูปแบบดังกล่าวจะต้องพิจารณาลำดับการเกิดของข้อมูลด้วย ปัจจุบันมีการนำเสนอขั้นตอนวิธีต่างๆ สำหรับการทำเหมืองรูปแบบลำดับเหตุการณ์ ขั้นตอนวิธี PrefixSpan เป็นขั้นตอนวิธีหนึ่งที่มีประสิทธิภาพในการทำเหมืองรูปแบบลำดับเหตุการณ์ ทำการค้นหารูปแบบลำดับเหตุการณ์แนวลึกแบบเรียกซ้ำ เพื่อทำให้ขนาดของข้อมูลที่ใช้ในการค้นหารูปแบบลำดับเหตุการณ์มีขนาดลดลงเรื่อยๆ นอกจากนี้ขั้นตอนวิธี PrefixSpan ทำการค้นหารูปแบบลำดับเหตุการณ์โดยไม่สร้างรูปแบบลำดับเหตุการณ์คู่แข่ง

การทำเหมืองรูปแบบลำดับเหตุการณ์แบบปิด การทำเหมืองรูปแบบลำดับเหตุการณ์ k และการทำเหมืองรูปแบบลำดับเหตุการณ์ความยาวสูงสุด ถูกนำเสนอขึ้นมาเพื่อแก้ปัญหาการสร้างรูปแบบลำดับเหตุการณ์จำนวนมากในการทำเหมืองรูปแบบลำดับเหตุการณ์

แบบฝึกหัดท้ายบท

1. จงอธิบายความหมายของการทำเหมืองรูปแบบลำดับเหตุการณ์

2. ถ้ารูปแบบเหตุการณ์ทางธรรมชาติเป็นดังนี้ <(A B C) (A C) (C)> จงอธิบายลำดับเหตุการณ์ที่เกิดขึ้นว่าเป็นอย่างไร

3. จงยกตัวอย่างการประยุกต์ใช้การทำเหมืองรูปแบบลำดับเหตุการณ์

4. ถ้าข้อมูลปรากฏการณ์ทางธรรมชาติเป็นดังตารางชุดข้อมูลข้างล่าง จงค้นหาฐานข้อมูลโปรเจคของรูปแบบลำดับเหตุการณ์ <(c)>

รายการเปลี่ยนแปลง	ปรากฏการณ์ธรรมชาติ
1	< (a c) (b c) >
2	< (c d) (e f) >
3	< (c) (c d f) >
4	< (a c d) (d e f) >

5. จากชุดข้อมูลในข้อ 4 ถ้ากำหนดค่าสนับสนุนขั้นต่ำแบบสัมบูรณ์ให้มีค่าเท่ากับ 2 จงแสดงวิธีการค้นหารูปแบบลำดับเหตุการณ์ที่เกิดจากการขยายรูปแบบลำดับเหตุการณ์ <(a)>

6. จากชุดข้อมูลในข้อ 4 จงแปลงข้อมูลสำหรับการทำเหมืองรูปแบบลำดับเหตุการณ์ เพื่อนำเข้า SPMF

7. จงเขียนโปรแกรมโดยใช้ SPMF เพื่อค้นหารูปแบบลำดับเหตุการณ์ โดยใช้ไฟล์ชุดข้อมูลที่ได้จากการแปลงข้อมูลในข้อ 6 และกำหนดให้ค่าสนับสนุนขั้นต่ำแบบสัมบูรณ์มีค่าเท่ากับ 3

8. ถ้าเซตของรูปแบบลำดับเหตุการณ์ คือ *FS* = {<(ab) (cd)>:2 , <(ab) (c)>:2, <(b) (c)>:2, <(ab)>:2, <(b) (ef)>:2, <(b) (f)>:3} จงหาเซตของรูปแบบลำดับเหตุการณ์แบบปิด

9. จงอธิบายความหมายของการทำเหมืองรูปแบบลำดับเหตุการณ์ความยาวสูงสุด

10. จงอธิบายความหมายของการทำเหมืองรูปแบบลำดับเหตุการณ์ k

บทที่ 4
การทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติ
(Multidimensional Sequential Pattern Mining)

การทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติ คือ กระบวนการค้นหารูปแบบที่น่าสนใจจากฐานข้อมูลที่ประกอบไปด้วยส่วนที่พิจารณาลำดับการเกิดและส่วนที่ไม่พิจารณาลำดับการเกิด เทคนิคการทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติเป็นการผสมผสานเทคนิคการทำเหมืองเซตรายการความถี่และการทำเหมืองรูปแบบลำดับเหตุการณ์ ทำให้ได้รูปแบบที่มีประโยชน์มากกว่าการทำเหมืองรูปแบบลำดับเหตุการณ์อย่างเดียวหรือการทำเหมืองเซตรายการความถี่อย่างเดียว เช่น รูปแบบลำดับเหตุการณ์หลายมิติ (นักธุรกิจ, กรุงเทพฯ, วัยกลางคน, <(สูทยี่ห้อ Arrow, ไทน์ ยี่ห้อ Dapper) (รองเท้ายี่ห้อ Chap)> รูปแบบลำดับเหตุการณ์หลายมิติดังกล่าวทำให้รู้ว่า กลุ่มลูกค้าที่เป็นนักธุรกิจอาศัยอยู่ในกรุงเทพฯ วัยกลางคนจะชอบซื้อชุดสูทยี่ห้อ Arrow และเนคไทยี่ห้อ Dapper พร้อมกัน แล้วต่อมาจะซื้อรองเท้ายี่ห้อ Chap ซึ่งทำให้ร้านสามารถแนะนำสินค้าให้ตรงกับลักษณะของลูกค้าได้ เป็นต้น

4.1 ลักษณะข้อมูลสำหรับการทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติ

ชุดข้อมูลที่ใช้สำหรับการทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติ แบ่งข้อมูลออกเป็น 2 ส่วน คือ ข้อมูลสารสนเทศหลายมิติ (Multidimensional information) และข้อมูลลำดับเหตุการณ์ (Sequence) โดยรายละเอียดของแต่ละส่วนมีดังนี้

ส่วนที่ 1 ข้อมูลสารสนเทศหลายมิติเป็นข้อมูลที่ไม่พิจารณาลำดับการเกิดของข้อมูลเหมือนกับข้อมูลเซตรายการ ข้อมูลถูกเก็บอยู่ในแอตทริบิวต์หลายๆ แอตทริบิวต์ ซึ่งหนึ่งแอตทริบิวต์ถูกพิจารณาเป็นหนึ่งมิติ

ส่วนที่ 2 ข้อมูลลำดับเหตุการณ์เป็นข้อมูลที่ต้องพิจารณาลำดับการเกิด

สมมติชุดข้อมูลลำดับเหตุการณ์หลายมิติเป็นดังตารางที่ 4.1 ประกอบไปด้วย 4 รายการเปลี่ยนแปลง และชุดข้อมูลประกอบไปข้อมูล 2 ส่วน คือ

ส่วนที่ 1 เป็นข้อมูลพื้นฐานของผู้ป่วย ซึ่งเป็นข้อมูลสารสนเทศหลายมิติที่ประกอบไป 5 มิติ หรือ 5 แอตทริบิวต์ ดังต่อไปนี้

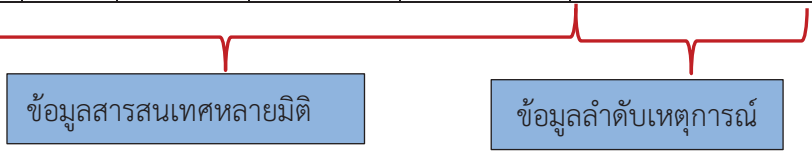
- มิติที่ 1 คือ ข้อมูลภูมิลำเนา ซึ่งประกอบไปด้วยค่า {มหาสารคาม, กาฬสินธุ์, ร้อยเอ็ด} (หมายถึงค่าที่เป็นไปได้ในมิติที่ 1)

- มิติที่ 2 คือ ข้อมูลเพศ ซึ่งประกอบไปด้วยค่า {หญิง, ชาย}
- มิติที่ 3 คือ ข้อมูลระดับการศึกษา ซึ่งประกอบไปด้วยค่า {ตรี, โท}
- มิติที่ 4 คือ ข้อมูลสถานภาพ ซึ่งประกอบไปด้วยค่า {โสด, แต่งงาน}
- มิติที่ 5 คือ ข้อมูลอาชีพ ซึ่งประกอบไปด้วยค่า {เกษตรกร, อาจารย์}

ส่วนที่ 2 เป็นข้อมูลลำดับการจ่ายยาแก่ผู้ป่วย ซึ่งเป็นข้อมูลลำดับเหตุการณ์ โดยประกอบไปด้วยเซตของยา คือ {A, B, C, D, E, F}

ตารางที่ 4.1 ตัวอย่างชุดข้อมูลลำดับเหตุการณ์หลายมิติ

รายการเปลี่ยนแปลง	ภูมิลำเนา	เพศ	ระดับการศึกษา	สถานภาพ	อาชีพ	ประวัติการจ่ายยา
1	มหาสารคาม	หญิง	ตรี	แต่งงาน	อาจารย์	< (A B) (A C) (D) >
2	มหาสารคาม	ชาย	ตรี	โสด	อาจารย์	< (A) (C D) (E) >
3	มหาสารคาม	ชาย	โท	แต่งงาน	อาจารย์	< (A) (C D) (F) >
4	ร้อยเอ็ด	ชาย	โท	โสด	เกษตรกร	< (C D) (E F) >



จากตารางที่ 4.1 ในรายการเปลี่ยนแปลงที่ 1 ส่วนของข้อมูลสารสนเทศหลายมิติ คือ มหาสารคาม หญิง ตรี แต่งงาน อาจารย์ ส่วนของข้อมูลลำดับเหตุการณ์ คือ < (A B) (A C) (D) >

4.2 นิยามที่เกี่ยวข้อง

กำหนดให้ $F = \{A_1, A_2, \ldots, A_p\}$ คือ ข้อมูลสารสนเทศหลายมิติหรือเซตของแอตทริบิวต์ในชุดข้อมูล

กำหนดให้ * แทนค่าใดๆ ที่ไม่ได้อยู่ใน $A_1, A_2, \ldots, A_p$ และ $T = \{t_1, t_2, \ldots, t_k\}$ คือ เซตของรายการเปลี่ยนแปลงทั้งหมดในชุดข้อมูล ในแต่ละรายการเปลี่ยนแปลง t_i มีลำดับเหตุการณ์หลายมิติ $M_i = (a_1, a_2, \ldots, a_p, s_i)$ โดยที่ $a_i \in (A_i \cup \{*\})$ สำหรับ $(1 \le i \le p)$ และ s_i คือ ลำดับเหตุการณ์

นิยามที่ 4.1 ลำดับเหตุการณ์หลายมิติ M_i ตรงกับลำดับเหตุการณ์หลายมิติ M_j ก็ต่อเมื่อ $a_i = a_j$ หรือ $a_i = *$ และ $s_i = s_j$

นิยามที่ 4.2 ค่าสนับสนุนของลำดับเหตุการณ์หลายมิติ M_i คือ จำนวนรายการเปลี่ยนแปลงที่มีลำดับเหตุการณ์หลายมิติตรงกับ M_i

ตัวอย่างที่ 4.1 สมมติลำดับเหตุการณ์หลายมิติ M = (มหาสารคาม, *, *, *, <(CD)>) หมายความว่า ค่าแอตทริบิวต์ที่ 1 มีค่า “มหาสารคาม” ส่วนค่าในแอตทริบิวต์ที่ 2 3 และ 4 เป็น * หมายถึงค่าใดๆ และมีลำดับเหตุการณ์ <(CD)> จากตารางที่ 4.1 ลำดับเหตุการณ์หลายมิติที่ตรงกับลำดับเหตุการณ์หลายมิติ M อยู่ในรายการเปลี่ยนแปลงที่ 2 กับ 3 แสดงว่าค่าสนับสนุนของ M มีค่าเท่ากับ 2

นิยามที่ 4.3 ลำดับเหตุการณ์หลายมิติ M เรียกว่า รูปแบบลำดับเหตุการณ์หลายมิติ ก็ต่อเมื่อ มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ

ตัวอย่างที่ 4.2 ลำดับเหตุการณ์หลายมิติ M = (มหาสารคาม, *, *, *, <(CD)>) ถือว่าเป็นรูปแบบลำดับเหตุการณ์หลายมิติ ถ้ากำหนดค่าสนับสนุนขั้นต่ำเท่ากับ 2 เนื่องจาก M มีค่าสนับสนุนเท่ากับค่าสนับสนุนขั้นต่ำ

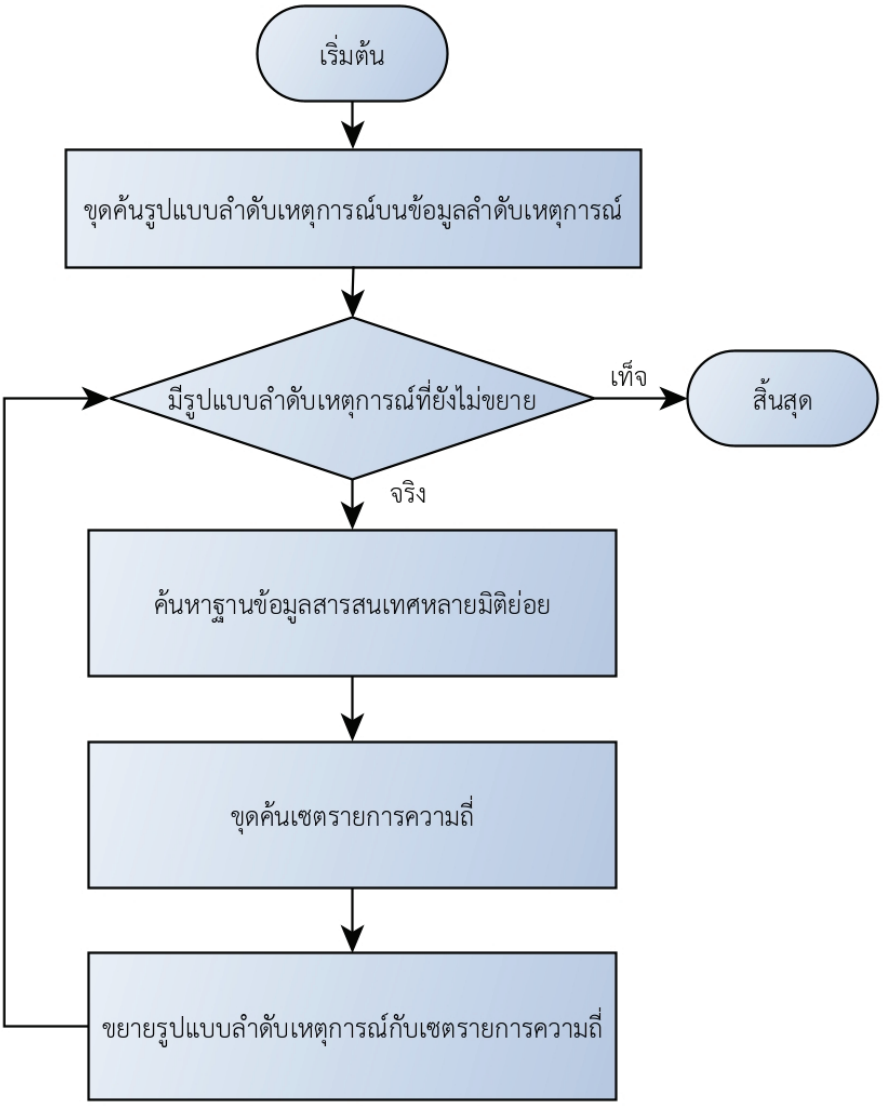
4.3 ขั้นตอนวิธีสำหรับการทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติ

การทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติถูกนำเสนอขึ้นครั้งแรกโดย Helen Pinto และคณะ ซึ่งได้นำเสนอ 3 ขั้นตอนวิธีที่มีชื่อว่า Seq-Dim, Dim-Seq และ UniSeq ซึ่งทั้งสามขั้นตอนวิธีมีลักษณะการทำงานดังนี้

4.3.1 ขั้นตอนวิธี Seq-Dim

ขั้นตอนวิธี Seq-Dim เป็นขั้นตอนวิธีที่ผสมผสานการทำเหมืองเซตรายการความถี่และการทำเหมืองรูปแบบลำดับเหตุการณ์ไว้ด้วยกัน โดยขั้นตอน Seq-Dim เริ่มขุดค้นรูปแบบลำดับเหตุการณ์บนข้อมูลลำดับเหตุการณ์ก่อน จากนั้นทำการขุดค้นเซตรายการความถี่บนข้อมูลสารสนเทศหลายมิติย่อย

ขั้นตอนการทำงานของขั้นตอนวิธี Seq-Dim แสดงดังรูปที่ 4.1 และมีรายละเอียดต่อไปนี้



รูปที่ 4.1 ขั้นตอนการทำงานของขั้นตอนวิธี Seq-Dim

ขั้นตอนที่ 1 ทำการขุดค้นรูปแบบลำดับเหตุการณ์บนข้อมูลลำดับเหตุการณ์ทั้งหมดโดยใช้การทำเหมืองรูปแบบลำดับเหตุการณ์

ตัวอย่างที่ 4.3 ตารางที่ 4.1 ถ้ากำหนดค่าสนับสนุนขั้นต่ำเท่ากับ 3 เมื่อขุดค้นรูปแบบลำดับเหตุการณ์บนข้อมูลลำดับเหตุการณ์แล้ว จะได้รูปแบบลำดับเหตุการณ์ทั้งหมด 6 รูปแบบ คือ <(A)>: 3, <(C)>: 4, <(CD)>: 3, <(D)>: 4, <(A) (D)>: 3, <(A) (C)>: 3

ขั้นตอนที่ 2 ทำการตรวจสอบว่ามีรูปแบบลำดับเหตุการณ์ที่ต้องขยายหรือไม่ ถ้าไม่มี จะหยุดการค้นหา แต่ถ้ายังมีรูปแบบลำดับเหตุการณ์ที่ต้องขยาย จะทำการค้นหาฐานข้อมูลสารสนเทศหลายมิติย่อยของรูปแบบลำดับเหตุการณ์นั้น

ตัวอย่างที่ 4.4 พิจารณารูปแบบลำดับเหตุการณ์ <(A)>: 3 ปรากฏในรายการเปลี่ยนแปลงที่ 1 2 และ 3 (ดังตารางที่ 4.1) ดังนั้นฐานข้อมูลสารสนเทศหลายมิติย่อยของ <(A)>: 3 แสดงได้ดังตารางที่ 4.2

ตารางที่ 4.2 ฐานข้อมูลสารสนเทศหลายมิติย่อยของ <(A)>: 3

รายการเปลี่ยนแปลง	ภูมิลำเนา	เพศ	ระดับการศึกษา	สถานภาพ	อาชีพ
1	มหาสารคาม	หญิง	ตรี	แต่งงาน	อาจารย์
2	มหาสารคาม	ชาย	ตรี	โสด	อาจารย์
3	มหาสารคาม	ชาย	โท	แต่งงาน	อาจารย์

ขั้นตอนที่ 3 ทำการขุดค้นเซตรายการความถี่บนฐานข้อมูลสารสนเทศหลายมิติย่อย โดยใช้การทำเหมืองเซตรายการความถี่

ตัวอย่างที่ 4.5 เมื่อขุดค้นเซตรายการความถี่บนข้อมูลสารสนเทศหลายมิติย่อยของ <(A)>: 3 จะได้เซตรายการความถี่ทั้งหมด 4 รูปแบบ คือ

(*, *, *, *, *): 3

(มหาสารคาม, *, *, *, *): 3

(*, *, *, *, อาจารย์):3

(มหาสารคาม, *, *, *, อาจารย์):3

ขั้นตอนที่ 4 สร้างรูปแบบลำดับเหตุการณ์หลายมิติโดยขยายรูปแบบลำดับเหตุการณ์ *S* กับเซตรายการความถี่ทั้งหมดที่ได้จากการขุดค้นบนฐานข้อมูลสารสนเทศหลายมิติย่อยของ *S* โดยค่าสนับสนุนของรูปแบบลำดับเหตุการณ์หลายมิติจะเท่ากับค่าสนับสนุนของเซตรายการความถี่

ตัวอย่างที่ 4.6 ขยายรูปแบบลำดับเหตุการณ์ <(A)>:3 กับเซตรายการความถี่ทั้งหมด สามารถสร้างรูปแบบลำดับเหตุการณ์หลายมิติได้ทั้งหมด 4 รูปแบบ คือ

(*, *, *, *, *, <(A)>): 3

(มหาสารคาม, *, *, *, *, <(A)>): 3

(*, *, *, *, อาจารย์, <(A)>):3

(มหาสารคาม, *, *, *, อาจารย์, <(A)>):3

เมื่อทำขั้นตอนที่ 4 เสร็จแล้ววนกลับไปทำซ้ำตามขั้นตอนที่ 2-4 เพื่อขยายรูปแบบลำดับเหตุการณ์ตัวถัดไป วนทำซ้ำไปเรื่อยๆ จนกว่าจะขยายรูปแบบลำดับเหตุการณ์หมดทุกรูปแบบ

ตัวอย่างที่ 4.7 ทำการขยายรูปแบบลำดับเหตุการณ์ <(C)>: 4, <(CD)>: 3, <(D)>: 4, <(A) (D)>: 3 และ <(A) (C)>: 3 ตามลำดับ

เมื่อขยายครบทุกรูปแบบลำดับเหตุการณ์จะได้รูปแบบลำดับเหตุการณ์หลายมิติทั้งหมด 24 รูปแบบดังตารางที่ 4.3

ตารางที่ 4.3 รูปแบบลำดับเหตุการณ์หลายมิติทั้งหมด

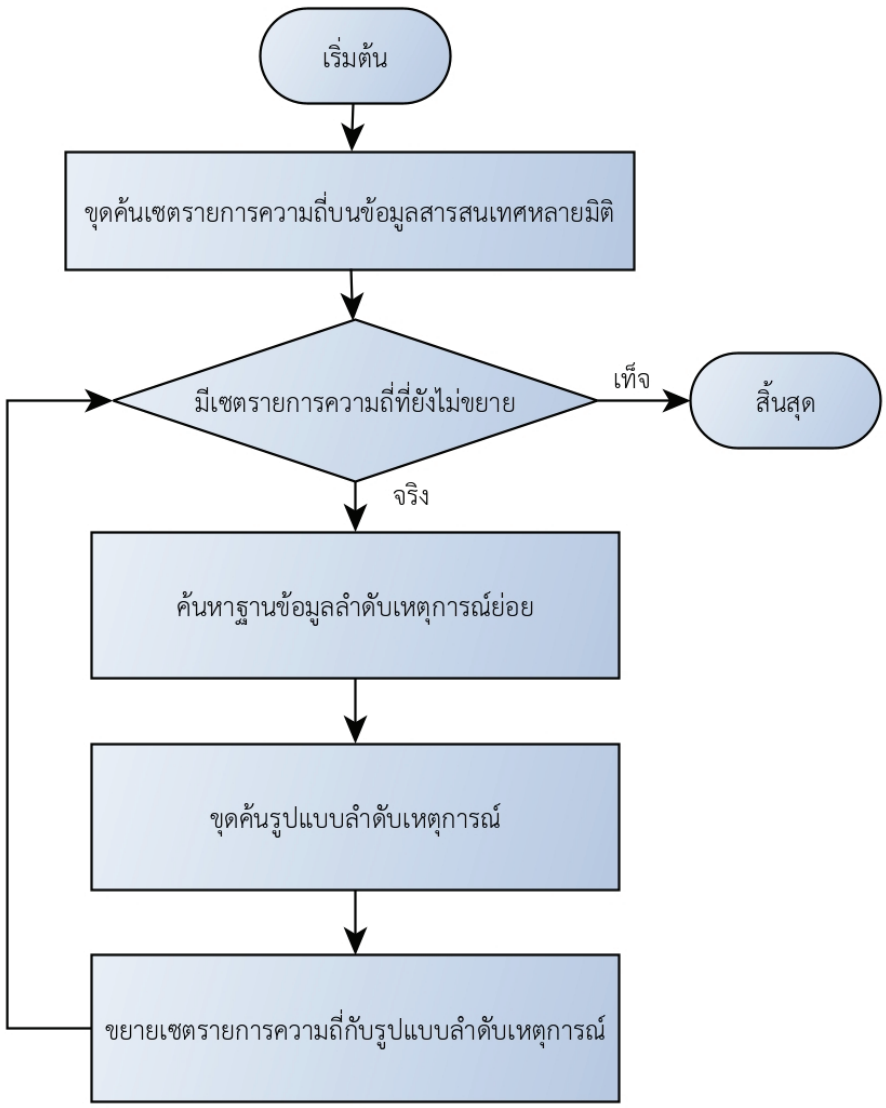
รูปแบบลำดับเหตุการณ์	**เซตรายการความถี่**	**รูปแบบลำดับเหตุการณ์หลายมิติ**
<(A)>:3	(*, *, *, *, *): 3 (มหาสารคาม, *, *, *, *): 3 (*, *, *, *, อาจารย์):3 (มหาสารคาม, *, *, *, อาจารย์):3	(*, *, *, *, *, <(A)>): 3 (มหาสารคาม, *, *, *, *): 3 (*, *, *, *, อาจารย์, <(A)>):3 (มหาสารคาม, *, *, *, อาจารย์, <(A)>):3
<(C)>: 4	(*, *, *, *, *): 4 (มหาสารคาม, *, *, *, *): 3 (*, *, *, *, อาจารย์):3 (*, ชาย, *, *, *): 3 (มหาสารคาม, *, *, *, อาจารย์):3	(*, *, *, *, *, <(C)>): 4 (มหาสารคาม, *, *, *, *, <(C)>): 3 (*, *, *, *, อาจารย์, <(C)>):3 (*, ชาย, *, *, *, <(C)>): 3 (มหาสารคาม, *, *, *, อาจารย์, <(C)>):3
<(CD)>: 3	(*, *, *, *, *): 3 (*, ชาย, *, *, *): 3	(*, *, *, *, *, <(CD)>): 3 (*, ชาย, *, *, *, <(CD)>): 3

รูปแบบลำดับเหตุการณ์	เซตรายการความถี่	รูปแบบลำดับเหตุการณ์หลายมิติ
<(D)>: 4	(*, *, *, *, *): 4 (มหาสารคาม, *, *, *, *): 3 (*, *, *, *, อาจารย์):3 (*, ชาย, *, *, *): 3 (มหาสารคาม, *, *, *, อาจารย์):3	(*, *, *, *, *, <(D)>): 4 (มหาสารคาม, *, *, *, *, <(D)>): 3 (*, *, *, *, อาจารย์, <(D)>):3 (*, ชาย, *, *, *, <(D)>): 3 (มหาสารคาม, *, *, *, อาจารย์, <(D)>):3
<(A)(D)>: 3	(*, *, *, *, *): 3 (มหาสารคาม, *, *, *, *): 3 (*, *, *, *, อาจารย์):3 (มหาสารคาม, *, *, *, อาจารย์):3	(*, *, *, *, *, <(A)(D)>): 3 (มหาสารคาม, *, *, *, *, <(A)(D)>): 3 (*, *, *, *, อาจารย์, <(A)(D)>):3 (มหาสารคาม, *, *, *, อาจารย์, <(A)(D)>):3
<(A)(C)>: 3	(*, *, *, *, *): 3 (มหาสารคาม, *, *, *, *): 3 (*, *, *, *, อาจารย์):3 (มหาสารคาม, *, *, *, อาจารย์):3	(*, *, *, *, *, <(A)(C)>): 3 (มหาสารคาม, *, *, *, *, <(A)(C)>): 3 (*, *, *, *, อาจารย์, <(A)(C)>):3 (มหาสารคาม, *, *, *, อาจารย์, <(A)(C)>):3

4.3.2 ขั้นตอนวิธี Dim-Seq

ขั้นตอนวิธี Dim-Seq คล้ายกับขั้นตอนวิธี Seq-Dim คือ เป็นขั้นตอนวิธีที่ผสมผสานการทำเหมืองเซตรายการความถี่และการทำเหมืองรูปแบบลำดับเหตุการณ์ไว้ด้วยกัน แต่แตกต่างกันตรงที่ขั้นตอน Dim-Seq ทำการขุดค้นเซตรายการความถี่บนข้อมูลสารสนเทศหลายมิติก่อน จากนั้นขุดค้นรูปแบบลำดับเหตุการณ์บนฐานข้อมูลลำดับเหตุการณ์ย่อย

ขั้นตอนการทำงานของขั้นตอนวิธี Dim-Seq แสดงดังรูปที่ 4.2 และมีรายละเอียดต่อไปนี้



รูปที่ 4.2 ขั้นตอนการทำงานของขั้นตอนวิธี Dim-Seq

ขั้นตอนที่ 1 ทำการขุดค้นเซตรายการความถี่บนข้อมูลสารสนเทศหลายมิติทั้งหมด โดยใช้การทำเหมืองเซตรายการความถี่

ตัวอย่างที่ 4.8 ตารางที่ 4.1 ถ้ากำหนดค่าสนับสนุนขั้นต่ำเท่ากับ 3 เมื่อขุดค้นเซตรายการความถี่บนข้อมูลสารสนเทศหลายมิติแล้ว จะได้เซตรายการความถี่ทั้งหมด 5 เซตรายการ คือ

(มหาสารคาม, *, *, *, *): 3

(มหาสารคาม, *, *, *, อาจารย์): 3

(*, ชาย, *, *, *): 3

(*, *, *, *, อาจารย์): 3

(*, *, *, *, *): 4

ขั้นตอนที่ 2 ทำการตรวจสอบว่ามีเซตรายการความถี่ที่ยังไม่ได้ขยายหรือไม่ ถ้าไม่มีจะทำการหยุดค้นหา แต่ถ้ายังมีเซตรายการความถี่ที่ต้องขยาย จะทำการค้นหาฐานข้อมูลลำดับเหตุการณ์ย่อยของเซตรายการความถี่นั้น

ตัวอย่างที่ 4.9 พิจารณาเซตรายการความถี่ (มหาสารคาม, *, *, *, *) ปรากฏในรายการเปลี่ยนแปลงที่ 1 2 และ 3 ในตารางที่ 4.1 ดังนั้นฐานข้อมูลลำดับเหตุการณ์ย่อยของ (มหาสารคาม, *, *, *, *) คือ ลำดับเหตุการณ์ที่อยู่ในรายการเปลี่ยนแปลงดังกล่าว ซึ่งก็คือ {<(A B) (A C) (D) >, < (A) (C D) (E) >, < (A) (C D) (F) >} ดังตารางที่ 4.4

ตารางที่ 4.4 ฐานข้อมูลลำดับเหตุการณ์ย่อยของ (มหาสารคาม, *, *, *, *): 3

รายการเปลี่ยนแปลง	ฐานข้อมูลลำดับเหตุการณ์ย่อย
1	< (A B) (A C) (D) >
2	< (A) (C D) (E) >
3	< (A) (C D) (F) >

ขั้นตอนที่ 3 ทำการขุดค้นรูปแบบลำดับเหตุการณ์บนฐานข้อมูลลำดับเหตุการณ์ย่อย โดยใช้การทำเหมืองรูปแบบลำดับเหตุการณ์

ตัวอย่างที่ 4.10 เมื่อขุดค้นรูปแบบลำดับเหตุการณ์จากฐานข้อมูลลำดับเหตุการณ์ย่อยของ (มหาสารคาม, *, *, *, *) ได้รูปแบบลำดับเหตุการณ์ทั้งหมด 5 รูปแบบ คือ <(A)>:3, <(A)(C)>:3, <(A)(D)>:3, <(C)>:3, <(D)>:3

ขั้นตอนที่ 4 สร้างรูปแบบลำดับเหตุการณ์หลายมิติโดยขยายเซตรายการความถี่ X และรูปแบบลำดับเหตุการณ์ทั้งหมดที่ขุดค้นจากฐานข้อมูลลำดับเหตุการณ์ย่อยของ X โดยกำหนดให้ค่าสนับสนุนของรูปแบบลำดับเหตุการณ์หลายมิติเท่ากับค่าสนับสนุนของรูปแบบลำดับเหตุการณ์

ตัวอย่างที่ 4.11 เมื่อขยายเซตรายการความถี่ (มหาสารคาม, * , *, *, *): 3 กับรูปแบบลำดับเหตุการณ์ <(A)>:3, <(A)(C)>:3, <(A)(D)>:3, <(C)>:3 และ<(D)>:3 สามารถสร้างรูปแบบลำดับเหตุการณ์หลายมิติได้ทั้งหมด 5 รูปแบบ คือ

(มหาสารคาม, *, *, *, *, <(A)>):3

(มหาสารคาม, *, *, *, *, <(A)(C)>):3

(มหาสารคาม, *, *, *, *, <(A)(D)>):3

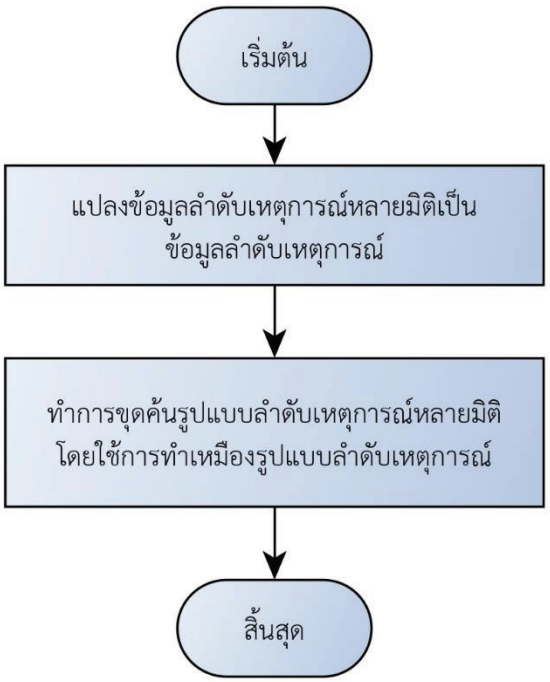
(มหาสารคาม, *, *, *, *, <(C)>):3

(มหาสารคาม, *, *, *, *, <(D)>):3

เมื่อทำขั้นตอนที่ 4 เสร็จแล้ววนกลับไปทำซ้ำตามขั้นตอนที่ 2-4 เพื่อขยายเซตรายการความถี่ตัวถัดไป จะวนทำซ้ำไปเรื่อยๆ จนกว่าจะขยายเซตรายการความถี่หมดทุกเซตรายการ

4.3.3 ขั้นตอนวิธี UniSeq

ขั้นตอนวิธี UniSeq เป็นขั้นตอนวิธีที่ง่าย โดยทำการแปลงข้อมูลลำดับเหตุการณ์หลายมิติเป็นข้อมูลลำดับเหตุการณ์ จากนั้นใช้ขั้นตอนวิธีการทำเหมืองรูปแบบลำดับเหตุการณ์ในการขุดค้นรูปแบบลำดับเหตุการณ์หลายมิติ (ดังรูปที่ 4.3) โดยมีรายละเอียดดังต่อไปนี้



รูปที่ 4.3 ขั้นตอนการทำงานของขั้นตอนวิธี UniSeq

ขั้นตอนที่ 1 ทำการพิจารณาข้อมูลสารสนเทศหลายมิติในแต่ละรายการเปลี่ยนแปลงเป็นหนึ่งเหตุการณ์ จากนั้นนำไปรวมกับข้อมูลลำดับเหตุการณ์ แล้วพิจารณาข้อมูลทั้งหมดเป็นข้อมูลลำดับเหตุการณ์

ตัวอย่างที่ 4.12 รายการเปลี่ยนแปลงที่ 1 ในรูปที่ 4.4 ข้อมูลสารสนเทศหลายมิติ คือ มหาสารคาม, หญิง, ตรี, แต่งงาน และ อาจารย์ ข้อมูลเหล่านี้จะถูกพิจารณาเป็นหนึ่งเหตุการณ์และเป็นเหตุการณ์แรก แล้วตามด้วยข้อมูลลำดับเหตุการณ์ (A B) (A C) (D) ดังนั้นข้อมูลในรายการเปลี่ยนแปลงที่ 1 สามารถแปลงเป็นข้อมูลลำดับเหตุการณ์ คือ <(มหาสารคาม หญิง ตรี แต่งงาน อาจารย์) (A B) (A C) (D) >

รายการเปลี่ยนแปลง	ภูมิลำเนา	เพศ	ระดับการศึกษา	สถานภาพ	อาชีพ	ประวัติการจ่ายยา
1	มหาสารคาม	หญิง	ตรี	แต่งงาน	อาจารย์	< (A B) (A C) (D) >
2	มหาสารคาม	ชาย	ตรี	โสด	อาจารย์	< (A) (C D) (E) >
3	มหาสารคาม	ชาย	โท	แต่งงาน	อาจารย์	< (A) (C D) (F) >
4	ร้อยเอ็ด	ชาย	โท	โสด	เกษตรกร	< (C D) (E F) >

ลำดับเหตุการณ์
<(มหาสารคาม หญิง ตรี แต่งงาน อาจารย์) (A B) (A C) (D)>
<(มหาสารคาม ชาย ตรี โสด อาจารย์) (A) (C D) (E)>
<(มหาสารคาม ชาย โท แต่งงาน อาจารย์) (A) (C D) (F)>
<(ร้อยเอ็ด ชาย โท โสด เกษตรกร) (C D) (E F)>

รูปที่ 4.4 ตัวอย่างข้อมูลลำดับเหตุการณ์ที่ได้จากการแปลงข้อมูลลำดับเหตุการณ์หลายมิติ

ขั้นตอนที่ 2 ทำการขุดค้นรูปแบบลำดับเหตุการณ์หลายมิติด้วยขั้นตอนวิธีการทำเหมืองรูปแบบลำดับเหตุการณ์

ขั้นตอนวิธี Seq-Dim, Dim-Seq และ UniSeq ประยุกต์ใช้การทำเหมืองเซตรายการความถี่และการทำเหมืองรูปแบบลำดับเหตุการณ์ ดังนั้นทำให้เกิดปัญหาในลักษณะเดียวกัน คือ สร้างรูปแบบจำนวนมากเมื่อค่าสนับสนุนขั้นต่ำมีค่าน้อย

การทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติแบบปิดเป็นวิธีการหนึ่งที่พัฒนาขึ้นเพื่อลดจำนวนรูปแบบที่สร้างขึ้น โดยมีขั้นตอนการทำงานคล้ายกับการทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติ คือ ผสมผสานการทำเหมืองเซตรายการแบบปิดและการทำเหมืองรูปแบบลำดับเหตุการณ์แบบปิด ขั้นตอนวิธีการสำหรับการทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติแบบปิด เช่น CIS และ CSI เป็นต้น

4.4 ตัวอย่างการทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติด้วย SPMF

4.4.1 การเตรียมชุดข้อมูลลำดับเหตุการณ์หลายมิติ

ชุดข้อมูลลำดับเหตุการณ์หลายมิติแบ่งออกเป็น 2 ส่วน คือ ข้อมูลสารสนเทศหลายมิติ และข้อมูลลำดับเหตุการณ์ ส่วนของข้อมูลสารสนเทศหลายมิติทำการแปลงข้อมูลเหมือนชุดข้อมูลเซตรายการ ส่วนของลำดับเหตุการณ์ทำการแปลงข้อมูลเหมือนชุดข้อมูลลำดับเหตุการณ์ โดยเงื่อนไขการแปลงชุดข้อมูลลำดับเหตุการณ์หลายมิติเพื่อนำเข้าโปรแกรม SPMF เป็นดังนี้

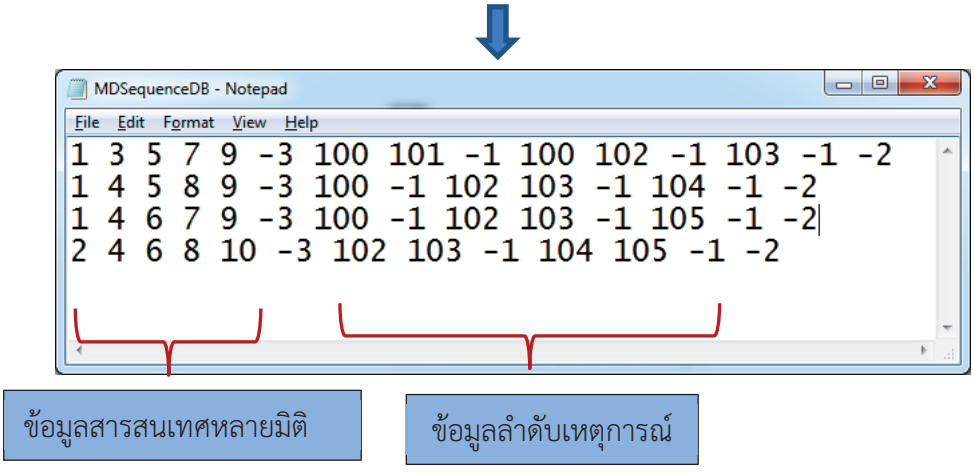
- ไฟล์นำเข้าอยู่ในรูปของไฟล์ข้อความ (Text file)
- รายการแต่ละรายการจะต้องแทนด้วยตัวเลขจำนวนเต็มบวก
- ลำดับเหตุการณ์หลายมิติใน 1 รายการเปลี่ยนแปลงแทนด้วยข้อมูล 1 บรรทัดในไฟล์
- ใช้ -3 เป็นตัวคั่นระหว่างข้อมูลสารสนเทศหลายมิติกับข้อมูลลำดับเหตุการณ์
- ใช้ -2 เป็นตัวระบุการสิ้นสุดของลำดับเหตุการณ์หลายมิติ
- ใช้ 1 เว้นวรรค สำหรับคั่นระหว่างตัวเลข
- ในเหตุการณ์เดียวกัน ตัวเลขจะต้องเรียงลำดับจากน้อยไปมาก
- ใช้ -1 คั่นระหว่างเหตุการณ์
- ห้ามมีบรรทัดที่มีข้อมูล (ลำดับเหตุการณ์) เหมือนกัน

รูปที่ 4.5 เป็นตัวอย่างการแทนค่าชุดข้อมูลลำดับเหตุการณ์หลายมิติ โดยแต่ละรายการถูกแทนเป็นตัวเลขตามตารางที่ 4.5 เช่น ในรายการเปลี่ยนแปลงที่ 1 ประกอบไปด้วย <(มหาสารคาม หญิง ตรี แต่งงาน อาจารย์) (A B) (A C) (D)> ในส่วนของ (มหาสารคาม หญิง ตรี แต่งงาน อาจารย์) คือ ข้อมูลสารสนเทศหลายมิติ สามารถแทนค่าเป็นตัวเลขจำนวนเต็มบวกได้เลย โดยตัวเลขแต่ละตัวให้คั่นด้วยเว้นวรรค ดังนั้นส่วนของ (มหาสารคาม หญิง ตรี แต่งงาน อาจารย์) แทนค่าเป็น 1 3 5 7 9

จากนั้นใช้ -3 เพื่อคั่นระหว่างข้อมูลสารสนเทศหลายมิติกับลำดับเหตุการณ์ ต่อมาแทนค่าลำดับหตุการณ์ (A B) (A C) (D) ด้วย 100 101 -1 100 102 -1 103 -1 (ใช้ -1 คั่นระหว่างเหตุการณ์) แล้วใช้ -2 บ่งบอกการสิ้นสุดของข้อมูลลำดับเหตุการณ์หลายมิติใน 1 รายการเปลี่ยนแปลง

เมื่อแทนข้อมูลในรายการเปลี่ยนแปลงที่ 1 แล้วจะได้ 1 3 5 7 9 -3 100 101 -1 100 102 -1 103 -1 -2 ทุกรายการเปลี่ยนแปลงจะถูกแทนค่าในลักษณะเดียวกัน จากข้อมูลทั้งหมดในตารางที่ 4.1 แปลงข้อมูลได้ดังรูปที่ 4.5 จากนั้นทำการบันทึกไฟล์ชื่อ MDsequenceDB.txt และจัดเก็บในโฟลเดอร์ dataset

รายการเปลี่ยนแปลง	ภูมิลำเนา	เพศ	ระดับการศึกษา	สถานภาพ	อาชีพ	ประวัติการจ่ายยา
1	มหาสารคาม	หญิง	ตรี	แต่งงาน	อาจารย์	< (A B) (A C) (D) >
2	มหาสารคาม	ชาย	ตรี	โสด	อาจารย์	< (A) (C D) (E) >
3	มหาสารคาม	ชาย	โท	แต่งงาน	อาจารย์	< (A) (C D) (F) >
4	ร้อยเอ็ด	ชาย	โท	โสด	เกษตรกร	< (C D) (E F) >



รูปที่ 4.5 ตัวอย่างไฟล์นำเข้า

ตารางที่ 4.5 การแทนค่าด้วยตัวเลขจำนวนเต็มในชุดข้อมูลลำดับเหตุการณ์หลายมิติ

รายการ	การแทนค่า
มหาสารคาม	1
ร้อยเอ็ด	2
หญิง	3
ชาย	4
ตรี	5
โท	6
แต่งงาน	7
โสด	8
อาจารย์	9
เกษตรกร	10
A	100
B	101
C	102
D	103
E	104
F	105

4.4.2 ตัวอย่างคำสั่งสำหรับการทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติ

ในส่วนของคำสั่งจะขอยกตัวอย่างการทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติด้วยขั้นตอนวิธี Seq-Dim ซึ่งทำการสร้างรูปแบบลำดับเหตุการณ์ก่อนด้วยใช้ขั้นตอนวิธี PrefixSpan จากนั้นทำการทำเหมืองเซตรายการความถี่บนข้อมูลสารสนเทศหลายมิติด้วยขั้นตอนวิธี Apriori คำสั่งแสดงได้ดังตัวอย่างคำสั่งที่ 4.1 โดยแต่ละคำสั่งสามารถอธิบายได้ดังนี้

ตัวอย่างคำสั่งที่ 4.1

```
1.  package mySpmfProject;
2.
3.  import java.io.IOException;
4.  import ca.pfv.spmf.algorithms.sequentialpatterns.fournier2008_seqdim.AlgoPrefixSpanMDSPM;
5.  import ca.pfv.spmf.algorithms.sequentialpatterns.fournier2008_seqdim.multidimensionalpatterns.
    AlgoDim;
6.  import ca.pfv.spmf.algorithms.sequentialpatterns.fournier2008_seqdim.
    multidimensionalsequentialpatterns.AlgoSeqDim;
7.  import ca.pfv.spmf.algorithms.sequentialpatterns.fournier2008_seqdim.
    multidimensionalsequentialpatterns.MDSequenceDatabase;
8.
9.  public class SeqDimTest {
10.     public static void main(String [] arg) throws IOException{
11.
12.             String input = ".//dataset/MDSequenceDB.txt";
13.             String output = ".//output.txt";
14.
15.             double minsupp = 0.75;
16.
17.             MDSequenceDatabase contextMDDatabase  = new MDSequenceDatabase();
18.             contextMDDatabase.loadFile(input);
19.
20.             AlgoDim algoDim = new AlgoDim(false, false);
21.
22.             AlgoSeqDim algoSeqDim = new AlgoSeqDim();
23.
24.             AlgoPrefixSpanMDSPM prefixSpan = new AlgoPrefixSpanMDSPM(minsupp);
25.             algoSeqDim.runAlgorithm(contextMDDatabase, prefixSpan, algoDim, false, output);
26.
27.             algoSeqDim.printStatistics(contextMDDatabase.size());
28.     }
29. }
```

บรรทัดที่ 4-7 เป็นการ import คลาสที่เกี่ยวข้อง ซึ่งมีดังต่อไปนี้

- AlgoPrefixSpanMDSPM เป็นคลาสสำหรับขั้นตอนวิธี PrefixSpan
- AlgoDim เป็นคลาสสำหรับขุดค้นเซตรายการความถี่
- AlgoSeqDim เป็นคลาสสำหรับขั้นตอนวิธี Seq-Dim
- MDSequenceDatabase เป็นคลาสสำหรับจัดเก็บข้อมูลลำดับเหตุการณ์หลายมิติ

บรรทัดที่ 12 เป็นการกำหนดตำแหน่งชุดข้อมูลนำเข้า ในตัวอย่างคือไฟล์ MDsequenceDB.txt ซึ่งอยู่ในโฟลเดอร์ dataset

บรรทัดที่ 13 เป็นการกำหนดตำแหน่งไฟล์สำหรับเก็บผลลัพธ์

บรรทัดที่ 15 เป็นการกำหนดค่าสนับสนุนขั้นต่ำ ซึ่งในตัวอย่างกำหนดค่าสนับสนุนขั้นต่ำแบบสัมพัทธ์ให้มีค่าเท่ากับ 0.75 หรือ 75% เมื่อเทียบกับค่าสนับสนุนขั้นต่ำแบบสัมบูรณ์จะมีค่าเท่ากับ 3 (สำหรับตัวอย่างชุดข้อมูล)

บรรทัดที่ 17-18 เป็นการโหลดชุดข้อมูลลำดับเหตุการณ์หลายมิติ

บรรทัดที่ 20 เป็นการสร้างอ๊อบเจ็กต์ของคลาส AlgoDim เพื่อขุดค้นเซตรายการความถี่โดยใช้ขั้นตอนวิธี Apriori ถ้าพารามิเตอร์ที่ 2 ถูกกำหนดเป็น true เป็นการกำหนดให้ขุดค้นเซตรายการความถี่แบบปิดโดยใช้ขั้นตอนวิธี CHARM

บรรทัดที่ 22 เป็นการสร้างอ๊อบเจ็กต์ของคลาส AlgoSeqDim

บรรทัดที่ 24 เป็นการสร้างอ๊อบเจ็กต์ของคลาส AlgoPrefixSpanMDSPM

บรรทัดที่ 25 เป็นการเรียกใช้เมทธอด runAlgorithm เพื่อสั่งให้ขั้นตอนวิธี Seq-Dim ทำการประมวลผล โดยมีพารามิเตอร์ 5 ตัว คือ

1) ชุดข้อมูล (contextMDDatabase)

2) ขั้นตอนวิธีสำหรับขุดค้นรูปแบบลำดับเหตุการณ์ (prefixSpan)

3) ขั้นตอนวิธีสำหรับขุดค้นเซตรายการความถี่ (algoDim)

4) ค่า true หรือ false ถ้ากำหนดค่า false เป็นการทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติ ถ้ากำหนดค่า true เป็นการทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติแบบปิด

5) ไฟล์ผลลัพธ์ (output)

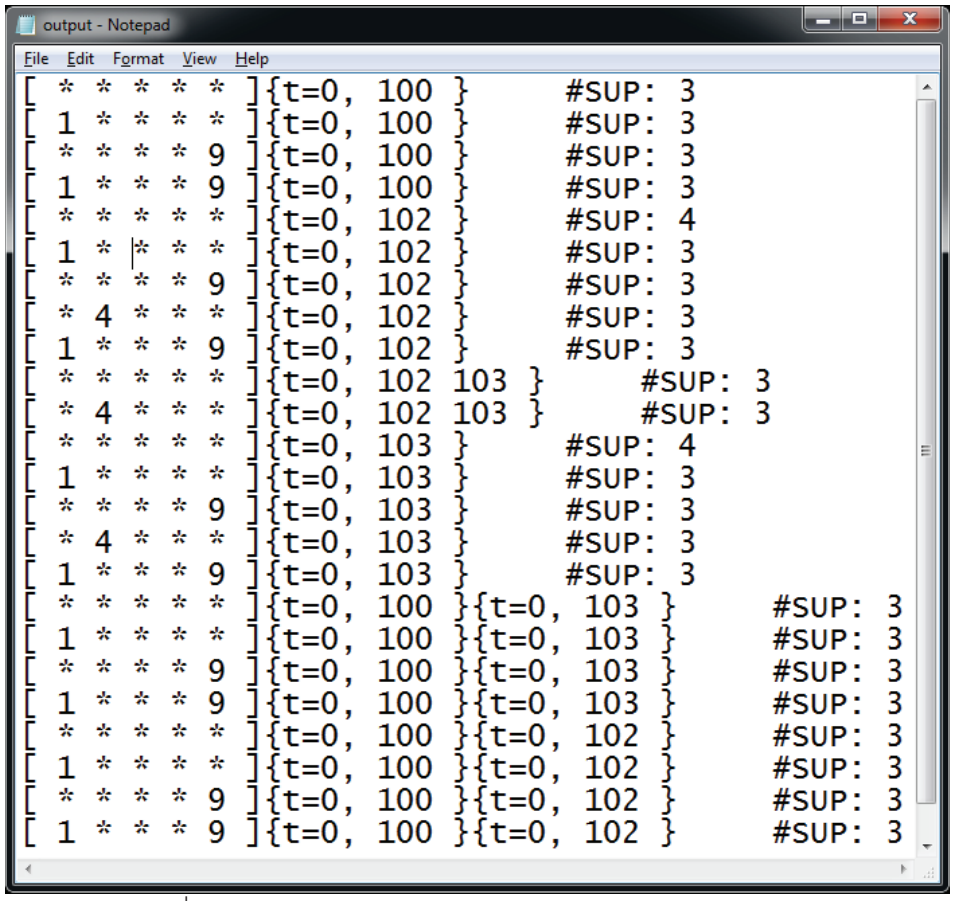
บรรทัดที่ 27 เป็นการแสดงค่าสถิติต่างๆ ที่ได้จากการประมวลผลดังรูปที่ 4.6 โดยแสดงรายละเอียดดังนี้

- เวลาในการประมวลผล (Total time)
- หน่วยความจำที่ใช้ (Max memory)
- จำนวนรูปแบบลำดับเหตุการณ์หลายมิติ (Frequent sequences count)

```
=============  SEQ-DIM - STATISTICS =============
 Total time ~ 9 ms
 max memory : 11.489959716796875
 Frequent sequences count : 24
=================================================
```

รูปที่ 4.6 แสดงค่าทางสถิติจากการประมวลผล SeqDimTest.java

ไฟล์ผลลัพธ์ประกอบไปด้วยรูปแบบลำดับเหตุการณ์หลายมิติและค่าสนับสนุนแบบสัมบูรณ์ดังรูปที่ 4.7 โดยใช้ [....] แสดงเซตรายการความถี่ที่ขุดค้นได้ และใช้ {t=0 ....} แสดงรูปแบบลำดับเหตุการณ์ที่ขุดค้นได้ เช่น [1 * * * *]{t=0, 100 } #SUP: 3 ในบรรทัดที่ 2 หมายถึง รูปแบบลำดับเหตุการณ์หลายมิติ (มหาสารคาม, *, *, *, *, <(A)>):3 และ [1 * * * 9]{t=0, 100 }{t=0, 102 } #SUP: 3 ในบรรทัดสุดท้าย หมายถึง รูปแบบลำดับเหตุการณ์หลายมิติ (มหาสารคาม, * , *, *, อาจารย์, <(A)(C)>):3 เป็นต้น จากรูปที่ 4.7 เมื่อแปลงผลลัพธ์กลับคืนจะได้รูปแบบลำดับเหตุการณ์หลายมิติดังแสดงในตารางที่ 4.6



รูปที่ 4.7 ไฟล์ผลลัพธ์จากการประมวลผล SeqDimTest.java

ตารางที่ 4.6 การแปลงผลลัพธ์รูปแบบลำดับเหตุการณ์หลายมิติ

ลำดับ	รูปแบบลำดับเหตุการณ์หลายมิติ
1	(*, *, *, *, *, <(A)>): 3
2	(มหาสารคาม, *, *, *, *): 3
3	(*, *, *, *, อาจารย์, <(A)>):3
4	(มหาสารคาม, *, *, *, อาจารย์, <(A)>):3
5	(*, *, *, *, *, <(C)>): 4
6	(มหาสารคาม, *, *, *, *, <(C)>): 3
7	(*, *, *, *, อาจารย์, <(C)>):3
8	(*, ชาย, *, *, *, <(C)>): 3
9	(มหาสารคาม, *, *, *, อาจารย์, <(C)>):3
10	(*, *, *, *, *, <(CD)>): 3
11	(*, ชาย, *, *, *, <(CD)>): 3
12	(*, *, *, *, *, <(D)>): 4
13	(มหาสารคาม, *, *, *, *, <(D)>): 3
14	(*, *, *, *, อาจารย์, <(D)>):3
15	(*, ชาย, *, *, *, <(D)>): 3
16	(มหาสารคาม, *, *, *, อาจารย์, <(D)>):3
17	(*, *, *, *, *, <(A)(D)>): 3
18	(มหาสารคาม, *, *, *, *, <(A)(D)>): 3
19	(*, *, *, *, อาจารย์, <(A)(D)>):3
20	(มหาสารคาม, *, *, *, อาจารย์, <(A)(D)>):3
21	(*, *, *, *, *, <(A)(C)>): 3
22	(มหาสารคาม, *, *, *, *, <(A)(C)>): 3
23	(*, *, *, *, อาจารย์, <(A)(C)>):3
24	(มหาสารคาม, *, *, *, อาจารย์, <(A)(C)>):3

รูปแบบลำดับเหตุการณ์หลายมิติที่ได้ สามารถแปลความหมายได้ดังตัวอย่างต่อไปนี้

รูปแบบลำดับเหตุการณ์หลายมิติที่ 2 จำนวนคนที่มีภูมิลำเนาอยู่ที่ มหาสารคาม คือ 3 คน

รูปแบบลำดับเหตุการณ์หลายมิติที่ 3 คนที่มีอาชีพเป็นอาจารย์ และได้รับยา A มีจำนวน 3 คน

รูปแบบลำดับเหตุการณ์หลายมิติที่ 4 คนที่มีภูมิลำเนาอยู่ที่ มหาสารคาม และมีอาชีพเป็นอาจารย์ และได้รับยา A มีจำนวน 3 คน

.....

รูปแบบลำดับเหตุการณ์หลายมิติที่ 24 คนที่มีภูมิลำเนาอยู่ที่ มหาสารคาม และมีอาชีพเป็นอาจารย์ ได้รับยา A แล้วตามด้วยยา C มีจำนวน 3 คน

บทสรุป

การทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติเป็นการผสมผสานการทำเหมืองเซตรายการความถี่และการทำเหมืองรูปแบบลำดับเหตุการณ์เข้าด้วยกัน การทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติค้นหารูปแบบที่น่าสนใจจากข้อมูลที่ประกอบไปด้วย 2 ส่วน คือ ข้อมูลสารสนเทศหลายมิติ ซึ่งเป็นข้อมูลที่ไม่ได้พิจารณาลำดับการเกิด และข้อมูลลำดับเหตุการณ์ ซึ่งเป็นข้อมูลที่ต้องพิจารณาลำดับการเกิด

การทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติสามารถใช้ขั้นตอนวิธี Seq-Dim, Dim-Seq และ UniSeq โดยขั้นตอน Seq-Dim เริ่มขุดค้นรูปแบบลำดับเหตุการณ์บนข้อมูลลำดับเหตุการณ์ก่อน จากนั้นทำการขุดค้นเซตรายการความถี่บนข้อมูลสารสนเทศหลายมิติย่อย ส่วนขั้นตอน Dim-Seq ทำการขุดค้นเซตรายการความถี่บนข้อมูลสารสนเทศหลายมิติก่อน จากนั้นขุดค้นรูปแบบลำดับเหตุการณ์บนข้อมูลลำดับเหตุการณ์ย่อย ส่วนขั้นตอนวิธี UniSeq ทำการแปลงข้อมูลลำดับเหตุการณ์หลายมิติเป็นข้อมูลลำดับเหตุการณ์ จากนั้นใช้ขั้นตอนวิธีการทำเหมืองรูปแบบลำดับเหตุการณ์ในการขุดค้นรูปแบบลำดับเหตุการณ์หลายมิติ

แบบฝึกหัดท้ายบท

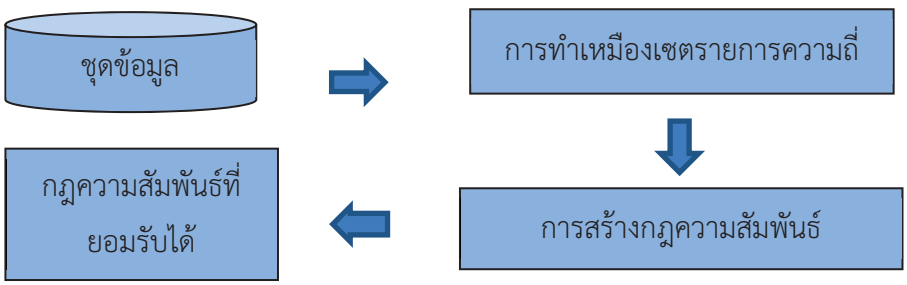
1. จงอธิบายความหมายของการทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติ
2. จงอธิบายลักษณะข้อมูลสำหรับการทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติ
3. จงอธิบายประโยชน์ของการทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติ ว่าแตกต่างอย่างไรกับการทำเหมืองรูปแบบลำดับเหตุการณ์และการทำเหมืองเซตรายการความถี่
4. จงยกตัวอย่างการประยุกต์ใช้การทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติ
5. จงอธิบายการทำงานของขั้นตอนวิธี UniSeq
6. จงอธิบายการทำงานของขั้นตอนวิธี Dim-Seq
7. จงอธิบายการทำงานของขั้นตอนวิธี Seq-Dim
8. ร้านค้าออนไลน์ทำการเก็บข้อมูลการซื้อสินค้าของลูกค้าดังตารางชุดข้อมูลต่อไปนี้ จงค้นหาฐานข้อมูลลำดับเหตุการณ์ย่อยของคนที่อยู่กรุงเทพฯ

รหัสลูกค้า	อาชีพ	ภูมิลำเนา	ช่วงอายุ	ลำดับการซื้อสินค้า
1001	นักศึกษา	กรุงเทพฯ	วัยรุ่น	< (a c) (b c) >
1002	นักธุรกิจ	กรุงเทพฯ	วัยกลางคน	< (c d) (e f) >
1003	อาจารย์	อยุธยา	วัยชรา	< (c) (c d f) >
1004	นักธุรกิจ	กรุงเทพฯ	วัยกลางคน	< (a c d) (d e f) >

9. จากตารางข้อมูลการซื้อสินค้าในข้อ 8 จงแปลงเป็นชุดข้อมูลนำเข้า เพื่อใช้ใน SPMF
10. จงเขียนโปรแกรมโดยใช้ SPMF เพื่อค้นหารูปแบบลำดับเหตุการณ์หลายมิติจากชุดข้อมูลที่แปลงเรียบร้อยแล้วในข้อ 9 โดยกำหนดให้ค่าสนับสนุนขั้นต่ำแบบสัมบูรณ์มีค่าเท่ากับ 2

บทที่ 5
การทำเหมืองกฎความสัมพันธ์
(Association Rule Mining)

การทำเหมืองกฎความสัมพันธ์เป็นการค้นหาความสัมพันธ์ของข้อมูลที่ปรากฏบ่อยจากข้อมูลขนาดใหญ่ ผลลัพธ์ที่ได้จะอยู่ในรูปของกฎที่แสดงเหตุที่นำไปสู่ผล โดยรูปแบบของกฎแสดงอยู่ในรูป $X \rightarrow Y$ โดยที่ X คือ เหตุ (Antecedent) และ Y เป็นผลที่ตามมา (Consequence) โดยทั่วไปการสร้างกฎความสัมพันธ์จะสร้างจากเซตรายการความถี่ ซึ่งประกอบไปด้วย 2 ขั้นตอนหลัก (ดังรูปที่ 5.1) ขั้นตอนแรก คือ การทำเหมืองเซตรายการความถี่ทั้งหมดเพื่อหากลุ่มข้อมูลที่เกิดขึ้นบ่อย โดยเซตรายการความถี่เป็นเซตรายการที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ ขั้นตอนที่สอง คือ การสร้างกฎความสัมพันธ์ โดยการนำเซตรายการความถี่ที่ได้จากขั้นตอนแรกมาสร้างกฎความสัมพันธ์ กฎความสัมพันธ์ที่ยอมรับได้เป็นกฎที่มีค่าความเชื่อมั่นมากกว่าหรือเท่ากับค่าความเชื่อมั่นขั้นต่ำ



รูปที่ 5.1 กระบวนการค้นหากฎความสัมพันธ์

กฎความสัมพันธ์ที่สร้างจากเซตรายการความถี่แสดงให้เห็นถึงความสัมพันธ์ที่เกิดร่วมกัน แต่ไม่ได้แสดงให้เห็นถึงความสัมพันธ์ที่เกิดขึ้นตามลำดับเหตุการณ์ กฎความสัมพันธ์สามารถสร้างได้จากรูปแบบลำดับเหตุการณ์ ซึ่งเป็นกฎความสัมพันธ์ที่แสดงให้เห็นถึงความสัมพันธ์ของเหตุการณ์ที่เกิดก่อนและเหตุการณ์ที่เกิดตามหลัง กฎที่สร้างจากรูปแบบลำดับเหตุการณ์ เรียกว่า กฎความสัมพันธ์เชิงลำดับ (Sequential association rule) การสร้างกฎความสัมพันธ์เชิงลำดับในปัจจุบันแบ่งออกเป็น 2 กลุ่มหลักๆ คือ

กลุ่มที่ 1 ทำขุดค้นรูปแบบลำดับเหตุการณ์ก่อน แล้วนำรูปแบบลำดับเหตุการณ์ไปสร้างกฎความสัมพันธ์เชิงลำดับ โดยลำดับเหตุการณ์ที่อยู่ฝั่งซ้ายและฝั่งขวาของกฎสร้างมาจากรูปแบบลำดับเหตุการณ์ เช่น ถ้ารูปแบบลำดับเหตุการณ์ คือ <(a) (f) (e)> จะสามารถสร้างกฎความสัมพันธ์เชิงลำดับได้ (a) $\rightarrow$ (f) (e) และ (a) (f) $\rightarrow$ (e)

กลุ่มที่ 2 สร้างกฎความสัมพันธ์เชิงลำดับจากเซตรายการ X และ Y ในรูปแบบ $X \longrightarrow Y$ โดยที่เซตรายการ X จะต้องเกิดก่อนเซตรายการ Y (รายการที่อยู่ใน X และ Y ไม่ได้พิจารณาลำดับ) กฎที่สร้างขึ้นในกลุ่มนี้เป็นกฎที่สามารถทำนายข้อมูลได้ถูกต้องกว่ากฎที่สร้างขึ้นจากกลุ่มแรก และกฎดังกล่าวสามารถนำไปประยุกต์ใช้ในด้านต่างๆ ได้มากกว่า เช่น การเรียนทางอิเล็กทรอนิกส์ (E-Learning) การควบคุมการผลิต การวิเคราะห์ลำดับเหตุการณ์สำหรับการเตือนภัย ระบบแนะนำในร้านอาหาร เป็นต้น ดังนั้นในหนังสือเล่มนี้จะกล่าวถึงกฎความสัมพันธ์เชิงลำดับในกลุ่มที่ 2

5.1 นิยามที่เกี่ยวข้อง

กำหนดให้ $I = \{i_1, i_2, \ldots, i_m\}$ คือ เซตของรายการทั้งหมดในชุดข้อมูล และ $T = \{t_1, t_2, \ldots t_m\}$ คือ เซตของรายการเปลี่ยนแปลงทั้งหมดในชุดข้อมูล และ X และ Y คือ เซตรายการ โดยที่ $X, Y \subseteq I$

นิยามที่ 5.1 กฎความสัมพันธ์ $X \longrightarrow Y$ คือ กฎที่แสดงความพันธ์ระหว่างสองเซตรายการที่เกิดร่วมกัน โดยที่ $X, Y \subseteq I$ และ $X \cap Y = \varnothing$

ตัวอย่างที่ 5.1 กฎ D⟶F แสดงให้เห็นว่าเมื่อเกิด D จะมี F เกิดร่วมด้วย

นิยามที่ 5.2 ค่าสนับสนุนของกฎความสัมพันธ์ $X \longrightarrow Y$ คือ จำนวนรายการเปลี่ยนแปลงที่เกิด X และ Y ร่วมกัน แทนด้วย $supp(X \cup Y)$

ตัวอย่างที่ 5.2 จากตารางที่ 5.1 ค่าสนับสนุนของกฎความสัมพันธ์ D⟶F คือ 2 เนื่องจากเซตรายการ D และ F ปรากฏใน 2 รายการเปลี่ยนแปลง คือ รายการเปลี่ยนแปลงที่ 3 และ 4

ตารางที่ 5.1 ตัวอย่างชุดข้อมูลเซตรายการ

รายการเปลี่ยนแปลง	เซตรายการ
1	(A C F)
2	(A B C)
3	(A C D F)
4	(B D E F)

นิยามที่ 5.3 ค่าความเชื่อมั่นของกฎความสัมพันธ์ $X \longrightarrow Y$ คือ ค่าที่แสดงให้เห็นถึงโอกาสการเกิด X แล้วเกิด Y ร่วมกัน ค่าความเชื่อมั่นสามารถคำนวณได้จากสมการ (5.1)

$$conf(r) = \frac{supp(X \cup Y)}{supp(X)} \times 100 \tag{5.1}$$

ตัวอย่างที่ 5.3 กฎความสัมพันธ์ D⟶F ที่สร้างจากเซตรายการความถี่ (DF) สามารถคำนวณค่าความเชื่อมั่นได้จาก

$$supp(D \cup F)/supp(D) \times 100 = 2/2 \times 100 = 100\%$$

ซึ่งแสดงให้เห็นว่าเมื่อเกิด D จะมีโอกาสเกิด F ร่วมด้วยถึง 100%

ส่วนค่าความเชื่อมั่นของกฎความสัมพันธ์ F⟶D สามารถคำนวณได้จาก

$$supp(F \cup D)/supp(F) \times 100 = 2/3 \times 100 = 67\%$$

ซึ่งแสดงให้เห็นว่าเมื่อเกิด F มีโอกาสเกิด D ร่วมด้วยแค่ 67%

นิยามที่ 5.4 กฎความสัมพันธ์ r ถือว่าเป็นกฎที่สามารถยอมรับได้ เมื่อค่าสนับสนุนของ r มีค่ามากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ และความเชื่อมั่นของ r มีค่ามากกว่าหรือเท่ากับค่าความเชื่อมั่นขั้นต่ำ

ตัวอย่างที่ 5.4 ถ้ากำหนดค่าสนับสนุนขั้นต่ำเท่ากับ 2 และค่าความเชื่อมั่นขั้นต่ำเท่ากับ 80%

กฎความสัมพันธ์ D⟶F ถือว่าเป็นกฎที่ยอมรับได้ เนื่องจากค่าสนับสนุนของ D⟶F คือ 2 ซึ่งเท่ากับค่าสนับสนุนขั้นต่ำและค่าความเชื่อมั่นของกฎความสัมพันธ์ D⟶F เท่ากับ 100% ซึ่งมีค่ามากกว่าค่าความเชื่อมั่นขั้นต่ำ

ในขณะที่กฎความสัมพันธ์ F⟶D จะถูกตัดทิ้งเนื่องมาจากค่าความเชื่อมั่นของกฎ F⟶D เท่ากับ 67% ซึ่งมีค่าน้อยกว่าค่าความเชื่อมั่นขั้นต่ำ

นิยามที่ 5.5 กฎความสัมพันธ์เชิงลำดับ $X \longrightarrow Y$ คือ กฎที่แสดงความพันธ์ระหว่างสองเซตรายการที่เกิดตามลำดับ หมายถึง X เกิดก่อน Y โดยที่ $X, Y \subseteq I$ และ $X \cap Y = \varnothing$

ตัวอย่างที่ 5.5 กฎความสัมพันธ์เชิงลำดับ (a b c)⟶(e) แสดงให้เห็นว่า เมื่อเกิดเซตรายการ (a b c) จะเกิดเซตรายการ (e) ตามมา โดยรายการที่อยู่ในเซตรายการ (a b c) เกิดรายการใดก่อนหลังก็ได้ แต่ว่าต้องเกิดก่อนรายการ e

นิยามที่ 5.6 ค่าสนับสนุนของกฎความสัมพันธ์เชิงลำดับ $X \longrightarrow Y$ คือ จำนวนรายการเปลี่ยนแปลงที่เกิด X แล้วตามด้วย Y แทนด้วย $supp(X \longrightarrow Y)$

ตัวอย่างที่ 5.6 จากชุดข้อมูลลำดับเหตุการณ์ในตารางที่ 5.2 ค่าสนับสนุนของกฎความสัมพันธ์เชิงลำดับ (a b c)⟶(e) สามารถพิจารณาจากจำนวนรายการเปลี่ยนแปลงที่ปรากฏเซตรายการ (a b c) แล้วตามด้วยเซตรายการ (e) ซึ่งก็คือ รายการเปลี่ยนแปลงที่ 1 และ 2 ดังนั้น $supp((a\ b\ c) \rightarrow (e)) = 2$

ตารางที่ 5.2 ตัวอย่างชุดข้อมูลลำดับเหตุการณ์

รายการเปลี่ยนแปลง	ลำดับเหตุการณ์
1	<(a b) (c) (f) (g) (e)>
2	<(a d) (c) (b) (e f)>
3	<(a) (b) (f) (e)>
4	<(b) (f g)>

นิยามที่ 5.7 ค่าความเชื่อมั่นของกฎความสัมพันธ์เชิงลำดับ $X \longrightarrow Y$ คือ ค่าที่แสดงให้เห็นถึงโอกาสการเกิด X แล้วตามด้วย Y ค่าความเชื่อมั่นสามารถคำนวณได้จากสมการ (5.2)

$$conf(r) = \frac{supp(X \rightarrow Y)}{supp(X)} \times 100 \qquad (5.2)$$

ตัวอย่างที่ 5.7 กฎความสัมพันธ์เชิงลำดับ (a b c)⟶(e) สามารถคำนวณได้จาก

$$supp((a\ b\ c) \rightarrow (e))/supp(e) \times 100 = 2/3 \times 100 = 67\%$$

ซึ่งแสดงให้เห็นว่าเมื่อเกิดเซตรายการ (a b c) แล้วจะมีโอกาสเกิดเซตรายการ (e) ตามมา 67%

นิยามที่ 5.8 กฎความสัมพันธ์เชิงลำดับ r ถือว่าเป็นกฎที่สามารถยอมรับได้ เมื่อค่าสนับสนุนของ r มีค่ามากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ และความเชื่อมั่นของ r มีค่ามากกว่าหรือเท่ากับค่าความเชื่อมั่นขั้นต่ำ

ตัวอย่างที่ 5.8 ถ้ากำหนดค่าสนับสนุนขั้นต่ำเท่ากับ 2 ความเชื่อมั่นขั้นต่ำเท่ากับ 60% กฎความสัมพันธ์เชิงลำดับ (a b c)⟶(e) มีค่าสนับสนุนเท่ากับ 2 ซึ่งเท่ากับค่าสนับสนุนขั้นต่ำ และมีค่าความเชื่อมั่น

เท่ากับ 67% ซึ่งมากกว่าค่าความเชื่อมั่นขั้นต่ำ ดังนั้น (a b c)⟶(e) เป็นกฎความสัมพันธ์เชิงลำดับที่ยอมรับได้

5.2 การทำเหมืองความสัมพันธ์จากเซตรายการความถี่

กฎความสัมพันธ์ที่สร้างจากเซตรายการความถี่ เหมาะกับการนำไปประยุกต์ใช้กับข้อมูลที่ไม่ได้สนใจเรื่องลำดับการเกิด เช่น พิจารณาว่ารายการไหนถูกซื้อไปพร้อมกัน เป็นต้น ขั้นตอนการทำเหมืองกฎความสัมพันธ์แบ่งออกเป็น 2 ขั้นตอนหลัก คือ การขุดค้นเซตรายการความถี่ และ การสร้างกฎความสัมพันธ์ ซึ่งมีรายละเอียดดังต่อไปนี้

ขั้นตอนที่ 1 การขุดค้นเซตรายการความถี่ โดยใช้การทำเหมืองเซตรายการความถี่ดังที่กล่าวไว้ในบทที่ 2 ซึ่งสามารถใช้ขั้นตอนวิธีใดก็ได้สำหรับการทำเหมืองเซตรายการความถี่ เช่น Apriori หรือ FP-Growth เป็นต้น เนื่องจากทุกขั้นตอนวิธีสร้างเซตรายการความถี่เหมือนกัน แตกต่างกันแค่วิธีการและโครงสร้างที่นำมาใช้

ขั้นตอนที่ 2 การสร้างกฎความสัมพันธ์จากเซตรายการความถี่ จะสร้างจากเซตรายการความถี่ที่มีความยาว 2 รายการขึ้นไป โดยสามารถพิจารณาจากเซตย่อยของเซตรายการความถี่

สมมติเซตย่อยของเซตรายการความถี่ l เท่ากับ $\{S_1, S_2, S_3, ..., S_m\}$ จะสามารถสร้างกฎได้ $S_i \longrightarrow (l - S_i)$ โดยที่ i = 1 ถึง m ดังนั้นสามารถสร้างจำนวนกฎความสัมพันธ์จากเซตรายการความถี่ที่มีความยาว n ได้ทั้งหมด $2^n - 2$ กฎ หรือเท่ากับจำนวนเซตย่อยของเซตรายการความถี่ เช่น เซตรายการความถี่ l = (ACF) มีความยาวเท่ากับ 3 และประกอบด้วยเซตรายการย่อย คือ {A, C, F, AC, AF, CF} สามารถสร้างกฎความสัมพันธ์ได้ทั้งหมด $2^3 - 2 = 6$ กฎ คือ r_1: A⟶CF, r_2: C⟶AF, r_3: F⟶AC, r_4: AC⟶F, r_5: AF⟶C และ r_6: CF⟶A เป็นต้น เมื่อได้กฎแล้ว ทำการคำนวณหาค่าความเชื่อมั่นของแต่ละกฎ ซึ่งกฎที่ยอมรับได้ คือ กฎที่ค่าความเชื่อมั่นมากกว่าหรือเท่ากับค่าความเชื่อมั่นขั้นต่ำ

ตัวอย่างที่ 5.9 ถ้ากำหนดให้เซตรายการความถี่ที่ได้มีทั้งหมด 10 เซตรายการ คือ (D):2, (FD):2, (B):3, (F):3, (AF):2, (CF):2, (ACF):2, (C):3, (AC):3 และ (A):3 เซตรายการความถี่ที่มีความยาวตั้งแต่ 2 รายการขึ้นไปมีจำนวน 5 เซตรายการ คือ (FD):2, (AF):2, (CF):2, (ACF):2, (AC):3 ทั้ง 5 เซตรายการถูกนำไปสร้างกฎความสัมพันธ์ได้ดังแสดงในตารางที่ 5.3

ตารางที่ 5.3 ตัวอย่างกฎความสัมพันธ์

เซตรายการความถี่	กฎที่ได้	ค่าความเชื่อมั่น (%)	
FD	F⟶D	*supp*(FD)/*supp*(F)x100 = 2/3 x 100 = 67%	✘
	D⟶F	*supp*(FD)/*supp*(D)x100 = 2/2 x 100 = 100%	
AF	A⟶F	*supp*(AF)/*supp*(A)x100 = 2/3 x 100 = 67%	✘
	F⟶A	*supp*(AF)/*supp*(F)x100 = 2/3 x 100 = 67%	✘
CF	C⟶F	*supp*(CF)/*supp*(C)x100 = 2/3 x 100 = 67%	✘
	F⟶C	*supp*(CF)/*supp*(F)x100 = 2/3 x 100 = 67%	✘
ACF	A⟶CF	*supp*(ACF)/*supp*(A)x100 = 2/3 x 100 = 67%	✘
	C⟶AF	*supp*(ACF)/*supp*(C)x100 = 2/3 x 100 = 67%	✘
	F⟶AC	*supp*(ACF)/*supp*(F)x100 = 2/3 x 100 = 67%	✘
	AC⟶F	*supp*(ACF)/*supp*(AC)x100 = 2/3 x 100 = 67%	✘
	AF⟶C	*supp*(ACF)/*supp*(AF)x100 = 2/2 x 100 = 100%	
	CF⟶A	*supp*(ACF)/*supp*(CF)x100 = 2/2 x 100 = 100%	
AC	A⟶C	*supp*(AC)/*supp*(A)x100 = 3/3 x 100 = 100%	
	C⟶A	*supp*(AC)/*supp*(C)x100 = 3/3 x 100 = 100%	

จากตารางที่ 5.3 ถ้ากำหนดค่าความเชื่อมั่นขั้นต่ำเท่ากับ 80% กฎที่ถูกตัดทิ้งมีทั้งหมด 9 กฎ (กฎที่ระบายสีเทา) จะเห็นได้ว่ากฎความสัมพันธ์ที่ผ่านค่าความเชื่อมั่นขั้นต่ำมีจำนวน 5 กฎ คือ D⟶F, AF⟶C, CF⟶A, A⟶C และ C⟶A ซึ่งกฎทั้ง 5 กฎนี้จะถูกนำไปใช้ประโยชน์ต่อไป

จากขั้นตอนการสร้างกฎความสัมพันธ์ที่กล่าวมา จะเห็นได้ว่ากฎความสัมพันธ์ทั้งหมดจะถูกสร้างขึ้นมา แล้วค่อยตรวจสอบค่าความเชื่อมั่นของกฎว่าผ่านค่าความเชื่อมั่นขั้นต่ำหรือไม่ ทำให้เสียเวลาในการค้นหากฎที่ผ่านค่าความเชื่อมั่นขั้นต่ำ จึงได้มีการนำเสนอขั้นตอนวิธี Faster เพื่อให้สามารถสร้างกฎความสัมพันธ์ที่ผ่านค่าความเชื่อมั่นขั้นต่ำได้เร็วขึ้น

แนวคิดของขั้นตอนวิธี Faster คือ ถ้ามีกฎ $(I - S) \longrightarrow S$ ผ่านค่าความเชื่อมั่นขั้นต่ำแล้ว ทุกกฎ $(I - V) \longrightarrow V$ จะผ่านค่าความเชื่อมั่นขั้นต่ำด้วย โดยที่ $V \subset S$ ดังนั้นถ้ามีกฎ $(I - V) \longrightarrow V$ ไม่ผ่านค่าความเชื่อมั่นขั้นต่ำแล้ว กฎ $(I - S) \longrightarrow S$ ก็จะไม่ผ่านค่าความเชื่อมั่นขั้นต่ำด้วย เนื่องจากค่าสนับสนุนของ $(I - S)$ มีค่าได้มากกว่าหรือเท่ากับค่าสนับสนุนของ $(I – V)$ เท่านั้น ซึ่งทำให้ค่าความเชื่อมั่นของกฎ $(I - S) \longrightarrow S$ มีค่าเท่ากับหรือน้อยกว่าค่าความเชื่อมั่นของกฎ $(I - V) \longrightarrow V$

ดังนั้นไม่จำเป็นต้องสร้างกฎ $(I - S) \longrightarrow S$ ขึ้นมา ถ้าปรากฏว่ามีกฎ $(I - V) \longrightarrow V$ ไม่ผ่านค่าความเชื่อมั่นขั้นต่ำโดยที่ $V \subset S$ ซึ่งจะทำให้เวลาในการค้นหากฎที่ผ่านค่าความเชื่อมั่นเร็วขึ้น เช่น จากเซตรายการความถี่ ACF ในตารางที่ 5.3 มีกฎ AC⟶F ไม่ผ่านค่าความเชื่อมั่นขั้นต่ำ กฎ A⟶CF และ C⟶AF ก็จะไม่ผ่านค่าความเชื่อมั่นด้วย เนื่องจาก F ⊂ CF และ F ⊂ AF ดังนั้นไม่จำเป็นต้องสร้างกฎ A⟶CF และ C⟶AF เป็นต้น

5.3 การทำเหมืองความสัมพันธ์เชิงลำดับ

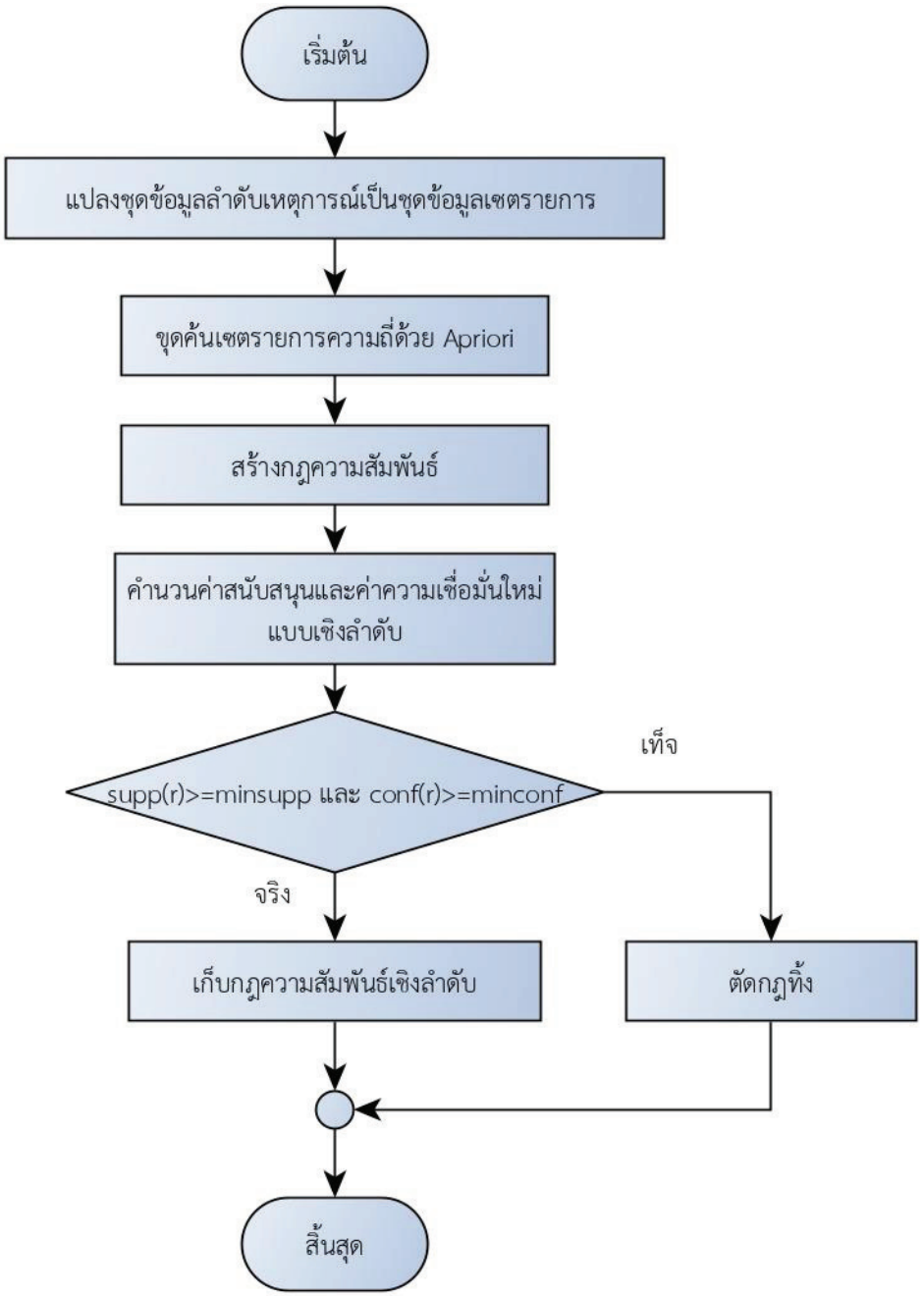
การทำเหมืองความสัมพันธ์เชิงลำดับเป็นการค้นหาความสัมพันธ์ของข้อมูลจากข้อมูลขนาดใหญ่ที่ปรากฏขึ้นบ่อย โดยความสัมพันธ์ดังกล่าวจะต้องพิจารณาเรื่องลำดับการเกิดของข้อมูลเป็นหลัก ซึ่งแตกต่างจากกฎความสัมพันธ์ที่สร้างจากเซตรายการความถี่ที่พิจารณาแค่การเกิดร่วมกัน กฎความสัมพันธ์เชิงลำดับสามารถนำไปประยุกต์ใช้ในหลายด้าน เช่น การหาความสัมพันธ์ของการเกิดปรากฏการณ์ธรรมชาติ การหาความสัมพันธ์ของการกดลิงก์หน้าเว็บไซต์ การทำนายตำแหน่งของโปรตีน การวิเคราะห์ทางการตลาด การตรวจสอบการโจมตีบนข่ายเน็ตเวิร์ค เป็นต้น ซึ่งทุกงานที่กล่าวจำเป็นต้องพิจารณาลำดับการเกิดของข้อมูล

ขั้นตอนวิธีการสร้างกฎความสัมพันธ์เชิงลำดับมีหลายวิธี บางขั้นตอนวิธีทำการหารูปแบบลำดับเหตุการณ์ก่อน จากนั้นนำรูปแบบลำดับเหตุการณ์ที่ได้มาใช้ในการสร้างกฎความสัมพันธ์เชิงลำดับ บางขั้นตอนวิธีสร้างกฎความสัมพันธ์เชิงลำดับโดยไม่ต้องหารูปแบบลำดับเหตุการณ์ความถี่ก่อน เพื่อลดขั้นตอนการทำเหมืองรูปแบบลำดับเหตุการณ์ เช่น RuleGrowth, CMRules และ ERMiner เป็นต้น

ในบทนี้จะอธิบายถึงขั้นตอนวิธี CMRules เป็นขั้นตอนหนึ่งที่มีประสิทธิภาพและไม่ซับซ้อน โดยทำการสร้างกฎความสัมพันธ์ก่อน แล้วค่อยพิจารณากฎความสัมพันธ์ว่าเป็นกฎความสัมพันธ์เชิงลำดับหรือไม่

5.3.1 การสร้างกฎความสัมพันธ์เชิงลำดับด้วยขั้นตอนวิธี CMRules

ขั้นตอนการทำงานของขั้นตอนวิธี CMRules แสดงได้ดังรูปที่ 5.2 สามารถอธิบายโดยละเอียดพร้อมกับยกตัวอย่างได้ดังนี้



รูปที่ 5.2 ขั้นตอนวิธี CMRules

ขั้นตอนที่ 1 แปลงชุดข้อมูลลำดับเหตุการณ์เป็นชุดข้อมูลเซตรายการ

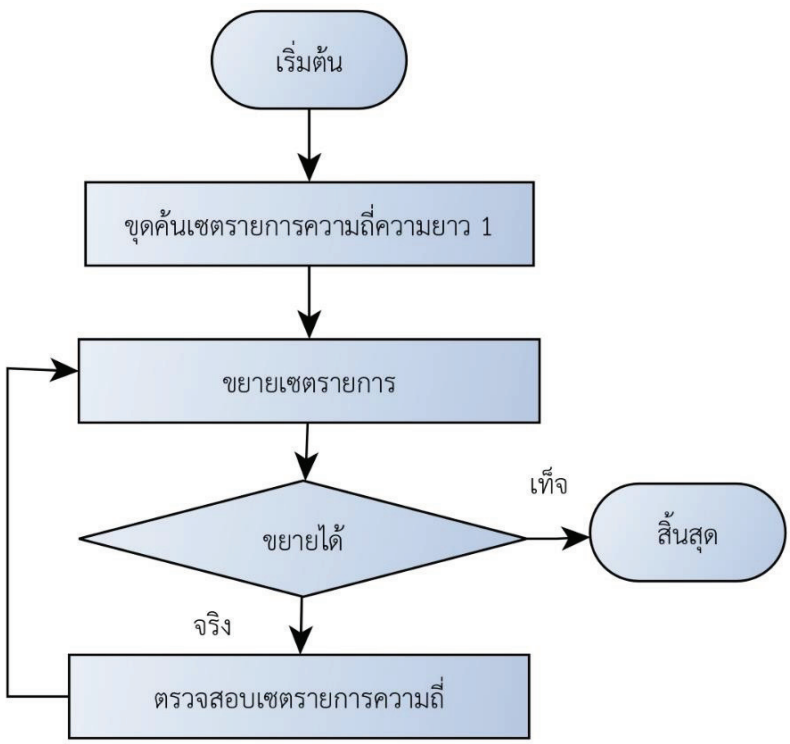
ตัวอย่างที่ 5.10 สามารถแปลงชุดข้อมูลลำดับเหตุการณ์ในตารางที่ 5.2 เป็นชุดข้อมูลเซตรายการที่ไม่พิจารณาลำดับการเกิดได้ดังตารางที่ 5.4

ตารางที่ 5.4 ข้อมูลเซตรายการ

รายการเปลี่ยนแปลง	เซตรายการ
1	a b c e f g
2	a b c d e f
3	a b e f
4	b f g

ขั้นตอนที่ 2 ทำการขุดค้นเซตรายการความถี่ด้วยขั้นตอนวิธี Apriori บนชุดข้อมูลที่แปลงแล้ว โดยขั้นตอนการทำงานของขั้นตอนวิธี Apriori แสดงได้ดังรูปที่ 5.3 และมีรายละเอียดดังต่อไปนี้

ขั้นตอนที่ 2.1 ค้นหาเซตรายการความถี่ความยาว 1 หรือรายการที่มีค่าสนับสนุนมากกว่าค่าสนับสนุนขั้นต่ำ



รูปที่ 5.3 ขั้นตอนวิธี Apriori

ตัวอย่างที่ 5.11 กำหนดค่าสนับสนุนขั้นต่ำเท่ากับ 2 จากตารางที่ 5.4 เซตรายการความยาว 1 มีค่าสนับสนุนดังตารางที่ 5.5 โดยเซตรายการที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำประกอบไปด้วย a, b, c, e, f และ g ส่วนเซตรายการที่มีค่าสนับสนุนน้อยกว่าค่าสนับสนุนขั้นต่ำจะถูกตัดออกไป (เซตรายการที่ระบายพื้นสีเทา) และจะไม่นำมาพิจารณาเพื่อขยายเซตรายการต่อไป เพราะไม่สามารถให้ค่าสนับสนุนที่มากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำได้ (เนื่องจากการขยายเซตรายการจะทำให้ค่าสนับสนุนน้อยกว่าหรือเท่าเดิม)

ดังนั้นเซตรายการความถี่ความยาว 1 ทั้งหมด คือ Fl_1 = {a, b, c, e, f, g}

ตารางที่ 5.5 เซตรายการความยาว 1 และค่าสนับสนุน

เซตรายการ	ค่าสนับสนุน
a	3
b	4
c	2
d	1
e	3
f	4
g	2

ขั้นตอนที่ 2.2 ทำการขยายเซตรายการ จากเซตรายการความถี่ที่มีความยาว *k*-1 เป็นเซตรายการความยาว *k*

ขั้นตอนที่ 2.3 ถ้าขยายเซตรายการไม่ได้ให้หยุดการค้นหา แต่ถ้าขยายได้ ให้ทำการตรวจสอบว่า เซตรายการที่ขยายแล้วเป็นเซตรายการความถี่หรือไม่ ในกรณีที่เซตรายการที่ต้องการตรวจสอบมีความยาวตั้งแต่ 3 ขึ้นไป จะพิจารณาว่าเซตรายการดังกล่าวละเมิดคุณสมบัติของ Apriori หรือไม่ ก่อนจะคำนวณค่าสนับสนุน ตัดเซตรายการที่ละเมิดคุณสมบัติ Apriori ออก เพื่อช่วยลดเวลาในการประมวลผลและการอ่านชุดข้อมูลเพื่อคำนวณหาค่าสนับสนุน

คุณสมบัติ Apriori คือ ถ้าเซตรายการ *X* คือ เซตรายการความถี่แล้ว ทุกเซตรายการย่อยของ *X* จะเป็นเซตรายการความถี่ด้วย ดังนั้นถ้าเซตรายการ *X* มีเซตรายการย่อยที่ไม่ใช่เซตรายการความถี่แสดงว่าละเมิดคุณสมบัติ Apriori โดยเซตรายการ *X* จะถือว่าไม่ใช่เซตรายการความถี่ (ไม่จำเป็นต้องคำนวณค่าสนับสนุนและตรวจสอบว่าผ่านค่าสนับสนุนขั้นต่ำหรือไม่) ส่วนเซตรายการที่ไม่ละเมิด

คุณสมบัติของ Apriori จะทำคำนวณหาค่าสนับสนุนก่อน แล้วตรวจสอบว่าค่าสนับสนุนของเซตรายการมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำหรือไม่

ตัวอย่างที่ 5.12 จากตารางที่ 5.5 นำเซตรายการความถี่แต่ละตัวมาจับคู่กัน เช่น เซตรายการความถี่ a สามารถจับคู่ได้กับเซตรายการความถี่ b, c, e, f, และ g ทำให้ได้เซตรายการที่มีความยาว 2 จำนวน 5 เซตรายการ คือ (a b), (a c), (a e), (a f) และ (a g) ส่วนเซตรายการความถี่ b สามารถจับคู่กับเซตรายการความถี่ c, e, f, และ g ทำให้ได้เซตรายการที่มีความยาว 2 จำนวน 4 เซตรายการ คือ (b c), (b e), (b f) และ (b g) ส่วนเซตรายการความถี่ c สามารถจับคู่กับเซตรายการความถี่ e, f, และ g ทำให้ได้เซตรายการที่มีความยาว 2 จำนวน 3 เซตรายการ คือ (c e), (c f) และ (c g) เป็นต้น จับคู่ลักษณะนี้ ไปเรื่อยจะทำให้ได้เซตรายการความยาว 2 ดังตารางที่ 5.6

ตารางที่ 5.6 เซตรายการความยาว 2 และค่าสนับสนุน

เซตรายการ	ค่าสนับสนุน	
(a b)	3	
(a c)	2	
(a e)	3	
(a f)	4	
(a g)	1	✘
(b c)	2	
(b e)	3	
(b f)	4	
(b g)	2	
(c e)	2	
(c f)	2	
(c g)	1	✘
(e f)	3	
(e g)	1	✘

จากนั้นคำนวนหาค่าสนับสนุนแต่ละเซตรายการ โดยพิจารณาจากชุดข้อมูล ทำการตัดเซตรายการความยาว 2 ที่ไม่ผ่านค่าสนับสนุนออก (เซตรายการที่ระบายพื้นสีเทา) ดังนั้นเซตรายการความถี่ความยาว 2 ทั้งหมด คือ FI_2 = {(a b), (a c), (a e), (a f), (b c), (b e), (b f), (b g), (c e), (c f), (e f)}

<u>ทำการขยายเซตรายการเป็นเซตรายการความยาว 3</u> โดยเอาเซตรายการความถี่ความยาว 2 ที่มีรายการแรกเหมือนกันมาจับคู่กัน เช่น เซตรายการ (a b) และ (a c) สามารถยุบรวมกันเป็น (a b c) เซตรายการ (a b) และ (a e) สามารถยุบรวมเป็น (a b e) เป็นต้น เมื่อยุบรวมเป็นเซตรายการความยาว 3 จะทำการพิจารณาว่าละเมิดคุณสมบัติของ Apriori หรือไม่ ถ้าละเมิดคุณสมบัติของ Apriori จะทำการตัดเซตรายการดังกล่าวทิ้งไป (เซตรายการที่ระบายพื้นสีเทาดังตารางที่ 5.7)

ตารางที่ 5.7 เซตรายการความยาว 3 และค่าสนับสนุน

เซตรายการ	ค่าสนับสนุน	
(a b c)	2	
(a b e)	3	
(a b f)	2	
(a c e)	2	
(a c f)	2	
(a e f)	3	
(b c e)	2	
(b c f)	2	
(b c g)	ละเมิดคุณสมบัติ Apriori	✘
(b e f)	3	
(b e g)	ละเมิดคุณสมบัติ Apriori	✘
(b f g)	2	
(c e f)	2	
(c e g)	ละเมิดคุณสมบัติ Apriori	✘
(c f g)	ละเมิดคุณสมบัติ Apriori	✘
(e f g)	ละเมิดคุณสมบัติ Apriori	✘

จากตารางที่ 5.7 จะเห็นได้ว่า เซตรายการ (b c g) มีเซตรายการ (c g) เป็นเซตรายการย่อย และเซตรายการ (c g) ไม่ใช่เซตรายการความถี่ (ดังแสดงในตารางที่ 5.6) ทำให้ละเมิดคุณสมบัติ Apriori ดังนั้น (b c g) ถือว่าไม่ใช่เซตรายการความถี่ จึงถูกตัดทิ้งไป เซตรายการ (b e g), (c e g), (c f g) และ (e f g) ละเมิดคุณสมบัติ Apriori เหมือนกัน ดังนั้นจึงถูกตัดทิ้ง

เซตรายการความยาว 3 ที่เหลือจะถูกนำไปคำนวนค่าสนับสนุนและตรวจสอบว่าผ่านค่าสนับสนุนขั้นต่ำหรือไม่ ถ้าไม่ผ่านค่าสนับสนุนขั้นต่ำเซตรายการดังกล่าวจะถูกตัดทิ้ง

ดังนั้นเซตรายการความถี่ความยาว 3 ทั้งหมด คือ Fl_3 = {(a b c), (a b e), (a b f), (a c e), (a c f), (a e f), (b c e), (b c f), (b e f), (b f g), (c e f)}

ทำการขยายเซตรายการเป็นเซตรายการความยาว 4 โดยเอาเซตรายการความถี่ความยาว 3 ที่มี 2 รายการแรกเหมือนกันมาจับคู่กัน เช่น เซตรายการ (a b c) และเซตรายการ (a b e) ยุบรวมเป็นเซตรายการ (a b c e) ถ้ายุบรวมกันแล้วละเมิดคุณสมบัติ Apriori เซตรายการดังกล่าวจะถูกตัดออก เซตรายการที่ไม่ถูกตัดออกจะนำไปคำนวณหาค่าสนับสนุนแล้วตรวจสอบว่าผ่านค่าสนับสนุนหรือไม่

เซตรายการความยาว 4 ทั้งหมดและค่าสนับสนุนแสดงได้ดังตารางที่ 5.8 ซึ่งทุกเซตรายการมีค่าสนับสนุนเท่ากับค่าสนับสนุนขั้นต่ำ ดังนั้นเซตรายการความถี่ความยาว 4 ทั้งหมด คือ Fl_4 = {(a b c e), (a b c f), (a b e f), (a c e f), (b c e f)}

ตารางที่ 5.8 เซตรายการความยาว 4 และค่าสนับสนุน

เซตรายการ	ค่าสนับสนุน
(a b c e)	2
(a b c f)	2
(a b e f)	2
(a c e f)	2
(b c e f)	2

ทำการขยายเซตรายการเป็นเซตรายการความยาว 5 โดยเอาเซตรายการความถี่ความยาว 4 ที่มีเซตรายการ 3 ตัวแรกเหมือนกันมาจับคู่กันกัน เช่น จากตารางที่ 5.8 จะเห็นได้ว่ามีแค่เซตรายการ (a b c e) และ (a b c f) สามารถยุบรวมกันเป็น (a b c e f) และค่าสนับสนุนเท่ากับ 2 ดังแสดงในตารางที่

5.9 ซึ่งเซตรายการ (a b c e f) มีค่าสนับสนุนเท่ากับค่าสนับสนุนขั้นต่ำ แสดงว่าเซตรายการ (a b c e f) เป็นเซตรายการความถี่

ดังนั้นเซตรายการความถี่ความยาว 5 ทั้งหมด คือ Fl_5 = {(a b c e f)}

ตารางที่ 5.9 เซตรายการความยาว 5 และค่าสนับสนุน

เซตรายการ	ค่าสนับสนุน
(a b c e f)	2

จากตารางที่ 5.9 จะเห็นได้ว่ามีเซตรายการความถี่แค่เซตรายการเดียว จึงไม่สามารถขยายต่อได้ ดังนั้นจึงสรุปได้ว่าเซตรายการความถี่ทั้งหมด คือ $Fl_1 \cup Fl_2 \cup Fl_3 \cup Fl_4 \cup Fl_5$

ขั้นตอนที่ 3 ทำการสร้างกฎความสัมพันธ์จากเซตรายการความถี่ที่มีความยาว 2 ขึ้นไป ตามขั้นตอนที่กล่าวไว้ในหัวข้อ 5.2 ด้วยขั้นตอนวิธี Faster จากนั้นทำการตรวจสอบค่าเชื่อมั่นของกฎว่ามากกว่าหรือเท่ากับค่าความเชื่อมั่นขั้นต่ำหรือไม่ กฎที่มีค่าความเชื่อมั่นน้อยกว่าค่าความเชื่อมั่นขั้นต่ำ จะถูกตัดทิ้งไป

ตัวอย่างที่ 5.13 ในส่วนนี้จะขอยกตัวอย่างการสร้างกฎจากเซตรายการความถี่ (b e f) โดยสามารถสร้างกฎความสัมพันธ์ได้ทั้งหมด 2^3-2 = 6 กฎ (แสดงได้ดังตารางที่ 5.10) เมื่อกำหนดค่าความเชื่อมั่นขั้นต่ำเท่ากับ 75% จะเห็นได้ว่าทุกกฎมีค่าความเชื่อมั่นมากกว่าหรือเท่ากับค่าความเชื่อมั่นขั้นต่ำ กฎทั้งหมดที่ผ่านค่าความเชื่อมั่นขั้นต่ำจะถูกนำไปพิจารณาต่อว่าเป็นกฎความสัมพันธ์เชิงลำดับหรือไม่

ตารางที่ 5.10 กฎความสัมพันธ์และค่าความเชื่อมั่น

กฎความสัมพันธ์	ค่าความเชื่อมั่น
(b)⟶(e f)	3 / 4 * 100 = 75%
(e)⟶(b f)	3 / 3 * 100 = 100%
(f)⟶(b e)	3 / 4 * 100 = 75%
(b e)⟶(f)	3 / 3 * 100 = 100%
(b f)⟶(e)	3 / 4 *100 = 75%
(e f)⟶(b)	3 / 3 * 100 = 100%

<u>ขั้นตอนที่ 4</u> ทำการอ่านชุดข้อมูลลำดับเหตุการณ์ เพื่อคำนวณค่าสนับสนุนและค่าความเชื่อมั่นของกฎใหม่ โดยการคำนวณค่าสนับสนุนและค่าความเชื่อมั่นของกฎจะต้องพิจารณาลำดับการเกิดของข้อมูลด้วย ทำให้ได้กฎความสัมพันธ์เชิงลำดับ จากนั้นทำการพิจารณาว่ากฎความสัมพันธ์เชิงลำดับ มีค่าสนับสนุนและค่าความเชื่อมั่นมากกว่าหรือเท่าค่าสนับสนุนขั้นต่ำและค่าความเชื่อมั่นขั้นต่ำหรือไม่ ถ้ากฎความสัมพันธ์เชิงลำดับผ่านค่าสนับสนุนขั้นต่ำและค่าความเชื่อมั่นขั้นต่ำ กฎดังกล่าวก็จะเป็นผลลัพธ์และเป็นกฎที่ยอมรับได้

ตัวอย่างที่ 5.14 พิจารณากฎความสัมพันธ์ (b)⟶(e f) ในตารางที่ 5.10 แบบเชิงลำดับจากชุดข้อมูลลำดับเหตุการณ์ในตารางที่ 5.2 จะได้ค่าสนับสนุนเท่ากับ 3 เนื่องจากมีรายการเปลี่ยนแปลงที่ปรากฏ b แล้วตามด้วย e f (e กับ f ไม่พิจารณาลำดับการเกิด) จำนวน 3 รายการเปลี่ยนแปลง คือ 1, 2 และ 3 ส่วนค่าความเชื่อมั่นของกฎความสัมพันธ์ (b)⟶(e f) เมื่อพิจารณาแบบเชิงลำดับจะมีค่าความเชื่อมั่นเท่ากับ *supp*((b)⟶(e f))/*supp*(b)x100 = 3/4x100 = 75% ดังนั้นจึงสรุปได้ว่ากฎ (b)⟶(e f) เป็นกฎความสัมพันธ์เชิงลำดับที่ยอมรับได้ เนื่องจากมีค่าสนับสนุนมากกว่าค่าสนับสนุนขั้นต่ำ (ค่าสนับสนุนขั้นต่ำเท่ากับ 2) และมีค่าความเชื่อมั่นเท่ากับค่าความเชื่อมั่นขั้นต่ำ (ค่าความเชื่อมั่นขั้นต่ำเท่ากับ 75%)

ส่วนกฎความสัมพันธ์ (e)⟶(b f) เมื่อพิจารณาเชิงลำดับ ปรากฏว่าไม่รายการเปลี่ยนแปลงใดที่ปรากฏ e แล้วตามด้วย b f ดังนั้นกฎ (e)⟶(b f) ไม่ผ่านค่าสนับสนุนขั้นต่ำ กฎ (e)⟶(b f) จึงไม่ใช่กฎความสัมพันธ์เชิงลำดับที่ยอมรับได้

ส่วนกฎความสัมพันธ์ (f) ⟶ (b e), (b e) ⟶ (f) และ (e f) ⟶ (b) เมื่อพิจารณาเชิงลำดับปรากฏว่าไม่มีในรายการเปลี่ยนแปลงใด ดังนั้นกฎดังกล่าวไม่ใช่กฎความสัมพันธ์เชิงลำดับที่ยอมรับได้

ส่วนกฎความสัมพันธ์ (b f) ⟶ (e) เมื่อพิจารณาเชิงลำดับ ปรากฏว่ามีค่าสนับสนุนเท่ากับ 2 เนื่องจากปรากฏรายการ b f ตามด้วย e ในรายการเปลี่ยนแปลงที่ 1 และ 3 แต่ค่าความเชื่อมั่นของกฎความสัมพันธ์ (b f) ⟶ (e) มีค่าเท่ากับ 2/4x100 = 50% ซึ่งไม่ผ่านค่าความเชื่อมั่นขั้นต่ำ ดังนั้นกฎดังกล่าวไม่ใช่กฎความสัมพันธ์เชิงลำดับที่ยอมรับได้

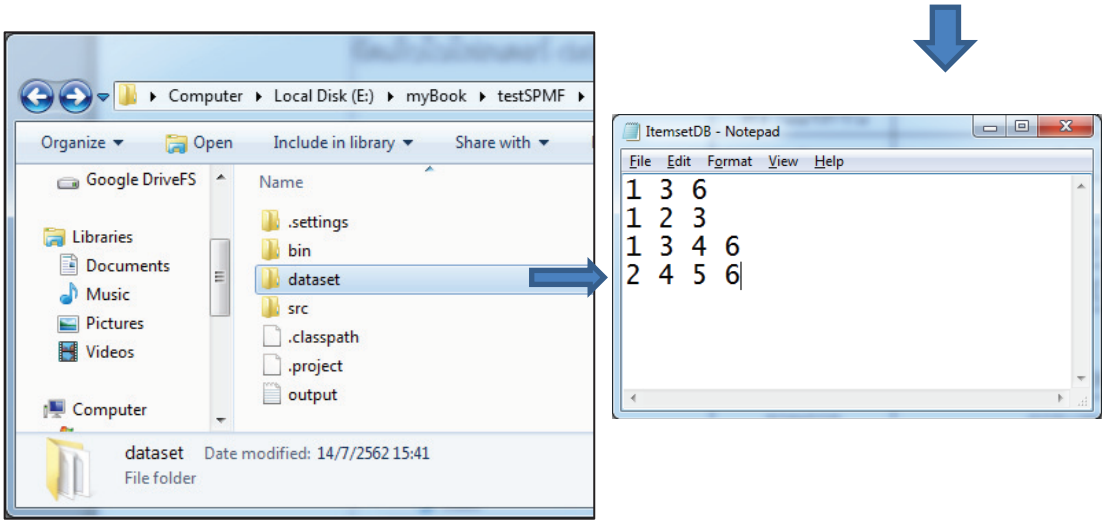
5.4 ตัวอย่างการสร้างกฎความสัมพันธ์โดยใช้ SPMF

การแปลงชุดข้อมูลสำหรับสร้างกฎความสัมพันธ์ สามารถทำได้เหมือนกับการแปลงชุดข้อมูลเซตรายการดังที่ได้กล่าวในบทที่ 2 ในตัวอย่างคำสั่งใช้ไฟล์ชุดข้อมูล ItemsetDB.txt อยู่ในโฟลเดอร์ dataset ดังรูปที่ 5.4 ซึ่งชุดข้อมูลนำเข้าแทนค่าด้วยตัวเลขดังตารางที่ 5.11

ตารางที่ 5.11 การแทนค่าด้วยตัวเลขเพื่อสร้างกฎความสัมพันธ์

รายการ	การแทนค่า
A	1
B	2
C	3
D	4
E	5
F	6

รายการเปลี่ยนแปลง	เซตรายการ
1	(A C F)
2	(A B C)
3	(A C D F)
4	(B D E F)



รูปที่ 5.4 ตัวอย่างไฟล์นำเข้าสำหรับสร้างกฎความสัมพันธ์และการจัดเก็บ

ส่วนตัวอย่างคำสั่งสำหรับการสร้างกฎความสัมพันธ์จากเซตรายการความถี่ที่ขุดค้นด้วยขั้นตอนวิธี F-Growth แสดงในตัวอย่างคำสั่งที่ 5.1 โดยแต่ละคำสั่งสามารถอธิบายได้ดังนี้

ตัวอย่างคำสั่งที่ 5.1

```
1.  package mySpmfProject;
2.
3.  import java.io.IOException;
4.  import ca.pfv.spmf.algorithms.associationrules.agrawal94_association_rules.AlgoAgrawalFaster94;
5.  import ca.pfv.spmf.algorithms.frequentpatterns.fpgrowth.AlgoFPGrowth;
6.  import ca.pfv.spmf.patterns.itemset_array_integers_with_count.Itemsets;
7.
8.  public class AssociationRuleGen {
9.          public static void main(String [] arg) throws IOException{
10                 String input = ".//dataset/ItemsetDB.txt";
11                 String output = ".//output.txt";
12         // STEP 1: Applying the FP-GROWTH algorithm to find frequent itemsets
13                 double minsupp = 0.5;
14                 AlgoFPGrowth fpgrowth = new AlgoFPGrowth();
15                 Itemsets patterns = fpgrowth.runAlgorithm(input, null, minsupp);
16                 fpgrowth.printStats();
17                 int databaseSize = fpgrowth.getDatabaseSize();
18         // STEP 2: Generating all rules from the set of frequent itemsets
19                 double  minconf = 0.80;
20                 AlgoAgrawalFaster94 algoAgrawal = new AlgoAgrawalFaster94();
21                 algoAgrawal.runAlgorithm(patterns, output, databaseSize, minconf);
22                 algoAgrawal.printStats();
23         }
24. }
```

บรรทัดที่ 4-6 เป็นการ import คลาสที่จำเป็นสำหรับการสร้างกฎความสัมพันธ์ ซึ่งมีดังนี้

- คลาส AlgoAgrawalFaster94 สำหรับขั้นตอนวิธี Faster
- คลาส AlgoFPGrowth สำหรับขั้นตอนตอนวิธี FP-Growth

บรรทัดที่ 10 ทำการระบุไฟล์ชุดข้อมูลนำเข้า

บรรทัดที่ 11 เป็นการระบุไฟล์ที่ต้องการบันทึกผลลัพธ์ที่ได้จากการประมวลผล

บรรทัดที่ 13 กำหนดค่าสนับสนุนขั้นต่ำแบบสัมพัทธ์ให้มีค่าเท่ากับ 0.5 หรือ 50%

บรรทัดที่ 14 สร้างอ๊อบเจ็กต์ของคลาส AlgoFPGrwoth เพื่อเรียกใช้ขั้นตอนวิธี FP-Growth

บรรทัดที่ 15 เรียกใช้เมทธอด runAlgorithm เพื่อประมวลผลด้วยขั้นตอนวิธี FP-Growth โดยเมทธอด runAlgorithm มีพารามิเตอร์มี 3 ตัว คือ

1) ไฟล์ชุดข้อมูลนำเข้า (input)

2) ไฟล์ผลลัพธ์(output) ในที่นี้กำหนดให้เป็น null เพื่อต้องการเก็บเซตรายการในโครงสร้างข้อมูล (patterns) (ถ้ากำหนดชื่อไฟล์ output จะเป็นการเขียนเซตรายการความถี่ลงในไฟล์)

3) ค่าสนับสนุนขั้นต่ำ (minsupp)

บรรทัดที่ 16 แสดงค่าทางสถิติออกมาดังรูปที่ 5.5 ในส่วนที่ 1 โดยแสดงรายละเอียดดังนี้

- จำนวนรายการเปลี่ยนแปลง (Transactions count from database)
- หน่วยความจำที่ใช้ (Max memory usage)
- จำนวนเซตรายการความถี่ (Frequent itemsets count)
- เวลาในการประมวลผล (Total time)

บรรทัดที่ 17 นับจำนวนรายการเปลี่ยนแปลง

บรรทัดที่ 19 เป็นการกำหนดค่าความเชื่อมั่นขั้นต่ำให้มีค่าเท่ากับ 0.80 หรือ 80%

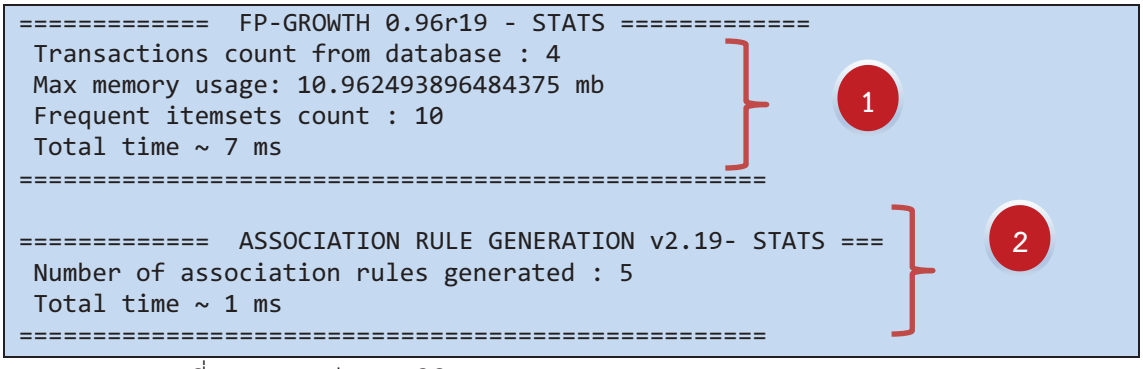
บรรทัดที่ 20 สร้างอ๊อบเจ็กต์ของคลาส AlgoAgrawalFaster94 เพื่อเรียกใช้ขั้นตอนวิธี Faster

บรรทัดที่ 21 เรียกเมทธอด runAlgorithm เพื่อให้สร้างกฎความสัมพันธ์ ซึ่งมีพารามิเตอร์ทั้งหมด 4 ตัว คือ

1) เซตรายการความถี่ (patterns)
2) ไฟล์ผลลัพธ์ (output) ซึ่งก็คือ ไฟล์ที่เก็บกฎความสัมพันธ์
3) จำนวนรายการเปลี่ยนแปลงในชุดข้อมูล (databbaseSize)
4) ค่าความเชื่อมั่นขั้นต่ำ (minconf)

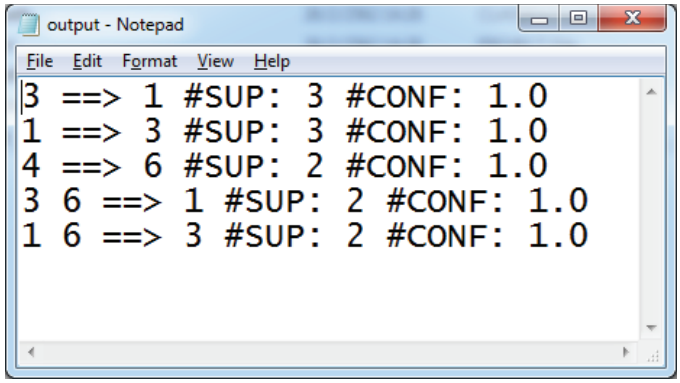
บรรทัดที่ 22 แสดงค่าทางสถิติออกมาดังรูปที่ 5.5 ในส่วนที่ 2 โดยแสดงรายละเอียดดังนี้

- จำนวนกฎความสัมพันธ์ (Number of association rules generated)
- เวลาในการสร้างกฎความสัมพันธ์ (Total time)

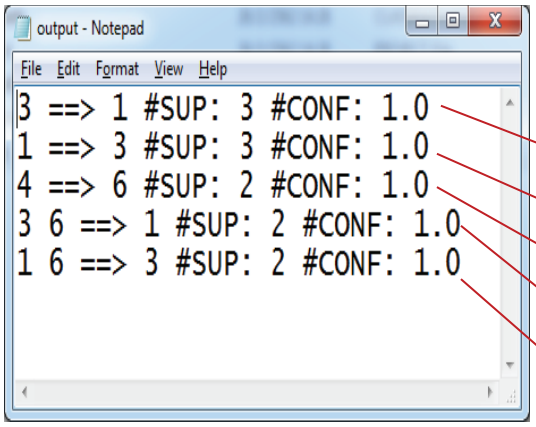


รูปที่ 5.5 แสดงค่าทางสถิติจากการประมวลผล AssociationRuleGen.java

กฎความสัมพันธ์ที่ได้จากการประมวลผลถูกเก็บไว้ในไฟล์ output.txt ดังรูปที่ 5.6 ซึ่งจะแสดงกฎความสัมพันธ์ ค่าสนับสนุนแบบสัมบูรณ์ ค่าความเชื่อมั่น เช่น ในบรรทัดแรก 3 ==>1 #SUPP:3 #CONF: 1.0 แสดงถึงกฎ 3 ==>1 มีค่าสนับสนุนแบบสัมบูรณ์เท่ากับ 3 และมีค่าความเชื่อมั่นเท่ากับ 1.0 หรือ 100% เมื่อแปลงผลลัพธ์กลับคืนจะได้ดังรูปที่ 5.7



รูปที่ 5.6 ไฟล์ผลลัพธ์จากการประมวลผล AssociationRuleGen.java



กฎที่ได้	ค่าสนับสนุน	ค่าความเชื่อมั่น (%)
C⟶A	3	100%
A⟶C	3	100%
D⟶F	2	100%
CF⟶A	2	100%
AF⟶C	2	100%

รูปที่ 5.7 แปลงผลลัพธ์กฎความสัมพันธ์

กฎความสัมพันธ์ที่ได้สามารถแปลความหมายดังตัวอย่างต่อไปนี้

กฎที่ 1 ความถี่ในการเกิดรายการ C ร่วมกับรายการ A คือ 3 และเมื่อปรากฏรายการ C มีโอกาสปรากฏรายการ A ร่วมด้วยถึง 100%

กฎที่ 2 ความถี่ในการเกิดรายการ A ร่วมกับรายการ C คือ 3 และเมื่อปรากฏรายการ A มีโอกาสปรากฏรายการ C ร่วมด้วยถึง 100%

กฎที่ 3 ความถี่ในการเกิดรายการ D ร่วมกับรายการ F คือ 2 และเมื่อปรากฏรายการ D มีโอกาสปรากฏรายการ F ร่วมด้วยถึง 100%

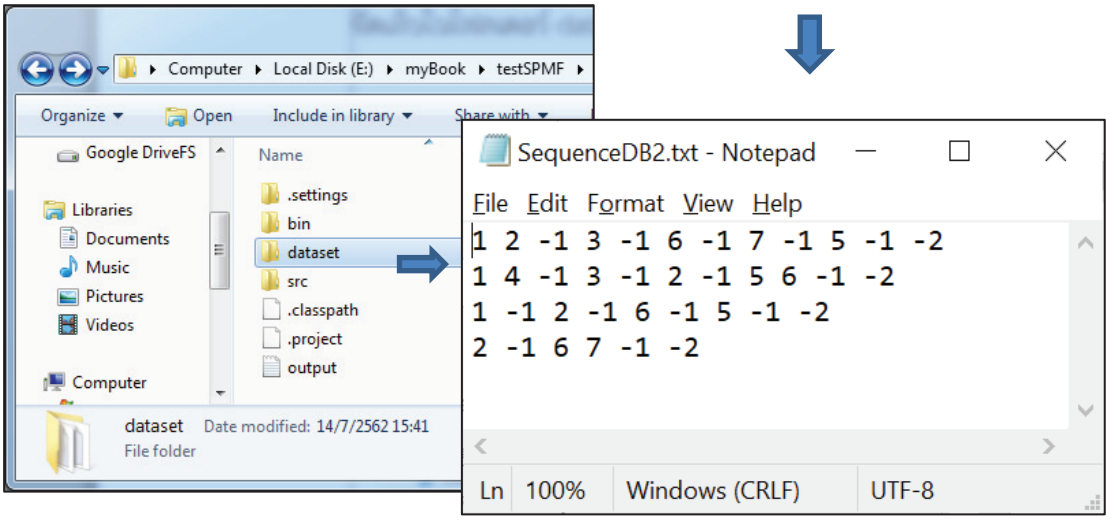
กฎที่ 4 ความถี่ในการเกิดรายการ CF ร่วมกับรายการ A คือ 2 และเมื่อปรากฏรายการ CF มีโอกาสปรากฏรายการ A ร่วมด้วยถึง 100%

กฎที่ 5 ความถี่ในการเกิดรายการ AF ร่วมกับรายการ C คือ 2 และเมื่อปรากฏรายการ AF มีโอกาสปรากฏรายการ C ร่วมด้วยถึง 100%

5.5 ตัวอย่างการสร้างกฎความสัมพันธ์เชิงลำดับโดยใช้ SPMF

การแปลงชุดข้อมูลสำหรับสร้างกฎความสัมพันธ์เชิงลำดับ สามารถทำได้เหมือนกับการแปลงชุดข้อมูลลำดับเหตุการณ์ดังที่ได้กล่าวในบทที่ 3 จากตัวอย่างชุดข้อมูลในตารางที่ 5.2 สามารถแทนค่าด้วยตัวเลขตามตารางที่ 5.12 แปลงเป็นชุดข้อมูลตัวเลขเพื่อนำเข้า SPMF ได้ดังรูปที่ 5.8 และบันทึกไฟล์ชื่อ SequenceDB2.txt จัดเก็บในโฟลเดอร์ dataset

รายการเปลี่ยนแปลง	ลำดับเหตุการณ์
1	<(a b) (c) (f) (g) (e)>
2	<(a d) (c) (b) (e f)>
3	<(a) (b) (f) (e)>
4	<(b) (f g)>



รูปที่ 5.8 ตัวอย่างไฟล์นำเข้าสำหรับสร้างกฎความสัมพันธ์เชิงลำดับและการจัดเก็บ

ตารางที่ 5.12 การแทนค่าด้วยตัวเลขเพื่อสร้างกฎความสัมพันธ์เชิงลำดับ

รายการ	การแทนค่า
a	1
b	2
c	3
d	4
e	5
f	6
g	7

คำสั่งสำหรับสร้างกฎความสัมพันธ์เชิงลำดับโดยใช้ขั้นตอนวิธี CMRules แสดงได้ดังตัวอย่างคำสั่งที่ 5.2 โดยแต่ละคำสั่งสามารถอธิบายได้ดังนี้

ตัวอย่างคำสั่งที่ 5.2

```
1.  package mySpmfProject;
2.
3.  import java.io.IOException;
4.  import ca.pfv.spmf.algorithms.sequential_rules.cmrules.AlgoCMRules;
5.
6.  public class CMRuleTest {
7.          public static void main(String [] arg) throws IOException{
8.                  String input = ".//dataset/SequenceDB2.txt";
9.                  String  output = ".//output.txt";
10
11.                 double minSup = 0.50;
12.                 double minConf = 0.75;
13.
14.                 AlgoCMRules algo = new AlgoCMRules();
15.                 algo.runAlgorithm(input, output, minSup, minConf);
16.                 algo.printStats();
17.         }
18. }
```

บรรทัดที่ 4 เป็นการ import คลาส AlgoCMRules เพื่อขุดค้นกฎความสัมพันธ์เชิงลำดับ

บรรทัดที่ 8 ระบุไฟล์ชุดข้อมูลนำเข้า โดยในตัวอย่างคำสั่งไฟล์ชุดข้อมูล คือ SequenceDB2.txt

บรรทัดที่ 9 เป็นการระบุไฟล์ที่ต้องการบันทึกผลลัพธ์ที่ได้จากการประมวลผล

บรรทัดที่ 11 กำหนดค่าสนับสนุนขั้นต่ำแบบสัมพัทธ์ให้มีค่าเท่ากับ 0.5 หรือ 50%

บรรทัดที่ 12 เป็นการกำหนดค่าความเชื่อมั่นขั้นต่ำให้มีค่าเท่ากับ 0.75 หรือ 75%

บรรทัดที่ 14 สร้างอ๊อบเจ็กต์ของคลาส AlgoCMRules

บรรทัดที่ 15 เรียกเมทธอด runAlgorithm เพื่อให้ขั้นตอนวิธี CMRules ประมวลผล โดยมีพารามิเตอร์ที่เกี่ยวข้อง 4 ตัว คือ

1) ไฟล์ชุดข้อมูลนำเข้า (input)

2) ไฟล์ผลลัพธ์ (output)

3) ค่าสนับสนุนขั้นต่ำ (minSup)

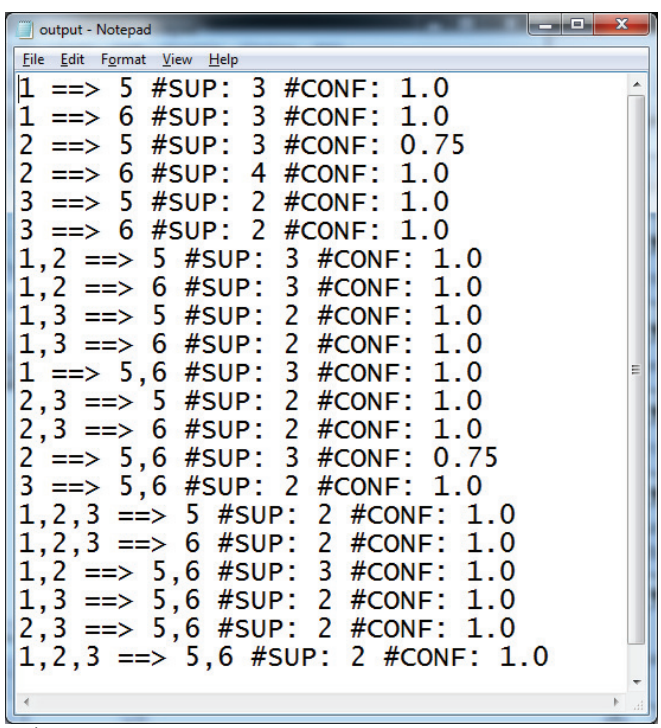
4) ค่าความเชื่อมั่นขั้นต่ำ (minConf)

บรรทัดที่ 16 แสดงค่าทางสถิติออกมาดังรูปที่ 5.9 โดยจะแสดงข้อมูลดังนี้

- จำนวนกฎความสัมพันธ์ (Association rules count)
- จำนวนกฎความสัมพันธ์เชิงลำดับ (Sequential rules count)
- เวลาในการสร้างกฎ (Total time)
- หน่วยความจำที่ใช้ (Max memory)

```
=============  CMRULES - STATS =============
Association rules count: 120
Sequential rules count: 21
Total time : 12 ms
Max memory: 10.495742797851562
===================================================
```

รูปที่ 5.9 แสดงค่าทางสถิติจากการประมวลผล CMRuleTest.java



รูปที่ 5.10 ไฟล์ผลลัพธ์จากการประมวลผล CMRuleTest.java

กฎความสัมพันธ์เชิงลำดับที่ได้จะถูกเก็บไว้ในไฟล์ output.txt ดังรูปที่ 5.10 ซึ่งจะแสดงกฎความสัมพันธ์เชิงลำดับ ค่าสนับสนุนแบบสัมบูรณ์ และค่าความเชื่อมั่น เช่น ในบรรทัดแรก 1 ==>3 #SUPP:3 #CONF: 1.0 แสดงถึงกฎความสัมพันธ์เชิงลำดับ 1 ==>3 มีค่าสนับสนุนแบบสัมบูรณ์เท่ากับ 3 และมีค่าความเชื่อมั่นเท่ากับ 1.0 หรือ 100% เมื่อแปลงผลลัพธ์กลับคืนจะได้ดังตารางที่ 5.13

ตารางที่ 5.13 การแปลงผลลัพธ์กฎความสัมพันธ์เชิงลำดับ

ลำดับ	กฎความสัมพันธ์เชิงลำดับ	ค่าสนับสนุน	ค่าความเชื่อมั่น (%)
1	(a)⟶(e)	3	100
2	(a)⟶(f)	3	100
3	(b)⟶(e)	3	75
4	(b)⟶(f)	3	100
5	(c)⟶(e)	2	100
6	(c)⟶(f)	2	100
7	(a b)⟶(e)	3	100
8	(a b)⟶(f)	3	100
9	(a c)⟶(e)	3	100
10	(a c)⟶(f)	3	100
11	(a)⟶(e f)	3	100
12	(b c)⟶(e)	2	100
13	(b c)⟶(f)	2	100
14	(b)⟶(e f)	3	75
15	(c)⟶(e f)	2	100
16	(a b c)⟶(e)	2	100
17	(a b c)⟶(f)	2	100
18	(a b)⟶(e f)	3	100
19	(a c)⟶(e f)	2	100
20	(b c)⟶(e f)	2	100
21	(a b c)⟶(e f)	2	100

กฎความสัมพันธ์เชิงลำดับที่ได้สามารถแปลความหมายของกฎได้ดังตัวอย่างต่อไปนี้

กฎที่ 1 ความถี่ในการเกิดรายการ a แล้วตามด้วยรายการ e คือ 3 และเมื่อปรากฏรายการ a มีโอกาสปรากฏรายการ e ตามหลังถึง 100%

กฎที่ 2 ความถี่ในการเกิดรายการ a แล้วตามด้วยรายการ f คือ 3 และเมื่อปรากฏรายการ a มีโอกาสปรากฏรายการ f ตามหลังถึง 100%

กฎที่ 3 ความถี่ในการเกิดรายการ b แล้วตามด้วยรายการ e คือ 3 และเมื่อปรากฏรายการ b มีโอกาสปรากฏรายการ e ตามหลังถึง 75%

....

กฎที่ 21 ความถี่ในการเกิดรายการ a b c แล้วตามด้วยรายการ e f คือ 2 และเมื่อปรากฏรายการ a b c มีโอกาสปรากฏรายการ e f ตามหลังถึง 100%

บทสรุป

การทำเหมืองกฎความสัมพันธ์ เป็นการค้นหาความสัมพันธ์ของข้อมูลที่ปรากฏร่วมกันบ่อย ผลลัพธ์ที่ได้จะอยู่ในรูปของกฎ $X \rightarrow Y$ โดยที่ X คือ เหตุ และ Y เป็นผลที่ตามมา โดยกฎที่ยอมรับได้จะต้องมีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ และมีค่าความเชื่อมั่นมากกว่าหรือเท่ากับค่าความเชื่อมั่นขั้นต่ำ กฎความสัมพันธ์ถูกสร้างขึ้นจากเซตรายการความถี่ที่มีความยาว 2 รายการขึ้นไป โดยนำเซตรายการย่อยของเซตรายการความถี่ไปสร้างกฎ เซตรายการความถี่ที่มีความยาว n จะสามารถสร้างกฎความสัมพันธ์ได้ทั้งหมด $2^n - 2$

ส่วนการทำเหมืองกฎความสัมพันธ์เชิงลำดับเป็นการค้นหาความสัมพันธ์ของข้อมูลที่เกิดขึ้นตามลำดับ แสดงอยู่ในรูป $X \rightarrow Y$ โดยที่เซตรายการ X จะต้องเกิดก่อนเซตรายการ Y โดยกฎความสัมพันธ์เชิงลำดับที่ยอมรับได้จะต้องมีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ และมีค่าความเชื่อมั่นมากกว่าหรือเท่ากับค่าความเชื่อมั่นขั้นต่ำ

แบบฝึกหัดท้ายบท

1. ร้านค้าแห่งหนึ่งจัดเก็บข้อมูลการซื้อสินค้าของลูกค้าดังตารางข้างล่าง และต้องการนำข้อมูลดังกล่าวไปหาความสัมพันธ์ของการซื้อสินค้า เพื่อใช้ในการส่งเสริมการขายสินค้า จงแปลงข้อมูลให้อยู่ในรูปแบบของรายการเปลี่ยนแปลงเพื่อใช้ในการสร้างกฎความสัมพันธ์

ใบเสร็จเลขที่	รายการสินค้า
3001200	นม
3001200	ผ้าอ้อมเด็ก
3001200	ไข่ไก่
3001201	นม
3001201	ขนมปังกรอบ
3001202	ขนมปังกรอบ
3001202	นม
3001202	ไข่ไก่
3001203	ผ้าอ้อม
3001203	ไข่ไก่
3001203	ผ้าอ้อม

2. จงอธิบายขั้นตอนการสร้างกฎความสัมพันธ์จากเซตรายการความถี่

3. จงอธิบายความหมายของค่าความเชื่อมั่นเมื่อกำหนดให้ค่าความเชื่อมั่นของกฎความสัมพันธ์ a$\rightarrow$b เท่ากับ 80%

4. เซตรายการความถี่ความยาว 10 สามารถสร้างกฎความสัมพันธ์ได้กี่กฎ

5. กำหนดให้เซตรายการความถี่และค่าสนับสนุนทั้งหมดประกอบไปด้วย (a):10, (b):8, (c):5, (ab): 8, (ac):5, (ac):6 และ (abc):5 จงแสดงกฎความสัมพันธ์ที่ได้จากเซตรายการความถี่ดังกล่าวและแสดงค่าความเชื่อมั่นของแต่ละกฎ

6. จากข้อ 5 กฎความสัมพันธ์ใดบ้างเป็นกฎที่ยอมรับได้ เมื่อกำหนดค่าความเชื่อมั่นขั้นต่ำเท่ากับ 80%

7. จงอธิบายความหมายของกฎความสัมพันธ์เชิงลำดับ

8. จากชุดข้อมูลลำดับเหตุการณ์ดังตารางข้างล่าง จงแสดงการคำนวณค่าสนับสนุนและค่าความเชื่อมั่นของกฎความสัมพันธ์เชิงลำดับ (A B) $\rightarrow$ (E)

ลำดับ	ลำดับเหตุการณ์
1	<(C D) (A B C) (E)>
2	<(A B F) (E) >
3	<(A B F) (A)>
4	<(A) (B) (E)>

9. จากชุดข้อมูลในข้อ 8 จงแสดงวิธีการหากฎความสัมพันธ์เชิงลำดับโดยใช้ขั้นตอนวิธี CMRules

10. จงยกตัวอย่างการนำกฎความสัมพันธ์เชิงลำดับนำไปใช้ประโยชน์

11. ฝึกเขียนโปรแกรมเพื่อค้นหากฎความสัมพันธ์เชิงลำดับจากชุดข้อมูลลำดับเหตุการณ์ในข้อ 8 โดยกำหนดให้ค่าสนับสนุนขั้นต่ำแบบสัมพัทธ์เท่ากับ 50% และค่าความเชื่อมั่นขั้นต่ำเท่ากับ 60%

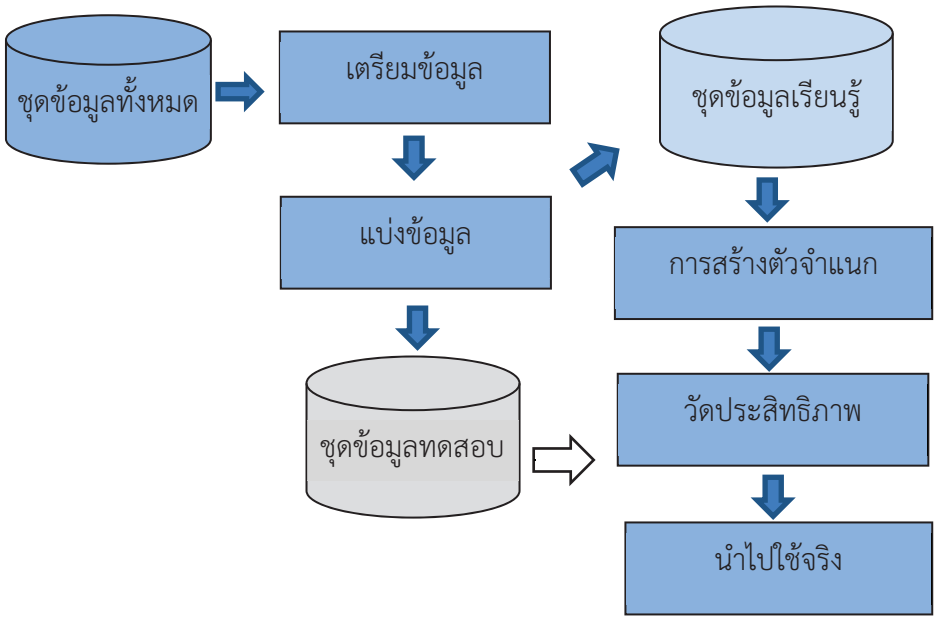
บทที่ 6
การจำแนกเชิงความสัมพันธ์
(Associative Classification)

การจำแนกเชิงความสัมพันธ์ (Associative classification) เป็นการผสมผสานระหว่างการทำเหมืองกฎความสัมพันธ์และการจำแนกข้อมูลเข้าด้วยกัน เพื่อสร้างกฎที่สามารถใช้ในการจำแนกข้อมูลได้ กฎที่ใช้ในการจำแนกอยู่ในรูปแบบ $r: X \rightarrow c$ โดยที่ X คือ เซตรายการ และ c คือ คลาสที่ใช้ในการทำนาย กฎถูกสร้างขึ้นบนพื้นฐานของการทำเหมืองกฎความสัมพันธ์ ค่าสนับสนุนขั้นต่ำและค่าความเชื่อมั่นขั้นต่ำถูกนำมาใช้เป็นตัวคัดกรองเอาเฉพาะกฎที่จะใช้ในการจำแนก จากนั้นกฎที่สามารถครอบคลุมข้อมูลได้จะถูกคัดเลือกเพื่อสร้างตัวจำแนกที่มีประสิทธิภาพ ตัวจำแนกข้อมูลที่สร้างขึ้นจากการจำแนกเชิงความสัมพันธ์ประกอบไปด้วยกฎที่สามารถเข้าใจง่ายโดยมนุษย์และให้ค่าความถูกต้องสูง ทำให้การจำแนกเชิงความสัมพันธ์เป็นวิธีการหนึ่งที่ได้รับความนิยมในการจำแนกข้อมูล

6.1 การจำแนกข้อมูล

การจำแนกข้อมูลเป็นการทำนายกลุ่มของข้อมูลหรือคลาสให้กับข้อมูลที่ไม่ทราบกลุ่มมาก่อน โดยการทำนายกลุ่มข้อมูลได้จากการเรียนรู้ข้อมูลที่มีการกำหนดกลุ่มไว้แล้ว การจำแนกข้อมูลเป็นวิธีการที่ได้รับความนิยมในการทำเหมืองข้อมูล และสามารถนำไปประยุกต์ใช้ในหลายด้าน เช่น การจำแนกผู้ป่วย การจำแนกดอกไม้ การจำแนกลูกค้า การจำแนกข้อความสแปม เป็นต้น ปัจจุบันได้มีการพัฒนาและนำเสนอวิธีการที่หลากหลายเพื่อสร้างตัวจำแนกที่มีประสิทธิภาพ เช่น นาอีฟเบส์ ต้นไม้ตัดสินใจ การค้นหาเพื่อนบ้านที่ใกล้ที่สุด k ตัว ซัพพอร์ตเวกเตอร์แมชชีน เป็นต้น

การนำตัวจำแนกไปใช้งานจริง จำเป็นต้องประเมินประสิทธิภาพของตัวจำแนกก่อนดังรูปที่ 6.1 โดยเริ่มต้นจากการเตรียมข้อมูล จากนั้นทำการแบ่งข้อมูลออกเป็น 2 ชุด คือ ชุดข้อมูลเรียนรู้ (Training set) และชุดข้อมูลทดสอบ (Testing set) ชุดข้อมูลเรียนรู้เป็นชุดข้อมูลที่มีการระบุกลุ่มไว้แล้ว ถูกนำไปใช้ในการสร้างตัวจำแนกเพื่อให้ตัวจำแนกเรียนรู้ลักษณะของแต่ละกลุ่ม ส่วนชุดข้อมูลทดสอบเป็นชุดข้อมูลที่ไม่มีการระบุกลุ่มไว้ นำไปทดสอบกับตัวจำแนก เพื่อให้ตัวจำแนกทำนายกลุ่มออกมา จากนั้นทำการเปรียบเทียบกับผลเฉลย แล้ววัดประสิทธิภาพตัวจำแนกว่ามีความสามารถในการทำนายมากน้อยเพียงใด โดยรายละเอียดของแต่ละขั้นตอนแสดงในหัวข้อย่อยต่อไปนี้



รูปที่ 6.1 ขั้นตอนการจำแนก

6.1.1 การเตรียมข้อมูล

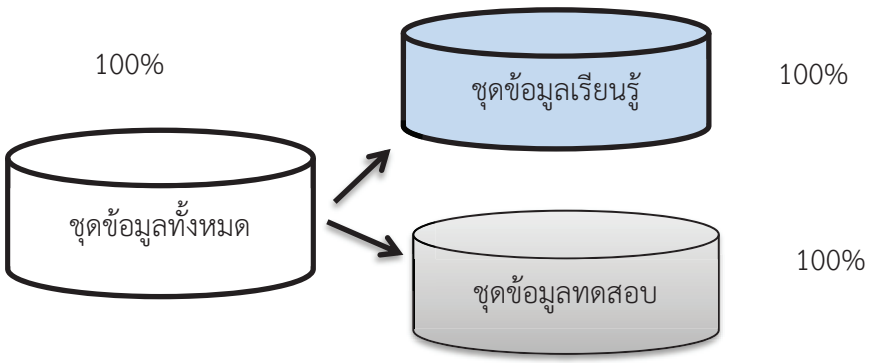
การเตรียมข้อมูลถือว่าเป็นสิ่งที่สำคัญ เนื่องจากการเตรียมข้อมูลที่ดีจะทำให้ตัวจำแนกมีประสิทธิภาพที่ดี โดยการเตรียมข้อมูลประกอบไปด้วย

- การทำความสะอาดข้อมูล ขจัดสิ่งรบกวน หรือเติมข้อมูล เพื่อให้ข้อมูลที่นำมาใช้มีความสมบูรณ์
- การเปลี่ยนรูปข้อมูลเพื่อช่วยลดความหลากหลายของข้อมูล
- ลดจำนวนคุณลักษณะหรือการคัดเลือกคุณลักษณะ เพื่อหาคุณลักษณะที่มีความสัมพันธ์กับกลุ่มข้อมูล ซึ่งจะช่วยให้สามารถจำแนกข้อมูลได้ดีขึ้น

6.1.2 การแบ่งข้อมูล

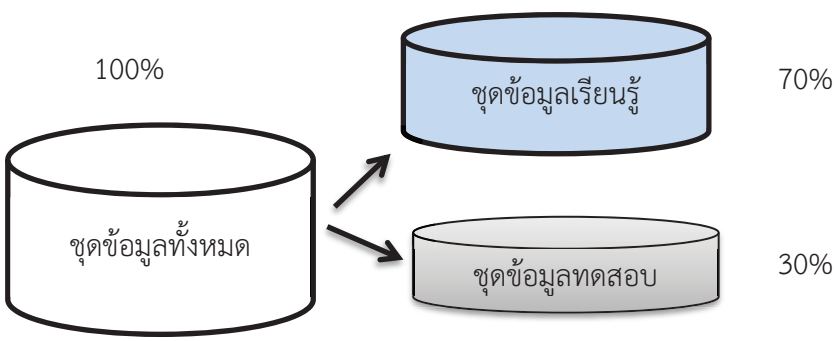
การแบ่งข้อมูลเพื่อทดสอบประสิทธิภาพของตัวจำแนก จะทำการแบ่งชุดข้อมูลทั้งหมดออกเป็น 2 ชุด คือ ชุดข้อมูลเรียนรู้และชุดข้อมูลทดสอบ ชุดข้อมูลเรียนรู้ใช้ในการสร้างตัวจำแนก ส่วนชุดข้อมูลทดสอบใช้ในการทดสอบตัวจำแนกว่ามีประสิทธิภาพมากน้อยเพียงใด การแบ่งข้อมูลมีหลายวิธีในบทนี้จะกล่าวถึง 3 วิธีโดยสังเขปดังต่อไปนี้

1. วิธีการแบ่งข้อมูลแบบ Self-consistency validation คือ การใช้ข้อมูลชุดเดียวกันในการสร้างตัวจำแนกและทดสอบตัวจำแนก ใช้ข้อมูลชุดเดียวกันเป็นทั้งชุดข้อมูลเรียนรู้และชุดข้อมูลทดสอบ (ดังรูปที่ 6.2) วิธีการแบ่งข้อมูลดังกล่าวเป็นวิธีการที่ง่ายในการทดสอบประสิทธิภาพตัวจำแนก



รูปที่ 6.2 การแบ่งข้อมูลแบบ Self-consistency validation

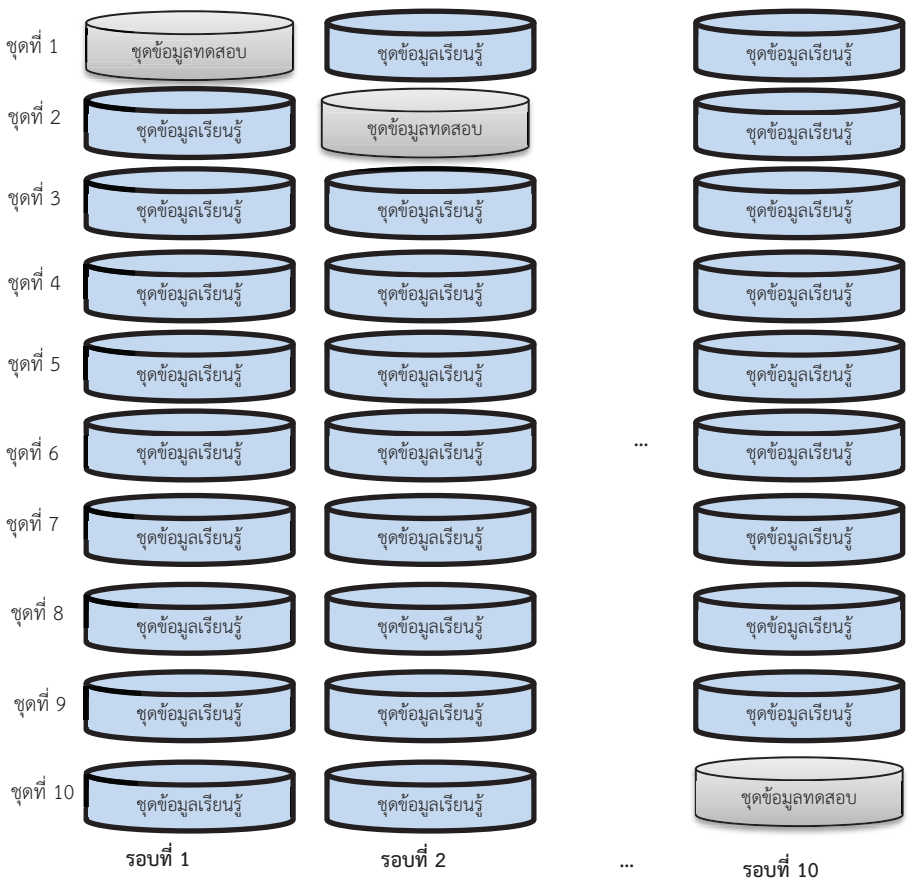
3. วิธีการแบ่งข้อมูลแบบ Hold-out validation คือ การแบ่งข้อมูลออกเป็น 2 ส่วนตามอัตราส่วนที่กำหนด โดยจะต้องทำการสุ่มข้อมูลก่อนทำการแบ่ง เช่น แบ่งชุดข้อมูลเรียนรู้เป็น 70% จากข้อมูลทั้งหมดและแบ่งชุดข้อมูลทดสอบเป็น 30% จากข้อมูลทั้งหมด (ดังรูปที่ 6.3) เป็นต้น อัตราส่วนในการแบ่งชุดข้อมูลเรียนรู้และชุดข้อมูลทดสอบขึ้นอยู่กับผู้ใช้หรือนักวิจัยที่จะกำหนดอัตราส่วนเองหรืออาจจะได้จากการทดลอง แต่นิยมแบ่งอัตราส่วนของชุดข้อมูลเรียนรู้มากกว่าชุดข้อมูลทดสอบ เพื่อให้ตัวจำแนกมีข้อมูลสำหรับการเรียนรู้ที่เพียงพอ การแบ่งข้อมูลโดยใช้วิธีนี้จะต้องทำการทดสอบหลายๆ ครั้ง และแต่ละครั้งจะต้องสุ่มชุดข้อมูลทดสอบและชุดข้อมูลเรียนรู้แตกต่างกันไป



รูปที่ 6.3 การแบ่งข้อมูลแบบ Hold-out validation

3. วิธีการแบ่งข้อมูลแบบ k-fold cross validation คือ การแบ่งข้อมูลออกเป็น *k* ชุด ชุดละเท่าๆ กัน และทำการทดสอบตัวจำแนกจำนวน *k* รอบ โดยในรอบที่ *k* จะใช้ข้อมูลชุดที่ *k* เป็นชุดข้อมูลทดสอบ ข้อมูลชุดที่เหลือเป็นชุดข้อมูลเรียนรู้ เช่น ถ้ากำหนด k = 10 ในรอบแรก ข้อมูลชุดที่ 1 เป็นชุดข้อมูลทดสอบและข้อมูลชุดที่เหลือเป็นชุดเรียนรู้ (ชุดที่ 2-10) ในรอบที่ 2 ข้อมูลชุดที่ 2 เป็นชุดข้อมูลทดสอบและข้อมูลชุดที่เหลือเป็นชุดข้อมูลเรียนรู้ (ชุดที่ 1 และชุดที่ 3-10) ทำวนไปเช่นนี้ จนครบ 10 รอบ (ดังรูปที่ 6.4) เป็นต้น เมื่อทำครบทุกรอบนำค่าประสิทธิภาพที่ได้ในแต่ละรอบมาคำนวณค่าเฉลี่ย

การแบ่งข้อมูลด้วยวิธีการ k-fold cross validation เป็นวิธีที่ได้รับความนิยมในการวัดประสิทธิภาพตัวจำแนก เนื่องจากข้อมูลทุกตัวถูกนำไปทดสอบกับตัวจำแนก



รูปที่ 6.4 การแบ่งข้อมูลแบบ 10-fold cross validation

6.1.3 การสร้างตัวจำแนก

ตัวจำแนกถูกสร้างขึ้นจากชุดข้อมูลเรียนรู้ ซึ่งเป็นข้อมูลที่มีการระบุกลุ่มข้อมูลไว้แล้ว ตัวจำแนกที่สร้างขึ้นได้จากการเรียนรู้ลักษณะหรือรูปแบบของข้อมูลที่มีการระบุกลุ่มข้อมูลไว้แล้ว ตัวจำแนกถูกสร้างขึ้นด้วยวิธีการที่หลากหลาย เช่น นาอีฟเบส์ ต้นไม้ตัดสินใจ การค้นหาเพื่อนบ้านที่ใกล้ที่สุด k ตัว ซัพพอร์ตเวกเตอร์แมชชีน และ การจำแนกเชิงความสัมพันธ์ เป็นต้น ซึ่งแต่ละวิธีมีกระบวนการแตกต่างกันในการสร้างตัวจำแนก เช่น นาอีฟเบส์ใช้พื้นฐานทฤษฎีเบส์เพื่อสร้างตัวจำแนก ซึ่งพิจารณาจากความเป็นไปได้ของข้อมูลและกลุ่ม การค้นหาเพื่อนบ้านที่ใกล้ที่สุด k ตัวเป็นจำแนกข้อมูลโดยคำนวณระยะความห่างระหว่างแอตทริบิวต์ ทำการเลือกแค่ k กลุ่มที่มีระยะห่างใกล้ที่สุด แล้วกำหนดกลุ่มจากกลุ่มที่มีจำนวนมากที่สุดใน k กลุ่ม ส่วนซัพพอร์ตเวกเตอร์แมชชีนจำแนกข้อมูลด้วยการหาเส้นแบ่งที่เหมาะสม การจำแนกเชิงความสัมพันธ์ใช้กฎความสัมพันธ์ในการจำแนกข้อมูล ซึ่งเป็นวิธีการที่มีประสิทธิภาพและง่ายต่อการเข้าใจ

6.1.4 การวัดประสิทธิภาพ

การวัดประสิทธิภาพในการจำแนกสามารถวัดได้หลายวิธี เช่น ความเร็วในการทำนายข้อมูลของตัวจำแนก ความทนทานต่อการทำนายข้อมูลที่มีสิ่งรบกวนหรือการขาดหายไป ความยืดหยุ่นต่อปริมาณข้อมูล ความสามารถที่ตัวจำแนกสามารถเข้าใจได้ง่ายจากผู้ใช้งาน และความสามารถในการทำนาย เป็นต้น โดยวิธีการที่ได้รับความนิยมในการวัดประสิทธิภาพในการจำแนก คือ ความสามารถในการทำนาย ซึ่งนิยมวัดด้วย ค่าความถูกต้อง (Accuracy) ค่าความแม่นยำ (Precision) ค่าระลึก (Recall) ค่าประสิทธิภาพโดยรวม (F-measure) ถ้าค่าดังกล่าวมีค่าสูง แสดงว่าตัวจำแนกมีประสิทธิภาพในการทำนายสูง ซึ่งค่าดังกล่าวสามารถคำนวณได้จากการพิจารณาค่าที่อยู่ในตารางเมทริกซ์ความสับสน (Confusion matrix) (ดังตารางที่ 6.1) โดยค่าที่อยู่ในตารางเมทริกซ์ความสับสนเป็นค่าที่แสดงผลลัพธ์ในการทำนายของแต่ละกลุ่มหรือคลาส

ตารางที่ 6.1 ตารางเมทริกซ์ความสับสน

		ผลการทำนาย			
		C_1	C_2	...	C_n
ค่าความจริง	C_1	p_{11}	p_{12}	...	p_{1n}
	C_2	p_{21}	p_{22}	...	p_{2n}
	...				
	C_n	p_{n1}	p_{n2}	...	p_{nn}

โดยที่ P_{ij} คือ จำนวนข้อมูลที่ทำนายว่าเป็นคลาส C_j แต่คำตอบจริงเป็นคลาส C_i

ตัวอย่างที่ 6.1 สมมติให้ข้อมูลมีทั้งหมด 3 คลาส ข้อมูลจริงในแต่ละคลาสมีทั้งหมด 10 ข้อมูล ผลการทำนายของตัวจำแนกแสดงในตารางเมทริกซ์ความสับสนดัง

ตารางที่ 6.2

ตารางที่ 6.2 ผลการทำนาย

		ผลการทำนาย			รวม
		C_1	C_2	C_3	
ค่าความจริง	C_1	p_{11} = 7	P_{12} = *1*	P_{13}= *2*	10
	C_2	p_{21} = 1	P_{22} = 8	P_{23}= 1	10
	C_3	P_{31} = 2	P_{32} = 3	P_{33} = 5	10

จากตารางที่ 6.2 ค่าแต่ละค่ามีความหมายดังนี้

p_{11} = 7 หมายความว่า ทำนายถูกต้องว่าเป็นคลาส C_1 จำนวน 7 ข้อมูล

p_{12} = 1 หมายความว่า ทำนายว่าเป็นคลาส C_2 จำนวน 1 ข้อมูล แต่จริงแล้วข้อมูลดังกล่าวเป็นคลาส C_1 (แสดงว่าทำนายผิด)

p_{13} = 2 หมายความว่า ทำนายว่าเป็นคลาส C_3 จำนวน 2 ข้อมูล แต่จริงแล้วข้อมูลดังกล่าวเป็นคลาส C_1 (แสดงว่าทำนายผิด)

p_{21} = 1 หมายความว่า ทำนายว่าเป็นคลาส C_1 จำนวน 1 ข้อมูล แต่จริงแล้วข้อมูลดังกล่าวเป็นคลาส C_2 (แสดงว่าทำนายผิด)

p_{22} = 8 หมายความว่า ทำนายถูกต้องว่าเป็นคลาส C_2 จำนวน 8 ข้อมูล

p_{23} = 1 หมายความว่า ทำนายว่าเป็นคลาส C_3 จำนวน 1 ข้อมูล แต่จริงแล้วข้อมูลดังกล่าวเป็นคลาส C_2 (แสดงว่าทำนายผิด)

p_{31} = 2 หมายความว่า ทำนายว่าเป็นคลาส C_1 จำนวน 2 ข้อมูล แต่จริงแล้วข้อมูลดังกล่าวเป็นคลาส C_3 (แสดงว่าทำนายผิด)

p_{32} = 3 หมายความว่า ทำนายว่าเป็นคลาส C_2 จำนวน 3 ข้อมูล แต่จริงแล้วข้อมูลดังกล่าวเป็นคลาส C_3 (แสดงว่าทำนายผิด)

p_{33} = 5 หมายความว่า ทำนายถูกต้องว่าเป็นคลาส C_3 จำนวน 5 ข้อมูล

นิยามที่ 6.1 ค่าความถูกต้องเป็นค่าที่บ่งบอกประสิทธิภาพในการทำนายโดยรวม สามารถคำนวณได้ดังสมการที่ (6.1) ซึ่งหาได้จากจำนวนข้อมูลที่ทำนายถูกต้องทั้งหมดหารด้วยจำนวนข้อมูลทั้งหมดที่ใช้ในการทดสอบ

$$AC = \frac{\sum_{i=1}^{n} p_{ii}}{N} \tag{6.1}$$

โดยที่ P_{ii} คือ จำนวนข้อมูลที่ทำนายถูกต้องว่าเป็นคลาส C_i

n คือ จำนวนคลาส

N คือ จำนวนข้อมูลทั้งหมดที่ใช้ในการทดสอบ

ตัวอย่างที่ 6.2 จากตารางที่ 6.2 ค่าความถูกต้องสามารถคำนวณได้ดังนี้

$$\text{AC} = \frac{p_{11} + p_{22} + p_{33}}{p_{11} + p_{12} + p_{13} + p_{21} + p_{22} + p_{23} + p_{31} + p_{32} + p_{33}}$$

$$\text{AC} = \frac{7 + 8 + 5}{7 + 1 + 2 + 1 + 8 + 1 + 2 + 3 + 5}$$

$$\text{AC} = \frac{20}{30} = 0.67$$

ดังนั้นสามารถสรุปได้ว่าค่าความถูกต้องในการทำนายของตัวจำแนก คือ 0.67 หรือ 67%

นิยามที่ 6.2 ค่าความแม่นยำเป็นค่าที่แสดงให้เห็นถึงความแม่นยำในการทำนายแต่ละคลาส พิจารณาจากอัตราส่วนข้อมูลที่ทำนายถูกต้องว่าเป็นคลาส C_i ต่อจำนวนข้อมูลที่ทำนายว่าเป็นคลาส C_i ค่าความแม่นยำในการทำนายแต่ละคลาสสามารถคำนวณดังสมการที่ (6.2)

$$precision_{C_i} = \frac{P_{ii}}{\sum_{i=1}^{n} p_{ij}} \tag{6.2}$$

โดยที่ P_{ii} คือ จำนวนข้อมูลที่ทำนายถูกต้องว่าเป็นคลาส C_i

P_{ij} คือ จำนวนข้อมูลทำนายว่าเป็นคลาส C_j แต่คำตอบจริงเป็นคลาส C_i

n คือ จำนวนคลาส

ตัวอย่างที่ 6.3 จากตารางที่ 6.2 ค่าความแม่นยำของแต่ละคลาสสามารถคำนวณได้ดังนี้

$$precision_{C_1} = \frac{p_{11}}{p_{11} + p_{21} + p_{31}}$$

$$precision_{C_1} = \frac{7}{7+1+2} = \frac{7}{10} = 0.70$$

$$precision_{C_2} = \frac{p_{22}}{p_{12} + p_{22} + p_{32}}$$

$$precision_{C_2} = \frac{8}{1+8+3} = \frac{8}{12} = 0.67$$

$$precision_{C_3} = \frac{p_{33}}{p_{13} + p_{23} + p_{33}}$$

$$precision_{C_3} = \frac{5}{2+1+5} = \frac{5}{8} = 0.63$$

สามารถสรุปได้ว่า ค่าความแม่นยำในการทำนายคลาส C_1 มีค่าเท่ากับ 0.70 หรือ 70%
ค่าความแม่นยำในการทำนายคลาส C_2 มีค่าเท่ากับ 0.67 หรือ 67%
ส่วนค่าความแม่นยำในการทำนายคลาส C_3 มีค่าเท่ากับ 0.63 หรือ 63%
แสดงให้เห็นว่าตัวจำแนกมีความแม่นยำในการทำนายคลาส C_1 มากที่สุด

นิยามที่ 6.3 ค่าระลึก หรือ ค่าความครบถ้วน แสดงถึงสามารถในการทำนายแต่ละคลาสว่ามีความถูกต้องเพียงใด โดยจะพิจารณาจากจำนวนข้อมูลที่ทำนายถูกต้องว่าเป็นคลาส C_i ต่อจำนวนข้อมูลจริงทั้งหมดที่เป็นคลาส C_i ค่าระลึกของแต่ละคลาสสามารถคำนวณดังสมการที่ (6.3)

$$recall_{C_i} = \frac{p_{ii}}{\sum_{j=1}^{n} p_{ij}} \tag{6.3}$$

โดยที่ P_{ii} คือ จำนวนข้อมูลที่ทำนายถูกต้องว่าเป็นคลาส C_i
P_{ij} คือ จำนวนข้อมูลทำนายว่าเป็นคลาส C_j แต่คำตอบจริงเป็นคลาส C_i
n คือ จำนวนคลาส

ตัวอย่างที่ 6.4 จากตารางที่ 6.2 ค่าระลึกของแต่ละคลาสสามารถคำนวณได้ดังนี้

$$recall_{C_1} = \frac{p_{11}}{p_{11} + p_{12} + p_{13}}$$

$$recall_{C_1} = \frac{7}{7+1+2} = \frac{7}{10} = 0.70$$

$$recall_{C_2} = \frac{p_{22}}{p_{21} + p_{22} + p_{23}}$$

$$recall_{C_2} = \frac{8}{1+8+1} = \frac{8}{10} = 0.80$$

$$recall_{C_3} = \frac{p_{22}}{p_{31} + p_{32} + p_{33}}$$

$$recall_{C_3} = \frac{5}{2+3+5} = \frac{5}{10} = 0.50$$

สามารถสรุปได้ว่า ค่าระลึกในการทำนายคลาส C_1 มีค่าเท่ากับ 0.70 หรือ 70%
ค่าระลึกในการทำนายคลาส C_2 มีค่าเท่ากับ 0.80 หรือ 80%
ส่วนค่าระลึกในการทำนายคลาส C_3 มีค่าเท่ากับ 0.50 หรือ 50%
แสดงให้เห็นว่าตัวจำแนกมีความสามารถในการทำนายคลาส C_2 ได้ดีที่สุด

นิยามที่ 6.4 ค่าประสิทธิภาพโดยรวมเป็นค่าที่แสดงภาพรวมของค่าความแม่นยำและความระลึกของแต่ละคลาส สามารถคำนวณได้ดังสมการที่ (6.4)

$$\text{F} - \text{Measure}_{C_i} = \frac{2\text{x Precision}_{C_i}\text{ x Recall}_{C_i}}{\text{Precision}_{C_i} + \text{Recall}_{C_i}} \tag{6.4}$$

ตัวอย่างที่ 6.5 จากตารางที่ 6.2 ค่าประสิทธิภาพโดยรวมของแต่ละคลาสสามารถคำนวณได้ดังนี้

$$\text{F} - \text{Measure}_{C_1} = \frac{2\text{x Precision}_{C_1}\text{ x Recall}_{C_1}}{\text{Precision}_{C_1} + \text{Recall}_{C_1}}$$

$$\text{F} - \text{Measure}_{C_1} = \frac{2\text{x } 0.7\text{x}0.7}{0.7 + 0.7} = 0.70$$

$$\text{F} - \text{Measure}_{C_2} = \frac{2\text{x Precision}_{C_2}\text{ x Recall}_{C_2}}{\text{Precision}_{C_2} + \text{Recall}_{C_2}}$$

$$\text{F} - \text{Measure}_{C_2} = \frac{2\text{x } 0.67\text{x}0.80}{0.67 + 0.80} = 0.73$$

$$\text{F}-\text{Measure}_{C_3} = \frac{2\text{x Precision}_{C_3}\text{ x Recall}_{C_3}}{\text{Precision}_{C_3} + \text{Recall}_{C_3}}$$

$$\text{F}-\text{Measure}_{C_3} = \frac{2\text{x } 0.63\text{x}0.50}{0.63 + 0.50} = 0.56$$

สามารถสรุปได้ว่า ค่าประสิทธิภาพโดยรวมในการทำนายคลาส C_1 มีค่าเท่ากับ 0.70 หรือ 70%
ค่าประสิทธิภาพโดยรวมในการทำนายคลาส C_2 มีค่าเท่ากับ 0.73 หรือ 73%
ส่วนค่าประสิทธิภาพโดยรวมในการทำนายคลาส C_3 มีค่าเท่ากับ 0.56 หรือ 56%
แสดงให้เห็นว่าตัวจำแนกมีประสิทธิภาพโดยรวมในการทำนายคลาส C_2 มากที่สุด

6.2 วิธีการจำแนกเชิงความสัมพันธ์

การสร้างกฎสำหรับจำแนกข้อมูลจะคล้ายกับการสร้างกฎความสัมพันธ์ แตกต่างกันตรงที่ฝั่งขวาของกฎเป็นคลาส ซึ่งในที่นี้จะขอเรียกว่า กฎความสัมพันธ์ระบุคลาส (Class association rule) โดยกฎที่ใช้ในการจำแนกอยู่ในรูปแบบของ r: $X \rightarrow c$ โดยที่ X คือ เซตรายการ และ c คือ คลาส ซึ่งกฎที่ได้เป็นรูปแบบที่มนุษย์สามารถเข้าใจได้ง่าย

เพื่อให้เข้าใจวิธีการจำแนกเชิงความสัมพันธ์จะขอนิยามคำที่เกี่ยวข้องดังต่อไปนี้

6.2.1 นิยามที่เกี่ยวข้อง

กำหนดให้ D เป็นชุดข้อมูลเรียนรู้ประกอบไปด้วยแอตทริบิวต์ทั้งหมด $A_1, A_2, ..., A_m$ และ $C=\{c_1, c_2, ..., c_n\}$ คือ เซตของคลาสที่ปรากฏใน D แต่ละรายการเปลี่ยนแปลง $d \in D$ ประกอบไปด้วยค่า a_{ij} ที่อยู่ในแอตทริบิวต์ A_i และมีคลาส c_j

ตัวอย่างที่ 6.6 จากตารางที่ 6.3 มีแอตทริบิวต์ทั้งหมด 3 แอตทริบิวต์ คือ A1, A2, A3 ซึ่งแต่ละแอตทริบิวต์ประกอบไปด้วยค่าต่อไปนี้

ค่า a1, a2, a3 เป็นค่าที่อยู่ในแอตทริบิวต์ A1

ค่า b1, b2, b3 เป็นค่าที่อยู่ในแอตทริบิวต์ A2

ค่า c1, c2 เป็นค่าที่อยู่ในแอตทริบิวต์ A3

คลาสประกอบไปด้วย 2 คลาส คือ Y และ N

ตารางที่ 6.3 ตัวอย่างข้อมูลเรียนรู้

รายการเปลี่ยนแปลง	A1	A2	A3	คลาส
1	a1	b1	c1	Y
2	a2	b1	c2	Y
3	a3	b1	c2	Y
4	a1	b2	c1	N
5	a1	b3	c1	N

นิยามที่ 6.5 รายการ คือ ค่า a_i ที่อยู่แอตทริบิวต์ A_i แทนค่าด้วย (A_i, a_i)

ตัวอย่างที่ 6.7 (A2, b1) เป็น รายการ โดยมีค่า b1 อยู่ในแอตทริบิวต์ A2

นิยามที่ 6.6 เซตรายการ *X* คือ เซตของรายการ แทนค่าด้วย $<(A_{i1}, a_{i1}), (A_{i2}, a_{i2}), ..., (A_{ik}, a_{ik})>$

ตัวอย่างที่ 6.8 <(A1,a1), (A3,c1)> เป็นเซตรายการ ซึ่งประกอบไปด้วย 2 รายการ คือ <(A1,a1) และ (A3,c1)>

นิยามที่ 6.7 กฎรายการ (ruleitem) คือ เซตรายการที่ปรากฏในคลาส *c* แทนด้วย <*X*, *c*> หรือสามารถเขียนให้อยู่ในรูปกฎ $r : X \rightarrow c$

ตัวอย่างที่ 6.9 กฎรายการ <(A2, b1), Y> หมายความว่า ค่า b1 ในแอตทริบิวต์ A2 ปรากฏในคลาส Y

นิยามที่ 6.8 ความยาวของกฎรายการ <*X*, *c*> คือ จำนวนรายการที่ปรากฏในเซตรายการ *X*

ตัวอย่างที่ 6.10 ความยาวของกฎรายการ <(A2, b1), Y> คือ 1 เนื่องจากมีรายการแค่รายการเดียว คือ (A2, b1) ส่วนความยาวของกฎรายการ <(A1,a1), (A3,c1), N> คือ 2 เนื่องจากมี 2 รายการ คือ <(A1,a1) และ (A3,c1)>

นิยามที่ 6.9 ค่าสนับสนุนของกฎรายการ <*X*, *c*> คือ จำนวนรายการเปลี่ยนแปลงที่ปรากฏกฎรายการ <*X*, *c*>

ตัวอย่างที่ 6.11 ค่าสนับสนุนของกฎรายการ <(A2, b1), Y> พิจารณาจากจำนวนรายการเปลี่ยนแปลงที่ปรากฏ b1 ในแอตทริบิวต์ A2 และอยู่ในคลาส Y ซึ่งปรากฏอยู่ในรายการเปลี่ยนแปลงที่ 1 2 และ 3 (ดังตารางที่ 6.3) ดังนั้นค่าสนับสนุนของกฎรายการ <(A2, b1), Y> คือ 3

นิยามที่ 6.10 กฎรายการความถี่ (Frequent ruleitemset) คือ กฎรายการที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำ

ตัวอย่างที่ 6.12 ถ้ากำหนดค่าสนับสนุนขั้นต่ำเท่ากับ 3 กฎรายการ (<(A2, b1), Y> ถือว่าเป็นกฎรายการความถี่เนื่องจากค่าสนับสนุนของกฎเท่ากับค่าสนับสนุนขั้นต่ำ

นิยามที่ 6.11 ค่าความเชื่อมั่นของกฎ $r: X \rightarrow c$ คือ ค่าที่แสดงให้เห็นถึงโอกาสการเกิด *X* แล้วจะเป็นคลาส *c* ค่าความเชื่อมั่นสามารถคำนวณได้จากสมการ (6.5)

$$conf(r) = \frac{supp(X \cup c)}{supp(X)} \times 100 \qquad (6.5)$$

นิยามที่ 6.12 กฎ (A2, b1)⟶ Y มีค่าความเชื่อมั่นเท่ากับ

supp(<(A2, b1), Y>)/*supp*((A2, b1)) * 100 = 3/3 *100 = 100%

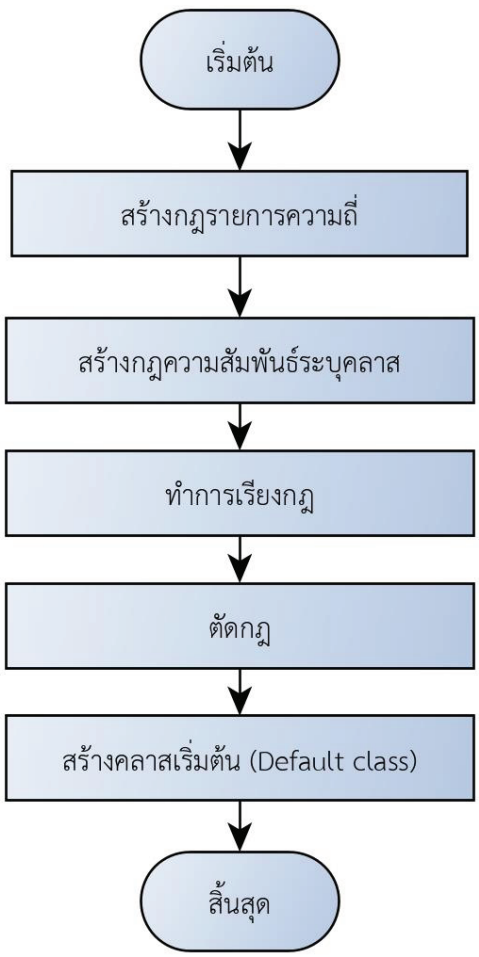
แสดงให้เห็นว่าถ้ามี b1 ในแอตทริบิวต์ A2 แล้วจะมีโอกาสเป็นคลาส Y ถึง 100%

นิยามที่ 6.13 กฎความสัมพันธ์ระบุคลาส $r: X \rightarrow c$ คือ กฎที่แสดงถึงความสัมพันธ์ของเซตรายการกับคลาส โดยกฎที่สามารถนำไปใช้ในการจำแนก คือ กฎที่มีค่าสนับสนุนมากกว่าหรือเท่ากับค่าสนับสนุนขั้นต่ำและค่าความเชื่อมั่นของกฎมีค่ามากกว่าหรือเท่ากับค่าความเชื่อมั่นขั้นต่ำ

ตัวอย่างที่ 6.13 กำหนดให้ค่าสนับสนุนขั้นต่ำเท่ากับ 2 และกำหนดให้ค่าความเชื่อมั่นขั้นต่ำเท่ากับ 60% กฎ (A2, b1)⟶ Y มีค่าสนับสนุนเท่ากับ 3 และมีค่าความเชื่อมั่นเท่ากับ 100% ดังนั้นกฎดังกล่าวถือว่าเป็นกฎความสัมพันธ์ระบุคลาส และสามารถนำไปใช้ในการจำแนก

6.2.2 การจำแนกเชิงความสัมพันธ์ด้วยขั้นตอนวิธี CBA

ปัจจุบันมีการนำเสนอขั้นตอนวิธีการสำหรับการจำแนกเชิงความสัมพันธ์หลายวิธี เช่น CBA, CMAR, CPAR, MMAC, MCAR, ACCF และ ACAC เป็นต้น ขั้นตอนวิธี CBA เป็นขั้นตอนวิธีแรกที่ถูกนำเสนอเพื่อการจำแนกเชิงความสัมพันธ์ ขั้นตอนวิธี CBA เป็นวิธีที่ง่ายและให้ประสิทธิภาพในการทำนายที่สูง ขั้นตอนการสร้างกฎความสัมพันธ์ระบุคลาสของขั้นตอนวิธี CBA ประกอบไปด้วย 5 ขั้นตอนหลัก (ดังรูปที่ 6.5) รายละเอียดแต่ละขั้นตอนสามารถอธิบายพร้อมยกตัวอย่างได้ดังนี้



รูปที่ 6.5 ขั้นตอนการสร้างตัวจำแนกของขั้นตอนวิธี CBA

ขั้นตอนที่ 1 ทำการสร้างกฎรายการความถี่ ซึ่งสามารถสร้างกฎรายการความถี่โดยใช้การทำเหมืองเซตรายการความถี่ ขั้นตอนวิธี CBA ทำการสร้างกฎรายการความถี่บนพื้นฐานของขั้นตอนวิธี Apriori

ตัวอย่างที่ 6.14 จากชุดข้อมูลตารางที่ 6.3 สามารถสร้างกฎรายการความถี่ได้ดังตารางที่ 6.4 ถ้ากำหนดค่าสนับสนุนขั้นต่ำเท่ากับ 2

ตารางที่ 6.4 กฎรายการความถี่

กฎรายการความถี่	ค่าสนับสนุน
<(A1, a1), N>	2
<(A2, b1), Y>	3
<(A3, c2), Y>	2
<(A3, c1), N>	2
<(A1, a1), (A3, c1) , N>	2
<(A2, b1), (A3, c2), Y>	2

ขั้นตอนที่ 2 ทำการสร้างกฎความสัมพันธ์ระบุคลาสจากกฎรายการความถี่ จากนั้นคำนวณหาค่าความเชื่อมั่นของแต่ละกฎ ทำการตัดกฎความสัมพันธ์ระบุคลาสที่มีค่าเชื่อมั่นน้อยกว่าค่าความเชื่อมั่นขั้นต่ำออก ถ้ามีกฎที่มีเซตรายการเหมือนกันแต่คลาสต่างกัน จะเลือกกฎที่มีค่าความเชื่อมั่นมากที่สุดไว้

ตัวอย่างที่ 6.15 ถ้ากำหนดความความเชื่อมั่นขั้นต่ำเท่ากับ 60% จากกฎรายการความถี่ในดังตารางที่ 6.4 สามารถสร้างกฎความสัมพันธ์ระบุคลาสและค่าความเชื่อมั่นได้ดังตารางที่ 6.5 กฎความสัมพันธ์ระบุคลาสทั้งหมดจะถูกนำไปพิจารณาเพื่อสร้างตัวจำแนก

ตารางที่ 6.5 กฎความสัมพันธ์ระบุคลาส

กฎความสัมพันธ์ระบุคลาส	ค่าสนับสนุน	ค่าความเชื่อมั่น
(A1, a1)⟶N	2	2/3 x 100 = 67%
(A2, b1) ⟶Y	3	3/3 x 100 = 100%
(A3, c2)⟶Y	2	2/2 x 100 = 100%
(A3, c1) ⟶ N	2	2/3 x 100 = 67%
(A1, a1), (A3, c1) ⟶N	2	2/3 x 100 = 67%
(A2, b1), (A3, c2) ⟶ Y	2	2/2 x 100 = 100%

ขั้นตอนที่ 3 ทำการเรียงกฎความสัมพันธ์ระบุคลาสทั้งหมด ซึ่งกฎจะถูกเรียงจากศักย์สูงไปหาต่ำตามเงื่อนไขดังต่อไปนี้

กำหนดให้ r_i และ r_j คือ กฎความสัมพันธ์ระบุคลาส กฎ r_i มีศักย์สูงกว่าหรือถูกเรียงก่อนกฎ r_j เมื่อตรงกับเงื่อนไขดังต่อไปนี้

1. $conf(r_i)>conf(r_j)$ (กฎที่ค่าความเชื่อมั่นมากกว่าจะมีศักย์สูงกว่า) หรือ

2. ถ้า $conf(r_i)=conf(r_j)$ แต่ $supp(r_i)>supp(r_j)$ (ถ้ากฎมีความเชื่อมั่นเท่ากัน กฎไหนที่มีค่าสนับสนุนมากกว่าจะมีศักย์สูงกว่า) หรือ

3. ถ้า $supp(r_i)=supp(r_j)$ แต่ $size(r_i)<size(r_j)$ (ถ้ากฎมีค่าสนับสนุนเท่ากัน กฎไหนที่มีความยาวกฎน้อยกว่าจะมีศักย์สูงกว่า) หรือ

4. ถ้า $size(r_i)= size(r_j)$ แต่ r_i ถูกสร้างก่อน r_j (ถ้ามีความยาวของกฎเท่ากัน กฎไหนถูกสร้างก่อนจะมีศักย์สูงกว่า)

ตัวอย่างที่ 6.16 จากตารางที่ 6.5 สามารถเรียงกฎความสัมพันธ์ระบุคลาสได้ดังตารางที่ 6.6 ซึ่งจะเห็นว่ามี 3 กฎ ที่มีค่าความเชื่อมั่นสูงสุด คือ (A2,b1) ⟶Y, (A3,c2)⟶Y และ (A2,b1), (A3,c2) ⟶ Y แต่กฎ (A2,b1) ⟶Y มีค่าสนับสนุนสูงสุดใน 3 กฎ ดังนั้น (A2,b1) ⟶Y จึงมีศักย์สูงที่สุด

ตารางที่ 6.6 กฎความสัมพันธ์ระบุคลาสที่เรียงแล้ว

ลำดับกฎ	กฎความสัมพันธ์ระบุคลาส	ค่าสนับสนุน	ค่าความเชื่อมั่น
1	(A2, b1) ⟶Y	3	100%
2	(A3, c2)⟶Y	2	100%
3	(A2, b1), (A3, c2) ⟶ Y	2	100%
4	(A1, a1)⟶N	2	67%
5	(A3, c1) ⟶ N	2	67%
6	(A1, a1), (A3, c1) ⟶N	2	67%

ขั้นตอนที่ 4 ทำการตัดกฎความสัมพันธ์ระบุคลาสที่ไม่สำคัญออกไป เนื่องจากกฎที่สร้างขึ้นมาอาจมีจำนวนมากและซ้ำซ้อน ส่งผลให้ตัวจำแนกมีขนาดใหญ่และไม่มีประสิทธิภาพ ในขั้นตอนวิธี CBA ใช้

วิธีการตัดกฎที่เรียกว่า การครอบคลุมฐานข้อมูล (Database coverage) โดยวิธีดังกล่าวมีหลักการทำงานดังนี้

- ทำการเปรียบเทียบกฎความสัมพันธ์ระบุคลาสที่เรียงแล้วกับชุดข้อมูลเรียนรู้ โดยจะทำการเปรียบเทียบทีละกฎตามลำดับ
- กฎที่ปรากฏในรายการเปลี่ยนแปลงใดๆ จะถูกเก็บไว้ในตัวจำแนก
- ข้อมูลรายการเปลี่ยนแปลงในชุดข้อมูลเรียนรู้ ที่ตรงกับกฎความสัมพันธ์ระบุคลาส จะไม่ถูกนำไปพิจารณากับกฎถัดไป
- ทำแบบนี้ซ้ำไปเรื่อยๆ จนกว่าชุดข้อมูลเรียนรู้ทั้งหมดถูกครอบคลุมด้วยกฎ หรือกฎถูกตรวจสอบทั้งหมด

ตัวอย่างที่ 6.17 ทำการตรวจสอบกฎที่ 1 (A2,b1) ⟶Y กับชุดข้อมูลเรียนรู้แล้วพบว่า กฎดังกล่าวตรงกับข้อมูลในรายการเปลี่ยนแปลงที่ 1 2 และ 3 (ดังรูปที่ 6.6) ดังนั้นกฎ (A2,b1) ⟶Y ถูกนำไปเก็บไว้ในตัวจำแนก และข้อมูลในรายการเปลี่ยนแปลงที่ 1 2 และ 3 จะไม่นำมาพิจารณาต่อไป เนื่องจากถูกครอบคลุมด้วยกฎ (A2,b1) ⟶Y แล้ว

ต่อมาพิจารณากฎ (A3,c2)⟶Y แล้วพบว่า ไม่ตรงกับข้อมูลในรายการเปลี่ยนแปลงที่เหลือเลย ดังนั้นกฎ (A3,c2)⟶Y จึงตัดกฎทิ้ง

จากนั้นทำการพิจารณากฎ (A2,b1), (A3,c2) ⟶ Y และพบว่าไม่ตรงกับข้อมูลในรายการเปลี่ยนแปลงที่เหลือเลย จึงตัดกฎทิ้ง

ต่อมาพิจารณากฎ (A1, a1)⟶N และพบว่าตรงกับข้อมูลในรายการเปลี่ยนแปลงที่ 4 และ 5 ดังนั้นเพิ่มกฎ (A1, a1)⟶N ในตัวจำแนก และไม่พิจารณาข้อมูลในรายการเปลี่ยนแปลงที่ 4 และ 5 ในการตัดกฎตัวถัดไป เมื่อข้อมูลในชุดข้อมูลเรียนรู้ถูกครอบคลุมหมดแล้วจึงหยุดการพิจารณา กฎที่เหลือจะถูกตัดทิ้งไป

ลำดับกฎ	กฎความสัมพันธ์ระบุคลาส
1	(A2,b1) →Y
2	(A3,c2)→Y
3	(A2,b1), (A3,c2) → Y
4	(A1,a1)→N
5	(A3,c1) → N
6	(A1,a1), (A3,c1) →N

รายการเปลี่ยนแปลง	A1	A2	A3	คลาส
1	a1	b1	c1	Y
2	a2	b1	c2	Y
3	a3	b1	c2	Y
4	a1	b2	c1	N
5	a1	b3	c1	N

รูปที่ 6.6 การครอบคลุมฐานข้อมูล

ดังนั้นสามารถสรุปได้ว่ากฎที่เพิ่มในตัวจำแนกมี 2 กฎ คือ (A2,b1) →Y และ (A1, a1)→N ซึ่งทั้ง 2 กฎครอบคลุมชุดข้อมูลเรียนรู้

ขั้นตอนที่ 5 ถ้าตรวจสอบทุกกฎกับชุดข้อมูลเรียนรู้แล้ว ปรากฏว่ามีข้อมูลที่ไม่มีกฎใดครอบคลุมได้ จะทำการสร้างคลาสเริ่มต้น (Default class) จากการพิจารณาข้อมูลดังกล่าว โดยคลาสเริ่มต้นจะเท่ากับคลาสที่มีจำนวนมากที่สุดในข้อมูลดังกล่าว เช่น สมมติข้อมูลในชุดข้อมูลเรียนรู้ที่ไม่ได้ถูกครอบคลุมมีทั้งหมด 5 รายการเปลี่ยนแปลง โดยมีคลาส Y ทั้งหมด 3 รายการเปลี่ยนแปลง และมีคลาส N ทั้งหมด 2 รายการเปลี่ยนแปลง คลาสเริ่มต้นจะเท่ากับ Y เนื่องจากคลาส Y มีจำนวนมากที่สุด เป็นต้น

ถ้าตรวจสอบทุกกฎกับชุดข้อมูลเรียนรู้แล้ว ปรากฏว่าชุดข้อมูลเรียนรู้ทั้งหมดถูกครอบคลุมโดยกฎ คลาสเริ่มต้นจะเท่ากับคลาสที่มีจำนวนมากที่สุดในชุดข้อมูลเรียนรู้

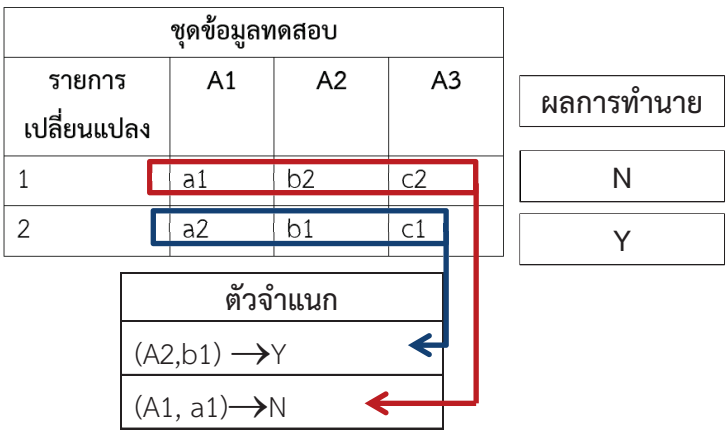
6.2.3 การทำนายคลาสด้วยกฎ

การจำแนกข้อมูลโดยใช้ตัวจำแนกที่สร้างขึ้น ทำได้โดยการเอาชุดข้อมูลทดสอบทีละรายการเปลี่ยนแปลง มาทำการตรวจสอบกับกฎความสัมพันธ์ระบุคลาสที่อยู่ในตัวจำแนกที่ละกฎตามลำดับ ถ้าข้อมูลทางด้านซ้ายของกฎเป็นเซตรายการย่อยของข้อมูลทดสอบ ใช้ข้อมูลทางด้านขวาของกฎนั้นเป็นผลการทำนาย เช่น กำหนดให้กฎความสัมพันธ์ระบุคลาสในตัวจำแนก คือ $X \longrightarrow c$ และ Y คือ เซตรายการในข้อมูลทดสอบ ถ้า $X \subseteq Y$ แล้วจะทำนายข้อมูลทดสอบว่าเป็นคลาส c เป็นต้น

ถ้าทำการตรวจสอบข้อมูลทดสอบกับทุกกฎแล้ว ปรากฏว่าไม่มีกฎใดเลยที่ตรงกับข้อมูลทดสอบจะใช้คลาสเริ่มต้นเป็นผลการทำนาย

ตัวอย่างที่ 6.18 สมมติชุดข้อมูลทดสอบมีทั้งหมด 2 รายการเปลี่ยนแปลง (ดังรูปที่ 6.7)

ข้อมูลทดสอบที่ 1 ตรวจสอบกับข้อมูลฝั่งซ้ายของกฎที่ 1 แล้วปรากฏว่าข้อมูลฝั่งซ้ายของกฎไม่ได้เป็นเซตรายการย่อยของข้อมูลทดสอบที่ 1 จึงนำข้อมูลทดสอบที่ 1 ไปตรวจสอบกับข้อมูลฝั่งซ้ายของกฎที่ 2 ปรากฏว่าข้อมูลฝั่งซ้ายของกฎเป็นเซตรายการย่อยของข้อมูลทดสอบที่ 1 คือ มีค่า a1 ที่อยู่ในแอตทริบิวต์ A1 ในข้อมูลทดสอบที่ 1 ดังนั้นข้อมูลทดสอบที่ 1 ทำนายว่าเป็นคลาส N (เอาฝั่งขวาของกฎที่ 2 มาทำนาย) ข้อมูลทดสอบที่ 2 ตรวจสอบกับข้อมูลฝั่งซ้ายของกฎที่ 1 แล้วปรากฏว่าข้อมูลฝั่งซ้ายของกฎเป็นเซตรายการย่อยของข้อมูลทดสอบที่ 2 ดังนั้นจึงนำข้อมูลฝั่งขวาของกฎที่ 1 เป็นผลการทำนาย ซึ่งก็คือ Y จึงสรุปได้ว่าข้อมูลทดสอบที่ 1 ทำนายว่าเป็นคลาส N ส่วนข้อมูลทดสอบที่ 2 ทำนายว่าเป็นคลาส Y



รูปที่ 6.7 การทำนายคลาสโดยใช้กฎ

6.3 การจำแนกเชิงความสัมพันธ์ด้วย Weka

การจำแนกเชิงความสัมพันธ์ด้วยขั้นตอนวิธี CBA สามารถเรียกใช้คลาส JCBA ซึ่งเป็นคลาสที่พัฒนาด้วยภาษาจาวา คลาส JCBA อยู่ในโปรแกรมสำเร็จ (Package) ที่ชื่อว่า classAssociationRules จำเป็นต้องติดตั้งเพิ่มเติมบน Weka โดยการติดตั้ง Weka และ classAssociationRules สามารถดูรายละเอียดได้ในภาคผนวก ข ส่วนการเตรียมข้อมูลสำหรับจำแนกเชิงความสัมพันธ์และตัวอย่างคำสั่งสำหรับการจำแนกเชิงความสัมพันธ์มีรายละเอียดดังหัวข้อย่อยต่อไปนี้

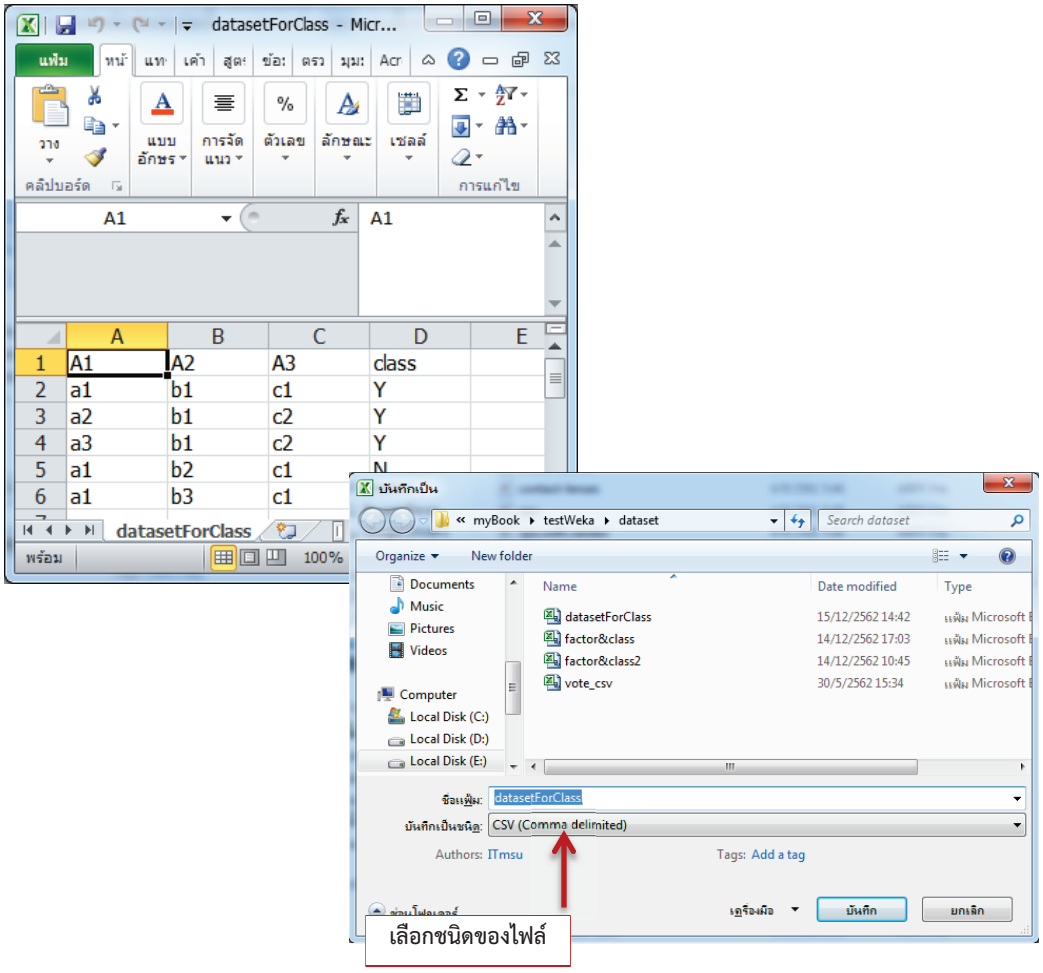
6.3.1 การเตรียมชุดข้อมูลนำเข้าสำหรับ Weka

ชุดข้อมูลนำเข้า Weka มี 2 รูปแบบ คือ ไฟล์ ARFF และไฟล์ CSV แต่ในหนังสือเล่มนี้แนะนำให้ใช้ไฟล์ CSV ซึ่งสามารถจัดเตรียมได้ง่าย โดยการเตรียมไฟล์ชุดข้อมูลมีเงื่อนไขดังนี้

- ไฟล์ CSV เป็นไฟล์ชุดข้อมูล ที่มีนามสกุล .csv
- แถวแรกต้องระบุชื่อแอตทริบิวต์ทั้งหมดและคลาส โดยแต่ละแอตทริบิวต์จะคั่นด้วยเครื่องหมายจุลภาค (,)
- แถวที่ 2 เป็นต้นไปเป็นส่วนของข้อมูล โดยข้อมูลในแต่ละแถว คือ ข้อมูลใน 1 รายการเปลี่ยนแปลง และแต่ละรายการเปลี่ยนแปลงจะประกอบไปด้วยค่าในแต่ละแอตทริบิวต์และค่าของคลาส ซึ่งค่าแต่ละค่าคั่นด้วยเครื่องหมายจุลภาค ดังตัวอย่างในรูปที่ 6.8

```
A1,A2,A3,class
a1,b1,c1,Y
a2,b1,c2,Y
a3,b1,c2,Y
a1,b2,c1,N
a1,b3,c1,N
```

รูปที่ 6.8 ตัวอย่างไฟล์ชุดข้อมูล CSV



รูปที่ 6.9 ไฟล์ .csv ในโปรแกรม Microsoft Excel

นอกจากนี้ยังสามารถเปิดและสร้างไฟล์ CSV โดยใช้โปรแกรม Microsoft excel ได้ โดยกำหนดให้แต่ละค่าอยู่ในเซลล์ จากนั้นทำการบันทึกไฟล์และเลือกชนิดของไฟล์เป็น CSV จากชุดข้อมูลเรียนรู้ในตารางที่ 6.3 สามารถเตรียมไฟล์ CSV ได้ดังรูปที่ 6.9

6.3.2 ตัวอย่างคำสั่งสำหรับการจำแนกเชิงความสัมพันธ์

ตัวอย่างคำสั่งสำหรับการจำแนกเชิงความสัมพันธ์ ประกอบไปด้วย 3 ตัวอย่างดังต่อไปนี้

ตัวอย่างคำสั่งที่ 6.1 เป็นคำสั่งสำหรับการสร้างกฎความสัมพันธ์ระบุคลาส และกฎที่ใช้ในการจำแนก ซึ่งมีรายละเอียดดังนี้

ตัวอย่างคำสั่งที่ 6.1

```
1.  package myWekaProject;
2.
3.  import weka.core.converters.CSVLoader;
4.  import weka.core.Instances;
5.  import java.io.File;
6.  import weka.associations.Apriori;
7.  import weka.classifiers.rules.car.JCBA;
8.
9.  public class CBAmine {
10
11.         public static void main(String args[]) throws Exception{
12.
13.           //load dataset
14.            CSVLoader loader = new CSVLoader();
15.            loader.setSource(new File("./dataset/datasetForClass.csv"));
16.            Instances dataset = loader.getDataSet();
17.
18.            dataset.setClassIndex(dataset.numAttributes()-1);
19.
20.             JCBA model = new JCBA();
21.             Apriori apriori = new Apriori();
22.             apriori.setMinMetric(0.6); // minimum confidence (60%)
23.             apriori.setLowerBoundMinSupport(0.4); //minimum support (40%)
24.             model.setCBA(true);
25.             model.setCarMiner(apriori);
26.             model.buildClassifier(dataset);
27.
28.              System.out.println(apriori.toString());
29.              System.out.println(model.toString());
30.         }
31. }
```

บรรทัดที่ 3-7 ทำการ import คลาสที่เกี่ยวข้อง โดยแต่ละคลาสมีรายละเอียดดังนี้

- CSVLoader เป็นคลาสสำหรับอ่านข้อมูลในไฟล์ CSV
- Instances เป็นคลาสสำหรับจัดการชุดข้อมูล
- File เป็นคลาสสำหรับจัดการไฟล์
- Apriori เป็นคลาสสำหรับขั้นตอนวิธี Apriori
- JCBA เป็นคลาสสำหรับขั้นตอนวิธี CBA

บรรทัดที่ 14-16 เป็นการโหลดชุดข้อมูล โดยในตัวอย่างนี้ใช้ไฟล์ datasetForClass.csv ซึ่งอยู่ในโฟลเดอร์ dataset (ดังรูปที่ 6.9)

บรรทัดที่ 18 เป็นการกำหนดตำแหน่งของแอตทริบิวต์คลาส โดยแอตทริบิวต์แรกอยู่ตำแหน่งที่ 0 จากไฟล์ datasetForClass.csv แอตทริบิวต์คลาสอยู่ที่ตำแหน่งที่ 3 หรือหาได้จากจำนวนแอตทริบิวต์ทั้งหมด – 1 ซึ่งก็คือ dataset.numAttributes()-1

บรรทัดที่ 20-21 เป็นการสร้างอ๊อบเจ็กต์ของคลาส JCBA และ Apriori

บรรทัดที่ 22 กำหนดค่าความเชื่อมั่นขั้นต่ำเท่ากับ 0.6 หรือ 60%

บรรทัดที่ 23 กำหนดค่าสนับสนุนขั้นต่ำเท่ากับ 0.4 หรือ 40%

บรรทัดที่ 24 กำหนดให้ประมวลผลโดยใช้ขั้นตอนวิธี CBA

บรรทัดที่ 25 กำหนดให้สร้างกฎความสัมพันธ์ระบุคลาสบนพื้นฐานของขั้นตอนวิธี Apriori

บรรทัดที่ 26 สร้างตัวจำแนก

บรรทัดที่ 28 แสดงรายละเอียดของการสร้างกฎความสัมพันธ์ระบุคลาส ดังแสดงในรูปที่ 6.10 ส่วนที่ 1

บรรทัดที่ 29 แสดงรายละเอียดของตัวจำแนกดังรูปที่ 6.10 ส่วนที่ 2 โดยจะแสดงข้อมูลดังนี้

- กฎที่อยู่ในตัวจำแนก (ในส่วนของ Classification Rules (ordered):)
- คลาสเริ่มต้น (ในส่วนของ Default Class:)

```
Apriori
=======
Minimum support: 0.4 (2 instances)
Minimum metric <confidence>: 0.6
Number of cycles performed: 12
Generated sets of large itemsets:
Size of set of large itemsets L(1): 4
Size of set of large itemsets L(2): 2

Best rules found:
 1. A2=b1 3 ==> class=Y 3    conf:(1)
 2. A3=c2 2 ==> class=Y 2    conf:(1)
 3. A2=b1 A3=c2 2 ==> class=Y 2    conf:(1)
 4. A1=a1 3 ==> class=N 2    conf:(0.67)
 5. A3=c1 3 ==> class=N 2    conf:(0.67)
 6. A1=a1 A3=c1 3 ==> class=N 2    conf:(0.67)

Classification Rules (ordered):
==========================
1.     A2=b1 1 0  ==> class=Y     conf:(1),  (3),
2.     A1=a1 0 0  ==> class=N     conf:(0.67),  (2),

Default Class: class=Y
Additional Information:
Number of Classification Associations Rules generated by Rule Miner: 6
Number of Classification Rules: 2

Mining Time in sec.: 0.004
Pruning Time in sec. : 0.002
```

1

2

รูปที่ 6.10 ผลลัพธ์จากการประมวลผล CBAmine.java

จากผลลัพธ์ในรูปที่ 6.10 จำนวนกฎความสัมพันธ์ระบุคลาสมีทั้งหมด 6 กฎ และกฎที่อยู่ในตัวจำแนกมี 2 กฎ คือ (A2,b1) ⟶Y และ (A1, a1)⟶N และคลาสเริ่มต้น คือ Y

ตัวอย่างคำสั่งที่ 6.2 เป็นคำสั่งสำหรับแสดงการประเมินประสิทธิภาพตัวจำแนกที่สร้างขึ้นจากวิธีการจำแนกเชิงความสัมพันธ์ โดยแบ่งข้อมูลแบบ hold out validation ซึ่งมีรายละเอียดดังต่อไปนี้

ตัวอย่างคำสั่งที่ 6.2

```
1.  package myWekaProject;
2.
3.  import weka.core.Debug;
4.  import weka.core.Instances;
5.  import weka.core.converters.CSVLoader;
6.  import java.io.File;
7.  import weka.associations.Apriori;
8.  import weka.classifiers.rules.car.JCBA;
9.  import weka.classifiers.Evaluation;
10
11. public class CBATest {
12.         public static void main(String args[]) throws Exception{
13.             CSVLoader loader = new CSVLoader();
14.             loader.setSource(new File("./dataset/vote_csv.csv"));
15.             Instances dataset = loader.getDataSet();
16.             dataset.setClassIndex(dataset.numAttributes()-1);
17.
18.             int trainSize = (int) Math.round(dataset.numInstances() * 0.8);
19.             int testSize = dataset.numInstances() - trainSize;
20.             dataset.randomize(new Debug.Random(1)); //random dataset
21.             Instances traindataset = new Instances(dataset, 0, trainSize);
22.             Instances testdataset = new Instances(dataset, trainSize, testSize);
23.
24.             JCBA model = new JCBA();
25.             Apriori apriori = new Apriori();
26.             apriori.setMinMetric(0.9); // minimum confidence (90%)
27.             apriori.setLowerBoundMinSupport(0.1); //minimum support (10%)
28.             model.setCBA(true);
29.             model.setCarMiner(apriori);
30.             model.buildClassifier(traindataset);
31.             Evaluation eval = new Evaluation(dataset);
32.             eval.evaluateModel(model, testdataset);
33.             System.out.println(eval.toSummaryString());
34.             System.out.println(eval.toMatrixString("Confusion matrix:"));
35.
36.             System.out.println("Precision of democrat: "+eval.precision(0)*100+" %");
37.             System.out.println("Recall of democrat: "+eval.recall(0)*100+" %");
38.             System.out.println("F-measure of democrat: "+eval.fMeasure(0)*100+" %");
39.             System.out.println("Precision of republican: "+eval.precision(1)*100+" %");
40.             System.out.println("Recall of republican: "+eval.recall(1)*100+" %");
41.             System.out.println("F-measure of republican: "+eval.fMeasure(1)*100+" %");
42.         }
43. }
```

บรรทัดที่ 3-9 เป็นการ import คลาสที่เกี่ยวข้อง

บรรทัดที่ 13-15 ทำการโหลดชุดข้อมูล ในตัวอย่างใช้ไฟล์ข้อมูล vote_csv.csv ซึ่งสามารถดาวน์โหลดได้ที่ https://datahub.io/machine-learning/vote จากนั้นนำมาเก็บไว้โฟลเดอร์ dataset

บรรทัดที่ 16 กำหนดตำแหน่งแอตทริบิวต์คลาส

บรรทัดที่ 18-19 กำหนดขนาดของชุดข้อมูลเรียนรู้ กับชุดข้อมูลทดสอบตามลำดับ โดยในตัวอย่างกำหนดให้ชุดข้อมูลเรียนรู้เท่ากับ 80% และชุดข้อมูลทดสอบ 20%

บรรทัดที่ 20 ทำการสุ่มข้อมูลเพื่อไม่ให้คลาสเหมือนกันเรียงต่อกัน

บรรทัดที่ 21-22 กำหนดให้ชุดข้อมูลเรียนรู้อยู่ในตัวแปร traindataset และชุดข้อมูลทดสอบอยู่ในตัวแปร testdataset

บรรทัดที่ 24-30 เป็นการสร้างตัวจำแนกโดยกำหนดค่าความเชื่อมั่นขั้นต่ำเท่ากับ 90% กำหนดค่าสนับสนุนขั้นต่ำเท่ากับ 10%

บรรทัดที่ 31-32 ประเมินตัวจำแนกกับชุดข้อมูลทดสอบ (testdataset)

บรรทัดที่ 33 แสดงผลการประเมินภาพรวม ดังแสดงในรูปที่ 6.11 ส่วนที่ 1

บรรทัดที่ 34 แสดงตารางเมทริกซ์ความสับสน ดังแสดงในรูปที่ 6.11 ส่วนที่ 2

บรรทัดที่ 36-41 เป็นการแสดงค่าความแม่นยำ ค่าระลึก และค่าประสิทธิภาพโดยรวมของแต่ละคลาสได้ดังรูปที่ 6.11 ส่วนที่ 3

```
Correctly Classified Instances          80               91.954  %
Incorrectly Classified Instances         7                8.046  %
Kappa statistic                          0.835
Mean absolute error                      0.0805
Root mean squared error                  0.2837
Relative absolute error                 16.8373 %
Root relative squared error             57.8168 %
Total Number of Instances               87

Confusion matrix:
  a  b   <-- classified as
 33  2 |  a = republican
  5 47 |  b = democrat

Precision of democrat: 86.8421052631579 %
Recall of democrat: 94.28571428571428 %
F-measure of democrat: 90.41095890410959 %
Precision of republican: 95.91836734693877 %
Recall of republican: 90.38461538461539 %
F-measure of republican: 93.06930693069307 %
```


รูปที่ 6.11 ผลลัพธ์ที่ได้จากการประมวลผล CBATest.java

ตัวอย่างคำสั่งที่ 6.3 เป็นคำสั่งแสดงการประเมินประสิทธิภาพตัวจำแนกที่สร้างขึ้นจากวิธีการจำแนกเชิงความสัมพันธ์ โดยแบ่งข้อมูลแบบ 10-fold cross validation ซึ่งมีรายละเอียดดังต่อไปนี้

ตัวอย่างคำสั่งที่ 6.3

```
1.  package myWekaProject;
2.
3.  import weka.core.Debug;
4.  import weka.core.Instances;
5.  import weka.core.converters.CSVLoader;
6.  import java.io.File;
7.  import weka.associations.Apriori;
8.  import weka.classifiers.rules.car.JCBA;
9.  import weka.classifiers.Evaluation;
10
11. public class CBATestWith10Fold {
12.         public static void main(String args[]) throws Exception{
13.             CSVLoader loader = new CSVLoader();
14.             loader.setSource(new File("./dataset/vote_csv.csv"));
15.            Instances dataset = loader.getDataSet();
16.
17.             dataset.setClassIndex(dataset.numAttributes()-1);
18.
19.             dataset.randomize(new Debug.Random(1)); //random dataset
20.
21.            double precisionC1 = 0;
22.            double recallC1 = 0;
23.            double fmeasureC1 = 0;
24.            double precisionC2 = 0;
25.            double recallC2 = 0;
26.            double fmeasureC2 = 0;
27.            double accuracy = 0;
28.
29.             int folds = 10;
30.            for(int n=0; n<folds; n++) {
31.                     Instances traindataset = dataset.trainCV(folds,  n); //training set
32.                     Instances testdataset = dataset.testCV(folds,  n); //test set
33.
34.                     // create classifier
35.                     JCBA model = new JCBA();
36.                     Apriori apriori = new Apriori();
37.                     apriori.setMinMetric(0.9); // minimum confidence 90%
38.                     apriori.setLowerBoundMinSupport(0.1); //minimum support 10%
39.                     model.setCBA(true);
40.                     model.setCarMiner(apriori);
41.                     model.buildClassifier(traindataset);
42.
43.                     Evaluation eval = new Evaluation(testdataset);
44.                     eval.evaluateModel(model, testdataset);
45.
46.                     accuracy=accuracy+eval.pctCorrect();
47                      precisionC1=precisionC1+eval.precision(0);
48.                     recallC1 = recallC1+eval.recall(0);
49.                     fmeasureC1 = fmeasureC1+eval.fMeasure(0);
```

```
50.                         precisionC2 = precisionC2+eval.precision(1);
51.                         recallC2 = recallC2+eval.recall(1);
52.                         fmeasureC2 = fmeasureC2+eval.fMeasure(1);
53.                 }
54.                 accuracy=accuracy/folds;
55.                 precisionC1=precisionC1/folds*100;
56.                 recallC1 = recallC1/folds*100;
57.                 fmeasureC1 = fmeasureC1/folds*100;
58.                 precisionC2 = precisionC2/folds*100;
59.                 recallC2 = recallC2/folds*100;
60.                 fmeasureC2 = fmeasureC2/folds*100;
61.
62.                 System.out.println("Accuracy "+accuracy+" %");
63.                 System.out.println("Precision of democrat: "+precisionC1+" %");
64.                 System.out.println("Recall of democrat: "+recallC1+" %");
65.                 System.out.println("F-measure of democrat: "+fmeasureC1+" %");
66.
67.                 System.out.println("Precision of republican: "+precisionC2+" %");
68.                 System.out.println("Recall of republican: "+recallC2+" %");
69.                 System.out.println("F-measure of republican: "+fmeasureC2+" %");
70.             }
71. }
```

บรรทัดที่ 3-9 ทำการ import คลาสที่เกี่ยวข้อง

บรรทัดที่ 13-15 ทำการโหลดชุดข้อมูล ในตัวอย่างใช้ไฟล์ข้อมูล vote_csv.csv ซึ่งอยู่ในโฟลเดอร์ dataset

บรรทัดที่ 17 ระบุตำแหน่งของแอตทริบิวต์คลาส

บรรทัดที่ 19 ทำการสุ่มข้อมูล

บรรทัดที่ 21-27 กำหนดค่าเริ่มต้นให้กับตัวแปรที่ใช้ในการวัดประสิทธิภาพ

บรรทัดที่ 29 กำหนดค่า k (folds) ซึ่งในที่นี้กำหนด k=10

บรรทัดที่ 31 กำหนดชุดข้อมูลเรียนรู้ในรอบที่ n เช่น trainCV(10, 0) หมายถึง ชุดข้อมูลเรียนรู้ในรอบที่ 1 ซึ่งก็คือ ชุดข้อมูลที่ 2-10

บรรทัดที่ 32 กำหนดชุดข้อมูลทดสอบในรอบ n เช่น trainCV(10, 0) หมายถึง ชุดข้อมูลทดสอบในรอบที่ 1 ซึ่งก็คือ ชุดข้อมูลที่ 1

บรรทัดที่ 35-41 เป็นการสร้างตัวจำแนก

บรรทัดที่ 43-44 เป็นการประเมินตัวจำแนก

บรรทัดที่ 46-52 เป็นการหาผลรวมของค่าที่ใช้ในการประเมินตัวจำแนก

บรรทัดที่ 54-60 เป็นการหาค่าเฉลี่ยแต่ละค่าที่ใช้ในการประเมินตัวจำแนก

บรรทัดที่ 62-69 เป็นการแสดงค่าที่ใช้ในการประเมินตัวจำแนกโดยผลลัพธ์ที่ได้แสดงดังรูปที่ 6.12

```
Accuracy 92.43128964059196 %
Precision of democrat: 92.14141414141415 %
Recall of democrat: 87.98463919980948 %
F-measure of democrat: 89.10658199911354 %
Precision of republican: 93.43283211504847 %
Recall of republican: 95.4537634408602 %
F-measure of republican: 94.08465113156188 %
```

รูปที่ 6.12 ผลลัพธ์ที่ได้จากการประมวลผล CBATestWith10Fold.java

บทสรุป

การจำแนกข้อมูลเป็นการทำนายกลุ่มให้กับข้อมูลที่ไม่ทราบกลุ่มมาก่อน โดยการทำนายกลุ่มได้จากการเรียนรู้ข้อมูลที่มีการกำหนดกลุ่มไว้แล้ว การจำแนกข้อมูลจำเป็นต้องมีการเตรียมข้อมูลที่ดี และมีการทดสอบประสิทธิภาพของตัวจำแนกก่อนนำไปใช้ โดยทำการแบ่งข้อมูลออกเป็นชุดข้อมูลเรียนรู้และชุดข้อมูลทดสอบ ชุดข้อมูลเรียนรู้ถูกนำไปใช้ในการสร้างตัวจำแนก ส่วนชุดข้อมูลทดสอบถูกนำไปใช้เพื่อทดสอบประสิทธิภาพของตัวจำแนก การวัดประสิทธิภาพของตัวจำแนกสามารถพิจารณาจากความสามารถในการทำนาย ซึ่งนิยมวัดด้วย ค่าความถูกต้อง ค่าความแม่นยำ ค่าระลึก และค่าประสิทธิภาพโดยรวม ถ้าค่าดังกล่าวมีค่าสูง แสดงว่าตัวจำแนกมีประสิทธิภาพในการทำนายสูง

การจำแนกเชิงความสัมพันธ์เป็นการผสมผสานระหว่างการทำเหมืองกฎความสัมพันธ์และการจำแนกข้อมูลเข้าด้วยกัน เป็นวิธีการจำแนกข้อมูลโดยใช้กฎที่ง่ายต่อความเข้าใจและมีความถูกต้องสูง โดยกฎที่ใช้ในการจำแนกอยู่ในรูปแบบของ $X \rightarrow c$ โดยที่ X คือ เซตรายการ และ c คือ คลาส

ขั้นตอนวิธี CBA เป็นขั้นตอนวิธีหนึ่งในการจำแนกเชิงความสัมพันธ์ เป็นวิธีที่ง่ายและให้ประสิทธิภาพในการจำแนกข้อมูลสูง โดยขั้นตอนวิธี CBA ประกอบไปด้วย การค้นหากฎรายการความถี่ การสร้างกฎความสัมพันธ์ระบุคลาส การเรียงกฎ การตัดกฎ และการสร้างคลาสเริ่มต้น

แบบฝึกหัดท้ายบท

1. กำหนดให้ชุดข้อมูลมีทั้งหมด 5,000 รายการเปลี่ยนแปลง จงอธิบายการแบ่งข้อมูลแบบ 10-fold cross validation และการแบ่งข้อมูลแบบ Hold-out validation ที่ประกอบไปด้วยชุดข้อมูลเรียนรู้ 70% และชุดข้อมูลทดสอบ 30%

2. จงแสดงค่าในตารางเมทริกซ์ความสับสน เมื่อกำหนดคลาสจริงและคลาสที่ได้จากการทำนายเป็นดังตารางต่อไปนี้

รายการเปลี่ยนแปลง	คลาสจริง	คลาสที่ได้จากการทำนาย
1	c_1	c_1
2	c_2	c_1
3	c_1	c_2
4	c_2	c_1
5	c_1	c_2
6	c_2	c_1
7	c_1	c_1
8	c_2	c_2
9	c_2	c_2
10	c_2	c_2

3. กำหนดให้ค่าที่อยู่ในตารางเมทริกซ์ความสับสนเป็นดังตารางต่อไปนี้ จงแสดงการคำนวณค่าความถูกต้อง และ ค่าความแม่นยำ ค่าระลึก ค่าประสิทธิภาพโดยรวมของแต่ละคลาส

		ผลการทำนาย	
		C_1	C_2
ค่าความจริง	C_1	1,000	50
	C_2	30	1,020

4. จงอธิบายความแตกต่างระหว่างกฎความสัมพันธ์ระบุคลาสและกฎความสัมพันธ์

5. จงอธิบายขั้นตอนการทำงานของขั้นตอนวิธี CBA โดยสังเขป

6. จากชุดข้อมูลเรียนรู้ต่อไปนี้ สามารถสร้างกฎที่ใช้ในการจำแนกได้ทั้งหมดกี่กฎ ด้วยขั้นตอนวิธี CBA แต่ละกฎมีค่าสนับสนุนและค่าความเชื่อมั่นเท่าไร เมื่อกำหนดค่าสนับสนุนขั้นต่ำแบบสัมพัทธ์เท่ากับ 20% และค่าความเชื่อมั่นขั้นต่ำเท่ากับ 80%

รายการเปลี่ยนแปลง	A1	A2	A3	A4	A5	A6	คลาส
1	a2	b1	c2	d1	e1	f1	y
2	a3	b1	c2	d2	e1	f2	y
3	a1	b1	c1	d3	e1	f2	n
4	a1	b3	c1	d1	e2	f1	y
5	a1	b1	c1	d1	e3	f1	n

7. กำหนดให้กฎความสัมพันธ์ระบุคลาสมีดังต่อไปนี้ จงเรียงกฎความสัมพันธ์ระบุคลาสดังกล่าวตามขั้นตอนการเรียงกฎของขั้นตอนวิธี CBA

กฎรายการความถี่	ค่าสนับสนุนแบบสัมพัทธ์	ค่าความเชื่อมั่น
(P, p1) (Q, q2) (R, r2)⟶ c1	75%	60%
(R, r2)⟶ c2	40%	100%
(Q, q1) (R, r2)⟶ c1	40%	80%
(Q, q1)⟶ c2	40%	67%
(P, p1) (Q, q1) ⟶ c1	50%	100%
(P, p2) (Q, q2) ⟶ c2	40%	80%

8. จงอธิบายวิธีการตัดกฎด้วยขั้นตอนวิธี CBA พร้อมกับยกตัวอย่าง

9. จงเขียนโปรแกรมเพื่อวัดประสิทธิภาพตัวจำแนกที่สร้างขึ้นจากขั้นตอนวิธี CBA โดยมีเงื่อนไขต่อไปนี้

   - ใช้ชุดข้อมูล vote_csv.csv
   - กำหนดให้แบ่งข้อมูลเป็นแบบ 5-fold cross validation
   - กำหนดค่าสนับสนุนขั้นต่ำแบบสัมพัทธ์เท่ากับ 20% และค่าความเชื่อมั่นเท่ากับ 70%
   - แสดงค่าความถูกต้องเฉลี่ย

- ค่าความแม่นยำเฉลี่ย ค่าระลึกเฉลี่ย และค่าประสิทธิภาพโดยรวมเฉลี่ยของแต่ละคลาส

10. กำหนดให้ชุดข้อมูลทดสอบเป็นดังตารางข้างล่าง และกฎที่อยู่ในตัวจำแนกประกอบไปด้วย 2 กฎ คือ รวย, โง่→แต่ง และ ฉลาด→ไม่แต่ง และคลาสเริ่มต้น คือ ไม่แต่ง จงทำนายว่าแต่ละบุคคลที่อยู่ในชุดข้อมูลทดสอบจะได้แต่งงานหรือไม่

คนที่	ฐานะ	สมอง	รูปลักษณ์	การศึกษา
1	รวย	โง่	สวย	ตรี
2	จน	ฉลาด	พอใช้	ตรี
3	รวย	ฉลาด	สวย	โท
4	รวย	ฉลาด	สวย	โท
5	จน	โง่	พอใช้	ตรี

บทที่ 7
การประยุกต์ใช้การทำเหมืองรูปแบบ
(Pattern Mining Application)

ในบทนี้จะขอยกตัวอย่างการประยุกต์ใช้เหมืองรูปแบบในด้านต่างๆ โดยจะยกตัวอย่างการหาความสัมพันธ์ของหมวดหมู่เพจบนเฟสบุ๊กด้วยการทำเหมืองกฎความสัมพันธ์ การระบุผู้มีอิทธิพลด้วยการทำเหมืองกฎความสัมพันธ์เชิงลำดับ จำแนกโรคหลอดเลือดสมองด้วยการจำแนกเชิงความสัมพันธ์ โดยในแต่ละส่วนจะกล่าวถึง การรวบรวมข้อมูล การเตรียมข้อมูล และการใช้ SPMF และ Weka เพื่อค้นหารูปแบบที่น่าสนใจ

7.1 การค้นหาความสัมพันธ์ของหมวดหมู่เพจ

เพจเป็นพื้นที่ที่ผู้ใช้สร้างขึ้นมาบนเฟสบุ๊ก เพื่อนำเสนอสิ่งต่างๆ แก่ผู้ใช้คนอื่น เช่น สินค้า บริการ แบรนด์ ภาพลักษณ์องค์กร เป็นต้น ปัจจุบันเพจบนเฟสบุ๊กถูกนำมาใช้อย่างแพร่หลายในหลายด้าน โดยเฉพาะด้านธุรกิจ ซึ่งสามารถประชาสัมพันธ์สินค้า บริการ สื่อสารกับผู้ที่สนใจได้อย่างง่ายดาย นอกจากนี้เพจบนเฟสบุ๊คสามารถใช้งานได้ฟรี จึงทำให้จำนวนเพจบนเฟสบุ๊คเพิ่มขึ้นมากในปัจจุบัน ผู้ใช้สามารถกดถูกใจเพื่อรับรู้ข่าวสารต่างๆ ที่เกี่ยวกับเพจ ส่วนเจ้าของเพจสามารถประชาสัมพันธ์ข่าวสารต่างๆ ไปยังผู้ที่สนใจได้ ดังนั้นเจ้าของเพจจึงพยายามโฆษณาเพจในเฟสบุ๊คเพื่อให้ผู้ใช้กดถูกใจเพจของตัวเอง โดยเจ้าของเพจสามารถกำหนดคุณลักษณะของผู้ใช้ที่ต้องการโฆษณาได้ เช่น กำหนดให้แสดงเพจบนเฟสบุ๊คของผู้ใช้ที่เป็นผู้หญิง มีอายุระหว่าง 20-30 ปี เป็นต้น ซึ่งการกำหนดคุณลักษณะของผู้ใช้อาจจะไม่ได้ส่งเสริมให้ผู้ใช้กดถูกใจเพจ และอาจจะทำผู้ใช้เกิดความรำคาญ เนื่องจากเป็นการโฆษณาเพจที่ตัวเองไม่ได้สนใจจริงๆ

การหาความสัมพันธ์หมวดหมู่ของเพจในเฟสบุ๊ค จะทำให้ได้กฎที่สามารถใช้แนะนำหมวดหมู่ของเพจที่ตรงกับความสนใจของผู้ใช้ได้ เช่น ผู้ใช้กดถูกใจเพจหมวดหมู่การบริการ มักจะกดถูกใจเพจหมวดหมู่กีฬา เป็นต้น การหาความสัมพันธ์หมวดหมู่เพจสามารถประยุกต์ใช้การทำเหมืองกฎความสัมพันธ์ โดยขั้นตอนการค้นหากฎความสัมพันธ์ของหมวดหมู่เพจมีดังต่อไปนี้

7.1.1 การรวบรวมข้อมูล

ข้อมูลที่ใช้ในการสร้างกฎความสัมพันธ์เป็นหมวดหมู่ของเพจที่ผู้ใช้กดถูกใจ โดยดึงข้อมูลจากผู้ใช้จำนวน 1,780 คน ในช่วงวันที่ 25 เมษายน – 30 เมษายน พ.ศ. 2560 โดยใช้เฟสบุ๊กกราฟเอพีไอ (Facebook Graph API) ทำการคัดเลือกเฉพาะหมวดหมู่ที่ผู้ใช้กดถูกใจจำนวน 10 ครั้งขึ้นไปเพื่อแสดง

ให้เห็นว่าผู้ใช้มีความสนใจในหมวดหมู่ดังกล่าวจริง เช่น ผู้ใช้กดถูกใจเพจทั้งหมด 10 เพจ ซึ่งเพจดังกล่าวอยู่ในหมวดหมู่กีฬา แสดงว่าผู้ใช้สนใจกีฬาจริง เป็นต้น ข้อมูลตัวอย่างแสดงได้ดังตารางที่ 7.1

ตารางที่ 7.1 ตัวอย่างข้อมูลที่ดึงได้จากเฟสบุ๊ก

ผู้ใช้	หมวดหมู่ที่กดถูกใจ
1	กีฬา, การขนส่ง, การช้อปปิ้ง/ขายปลีก
2	ทนายความ, การช้อปปิ้ง/ขายปลีก
3	ธุรกิจท้องถิ่น, การขนส่ง, กีฬา
4	การขนส่ง, ช๊อปปิ้ง/ขายปลีก

7.1.2 การเตรียมข้อมูล

การหากฎความสัมพันธ์ของหมวดหมู่เพจโดยประยุกต์ใช้การทำเหมืองกฎความสัมพันธ์ จะต้องแปลงข้อมูลให้อยู่ในรูปรายการเปลี่ยนแปลง โดยมีรายละเอียดดังต่อไปนี้

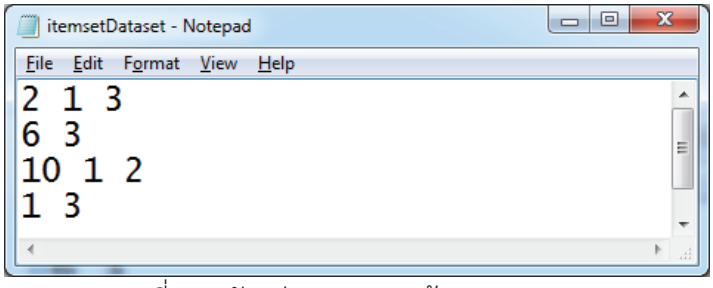
- 1 รายการเปลี่ยนแปลง หมายถึง ข้อมูลการกดถูกใจหมวดหมู่เพจของผู้ใช้หนึ่งคน
- 1 หมวดหมู่ คือ 1 รายการ
- แปลงหมวดหมู่แต่ละหมวดหมู่ให้อยู่ในรูปแบบตัวเลขจำนวนเต็ม (เหมือนการเตรียมชุดข้อมูลเซตรายการ) เพื่อให้สามารถใช้เทคนิคการทำเหมืองกฎความสัมพันธ์ที่อยู่ใน SPMF ได้
- ข้อมูลที่แปลงเรียบร้อยแล้ว จะถูกจัดเก็บลงในโปรแกรม Notepad และทำการบันทึกไฟล์ชื่อ itemsetDataset.txt

โดยในตัวอย่างนี้จะแปลงข้อมูลการกดถูกใจหมวดหมู่เพจเป็นตัวเลขตามตารางที่ 7.2 และสามารถแปลงตัวอย่างข้อมูลในตารางที่ 7.1 ได้ดังรูปที่ 7.1

ตารางที่ 7.2 การแปลงหมวดหมู่

ลำดับ	ชื่อหมวดหมู่	ลำดับ	ชื่อหมวดหมู่
1	การขนส่ง	39	อวกาศ/กองทัพ
2	กีฬา	40	อุตสาหกรรม
3	ช้อปปิ้ง/ขายปลีก	41	ชุมชน
4	ชุมชน/รัฐบาล,สมาคม	42	กระเป๋า/สัมภาระ
5	ที่อยู่	43	กล้อง/รูปภาพ
6	ทนายความ	44	เกมส์
7	ทัวร์/การเที่ยวชม	45	คอมพิวเตอร์/เทคโนโลยี
8	ท่าอากาศยาน	46	เครื่องใช้/อุปกรณ์
9	ธนาคาร/บริการทางการเงิน	47	เครื่องแต่งกาย,เครื่องประดับ/นาฬิกา
10	ธุรกิจท้องถิ่น	48	มือถือ/แท็บเล็ต
11	การบริการ	49	วัสดุก่อสร้าง
12	โบสถ์/องค์กรทางศาสนา	50	เครื่องดื่ม
13	พิพิธภัณฑ์/ห้องแสดงศิลปะ	51	สินค้าและบริการ
14	หนังสือ	52	สินค้าสำหรับทารกและเด็ก
15	อาหาร	53	อัลบั้ม
16	บันเทิง	54	ศิลปะการแสดง
17	สถานที่	55	สตูดิโอ
18	กฎหมาย	56	บุคคลสาธารณะ
19	การสื่อสารคมนาคม	57	ผู้จัดทำเว็บบล็อก
20	พลังงาน/สารเคมี	58	ผู้ประกอบการ,ผู้ประกอบการธุรกิจ
21	สถาบันการศึกษา	59	ผู้ผลิต
22	ท่องเที่ยว/พักผ่อน	60	นักแสดง
23	เทคโนโลยีชีวภาพ	61	ศิลปิน
24	เว็บไซต์	62	สัตว์เลี้ยง
25	บริษัท	63	โทรทัศน์/ภาพยนตร์

ลำดับ	ชื่อหมวดหมู่	ลำดับ	ชื่อหมวดหมู่
26	อาหารและเครื่องดื่ม	64	คอนเสิร์ต
27	นักธุรกิจ	65	นักดนตรี/วงดนตรี
28	ฟาร์ม/การเกษตร	66	ห้องครัว/การทำอาหาร
29	รถยนต์และอะไหล่	67	เพลง,ค่ายเพลง,อันดับเพลง
30	วิศวกรรม/การก่อสร้าง	67	อิเล็กทรอนิกส์
31	สาเหตุ	68	นิตยสาร
32	สินค้าปลีกและสินค้าบริโภค	70	อุปกรณ์ในเชิงพาณิชย์
33	สื่อ/ข่าวสาร/สิ่งพิมพ์	71	สวน/บ้าน
34	สุขภาพ/ความงาม	72	อาชีพ
35	หน่วยงานราชการ	73	นักการเมือง
36	เหมือง/แร่	74	นักเรียน
37	เพจของแอพ	75	สถานีวิทยุ
38	องค์กร		



รูปที่ 7.1 ตัวอย่างการแปลงข้อมูล

จากข้อมูลทั้งหมดที่รวบรวม เมื่อแปลงข้อมูลเรียบร้อยแล้ว จะได้ชุดข้อมูลที่มีรายการเปลี่ยนแปลงทั้งหมด 1,780 รายการเปลี่ยนแปลง ชุดข้อมูลดังกล่าวจะถูกนำไปประมวลผลเพื่อหากฎความสัมพันธ์ของหมวดหมู่เพจด้วย SPMF ต่อไป

7.1.3 การค้นหากฎความสัมพันธ์ของหมวดหมู่เพจด้วย FP-Growth

ในตัวอย่างนี้จะประยุกต์ใช้ขั้นตอนวิธี FP-Growth ที่อยู่ใน SPMF เพื่อค้นหากฎความสัมพันธ์ของหมวดหมู่เพจ โดยกำหนดให้ค่าสนับสนุนขั้นต่ำเท่ากับ 10% และค่าความเชื่อมั่นขั้นต่ำเท่ากับ 90% คำสั่งสำหรับการค้นหากฎความสัมพันธ์แสดงในตัวอย่างคำสั่งที่ 7.1 และผลลัพธ์ที่ได้จากการ

ประมวลผลแสดงดังรูปที่ 7.2 และจำนวนกฎที่สร้างได้ถูกบันทึกลงในไฟล์ output.txt ดังรูปที่ 7.3 ซึ่งแสดงกฎความสัมพันธ์ของหมวดหมู่เพจจำนวน 413 กฎ

ตัวอย่างคำสั่งที่ 7.1

```
1.  package mySpmfProject;
2.
3.  import java.io.IOException;
4.  import ca.pfv.spmf.algorithms.associationrules.agrawal94_association_rules.AlgoAgrawalFaster94;
    import ca.pfv.spmf.algorithms.frequentpatterns.fpgrowth.AlgoFPGrowth;
5.  import ca.pfv.spmf.patterns.itemset_array_integers_with_count.Itemsets;
6.
7.  public class FacebookPageGen {
8.          public static void main(String [] arg) throws IOException{
9.                  String input = ".//dataset/itemsetDataset.txt";
10                 String output = ".//output.txt";
11.
12.                double minsupp = 0.1;
13.                AlgoFPGrowth fpgrowth = new AlgoFPGrowth();
14.                Itemsets patterns = fpgrowth.runAlgorithm(input, null, minsupp);
15.                fpgrowth.printStats();
16.                int databaseSize = fpgrowth.getDatabaseSize();
17.                System.out.println(databaseSize);
18.
19.                double  minconf = 0.90;
20.                AlgoAgrawalFaster94 algoAgrawal = new AlgoAgrawalFaster94();
21.                algoAgrawal.runAlgorithm(patterns, output, databaseSize, minconf);
22.                algoAgrawal.printStats();
23.         }
24. }
```

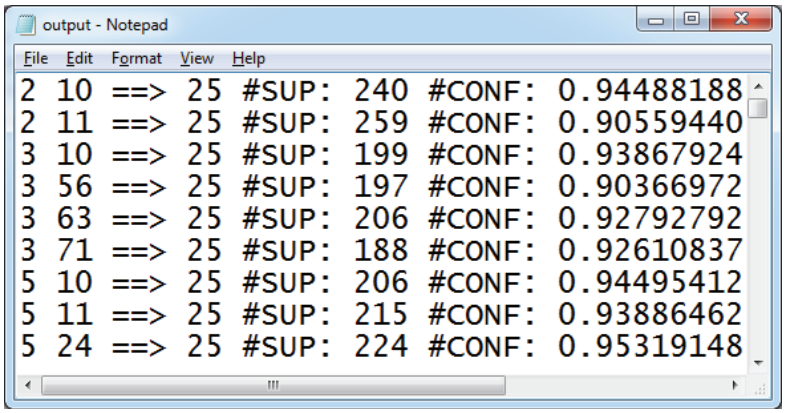
```
=============  FP-GROWTH 0.96r19 - STATS =============
 Transactions count from database : 1780
 Max memory usage: 10.962104797363281 mb
 Frequent itemsets count : 1282
 Total time ~ 142 ms
===================================================
1780
=============  ASSOCIATION RULE GENERATION v2.19- STATS =============
 Number of association rules generated : 413
 Total time ~ 19 ms
```

รูปที่ 7.2 ผลลัพธ์จากการประมวลผล FacebookPageGen.java

จากรูปที่ 7.2 สามารถแปลผลลัพธ์ที่ได้จากการประมวลผลได้ดังนี้

- จำนวนรายการเปลี่ยนแปลงในชุดข้อมูล คือ 1,780 รายการเปลี่ยนแปลง

- หน่วยความจำที่ใช้ในการประมวลผล คือ 10.962104797363281 เมกะไบต์
- เซตรายการความถี่มีทั้งหมด 1,282 เซตรายการ
- เวลาในการสร้างเซตรายการ คือ 142 มิลลิวินาที
- จำนวนกฎความสัมพันธ์ที่สร้างได้มีทั้งหมด 413 กฎ
- เวลาในการสร้างกฎความสัมพันธ์ เท่ากับ 19 มิลลิวินาที



รูปที่ 7.3 ไฟล์ผลลัพธ์จากการประมวลผล FacebookPageGen.java

เมื่อได้กฎความสัมพันธ์ของหมวดหมู่เพจแล้วดังรูปที่ 7.3 ทำการแปลงกฎที่เป็นชื่อหมวดหมู่ เพื่อนำกฎดังกล่าวไปใช้ประโยชน์ โดยจะขอยกตัวอย่างการแปลความหมาย 5 กฎแรกในรูปที่ 7.3 ได้ดังนี้

กฎที่ 1: 2 10 ⟶ 25 หมายถึง ผู้ใช้ที่สนใจเพจหมวดหมู่กีฬาและหมวดหมู่ธุรกิจท้องถิ่น มีโอกาสสนใจเพจหมวดหมู่บริษัท ด้วยความเชื่อมั่น 94% และมีค่าสนับสนุนแบบสัมบูรณ์เท่ากับ 240

กฎที่ 2: 2 11 ⟶ 25 หมายถึง ผู้ใช้ที่สนใจเพจหมวดหมู่กีฬาและการบริการ มีโอกาสสนใจเพจหมวดหมู่บริษัท ด้วยความเชื่อมั่น 90% และมีค่าสนับสนุนแบบสัมบูรณ์เท่ากับ 259

กฎที่ 3: 3 10 ⟶ 25 หมายถึง ผู้ใช้ที่สนใจเพจหมวดหมู่ช้อปปิ้ง/ขายปลีกและหมวดหมู่ธุรกิจท้องถิ่น มีโอกาสสนใจเพจหมวดหมู่บริษัท ด้วยความเชื่อมั่น 93% และมีค่าสนับสนุนแบบสัมบูรณ์เท่ากับ 199

กฎที่ 4: 3 56 ⟶ 25 หมายถึง ผู้ใช้ที่สนใจเพจหมวดหมู่ช้อปปิ้ง/ขายปลีกและหมวดหมู่บุคคลสาธารณะ มีโอกาสสนใจเพจหมวดหมู่บริษัท ด้วยความเชื่อมั่น 90% และมีค่าสนับสนุนแบบสัมบูรณ์เท่ากับ 197

กฎที่ 5: 3 63 ⟶ 25 หมายถึง ผู้ใช้ที่สนใจเพจหมวดหมู่ช้อปปิ้ง/ขายปลีกและหมวดหมู่โทรทัศน์/ภาพยนตร์ มีโอกาสสนใจเพจหมวดหมู่บริษัท ด้วยความเชื่อมั่น 92% และมีค่าสนับสนุนแบบสัมบูรณ์เท่ากับ 206

กฎความสัมพันธ์ที่ได้สามารถนำไปใช้ประโยชน์ในหลายด้าน เช่น เจ้าของเพจสามารถใช้กฎความสัมพันธ์เพื่อวางแผนประชาสัมพันธ์เพจตัวเองให้คนกดถูกใจมากขึ้น และสามารถรู้ถึงพฤติกรรมของผู้ใช้ ทำให้สามารถนำเสนอเพจได้ตรงกับความต้องการของผู้ใช้ เป็นต้น

7.2 การระบุผู้มีอิทธิพล

ปัจจุบันมีการศึกษาการระบุผู้มีอิทธิพลอย่างกว้างขวางในหลายด้าน เช่น ด้านการตลาด ด้านการเมือง ด้านการศึกษา ด้านการแพทย์ เป็นต้น ถ้าสามารถชักจูงให้ผู้มีอิทธิพลแสดงพฤติกรรมใดๆ ออกมา ก็จะทำให้ผู้ที่ถูกครอบงำแสดงพฤติกรรมดังกล่าวออกมาด้วย เช่น นางสาวสุดใจมักจะซื้อกระเป๋ายี่ห้อเดียวกับดาราที่เขาชื่นชอบ ซึ่งแสดงให้เห็นว่าดาราคนดังกล่าวมีอิทธิพลต่อนางสาวสุดใจ เป็นต้น

ปัจจุบันเครือข่ายสังคมออนไลน์เข้ามามีบทบาทในชีวิตประจำวัน ประชาชนใช้เครือข่ายออนไลน์ในการติดต่อสื่อสารกันมากขึ้น เนื่องจากสามารถพูดคุย แบ่งปัน แลกเปลี่ยนแนวคิด ความรู้ หรือเรื่องที่ตัวเองสนใจได้อย่างอิสระ ทำให้พฤติกรรมบางอย่างถูกแสดงออกมาบนเครือข่ายออนไลน์ การศึกษาอิทธิพลโดยใช้ข้อมูลบนเครือข่ายสังคมออนไลน์ จึงเป็นเรื่องที่น่าสนใจ สามารถพิจารณาการมีอิทธิพลจากพฤติกรรมต่างๆ ที่อยู่บนเครือข่ายสังคมออนไลน์ได้ เช่น เมื่อนายสมศักดิ์แสดงความคิดเห็นใดๆ ก็ตาม นางสาวสมหญิงมักจะแสดงความคิดเห็นตามเสมอ ซึ่งแสดงให้เห็นว่า นายสมศักดิ์มีอิทธิพลต่อนางสาวสมหญิง เป็นต้น

การระบุผู้มีอิทธิพลโดยใช้ข้อมูลการแสดงความคิดเห็น มีพื้นฐานแนวความคิดที่ว่า ผู้มีอิทธิพลโพสต์หรือแสดงความคิดเห็นในหัวข้อใดๆ ก็ตาม คนที่ถูกครอบงำจะแสดงความคิดเห็นในหัวข้อนั้นตาม ซึ่งจะเห็นได้ว่าการแสดงความคิดเห็นเกิดขึ้นตามลำดับ ดังนั้นการพิจารณากลุ่มผู้มีอิทธิพลจะต้องพิจารณาลำดับเหตุการณ์ที่เกิดขึ้นด้วย โดยในตัวอย่างนี้จะประยุกต์ใช้การทำเหมืองกฎความสัมพันธ์เชิง

ลำดับ เพื่อแสดงให้เห็นกฎความสัมพันธ์ของกลุ่มคนที่มีอิทธิพลและกลุ่มคนที่ถูกครอบงำ โดยขั้นตอนการค้นหาความสัมพันธ์ของกลุ่มคนที่มีอิทธิพลและกลุ่มคนที่ถูกครอบงำมีดังนี้

7.2.1 การเก็บรวบรวมข้อมูล

ข้อมูลที่ใช้ในการระบุผู้มีอิทธิพลเป็นข้อมูลแสดงความคิดเห็นที่อยู่บนกลุ่มเฟสบุ๊ก กลุ่มที่นำมาใช้เป็นตัวอย่าง เป็นกลุ่มที่เกี่ยวกับการเมือง มีจำนวนสมาชิก 57,090 คน โดยเก็บรวบรวมรหัสผู้ใช้งานและการกระทำ ที่มีการแสดงความคิดเห็นในหัวข้อต่างๆ ตั้งแต่วันที่ 6 มิถุนายน พ.ศ 2555 ถึง 23 เมษายน พ.ศ. 2558 ใช้เฟสบุ๊กกราฟเอพีไอในการดึงข้อมูลในเฟสบุ๊ก (ตัวอย่างที่ได้จากการเก็บรวบรวมแสดงได้ดังตารางที่ 7.3)

ตารางที่ 7.3 ตัวอย่างข้อมูลที่ดึงได้จากเฟสบุ๊ก

รหัสผู้ใช้	การกระทำ
u1	post
u2	comment
u3	comment
u2	post
u1	comment
u3	comment
u2	post
u3	comment
u1	comment
u5	comment
u3	post
u1	post
u2	comment
u4	comment
u3	comment
u5	comment

โดยสมมติให้ ui แทนรหัสผู้ใช้ในเฟสบุ๊ก และการกระทำ post หมายถึง ผู้ใช้คนนั้นได้โพสต์ในกลุ่มเฟสบุ๊ก ส่วนการกระทำ comment หมายถึง ผู้ใช้คนนั้นได้แสดงความคิดเห็นในโพสต์ ถ้ามีแต่ผู้โพสต์แต่ไม่มีใครแสดงความคิดเห็น ข้อมูลลักษณะแบบนี้จะถูกตัดทิ้งออกไป เนื่องจากมีเหตุการณ์เดียว ไม่สามารถพิจารณาผู้มีอิทธิพลได้ เช่น u3 โพสต์ข้อความบนเฟสบุ๊ก แต่ไม่มีใครแสดงความคิดเห็น ดังนั้นจึงถูกตัดทิ้งไป

7.2.2 การเตรียมข้อมูล

ในขั้นตอนนี้ทำการแปลงข้อมูลที่รวบรวมได้ให้อยู่ในรูปแบบของรายการเปลี่ยนแปลง โดยมีรายละเอียดดังต่อไปนี้

- 1 รายการเปลี่ยนแปลงจะประกอบไปด้วย รหัสผู้ใช้ที่โพสต์หัวข้อและรหัสผู้ใช้ที่แสดงความคิดเห็นในหัวข้อดังกล่าว

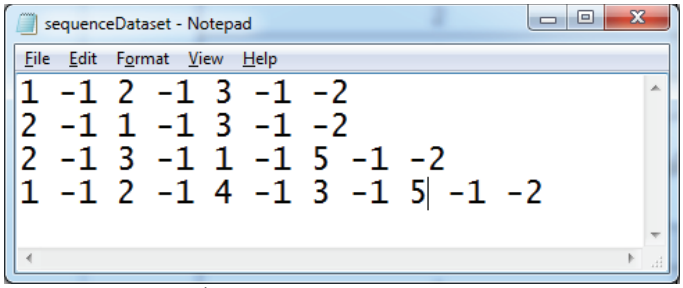
- ข้อมูลในแต่ละรายการเปลี่ยนแปลงจะถูกเรียงตามลำดับการแสดงความคิดเห็นของผู้ใช้ เช่น จากตารางที่ 7.3 ใน 3 แถวแรกแสดงให้เห็นได้ว่า u1 โพสต์ข้อความ แล้ว u2 กับ u3 เข้ามาแสดงความคิดเห็น ดังนั้น u1 u2 u3 อยู่ในรายการเปลี่ยนแปลงเดียวกันดังตารางที่ 7.4

- เมื่อจัดข้อมูลให้อยู่ในรูปรายการเปลี่ยนแปลงแล้ว ทำการเปลี่ยนรหัสผู้ใช้ให้เป็นตัวเลขจำนวนเต็มเพื่อนำเข้าสู่ SPMF (เหมือนการเตรียมชุดข้อมูลลำดับเหตุการณ์) เช่น ผู้ใช้ u1 ถูกแทนค่าด้วย 1 ผู้ใช้ u2 ถูกแทนค่าด้วย 2 เป็นต้น

- ข้อมูลที่แปลงเรียบร้อยแล้ว ถูกเก็บลงในโปรแกรม Notepad และทำการบันทึกไฟล์ชื่อ sequenceDataset.txt
  จากตัวอย่างข้อมูลในตารางที่ 7.4 เมื่อทำการแปลงข้อมูลเป็นตัวเลขจะได้ดังรูปที่ 7.4

ตารางที่ 7.4 ตัวอย่างการจัดข้อมูลให้อยู่ในรูปแบบรายการเปลี่ยนแปลง

รายการเปลี่ยนแปลง	ลำดับเหตุการณ์
1	u1 u2 u3
2	u2 u1 u3
3	u2 u3 u1 u5
4	u1 u2 u4 u3 u5



รูปที่ 7.4 ตัวอย่างการแปลงข้อมูล

จากข้อมูลทั้งหมดที่รวบรวม เมื่อทำการแปลงให้อยู่ในรูปแบบรายการเปลี่ยนแปลง และแปลงเป็นตัวเลข จะได้ชุดข้อมูลที่มีจำนวนรายการเปลี่ยนแปลงทั้งหมด 4,232 รายการเปลี่ยนแปลง โดยจะนำชุดข้อมูลดังกล่าวไปประมวลผลหากฎความสัมพันธ์เชิงลำดับโดยใช้ SPMF ต่อไป

7.2.3 การค้นหากฎความสัมพันธ์เชิงลำดับด้วย CMRules

การค้นหากฎความสัมพันธ์ที่แสดงถึงผู้มีอิทธิพลและคนที่ถูกครอบงำ จำเป็นต้องพิจารณาลำดับการแสดงความคิดเห็น ดังนั้นขั้นตอนวิธีที่นำมาใช้ต้องเป็นขั้นตอนวิธีการที่พิจารณาเรื่องลำดับเหตุการณ์ ในตัวอย่างนี้ประยุกต์ใช้ขั้นตอนวิธี CMRules ที่อยู่ใน SPMF เพื่อค้นหากฎความสัมพันธ์เชิงลำดับ ที่แสดงให้เห็นถึงความสัมพันธ์ของผู้มีอิทธิพลและผู้ที่ถูกครอบงำ

ในตัวอย่างนี้ใช้ค่าสนับสนุนขั้นต่ำแบบสัมพัทธ์เท่ากับ 0.1% (ค่าสนับสนุนแบบสัมบูรณ์เท่ากับ 5) และค่าความเชื่อมั่นขั้นต่ำเท่ากับ 80% คำสั่งสำหรับสร้างกฎความสัมพันธ์เชิงลำดับแสดงในตัวอย่างคำสั่งที่ 7.2 และผลลัพธ์ที่ได้จากการประมวลผลแสดงดังรูปที่ 7.5 และจะได้ไฟล์ output.txt ดังรูปที่ 7.6 ซึ่งแสดงกฎความสัมพันธ์เชิงลำดับจำนวน 25 กฎ

ตัวอย่างคำสั่งที่ 7.2

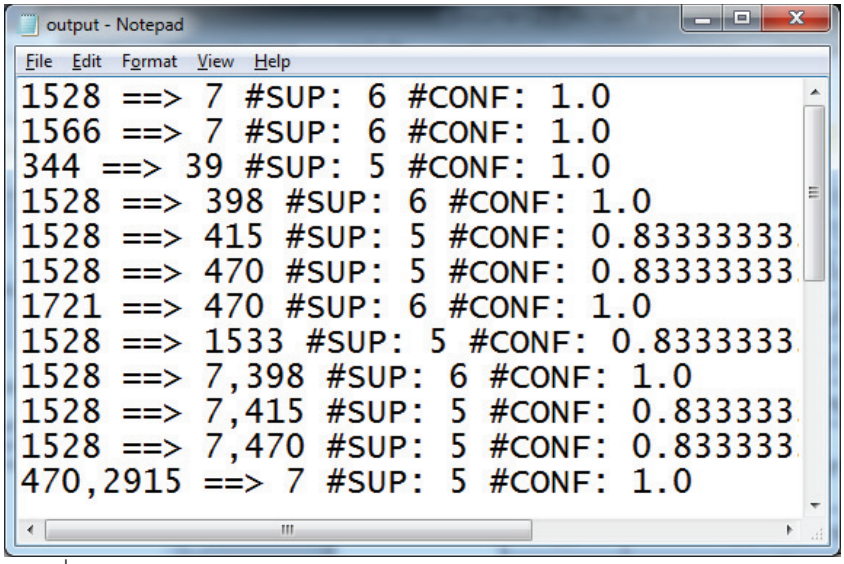
```
1.  package mySpmfProject;
2.  import java.io.IOException;
3.  import ca.pfv.spmf.algorithms.sequential_rules.cmrules.AlgoCMRules;
4.  public class CMRuleTestInfluen {
5.          public static void main(String [] arg) throws IOException{
6.                  String input = ".//dataset/sequenceDataset.txt";
7.                  String  output = ".//output.txt";
8.
9.                  double minSup = 0.001;
10                 double minConf = 0.80;
11.                AlgoCMRules algo = new AlgoCMRules();
12.                algo.runAlgorithm(input, output, minSup, minConf);
13.                algo.printStats();
14.         }
15. }
```

```
=============  CMRULES - STATS =============
Association rules count: 1449
Sequential rules count: 25
Total time : 3382 ms
Max memory: 233.37408447265625
===================================================
```

รูปที่ 7.5 ผลลัพธ์จากการประมวลผล CMRuleTestInfluen.java

จากรูปที่ 7.5 สามารถแปลผลลัพธ์ที่ได้จากการประมวลผลได้ดังนี้

- จำนวนกฎความสัมพันธ์เท่ากับ 1,449 กฎ
- จำนวนกฎความสัมพันธ์เชิงลำดับเท่ากับ 25 กฎ
- เวลาที่ใช้ในการประมวลผล คือ 3,382 มิลลิวินาที
- หน่วยความจำที่ใช้ในการประมวลผล คือ 233.37408447265625 เมกะไบต์



รูปที่ 7.6 ไฟล์ผลลัพธ์จากการประมวลผล CMRuleTestInfluen.java

จากรูปที่ 7.6 กฎที่ได้แสดงให้เห็นถึงว่าใครมีอิทธิพลกับใครบ้าง โดยสามารถแปลความหมายของ 5 กฎแรกได้ดังนี้

กฎที่ 1: 1528 ⟶ 7 แสดงให้เห็นว่าผู้ใช้รหัส 1528 โพสต์หรือแสดงความคิดเห็นใด ๆ ก็ตาม ผู้ใช้รหัส 7 จะแสดงความคิดเห็นตามเสมอ ด้วยค่าความเชื่อมั่นเท่ากับ 100% และค่าสนับสนุนแบบสัมบูรณ์เท่ากับ 6 แสดงว่าผู้ใช้รหัส 1528 มีอิทธิพลต่อผู้ใช้รหัส 7

กฎที่ 2: 1566 ⟶ 7 แสดงให้เห็นว่าผู้ใช้รหัส 1566 โพสต์หรือแสดงความคิดเห็นใด ๆ ก็ตาม ผู้ใช้รหัส 7 จะแสดงความคิดเห็นตามเสมอ ด้วยค่าความเชื่อมั่นเท่ากับ 100% และค่าสนับสนุนแบบสัมบูรณ์เท่ากับ 6 แสดงว่าผู้ใช้รหัส 1566 มีอิทธิพลต่อผู้ใช้รหัส 7

กฎที่ 3: 344 ⟶ 39 แสดงให้เห็นว่าผู้ใช้รหัส 344 โพสต์หรือแสดงความคิดเห็นใด ๆ ก็ตาม ผู้ใช้รหัส 39 จะแสดงความคิดเห็นตามเสมอ ด้วยค่าความเชื่อมั่นเท่ากับ 100% และค่าสนับสนุนแบบสัมบูรณ์เท่ากับ 5 แสดงว่าผู้ใช้รหัส 344 มีอิทธิพลต่อผู้ใช้รหัส 39

กฎที่ 4: 1528 ⟶ 398 แสดงให้เห็นว่าผู้ใช้รหัส 1528 โพสต์หรือแสดงความคิดเห็นใด ๆ ก็ตาม ผู้ใช้รหัส 398 จะแสดงความคิดเห็นตามเสมอ ด้วยค่าความเชื่อมั่นเท่ากับ 100% และค่าสนับสนุนแบบสัมบูรณ์เท่ากับ 6 แสดงว่าผู้ใช้รหัส 1528 มีอิทธิพลต่อผู้ใช้รหัส 398

กฎที่ 5: 1528 ⟶ 415 แสดงให้เห็นว่าผู้ใช้รหัส 1528 โพสต์หรือแสดงความคิดเห็นใด ๆ ก็ตาม ผู้ใช้รหัส 415 จะแสดงความคิดเห็นตามเสมอ ด้วยค่าความเชื่อมั่นเท่ากับ 83% และค่าสนับสนุนแบบสัมบูรณ์เท่ากับ 5 แสดงว่าผู้ใช้รหัส 1528 มีอิทธิพลต่อผู้ใช้รหัส 415

7.3 การจำแนกโรคหลอดเลือดสมอง

ประเทศไทยกำลังจะเข้าสู่สังคมผู้สูงอายุแบบเต็มรูปแบบในปี พ.ศ 2564 เนื่องจากจำนวนของประชากรผู้สูงอายุมีแนวโน้มเพิ่มขึ้น เมื่อเทียบกับจำนวนประชากรในวัยอื่น ทำให้รัฐบาลมีนโยบายที่พัฒนาระบบที่ช่วยสนับสนุนผู้สูงอายุให้อยู่ในสังคมอย่างมีความสุข การดูแลสุขภาพของผู้สูงอายุเป็นเป็นนโยบายหนึ่งที่รัฐบายให้ความสำคัญ

โรคหลอดเลือดสมอง (Cerebrovascular Disease, Stroke) เป็นสาเหตุลำดับต้นที่ทำให้เกิดความพิการและเสียชีวิตในผู้สูงอายุ โรคหลอดเลือดสมองเป็นภาวะที่สมองขาดเลือดไปเลี้ยงเนื่องจากหลอดเลือดตีบ หลอดเลือดอุดตัน หรือหลอดเลือดแตก ส่งผลให้เนื้อเยื่อในสมองถูกทำลายและการทำงานของสมองหยุดชะงัก ทำให้สมองขาดเลือด ส่งผลให้ผู้ป่วยเสียชีวิตถ้าไม่ได้รับการรักษาอย่าง

ทันท่วงที หรืออาจจะทำให้ผู้ป่วยเกิดความพิการไปตลอดชีวิต ดังนั้นการจำแนกผู้ป่วยที่มีโอกาสเป็นโรคหลอดเลือดสมองจึงเป็นเรื่องสำคัญ

การจำแนกโรคหลอดเลือดสมองสามารถประยุกต์ใช้การจำแนกเชิงความสัมพันธ์ ซึ่งจะได้กฎที่ใช้สำหรับจำแนกที่มีประสิทธิภาพ และกฎยังแสดงให้เห็นถึงความสัมพันธ์ของปัจจัยที่นำไปสู่โรคหลอดเลือดสมอง ทำให้สามารถหาทางป้องกันหรือส่งเสริมสุขภาพของคนไทยเพื่อไม่ให้เป็นโรคหลอดเลือ ด สมองในอนาคตได้ ขั้นตอนการจำแนกโรคหลอดเลือดสมองด้วยการจำแนกเชิงความสัมพันธ์มีดังต่อไปนี้

7.3.1 การรวบรวมข้อมูล

ข้อมูลที่ใช้ในการจำแนกโรคหลอดเลือดสมอง เป็นประวัติสุขภาพของผู้ป่วยที่เข้ารับบริการในโรงพยาบาลและถูกบันทึกลงในฐานข้อมูล โดยคัดเลือกเอาเฉพาะประวัติของผู้ป่วยที่มีอายุ 60 ปีขึ้นไป ซึ่งปัจจัยที่ใช้ในการจำแนกประกอบไปด้วย 7 ปัจจัย คือ เพศ สถานะภาพ การสูบบุหรี่ การดื่มสุรา การออกกำลังกาย ความดันโลหิต และคอเลสเตอรอล เก็บรวบรวมเฉพาะข้อมูลผู้ป่วยที่มีข้อมูลปัจจัยครบทั้ง 7 ปัจจัย จำนวน 1,000 ราย แบ่งเป็นผู้ป่วยโรคหลอดเลือดสมอง 500 ราย และผู้ป่วยที่ไม่เป็นโรคหลอดเลือดสมอง 500 ราย

7.3.2 การเตรียมข้อมูล

การจำแนกโรคหลอดเลือดสมองจะประยุกต์ใช้ขั้นตอน CBA ใน Weka ดังนั้นจำเป็นต้องเตรียมข้อมูลเพื่อให้สามารถนำเข้า Weka ได้ โดยการเตรียมข้อมูลมีรายละเอียดดังต่อไปนี้

- 1 รายการเปลี่ยนแปลง หมายถึง ข้อมูลทั้ง 7 ปัจจัยของผู้ป่วยหนึ่งคน เช่น ในตารางที่ 7.5 ประกอบไปด้วย 5 รายการเปลี่ยนแปลง ซึ่งเป็นข้อมูลปัจจัยของผู้ป่วย 5 คน เป็นต้น

- แทนค่าในแต่ปัจจัยเป็นตัวอักขระตามตารางที่ 7.6 และค่าในคลาสถูกแทนเป็นตัวอักขระตามตารางที่ 7.7 เช่น ข้อมูลผู้ป่วยลำดับที่ 1 ในตารางที่ 7.5 เป็นเพศหญิงถูกแปลงเป็น f สถานะภาพสมรสถูกแปลงเป็น y ไม่สูบบุหรี่ถูกแปลงเป็น n ดื่มสุราแต่เลิกแล้วถูกแปลงเป็น e ออกกำลังกายถูกแปลงเป็น y ความดันปกติถูกแปลงเป็น n คอเลสเตอรอลไม่ปกติถูกแปลงเป็น a และไม่เป็นโรคหลอดสมอง ดังนั้นคลาสเป็น n เป็นต้น

- ข้อมูลที่แปลงเรียบร้อยแล้ว ถูกจัดเก็บในโปรแกรม Microsoft excel และทำการบันทึกไฟล์ factor&class.csv (ดังรูปที่ 7.7)

จากข้อมูลทั้งหมดที่รวบรวม เมื่อทำการแปลงให้อยู่ในรูปแบบรายการเปลี่ยนแปลงและแปลงเป็นอักขระ จะได้ชุดข้อมูลที่มีจำนวนรายการเปลี่ยนแปลงทั้งหมด 1,000 รายการเปลี่ยนแปลง โดยจะนำชุดข้อมูลดังกล่าวไปจำแนกโรคหลอดเลือดสมองโดยใช้ขั้นตอนวิธี CBA ใน Weka ต่อไป

ตารางที่ 7.5 ตัวอย่างข้อมูลผู้ป่วย

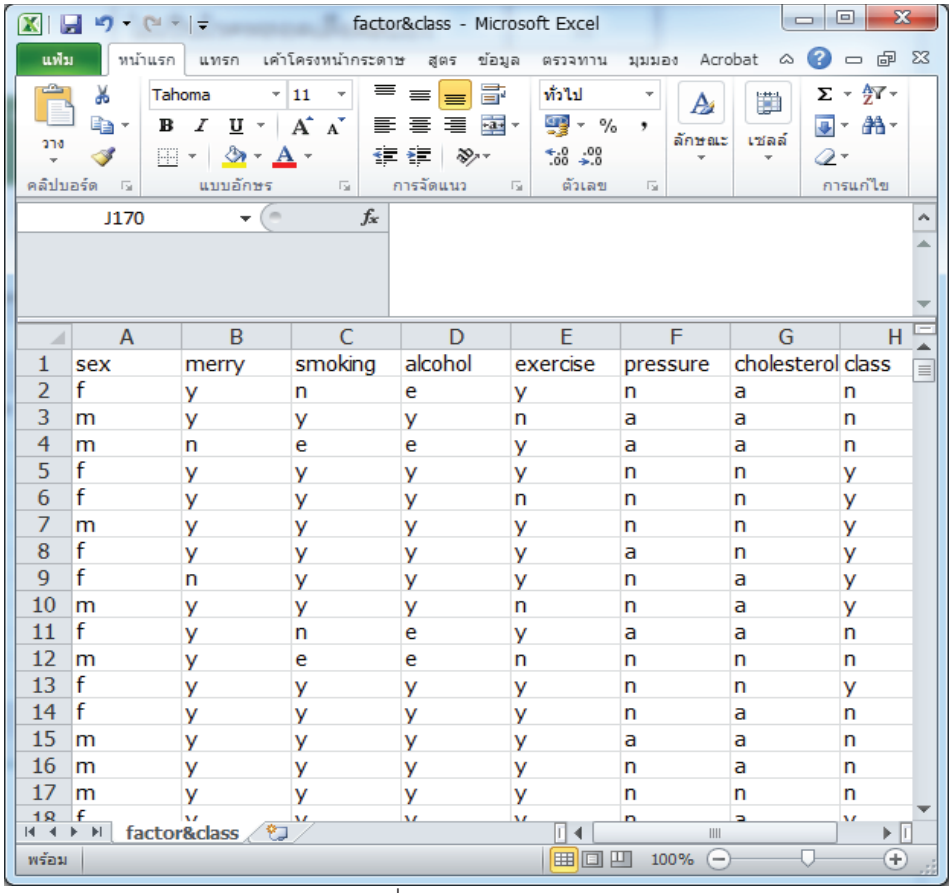
ลำดับ	เพศ	สถานะภาพ	สูบบุหรี่	ดื่มสุรา	ออกกำลังกาย	ความดัน	คอเลสเตอรอล	คลาส
1	หญิง	สมรส	ไม่สูบบุหรี่	ดื่มสุราแต่เลิกแล้ว	ออกกำลังกาย	ปกติ	ไม่ปกติ	ไม่เป็น
2	ชาย	สมรส	สูบบุหรี่	ดื่มสุรา	ไม่ออกกำลังกาย	ไม่ปกติ	ไม่ปกติ	ไม่เป็น
3	ชาย	โสด	เคยสูบแต่เลิกแล้ว	ดื่มสุราแต่เลิกแล้ว	ออกกำลังกาย	ไม่ปกติ	ไม่ปกติ	ไม่เป็น
4	หญิง	สมรส	สูบบุหรี่	ดื่มสุรา	ออกกำลังกาย	ปกติ	ปกติ	เป็น
5	หญิง	สมรส	สูบบุหรี่	ดื่มสุรา	ไม่ออกกำลังกาย	ปกติ	ปกติ	เป็น

ตารางที่ 7.6 การแปลงข้อมูล

ลำดับ	ปัจจัย	ความหมาย
1	เพศ	m=ชาย, f=หญิง
2	สถานะภาพ	n=โสด, y=สมรส
3	สูบบุหรี่	y=สูบบุหรี่, e=เคยสูบแต่เลิกแล้ว, n=ไม่สูบบุหรี่
4	ดื่มสุรา	y=ดื่มสุรา, e=ดื่มสุราแต่เลิกแล้ว, n=ไม่ดื่มสุรา
5	ออกกำลังกาย	y=ออกกำลังกาย, n=ไม่ออกกำลังกาย
7	ความดันโลหิต	n=ปกติ, a=ไม่ปกติ
8	คอเลสเตอรอล	n=ปกติ, a=ไม่ปกติ

ตารางที่ 7.7 การแทนค่าคลาส

ค่า	แทนค่า
เป็นโรคหลอดเลือดสมอง	y
ไม่เป็นโรคหลอดเลือดสมอง	n



	A	B	C	D	E	F	G	H
1	sex	merry	smoking	alcohol	exercise	pressure	cholesterol	class
2	f	y	n	e	y	n	a	n
3	m	y	y	y	n	a	a	n
4	m	n	e	e	y	a	a	n
5	f	y	y	y	y	n	n	y
6	f	y	y	y	n	n	n	y
7	m	y	y	y	y	n	n	y
8	f	y	y	y	y	a	n	y
9	f	n	y	y	y	n	a	y
10	m	y	y	y	n	n	a	y
11	f	y	n	e	y	a	a	n
12	m	y	e	e	n	n	n	n
13	f	y	y	y	y	n	n	y
14	f	y	y	y	y	n	a	n
15	m	y	y	y	y	a	a	n
16	m	y	y	y	y	n	a	n
17	m	y	y	y	y	n	n	n

รูปที่ 7.7 ไฟล์ชุดข้อมูล

7.3.3 การจำแนกโรคหลอดเลือดสมองด้วย CBA

การค้นหากฎสำหรับจำแนกโรคหลอดเลือดสมอง สามารถประยุกต์ใช้ขั้นตอนวิธี CBA ที่อยู่ใน Weka แสดงคำสั่งได้ดังตัวอย่างคำสั่งที่ 7.3

ตัวอย่างคำสั่งที่ 7.3

```
1.  package myWekaProject;
2.
3.  import weka.core.Debug;
4.  import weka.core.converters.CSVLoader;
5.  import weka.core.Instances;
6.  import java.io.File;
7.  import weka.associations.Apriori;
8.  import weka.classifiers.rules.car.JCBA;
9.  import weka.classifiers.Evaluation;
10 import java.text.DecimalFormat;
11.
12. public class CBATestStroke {
13.         public static void main(String args[]) throws Exception{
14.             CSVLoader loader = new CSVLoader();
15.             loader.setSource(new File("./dataset/factor&class.csv"));
16.             Instances dataset = loader.getDataSet();
17.
18.             dataset.setClassIndex(dataset.numAttributes()-1);
19.              int trainSize = (int) Math.round(dataset.numInstances() * 0.7);
20.              int testSize = dataset.numInstances() - trainSize;
21.
22.              dataset.randomize(new Debug.Random(1)); //random dataset
23.
24.               Instances traindataset = new Instances(dataset, 0, trainSize); //training set
25.               Instances testdataset = new Instances(dataset, trainSize, testSize); //test set
26.
27.             JCBA model = new JCBA();
28.             Apriori apriori = new Apriori();
29.             apriori.setMinMetric(0.7); // minimum confidence (70%)
30.             apriori.setLowerBoundMinSupport(0.3); //minimum support (30%)
31.             model.setCBA(true);
32.             model.setCarMiner(apriori);
33.             model.buildClassifier(traindataset);
34.
35.             Evaluation eval = new Evaluation(dataset);
36.             eval.evaluateModel(model, testdataset);
37.
38.             DecimalFormat f = new DecimalFormat("#0.00");
39.             System.out.println("Accuracy : "+f.format(eval.pctCorrect()));
40.             System.out.println(apriori.toString());
41.         }
42. }
```

จากตัวอย่างกำหนดให้ค่าสนับสนุนขั้นต่ำเท่ากับ 30% และค่าความเชื่อมั่นขั้นต่ำเท่ากับ 70% และทำการแบ่งข้อมูลเรียนรู้ 70% และข้อมูลทดสอบ 30% ผลลัพธ์ที่ได้จากการประมวลผลแสดงดังรูปที่ 7.8

```
Accuracy : 72.00
Apriori
=======
Minimum support: 0.3 (210 instances)
Minimum metric <confidence>: 0.7
Number of cycles performed: 14

Generated sets of large itemsets:

Size of set of large itemsets L(1): 13

Size of set of large itemsets L(2): 18

Size of set of large itemsets L(3): 21

Size of set of large itemsets L(4): 12

Size of set of large itemsets L(5): 2

Best rules found:

 1. smoking=y alcohol=y exercise=y pressure=n cholesterol=n 281 ==> class=y
213    conf:(0.76)
 2. smoking=y exercise=y pressure=n cholesterol=n 285 ==> class=y 216
conf:(0.76)
 3. merry=y smoking=y pressure=n cholesterol=n 287 ==> class=y 214
conf:(0.75)
 4. alcohol=y exercise=y pressure=n cholesterol=n 287 ==> class=y 214
conf:(0.75)
 5. merry=y smoking=y alcohol=y pressure=n cholesterol=n 283 ==> class=y
211    conf:(0.75)
 6. smoking=y pressure=n cholesterol=n 344 ==> class=y 253    conf:(0.74)
 7. smoking=y alcohol=y pressure=n cholesterol=n 340 ==> class=y 250
conf:(0.74)
 8. merry=y alcohol=y pressure=n cholesterol=n 289 ==> class=y 212
conf:(0.73)
 9. alcohol=y pressure=n cholesterol=n 347 ==> class=y 251    conf:(0.72)
10.  smoking=y  alcohol=y  exercise=y  cholesterol=n  333  ==>  class=y  239
conf:(0.72)
```

รูปที่ 7.8 ผลลัพธ์จากการประมวลผล CBATestStroke.java

รูปที่ 7.8 แสดงให้เห็นว่า การจำแนกโรคหลอดเลือดสมองด้วยขั้นตอนวิธี CBA ให้ค่าความถูกต้อง 72% กฎความสัมพันธ์ระบุคลาส 5 กฎแรก สามารถแปลความหมายได้ดังนี้

กฎที่ 1: smoking=y alcohol=y exercise=y pressure=n cholesterol=n 281 ==> class=y 213 conf:(0.76) หมายความว่า คนที่สูบบุหรี่ ดื่มสุรา ออกกำลังกาย ความดันโลหิตปกติ และคอเลสเตอรอล

ปกติ มีโอกาสเป็นโรคหลอดเลือดสมองด้วยค่าความเชื่อมั่น 76% และมีค่าสนับสนุนแบบสัมบูรณ์เท่ากับ 213

กฎที่ 2: smoking=y exercise=y pressure=n cholesterol=n 285 ==> class=y 216 conf:(0.76) หมายความว่า คนที่สูบบุหรี่ ออกกำลังกาย ความดันโลหิตปกติ และคอเลสเตอรอลปกติ มีโอกาสเป็นโรคหลอดเลือดสมองด้วยค่าความเชื่อมั่น 76% และมีค่าสนับสนุนแบบสัมบูรณ์เท่ากับ 216

กฎที่ 3: merry=y smoking=y pressure=n cholesterol=n 287 ==> class=y 214 conf:(0.75) หมายความว่า คนที่แต่งงาน สูบบุหรี่ ความดันโลหิตปกติ และคอเลสเตอรอลปกติ มีโอกาสเป็นโรคหลอดเลือดสมองด้วยค่าความเชื่อมั่น 75% และมีค่าสนับสนุนแบบสัมบูรณ์เท่ากับ 214

กฎที่ 4: alcohol=y exercise=y pressure=n cholesterol=n 287 ==> class=y 214 conf:(0.75) หมายความว่า คนที่ดื่มสุรา ออกกำลังกาย ความดันโลหิตปกติ และคอเลสเตอรอลปกติ มีโอกาสเป็นโรคหลอดเลือดสมองด้วยค่าความเชื่อมั่น 75% และมีค่าสนับสนุนแบบสัมบูรณ์เท่ากับ 214

กฎที่ 5: merry=y smoking=y alcohol=y pressure=n cholesterol=n 283 ==>class=y 211 conf:(0.75) หมายความว่า คนที่แต่งงาน สูบบุหรี่ ดื่มสุรา ความดันโลหิตปกติ และคอเลสเตอรอลปกติ มีโอกาสเป็นโรคหลอดเลือดสมองด้วยค่าความเชื่อมั่น 75% และมีค่าสนับสนุนแบบสัมบูรณ์เท่ากับ 211

บทสรุป

การทำเหมืองรูปแบบสามารถประยุกต์ใช้กับงานหลายด้าน ในบทนี้ได้ยกตัวอย่างการประยุกต์ใช้การทำเหมืองกฎความสัมพันธ์สำหรับค้นหาความสัมพันธ์ของหมวดหมู่เพจ การประยุกต์ใช้การทำเหมืองกฎความสัมพันธ์เชิงลำดับเพื่อค้นหาผู้มีอิทธิพล และการประยุกต์การจำแนกเชิงความสัมพันธ์ในการจำแนกโรคหลอดเลือดสมอง ในตัวอย่างการประยุกต์ใช้ ได้อธิบายการรวบรวมข้อมูล การเตรียมข้อมูล การใช้คำสั่งในการขุดค้นรูปแบบที่น่าสนใจ และแสดงให้เห็นถึงผลลัพธ์ที่ได้ โดยจำนวนรูปแบบที่ได้จะขึ้นอยู่กับค่าสนับสนุนขั้นต่ำและค่าความเชื่อมั่นขั้นต่ำที่ผู้ใช้กำหนดขึ้น ดังนั้นผู้อ่านสามารถกำหนดค่าทั้งสองที่แตกต่างจากตัวอย่างในหนังสือ เพื่อให้ได้รูปแบบที่มีประสิทธิภาพในงานแต่ละด้านได้

แบบฝึกหัดท้ายบท

1. จงอธิบายการเตรียมข้อมูลเพื่อค้นหาความสัมพันธ์ของหมวดหมู่เพจบนเฟสบุ๊ก
2. จงแปลความหมายกฎความสัมพันธ์ของหมวดหมู่เพจต่อไปนี้

   5 11 ⟶ 25 #SUPP:215 #CONF: 0.93

3. การระบุผู้ที่ถูกครอบงำจากกฎความสัมพันธ์เชิงลำดับ สามารถพิจารณาได้อย่างไร
4. จากกฎความสัมพันธ์เชิงลำดับสำหรับระบุผู้มีอิทธิพลต่อไปนี้ สามารถตีความหมายได้อย่างไร

   1528 ⟶ 7

   1528 ⟶ 398

   1528 ⟶415

5. จงแปลความหมายกฎต่อไปนี้ สำหรับการจำแนกโรคหลอดเลือดสมอง
   alcohol=y pressure=n cholesterol=n 347 ==> class=y 251 conf:(0.72)
6. จงยกตัวอย่างการประยุกต์ใช้การทำเหมืองกฎความสัมพันธ์
7. จงยกตัวอย่างการประยุกต์ใช้การทำเหมืองกฎความสัมพันธ์เชิงลำดับ
8. จงยกตัวอย่างการประยุกต์ใช้การจำแนกเชิงความสัมพันธ์
9. ถ้าต้องการค้นหาความสัมพันธ์ของโรคที่เกิดร่วมกัน วิธีการทำเหมืองรูปแบบแบบใดเหมาะสมที่สุด เพราะเหตุใด
10. ถ้าต้องการสรุปข้อความคิดเห็นบนเครือข่ายสังคมออนไลน์ที่มีต่อโรงแรมแห่งหนึ่งว่าเป็นข้อความคิดเห็นเชิงบวกหรือเชิงลบ สามารถประยุกต์ใช้การทำเหมืองรูปแบบแบบใด

ภาคผนวก

ภาคผนวก ก การติดตั้ง SPMF

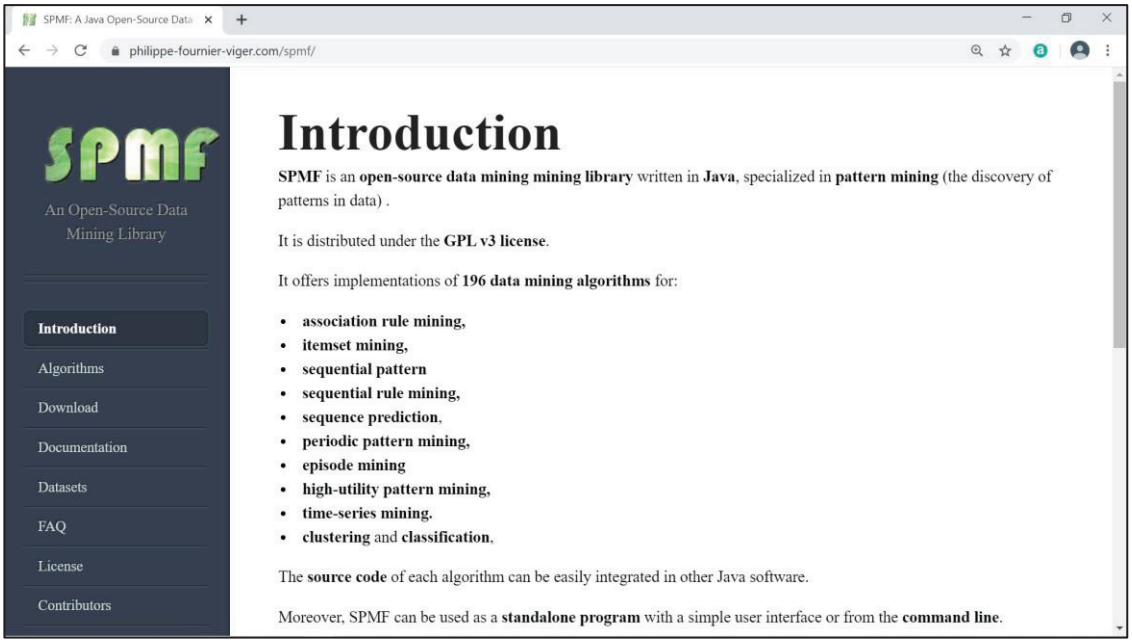
ปัจจุบันมีเครื่องมือที่ใช้ในการทำเหมืองรูปแบบที่น่าสนใจจำนวนมาก เช่น Weka, KNIME, Rapid-I, SPMF และ Mathlab เป็นต้น SPMF เป็นคลังโปรแกรม (Library) ที่พัฒนาขึ้นสำหรับการทำเหมืองรูปแบบโดยเฉพาะ สามารถใช้งานได้ฟรีและประกอบไปด้วยขั้นตอนวิธีต่างๆ เกี่ยวกับการทำเหมืองรูปแบบ เช่น ขั้นตอนวิธีสำหรับการสร้างกฎความสัมพันธ์ ขั้นตอนวิธีสำหรับการทำเหมืองเซตรายการความถี่ ขั้นตอนวิธีสำหรับการทำเหมืองรูปแบบลำดับเหตุการณ์ ขั้นตอนวิธีสำหรับการสร้างกฎความสัมพันธ์เชิงลำดับ เป็นต้น คลังโปรแกรม SPMF ถูกพัฒนาขึ้นด้วยภาษาจาวาโดยการนำทีมของ Philippe Fournier-Viger, Jerry Chun-Wei Lin และ Vincent S. Tseng ถูกพัฒนาขึ้นตั้งแต่ปี พ.ศ 2553 และมีจำนวนการอ้างอิงหรือใช้คลังโปรแกรม SPMF เพิ่มขึ้นทุกปี

การเรียกใช้คลังโปรแกรม SPMF สามารถทำได้ง่าย และสามารถปรับปรุงคำสั่งหรือเขียนคำสั่งเพิ่มเติมได้ นอกจากนี้ SPMF ยังมีส่วนที่เป็น GUI ที่รองรับการทำงานสำหรับผู้ใช้ที่ไม่มีความรู้เกี่ยวกับภาษาจาวา ในหนังสือเล่มนี้ใช้ SPMF ในการทำเหมืองเซตรายการความถี่ การทำเหมืองรูปแบบลำดับเหตุการณ์ การทำเหมืองรูปแบบลำดับเหตุการณ์หลายมิติ และการทำเหมืองกฎความสัมพันธ์ การเรียกใช้คลัง SPMF จำเป็นต้องติดตั้ง SPMF ก่อน ซึ่งจะกล่าวถึงรายละเอียดในหัวข้อย่อยต่อไปนี้

ก.1 เริ่มต้นใช้คลังโปรแกรม SPMF

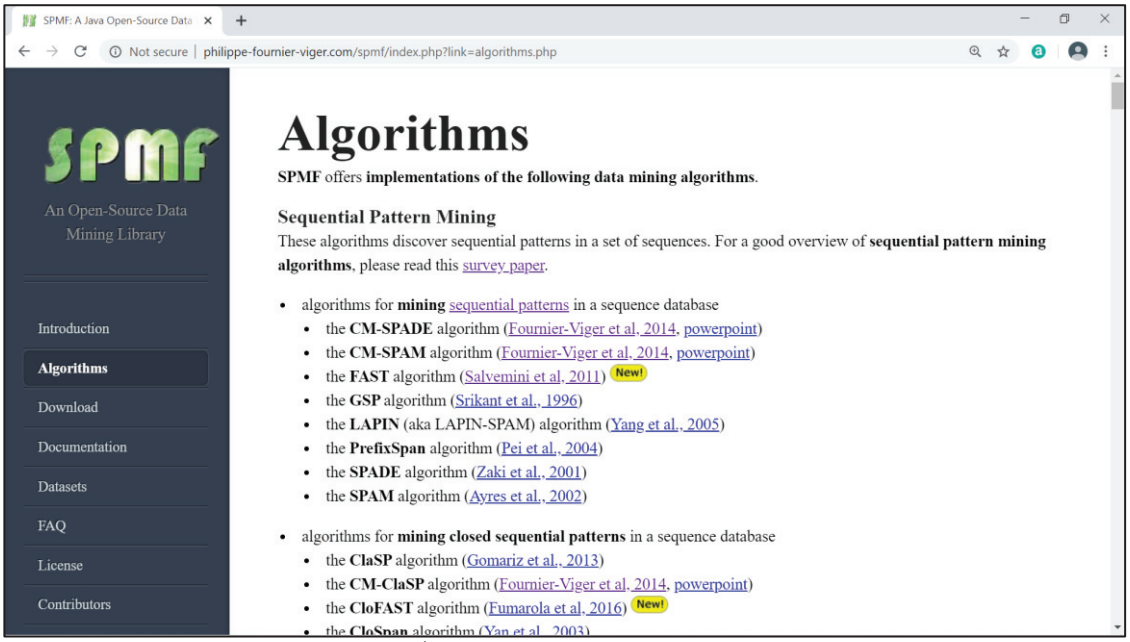
SPMF เป็นคลังโปรแกรมที่พัฒนาขึ้นจากภาษาจาวา ซึ่งใช้งานง่ายและสามารถปรับปรุงแก้ไขคำสั่งได้ง่าย มีคำอธิบายและตัวอย่างการใช้งานแต่ละขั้นตอนวิธีที่แสดงรายละเอียดอย่างชัดเจน สามารถดูรายละเอียดที่เว็บไซต์ http://www.philippe-fournier-viger.com/spmf/index.php เมื่อเข้าไปที่เว็บไซต์จะแสดงหน้าเว็บดังรูปที่ ก.1 ในเว็บไซต์ประกอบด้วยเมนูที่น่าสนใจดังต่อไปนี้

1. เมนู Introduction แสดงรายละเอียดเบื้องต้นเกี่ยวกับ SPMF ดังรูปที่ ก.1



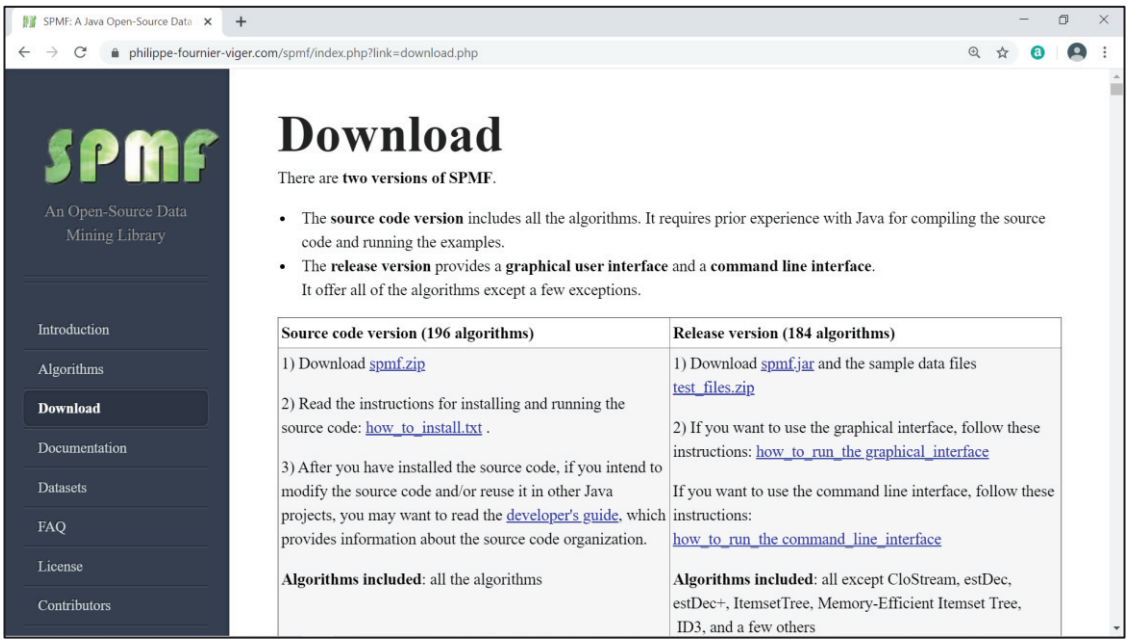
รูปที่ ก.1 หน้าจอเมนู Introduction

2. เมนู Algorithms แสดงรายละเอียดขั้นตอนวิธีที่มีอยู่ใน SPMF ซึ่งมีการแบ่งกลุ่มให้อย่างชัดเจน เช่น ขั้นตอนวิธีสำหรับการทำเหมืองรูปแบบลำดับเหตุการณ์ ประกอบไปด้วย CM-SPADE, CM-SPAM, FAST และ GSP เป็นต้น นอกจากนี้แล้วแต่ละขั้นตอนวิธียังมีลิงก์ไปยังบทความของแต่ละขั้นตอนวิธี ดังรูปที่ ก.2



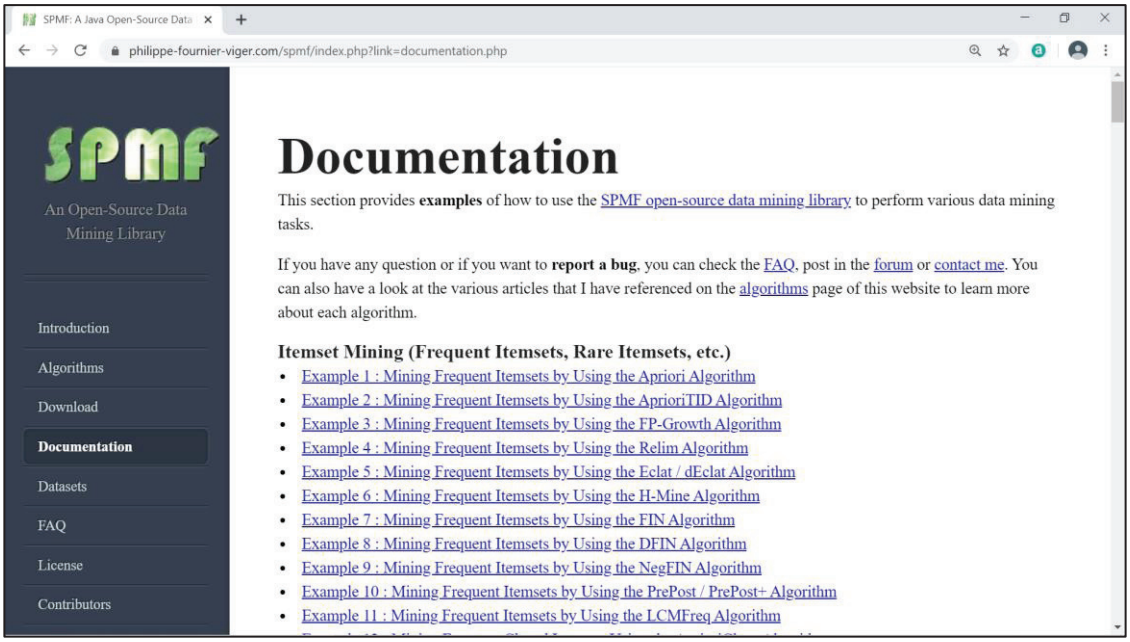
รูปที่ ก.2 หน้าจอเมนู Algorithms

3. เมนู Download แสดงรายละเอียดการดาวน์โหลด SPMF (ดังรูปที่ ก.3) ซึ่งมีอยู่ 2 ส่วน คือ ส่วนที่เป็นคำสั่งจาวา (Source code version) และส่วนของ GUI ส่วนที่เป็นคำสั่งจาวาเหมาะสำหรับผู้ที่มีความรู้ในการเขียนโปรแกรมด้วยภาษาจาวา ซึ่งสามารถเรียกใช้ไลบลารี่หรือปรับคำสั่งที่อยู่ใน SPMF ได้ ส่วน GUI หมาะสำหรับผู้ที่ไม่มีความรู้เกี่ยวกับภาษาจาวา

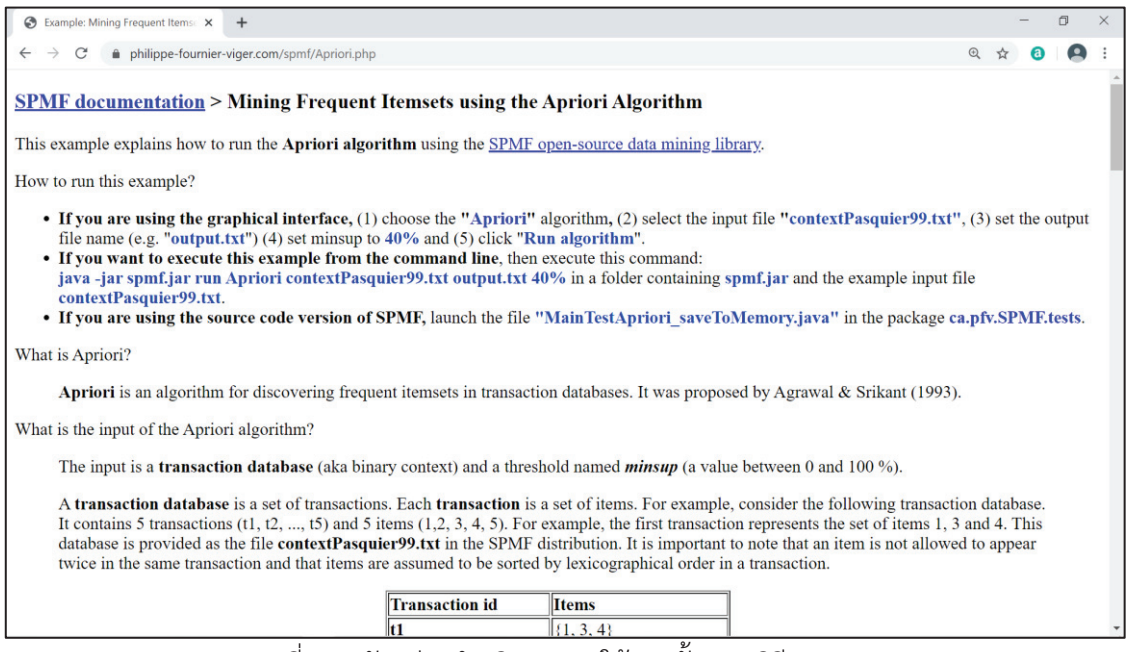


รูปที่ ก.3 หน้าจอเมนู Download

4. เมนู Document แสดงรายละเอียดคู่มือการเรียกใช้งานขั้นตอนวิธี (ดังรูปที่ ก.4) โดยสามารถคลิกเข้าไปที่ลิงก์ของขั้นตอนวิธีที่ต้องการเรียกใช้ เช่น ถ้าต้องการเรียกใช้ขั้นตอนวิธี Apriori ให้คลิกที่ลิงก์ Example 1 : Mining Frequent Itemsets by Using the Apriori Algorithm ซึ่งจะแสดงหน้าจอดังรูปที่ ก.5 โดยจะแสดงรายละเอียดต่างๆ เกี่ยวกับการเรียกใช้งานขั้นตอนวิธี Apriori เช่น การรันโปรแกรม บทความที่เกี่ยวข้อง ข้อมูลนำเข้า ผลลัพธ์ รายละเอียดการพัฒนาขั้นตอนวิธีด้วยจาวา เป็นต้น

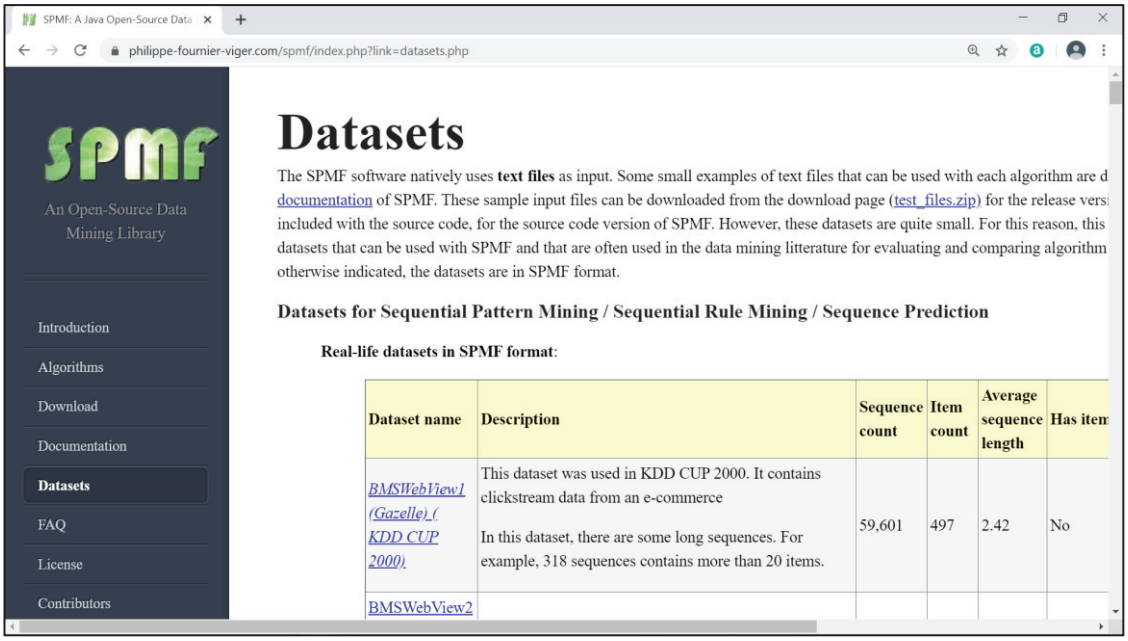


รูปที่ ก.4 หน้าจอเมนู Documentation



รูปที่ ก.5 ตัวอย่างคำอธิบายการใช้งานขั้นตอนวิธี Apriori

5. เมนู Dataset แสดงชุดข้อมูลต่างๆ ที่ใช้ในการทำเหมืองรูปแบบ (รูปที่ ก.6) สามารถดาวน์โหลดชุดข้อมูลได้ โดยชุดข้อมูลทั้งหมดมีการจัดรูปแบบให้สามารถใช้กับ SPMF และเป็นชุดข้อมูลมาตรฐานที่นิยมใช้ในการทำเหมืองข้อมูล

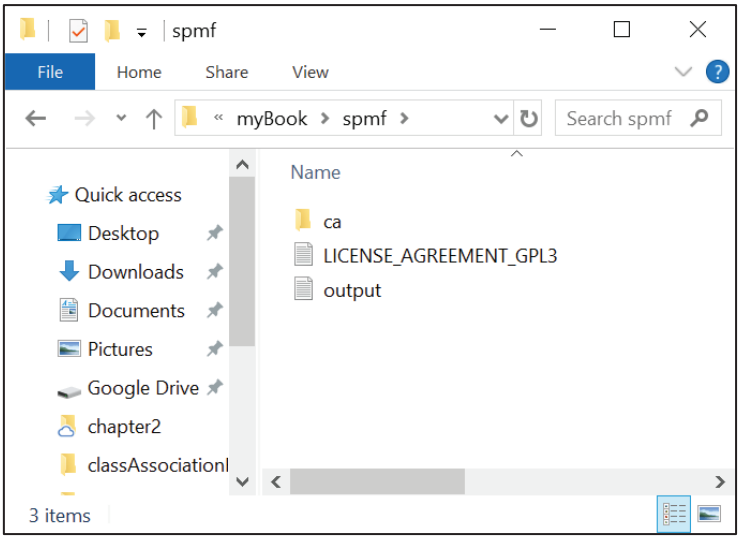


รูปที่ ก.6 หน้าจอเมนู Datasets

ก.2 การติดตั้ง SPMF

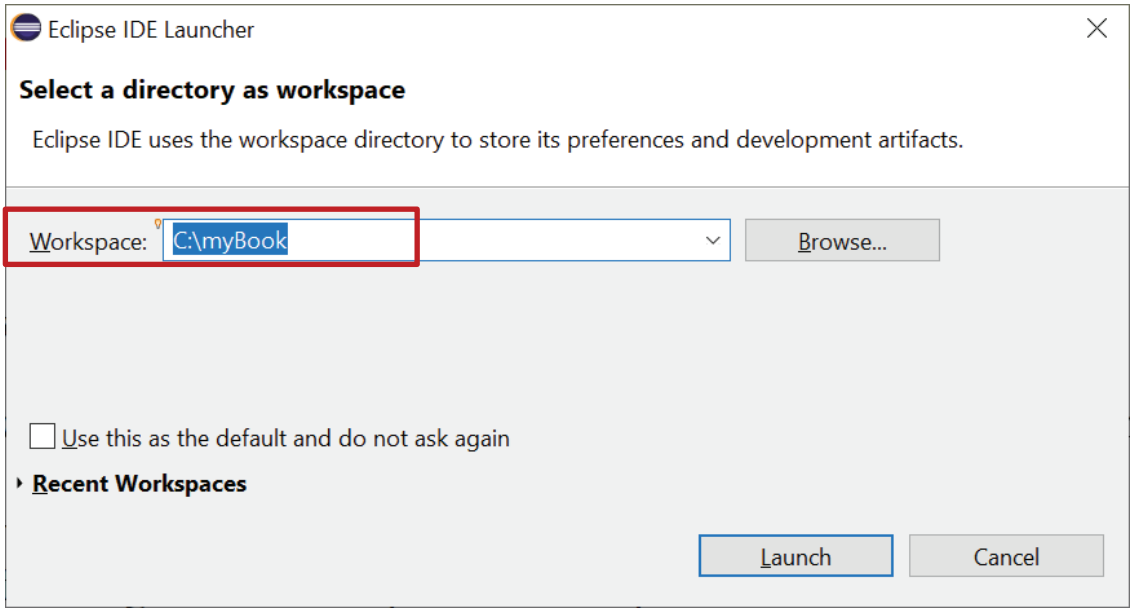
คำสั่งของ SPMF ถูกออกแบบขึ้นมาเพื่อให้ง่ายต่อการนำมาใช้และง่ายต่อการประยุกต์ใช้ร่วมกับขั้นตอนวิธีอื่นๆ การเรียกใช้คลังโปรแกรม SPMF จะต้องทำการติดตั้ง JDK ตั้งแต่เวอร์ชั่น 1.8 เป็นต้นไป สามารถติดตั้ง Eclipse หรือ Net Beans เพื่อประมวลผลคำสั่งของ SPMF ในบทนี้จะแสดงการติดตั้ง SPMF ใน Eclipse IDE 2019-09 (ดาวน์โหลด Eclipse ได้ฟรีที่ https://www.eclipse.org) โดยขั้นตอนการติดตั้งมีดังนี้

1. ทำการดาวน์โหลดไฟล์ spmf.zip ที่ http://www.philippe-fournier-viger.com/spmf/ แล้วเลือกเมนู Download ดังรูปที่ ก.3 ส่วนของ Source code version ให้เลือกดาวน์โหลด spmf.zip จากนั้นทำการแตกไฟล์ spmf.zip จะได้ไฟล์ดังรูปที่ ก.7 ซึ่งไฟล์ทั้งหมดที่ได้จากการแตกไฟล์อยู่ใน C:\myBook\spmf



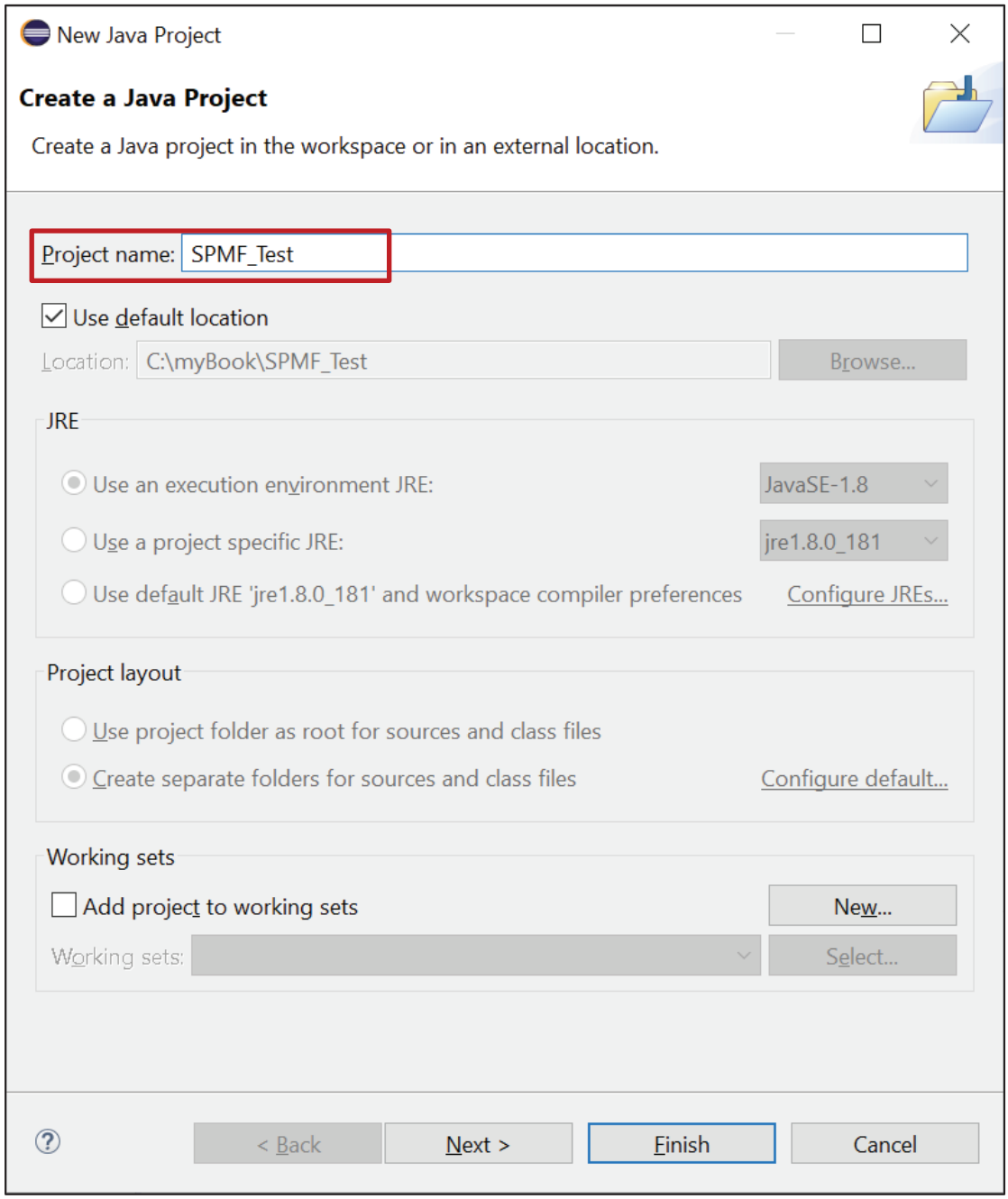
รูปที่ ก.7 ไฟล์ที่ได้จากการ unzip ไฟล์ spmf.zip

2. เปิดโปรแกรม Eclipse เลือกโฟลเดอร์ที่ต้องการเก็บไฟล์จากนั้นคลิกปุ่ม Lanch ดังรูปที่ ก.8



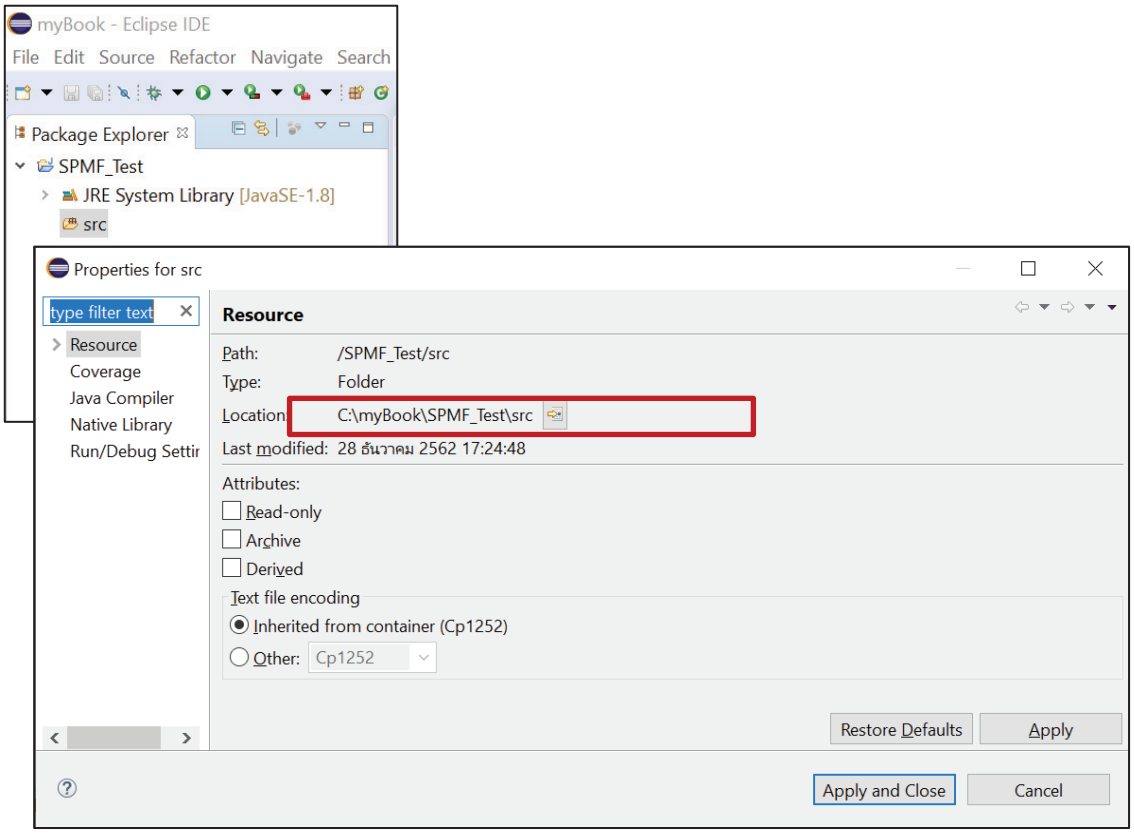
รูปที่ ก.8 การระบุตำแหน่ง workspace

3. ไปที่เมนู File->New->Java Project จากนั้นให้ป้อนชื่อโปรเจคในช่อง Project name: ในตัวอย่างรูปที่ ก.9 ป้อนชื่อโปรเจค คือ SPMF_Test จากนั้นคลิกปุ่ม Finish



รูปที่ ก.9 การสร้าง Java Project

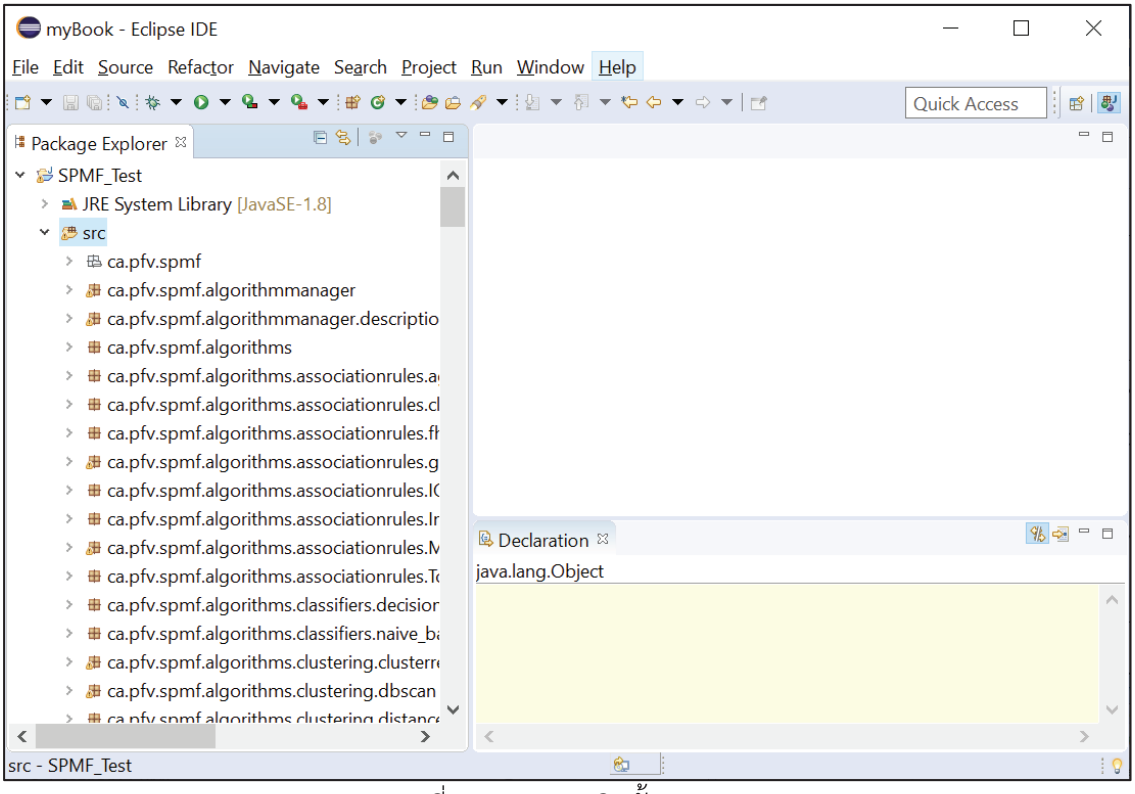
4. ทำการคลิกขวาบน "src" ที่อยู่ในหน้าต่าง Package Explorer แล้วเลือก "Properties" เพื่อดูตำแหน่งของโฟลเดอร์ src (ดังรูปที่ ก.10) เช่น ตำแหน่งของ src คือ C:\myBook\SPMF_Test\src เป็นต้น



รูปที่ ก.10 ตำแหน่งของ src

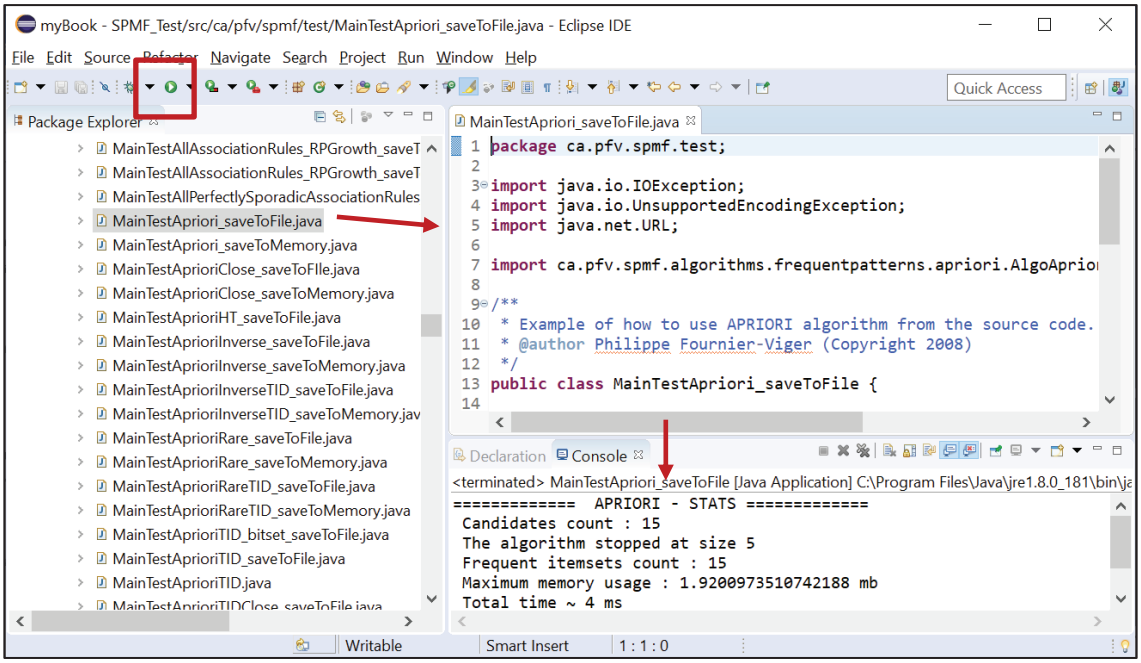
5. ทำการคัดลอกโฟลเดอร์ ca ไปไว้ในโฟลเดอร์ src เช่น คัดลอกโฟลเดอร์ ca ซึ่งอยู่ใน C:\myBook\spmf มาไว้ที่ตำแหน่ง C:\myBook\SPMF_Test\src เป็นต้น

6. กลับไปที่โปรแกรม Ecipse ทำการคลิกขวาบนโฟลเดอร์ src แล้วคลิกเลือก "Refresh" จะได้คลังโปรแกรมของ SPMF ดังรูปที่ ก.11 แสดงว่าทำการติดตั้ง SPMF เสร็จเรียบร้อยแล้ว



รูปที่ ก.11 ผลการติดตั้ง SPMF

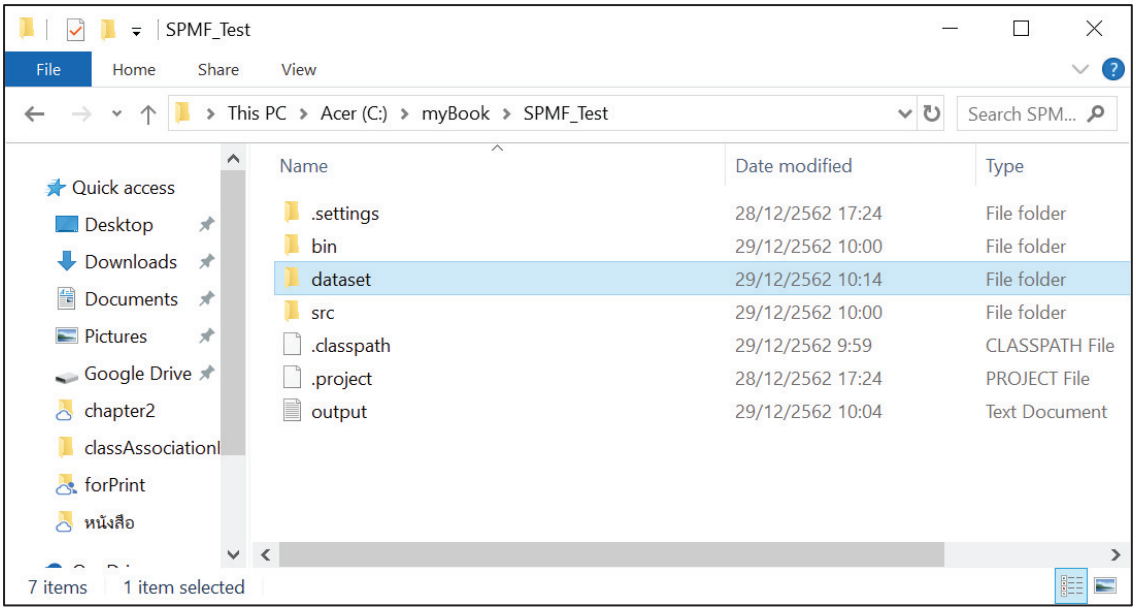
เมื่อทำการติดตั้ง SPMF เสร็จเรียบร้อยแล้ว สามารถทดสอบรันโปรแกรมโดยไปที่โปแกรม Eclipse ทำการเลือกแพ็กเกจ "ca.pfv.spmf.tests" ในหน้าต่าง Package Explorer ซึ่งแพ็กเกจดังกล่าวประกอบไปด้วยตัวอย่างการเรียกใช้ขั้นตอนวิธีต่างๆ เช่น ถ้าต้องการเลือกใช้ขั้นตอนวิธี Apriori ให้ทำการคลิกเลือกไฟล์ " MainTestApriori_saveToFile.java" (ดังรูปที่ ก.12) จากนั้นทำการประมวลผลโปรแกรมโดยคลิกที่ปุ่ม  หรือไปที่เมนู Run -> Run หรือกดปุ่ม Ctrl+F11



รูปที่ ก.12 ตัวอย่างการประมวลผล

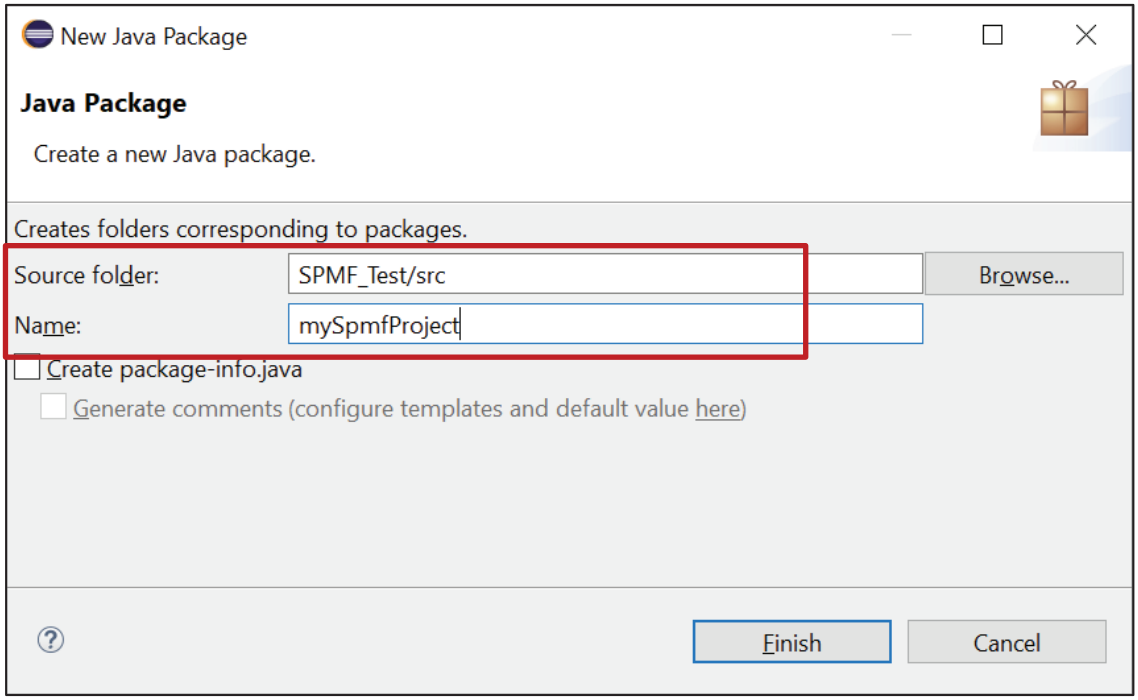
การสร้างไฟล์จาวาขึ้นมาใหม่ สามารถเรียกใช้คลาสต่างๆ ที่อยู่ในแพ็กเกจที่อยู่ใน SPMF ได้ (รายละเอียดของแพ็กเกจต่างๆ จะกล่าวถึงในหัวข้อ ก.3) ตัวอย่างการสร้างไฟล์จาวาและเรียกใช้คลาสใน SPMF มีขั้นตอนดังนี้

1. ทำการสร้างโฟลเดอร์สำหรับเก็บไฟล์ข้อมูล ดังรูปที่ ก.13 สร้างโฟลเดอร์ dataset ในโฟลเดอร์โปรเจค คัดลอกไฟล์ contextPasquier99.txt ใน C:\myBook\spmf\ca\pfv\spmf\test มาไว้ในโฟลเดอร์ dataset



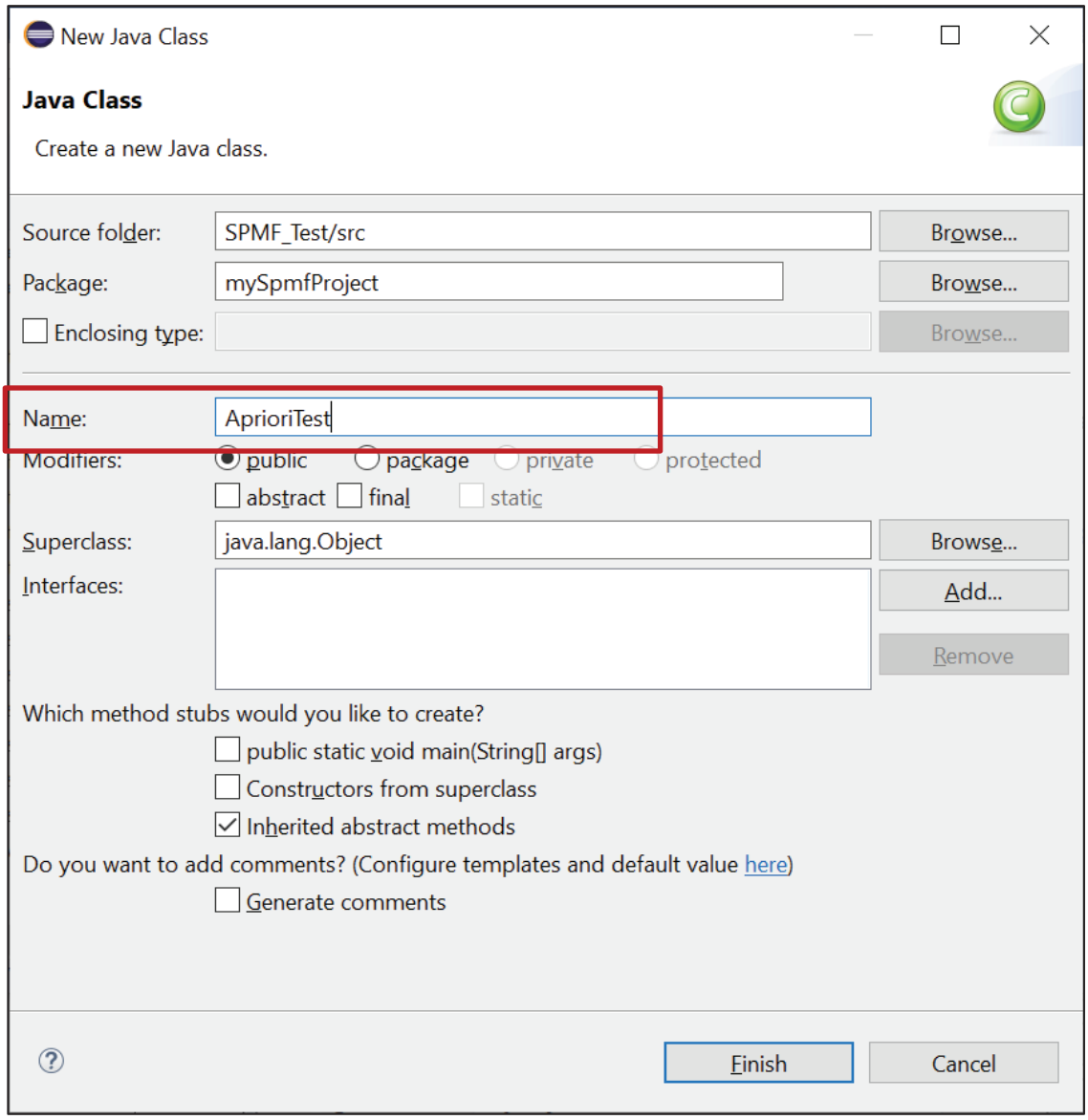
รูปที่ ก.13 ตำแหน่งสำหรับเก็บไฟล์ข้อมูล

2. สร้างแพ็กเกจขึ้นมาใหม่ โดยคลิกขวาที่ src แล้วเลือก New->Package จากนั้นป้อนชื่อแพ็กเกจในช่อง Name: จากนั้นคลิกปุ่ม Finish ดังรูปที่ ก.14 ตั้งชื่อแพ็กเกจว่า mySpmfProject



รูปที่ ก.14 ตัวอย่างการสร้างแพ็กเกจ

3. ทำการสร้างไฟล์จาวาขึ้นมาใหม่ในแพ็กเกจที่สร้างขึ้น โดยคลิกขวาที่ชื่อแพ็กเกจ แล้วเลือก New->Class จากนั้นป้อนชื่อคลาสในช่อง Name: จากนั้นคลิกปุ่ม Finish ดังรูปที่ ก.15 ชื่อคลาส คือ AprioriTest

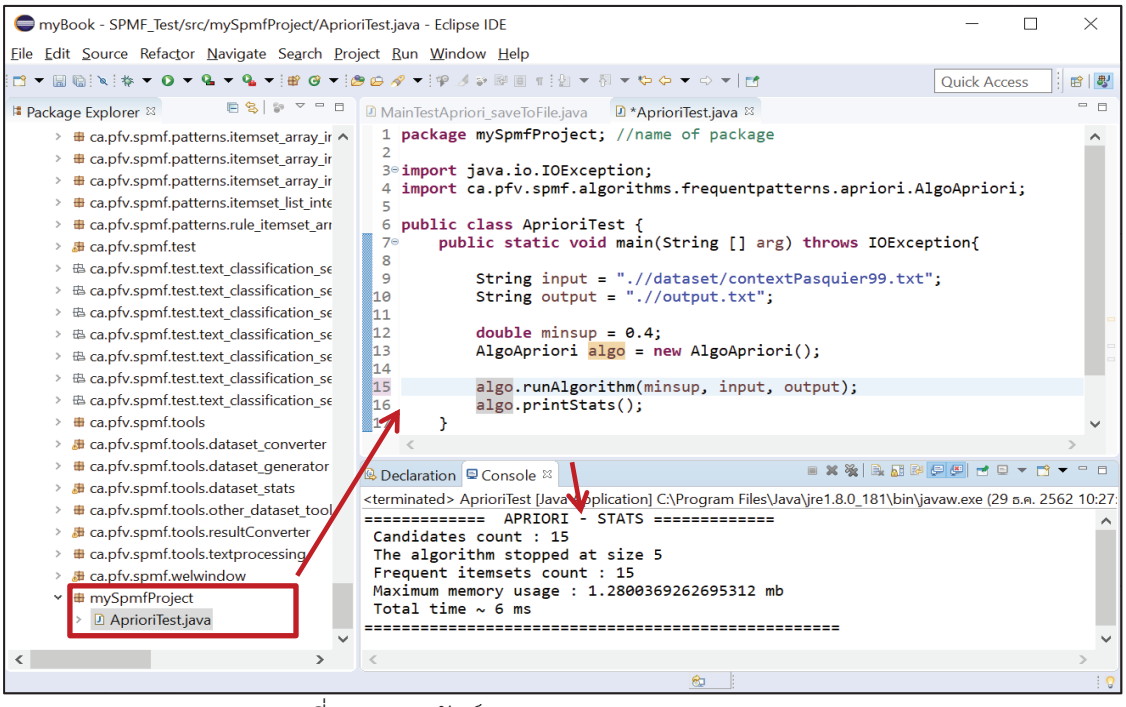


รูปที่ ก.15 ตัวอย่างการสร้างคลาส

3. ทดลองเขียนคำสั่งดังรูปที่ ก.16 ในโปรแกรม Eclipse แล้วลองรัน ผลลัพธ์ของโปรแกรมจะแสดงดังรูปที่ ก.17

```
1.  package myProject; // name of package
2.
3.  import java.io.IOException;
4.  import ca.pfv.spmf.algorithms.frequentpatterns.apriori.AlgoApriori;
5.
6.  public class AprioriTest {
7.          public static void main(String [] arg) throws IOException{
8.
9.                  String input = ".//dataset/contextPasquier99.txt";
10                 String output = ".//output.txt";
11.
12.                double minsup = 0.4;
13.                AlgoApriori algo = new AlgoApriori();
14.
15.                 algo.runAlgorithm(minsup, input, output);
16.                algo.printStats();
17.         }
18. }
```

รูปที่ ก.16 ตัวอย่างคำสั่ง



รูปที่ ก.17 ผลลัพธ์การประมวลผล AprioriTest.java

ก.3 โครงสร้างของ SPMF

ขั้นตอนวิธีต่างๆ ใน SPMF ถูกจัดอยู่ในแพ็กเกจเป็นลำดับชั้น (Package hierarchy) ซึ่งทำให้ง่ายในการดัดแปลงคำสั่งและนำไปใช้ โครงสร้างใน SPMF สามารถอธิบายได้ดังนี้

- ca/pfv/spmf/ : เป็นแพ็กเกจหลักซึ่งประกอบไปด้วยแพ็กเกจอื่นทั้งหมดของ SPMF
- ca/pfv/spmf/algorithms/ : เป็นแพ็กเกจที่ประกอบไปด้วยแพจเกจของขั้นตอนวิธีต่างๆ เช่น
    - ca/pfv/spmf/algorithms/associationrules/ : ประกอบไปด้วยคลาสสำหรับการสร้างกฎความสัมพันธ์
    - ca/pfv/spmf/algorithms/classifiers/ : ประกอบไปด้วยคลาสสำหรับการจำแนกข้อมูล
    - ca/pfv/spmf/algorithms/clustering/ : ประกอบไปด้วยคลาสสำหรับการจัดกลุ่มข้อมูล
    - ca/pfv/spmf/algorithms/frequentpatterns/ : ประกอบไปด้วยคลาสสำหรับการทำเหมืองเซตรายการความถี่
    - ca/pfv/spmf/algorithms/sequential_rules/ : ประกอบไปด้วยคลาสสำหรับการสร้างกฎความสัมพันธ์เชิงลำดับ
    - ca/pfv/spmf/algorithms/sequentialpatterns/ : ประกอบไปด้วยคลาสสำหรับการทำเหมืองรูปแบบลำดับเหตุการณ์
    - ca/pfv/spmf/algorithms/sort/ : ประกอบไปด้วยคลาสสำหรับการเรียงข้อมูล
- ca/pfv/spmf/datastructures : ประกอบไปด้วยคลาสสำหรับสร้างโครงสร้างข้อมูลที่ใช้สำหรับบางขั้นตอนวิธี เช่น Triangular matrix เป็นต้น
- ca/pfv/spmf/documentation/ : ประกอบไปด้วยไฟล์ข้อความ (Text file) ที่ลิงก์ไปยังคู่มือการใช้งานในเว็บไซต์
- ca/pfv/spmf/gui : ประกอบไปด้วยคลาสสำหรับ GUI ของ SPMF
- ca/pfv/spmf/input/ : ประกอบไปด้วยคลาสสำหรับอ่านข้อมูลนำเข้า เช่น ข้อมูลที่เป็นรายการเปลี่ยนแปลง (Transaction databases) ข้อมูลที่เป็นลำดับเหตุการณ์ (Sequence databases) เป็นต้น
- ca/pfv/spmf/patterns/ : ประกอบไปด้วยคลาสสำหรับกำหนดรูปแบบที่ถูกใช้ในขั้นตอนวิธีต่างๆ เช่น เซตรายการ กฎความสัมพันธ์ เป็นต้น

- ca/pfv/spmf/tests/ : ประกอบไปด้วยคลาสตัวอย่างการเรียกใช้แต่ละขั้นตอนวิธี สามารถดูเป็นตัวอย่างเพื่อนำไปประยุกต์ใช้งานได้
- ca/pfv/spmf/tools/ : ประกอบไปด้วยเครื่องมือต่างๆ เช่น เครื่องมือสำหรับการสุ่มข้อมูล เครื่องมือสำหรับสร้างสถิติบนข้อมูลลำดับเหตุการณ์ เครื่องมือสำหรับการแปลงข้อมูลให้เป็นรูปแบบที่ใช้งานกับ SPMF ได้ เป็นต้น

ขั้นตอนวิธีต่างๆ ใน SPMF มีคลาสหลักซึ่งชื่อคลาสขึ้นต้นด้วย Algo.... และสามารถเรียกใช้ขั้นตอนวิธีต่างๆ โดยใช้เมทอด runAlgorithm() เช่น คลาสของขั้นตอนวิธี Apriori คือ AlgoApriori และเมทอดสำหรับการเรียกใช้งานขั้นตอนวิธี Apriori คือ runAlgorithm() เป็นต้น สามารถดูตัวอย่างการเรียกใช้ขั้นตอนวิธีต่างๆ ได้ที่ไฟล์ "MainTestXXXX.java" ในโฟลเดอร์ ca/pfv/spmf/tests/ โดยที่ XXXX คือ ชื่อของขั้นตอนวิธี เช่น ไฟล์ MainTestApriori_saveToFile.java เป็นตัวอย่างการเรียกใช้ขั้นตอนวิธี Apriori ไฟล์ MainTestFPGrowth_saveToFile.java เป็นตัวอย่างการเรียกใช้ขั้นตอนวิธี FPGrowth เป็นต้น ทุกไฟล์ในโฟลเดอร์ ca/pfv/spmf/tests/ จะมีตัวอย่างคำสั่งการใช้งานที่คล้ายกัน เช่น ในไฟล์ "MainTestApriori_saveToFile.java" ประกอบไปด้วยคำสั่งดังนี้

String input = fileToPath("contextPasquier99.txt"); //เป็นคำสั่งสำหรับโหลดข้อมูลที่ชื่อว่า contextPasquier99.txt

String output = ".//output.txt"; // เป็นคำสั่งกำหนดไฟล์ผลลัพธ์

double minsup = 0.4; //เป็นคำสั่งกำหนดค่าสนับสนุนขั้นต่ำ 40%

AlgoApriori algo = new AlgoApriori(); //เป็นคำสั่งสร้างอ๊อบเจ็กต์ของขั้นตอนวิธี Apriori

algo.runAlgorithm (minsup, input, output); //เป็นคำสั่งเรียกใช้ขั้นตอนวิธี Apriori พร้อมกับกำหนดค่าที่จะส่งให้กับขั้นตอนวิธี Apriori ซึ่งประกอบไปด้วยค่าสนับสนุนขั้นต่ำ (minsup) ไฟล์ชุดข้อมูลนำเข้า (input) และไฟล์ผลลัพธ์ (output)

algo.printStatistics(); // เป็นคำสั่งสำหรับแสดงค่าทางสถิติออกมา เช่น จำนวนรูปแบบที่ได้ หน่วยความจำที่ใช้ เวลาที่ใช้ในการประมวลผล

การนำขั้นตอนวิธีไปใช้สามารถทำได้ง่าย เพราะขั้นตอนวิธีที่ขุดค้นรูปแบบเหมือนกันจะถูกเก็บในแพ็กเกจเดียวกัน เช่น ขั้นตอนวิธี Apriori และ FPGrowth เป็นขั้นตอนการทำเหมืองเซตรายการความถี่เหมือนกัน ดังนั้นคลาสที่เกี่ยวข้องจะถูกจัดเก็บในแพ็กเกจเดียวกัน เป็นต้น เมื่อจะเรียกใช้ขั้นตอนวิธีใดๆ ให้ตรวจสอบว่าขั้นตอนวิธีดังกล่าวอยู่ที่แพ็กเกจไหน จากนั้นทำการ import คลาสที่เกี่ยวข้องเพื่อนำมาใช้งานได้เลย

ภาคผนวก ข การติดตั้ง Weka

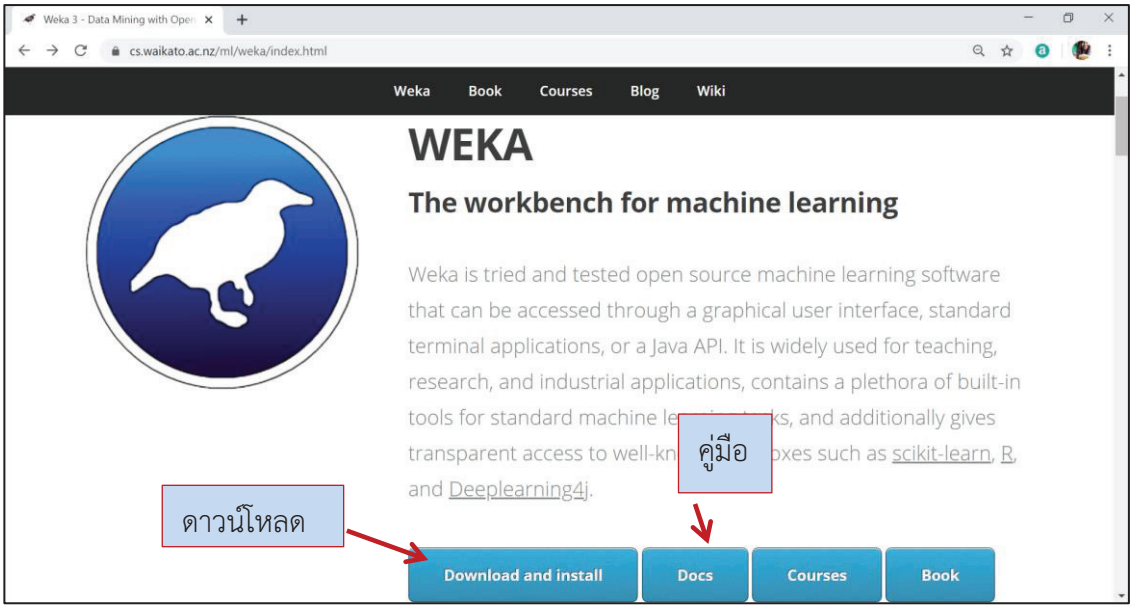
Weka เป็นเครื่องมือในการทำเหมืองข้อมูลที่ใช้อย่างแพร่หลาย ถูกพัฒนาขึ้นด้วยภาษาจาวา โดยกลุ่มคนซึ่งนำโดย Professor Albert Bifet ในมหาวิทยาลัย Waikato University ประเทศนิวซีแลนด์ Weka สามารถใช้งานได้ฟรี สามารถนำไปติดตั้งบนระบบปฏิบัติการ Windows, Linuk และ Max ได้ นอกจากนี้ยังมีเครื่องมือหลากหลายที่ช่วยในการทำเหมืองข้อมูล เช่น การเลือกคุณลักษณะ การจัดกลุ่ม การจำแนกข้อมูล การทำกราฟวิเคราะห์ข้อมูล การเตรียมข้อมูลข้อความ เป็นต้น โปรแกรม Weka มีหน้าจอรองรับการทำงานสำหรับผู้ใช้ที่ไม่มีความรู้เกี่ยวกับภาษาจาวา และสามารถเรียกใช้ API เพื่อเขียนโปรแกรมติดต่อกับ Weka API ได้

ในหนังสือเล่มนี้ประยุกต์ใช้ Weka API สำหรับการจำแนกเชิงความสัมพันธ์ ซึ่งจำเป็นต้องติดตั้ง Weka API ก่อนการเรียกใช้งาน โดยรายละเอียดการติดตั้ง Weka API มีดังหัวข้อย่อยต่อไปนี้

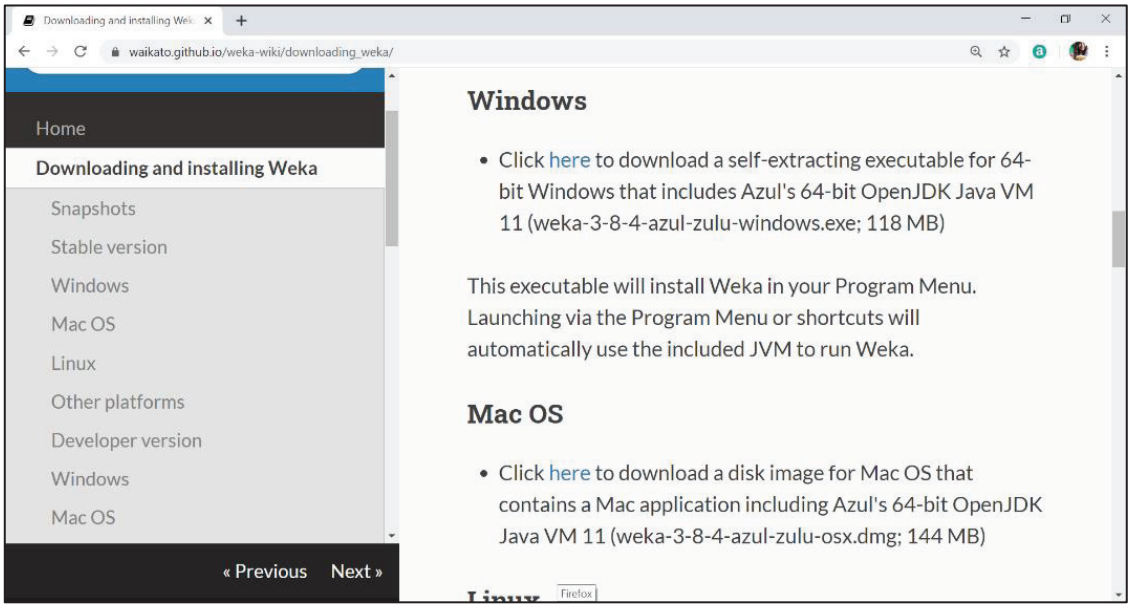
ข.1 เริ่มต้นใช้ Weka API

Weka API พัฒนาขึ้นเพื่อให้ผู้ใช้สามารถเรียกใช้คลาสหรือปรับปรุงคำสั่งต่างๆ ที่อยู่ใน Weka ได้ ทำให้มีความยืดหยุ่นกว่าการใช้งานโปรแกรม Weka ส่วนที่เป็น GUI นอกจากนี้ในเว็บไซต์ของ Weka (https://www.cs.waikato.ac.nz/ml/weka/) มีคำอธิบายการเรียกใช้คลาสต่างๆ โดยในเว็บไซต์ประกอบด้วยเมนูที่น่าสนใจดังต่อไปนี้

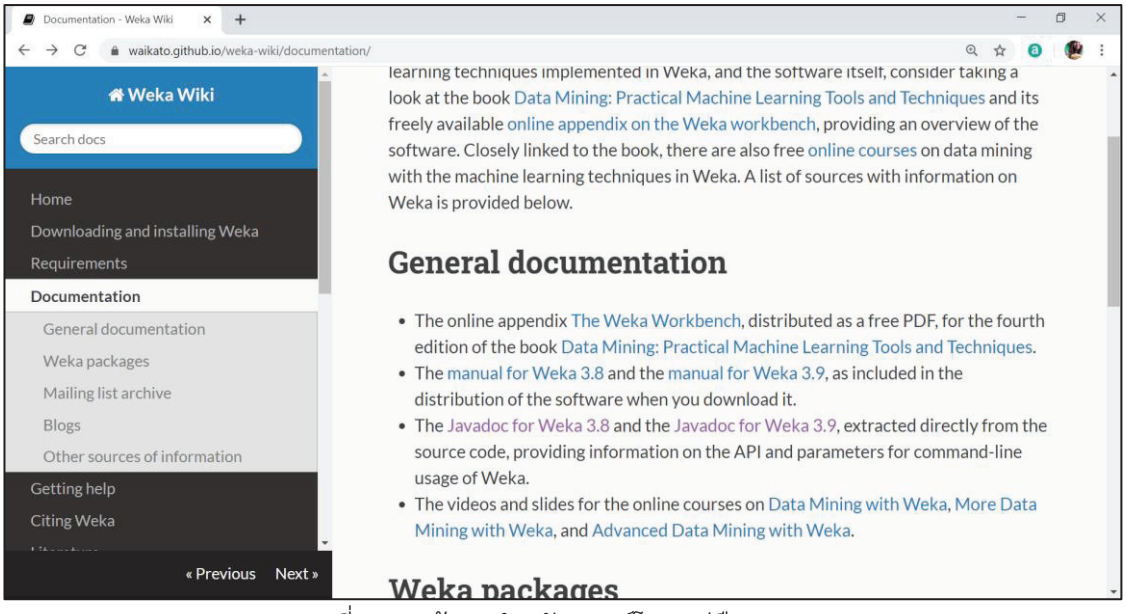
เมนู Weka แสดงรายละเอียดเบื้องต้นเกี่ยวกับ Weka (ดังรูปที่ ข.1) สามารถดาวน์โหลดโปรแกรม Weka โดยคลิกเข้าไปที่ปุ่ม Download and install จะได้หน้าเว็บดังรูปที่ ข.2 ซึ่งจะเห็นได้ว่าสามารถลง Weka ได้บนระบบปฏิบัติการที่หลากหลาย ถ้าผู้ใช้ต้องการทราบรายละเอียดการใช้งาน Weka สามารถคลิกที่ปุ่ม Docs (ในรูปที่ ข.1) จะแสดงหน้าเว็บดังรูปที่ ข.3 ซึ่งสามารถคลิกเข้าไปดูรายละเอียดการใช้งาน Weka API เวอร์ชั่นต่างๆ ได้ เช่น เมื่อคลิกไปที่ลิงก์ Weka 3.9 จะแสดงรายละเอียดการใช้งานแต่ละคลาสดังหน้าจอดังรูปที่ ข.4



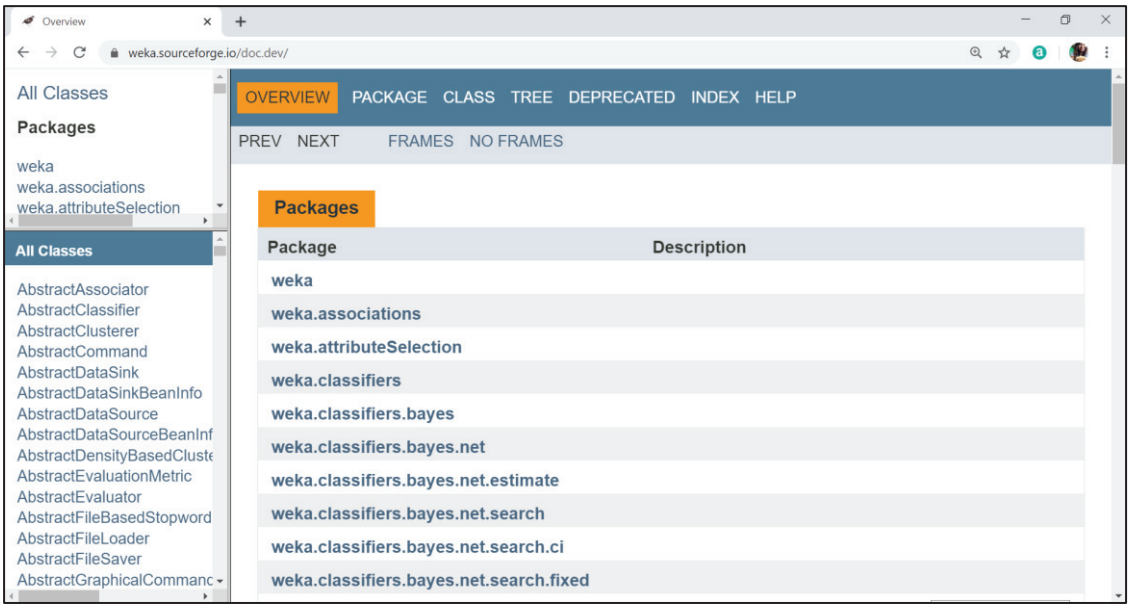
รูปที่ ข.1 แสดงหน้าเว็บในเมนู Software



รูปที่ ข.2 หน้าจอสำหรับดาวน์โหลด Weka

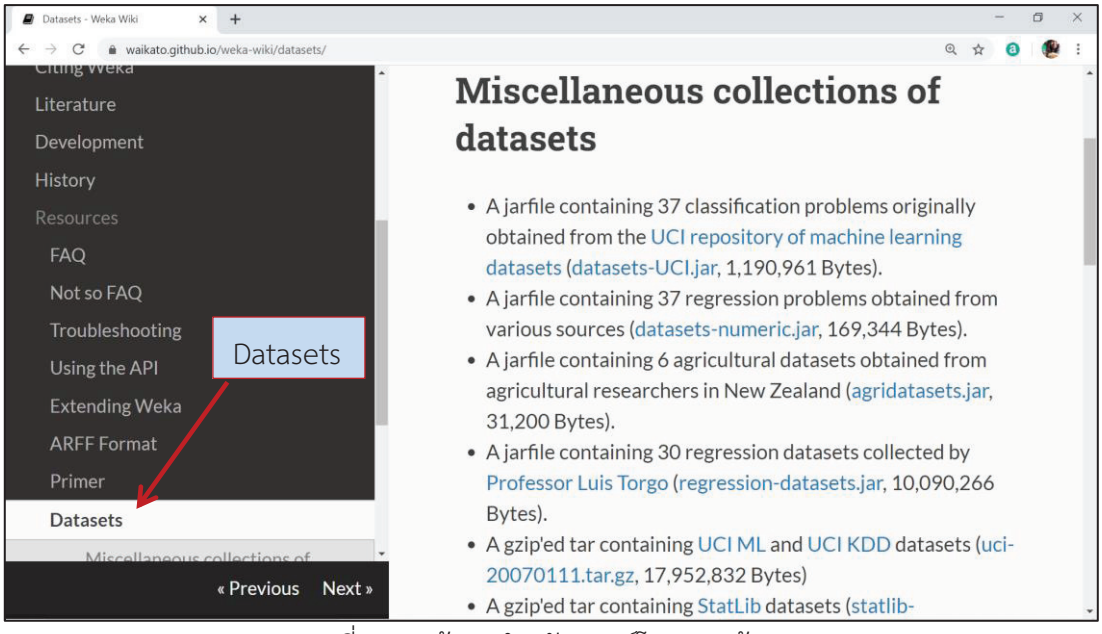


รูปที่ ข.3 หน้าจอสำหรับดาวน์โหลดคู่มือ Weka



รูปที่ ข.4 หน้าจอคู่มือการใช้งาน Weka API

นอกจากนี้แล้วยังสามารถดาวน์โหลดชุดข้อมูลมาตรฐานได้โดยคลิกเข้าไปที่ลิงก์ Datasets (ดังฝั่งซ้ายหน้าจอรูปที่ ข.5) ซึ่งชุดข้อมูลทั้งหมดมีการจัดรูปแบบให้สามารถใช้กับ Weka และชุดข้อมูลดังกล่าวเป็นชุดข้อมูลที่ใช้ในการทำเหมืองข้อมูลอย่างแพร่หลาย

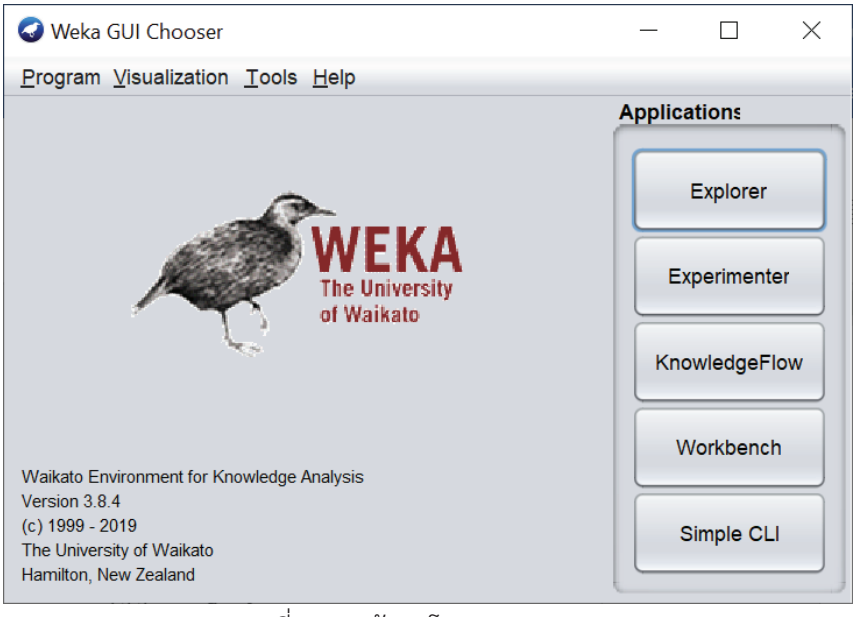


รูปที่ ข.5 หน้าจอสำหรับดาวน์โหลดชุดข้อมูล

ข.2 การติดตั้ง Weka API

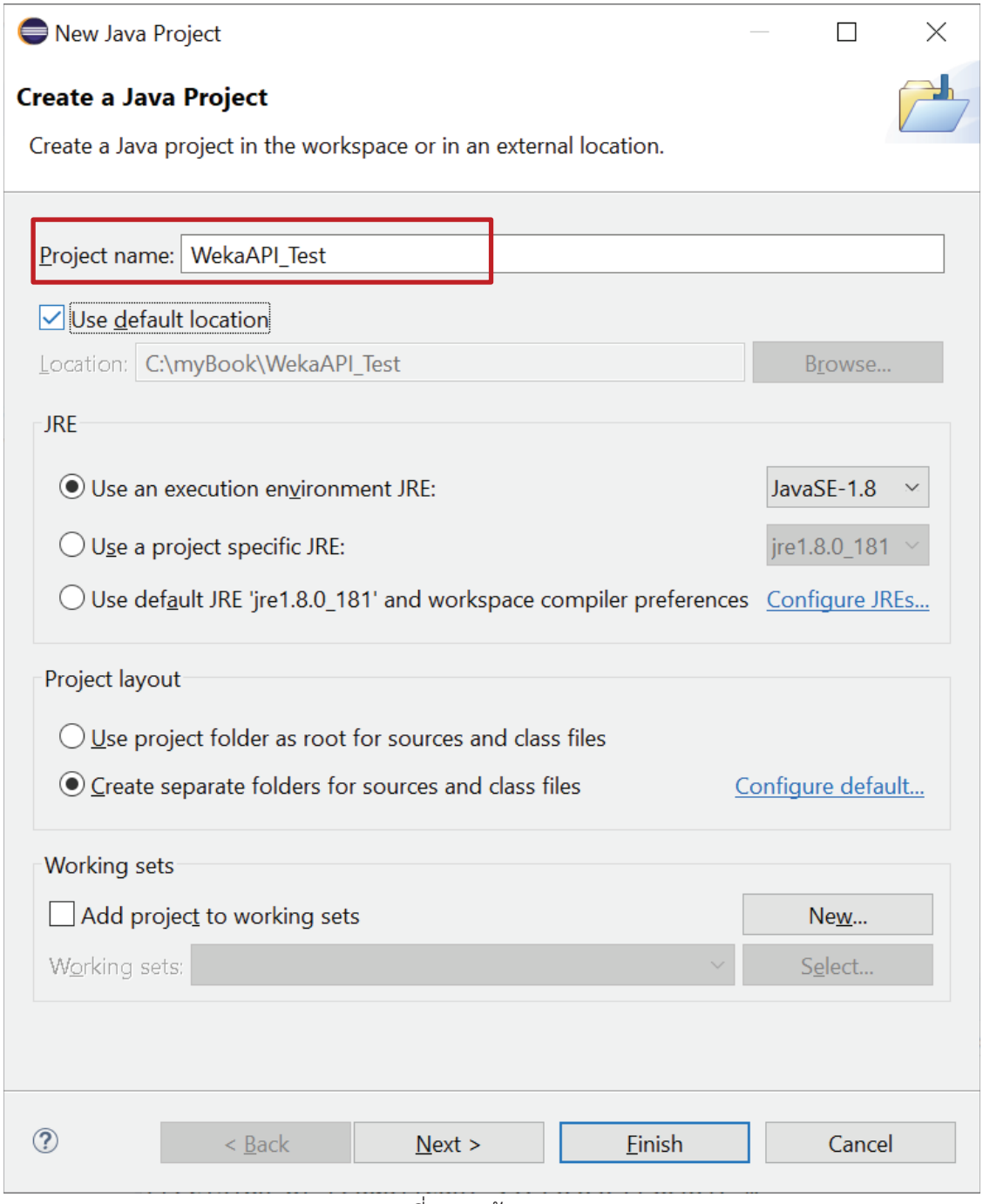
การเรียกใช้คลาสใน Weka จำเป็นจะต้องติดตั้ง Weka API โดยหนังสือเล่มนี้จะแสดงตัวอย่างการติดตั้ง Weka API บนโปรแกรม Eclipse โดยขั้นตอนการติดตั้งมีดังต่อไปนี้

1. ทำการดาวน์โหลดไฟล์ติดตั้งในเว็บไซต์ของ Weka จากนั้นทำการติดตั้ง Weka เมื่อติดตั้งเสร็จจะได้หน้าตาโปรแกรมดังรูปที่ ข.6 โดยเวอร์ชั่นที่ติดตั้งเป็นเวอร์ชั่น 3.8.4

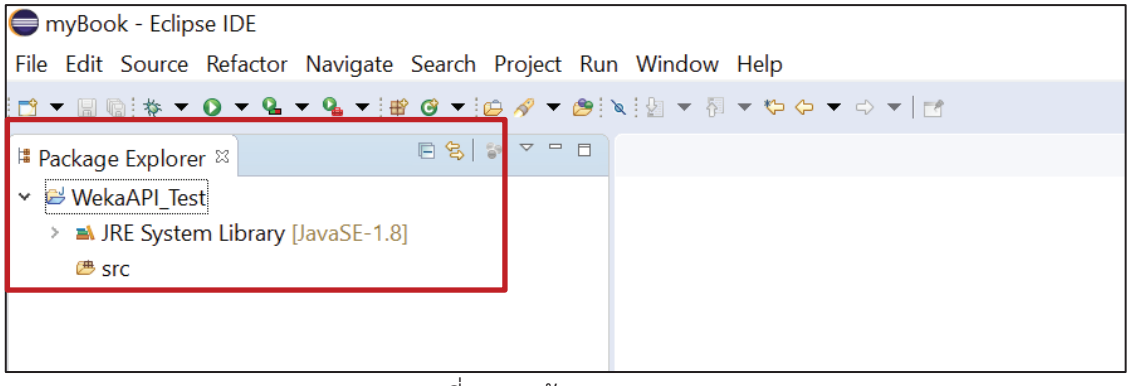


รูปที่ ข.6 หน้าจอโปรแกรม Weka

2. เปิดโปรแกรม Eclipse แล้วทำการสร้าง Project ใหม่ใน Eclipse โดยคลิกเข้าไปที่เมนู File->New->Java Project จากนั้นทำการป้อนชื่อ Project ในช่อง Project name: (ดังรูปที่ ข.7) แล้วคลิกปุ่ม Finish ชื่อ Project จะปรากฏในหน้าต่าง Explorer ดังรูปที่ ข.8

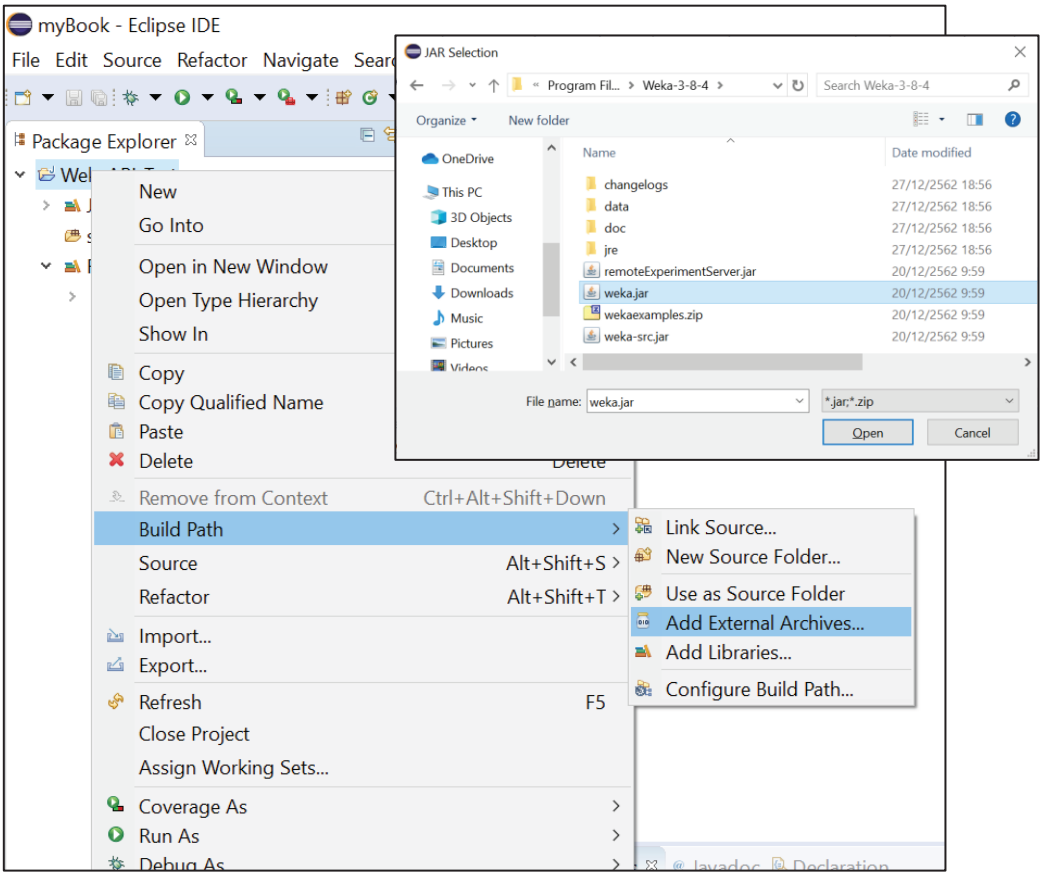


รูปที่ ข.7 สร้าง Project

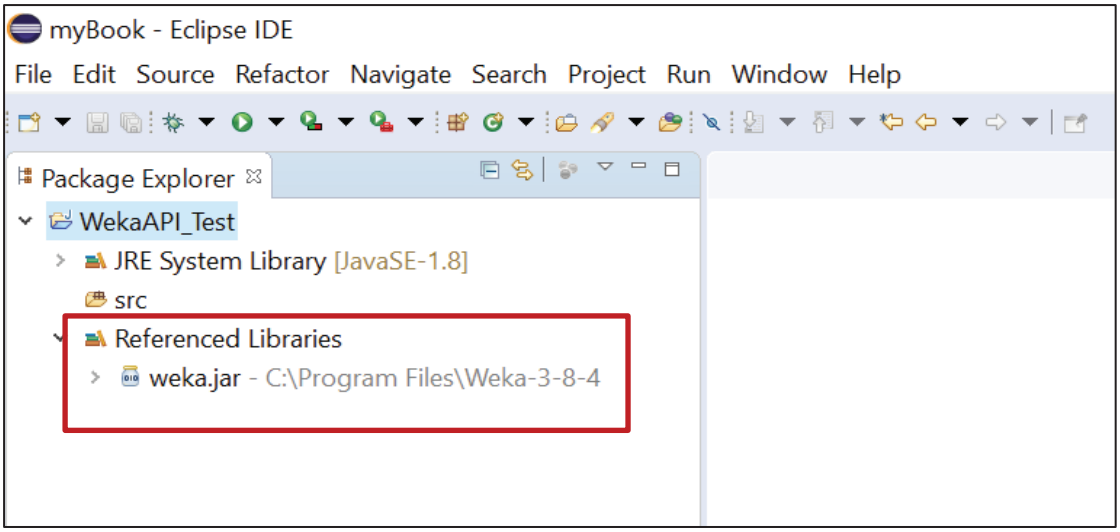


รูปที่ ข.8 หน้าจอ explorer

3. คลิกขวาที่ชื่อ Project (ตามตัวอย่างในรูปที่ ข.8 ให้คลิกตรงชื่อโปรเจค WekaAPI_Test) แล้วคลิกเลือก Build Path-> add External Archives.... ดังรูปที่ ข.9 แล้วทำการเลือกไฟล์ weka.jar (ไฟล์ weka.jar อยู่ในโฟลเดอร์ที่ติดตั้งโปรแกรม Weka เช่น C:\Program Files\Weka-3-8-4) คลิกปุ่ม Open ซึ่งจะได้ผลลัพธ์ดังรูปที่ ข.10 เป็นการเสร็จสิ้นการติดตั้ง Weka API



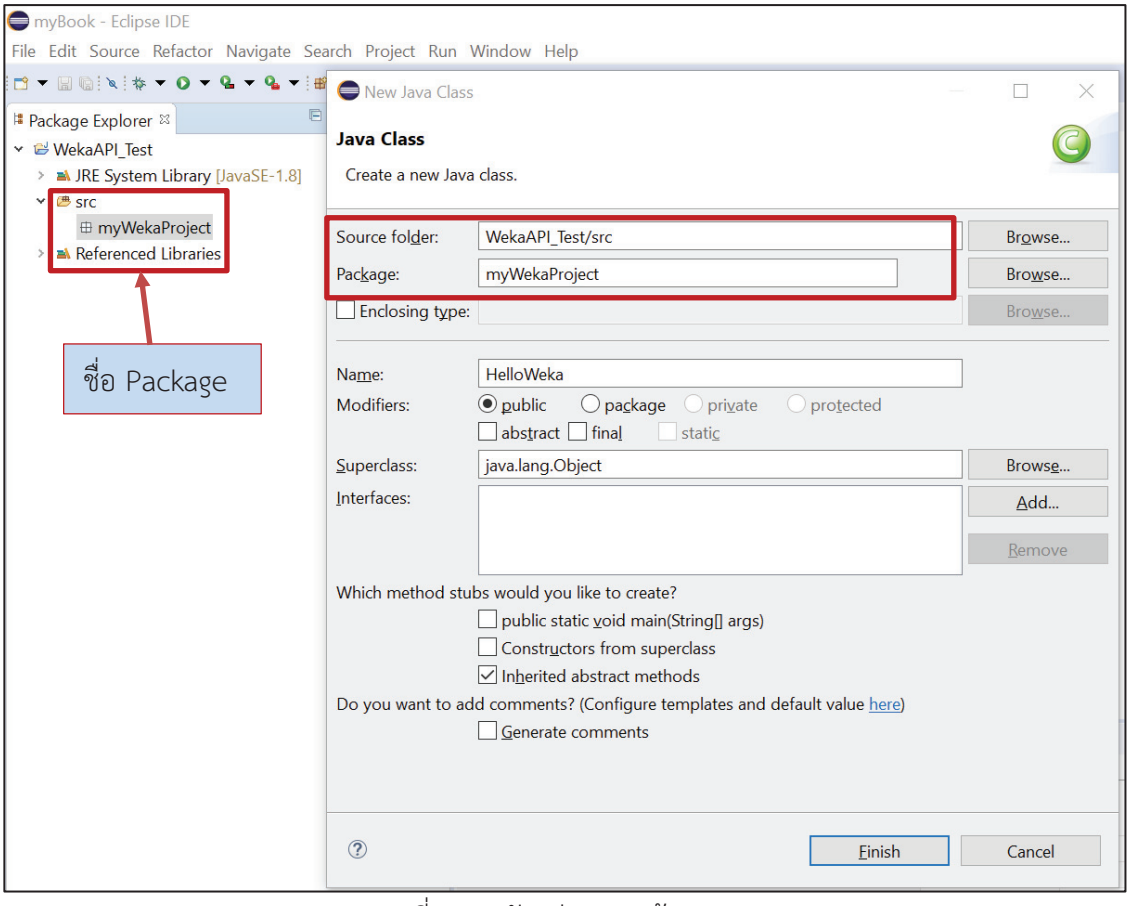
รูปที่ ข.9 สร้างเส้นทางเชื่อมต่อ weka.jar



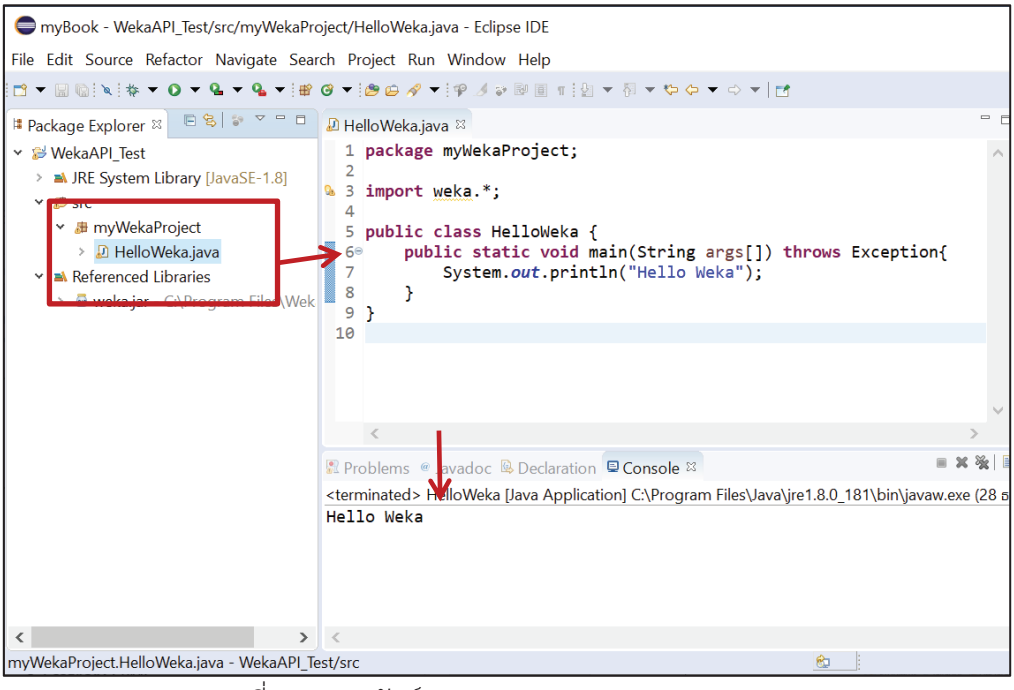
รูปที่ ข.10 การเชื่อมต่อ Weka API

เมื่อทำการติดตั้ง Weka API เรียบร้อยแล้ว สามารถทดสอบรันโปรแกรมดังนี้

- สร้าง Package โดยคลิกขวาที่ชื่อโปรเจค เลือกเมนู New->Package ป้อนชื่อ Package ในช่อง Name: จากนั้นคลิกปุ่ม Finish (ในตัวอย่างชื่อ Package คือ myWekaProject)
- สร้าง Class โดยคลิกขวาที่ชื่อ Package เลือกเมนู New->Class แล้วป้อนชื่อ Class ในช่อง Name: (ในตัวอย่างรูปที่ ข.11 ชื่อ Class คือ HelloWeka) จากนั้นคลิกที่ปุ่ม Finish
- ทดลองเขียนคำสั่งดังตัวอย่างในรูปที่ ข.12 จากนั้นคลิกที่ปุ่ม  เพื่อประมวลผล



รูปที่ ข.11 ตัวอย่างการสร้างคลาส



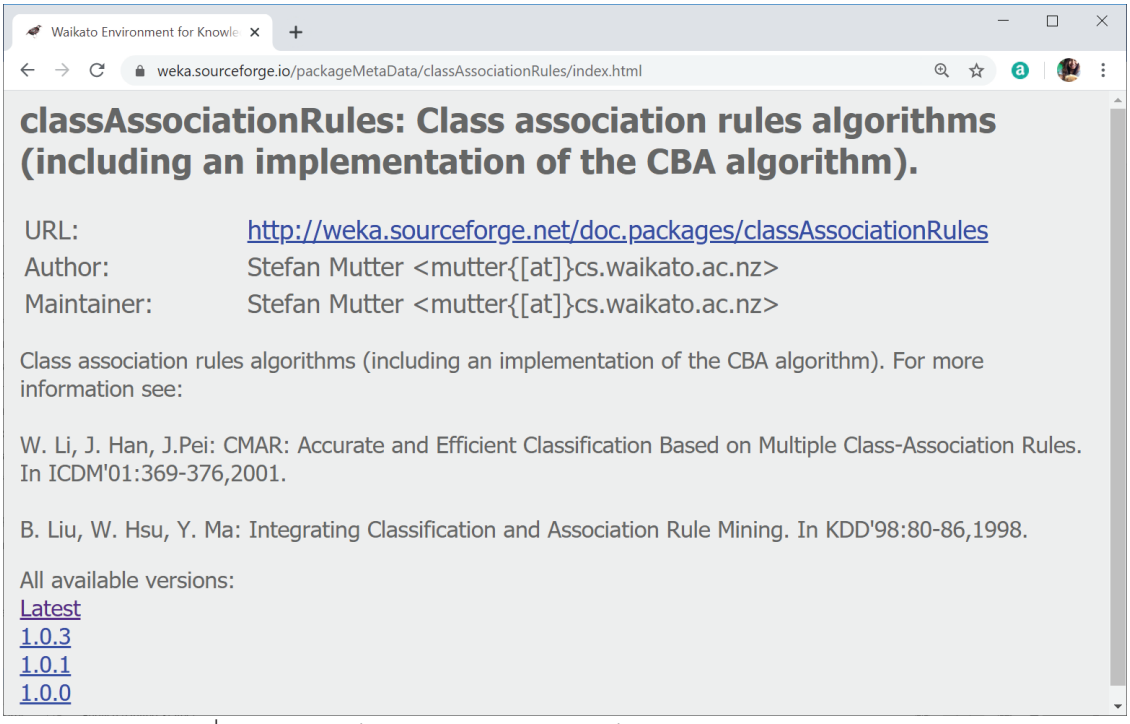
รูปที่ ข.12 ผลลัพธ์การประมวลผล HelloWeka.java

ข.3 การติดตั้ง JCBA บน Weka สำหรับการจำแนกเชิงความสัมพันธ์

Weka มีคลาส JCBA ที่ใช้ในการจำแนกเชิงความสัมพันธ์ ซึ่งคลาสดังกล่าวพัฒนาบนพื้นฐานของขั้นตอนวิธี CBA และพัฒนาโดยใช้ภาษาจาวา การเรียกใช้คลาสดังกล่าวจำเป็นต้องทำการติดตั้งแพ็กเกจ classAssociationRules เพิ่มเติมใน Weka เนื่องคลาสพื้นฐานที่อยู่ใน Weka ไม่มีคลาส JCBA การติดตั้ง classAssociationRules สามารถทำได้ง่าย โดยประกอบไปด้วยขั้นตอนดังต่อไปนี้

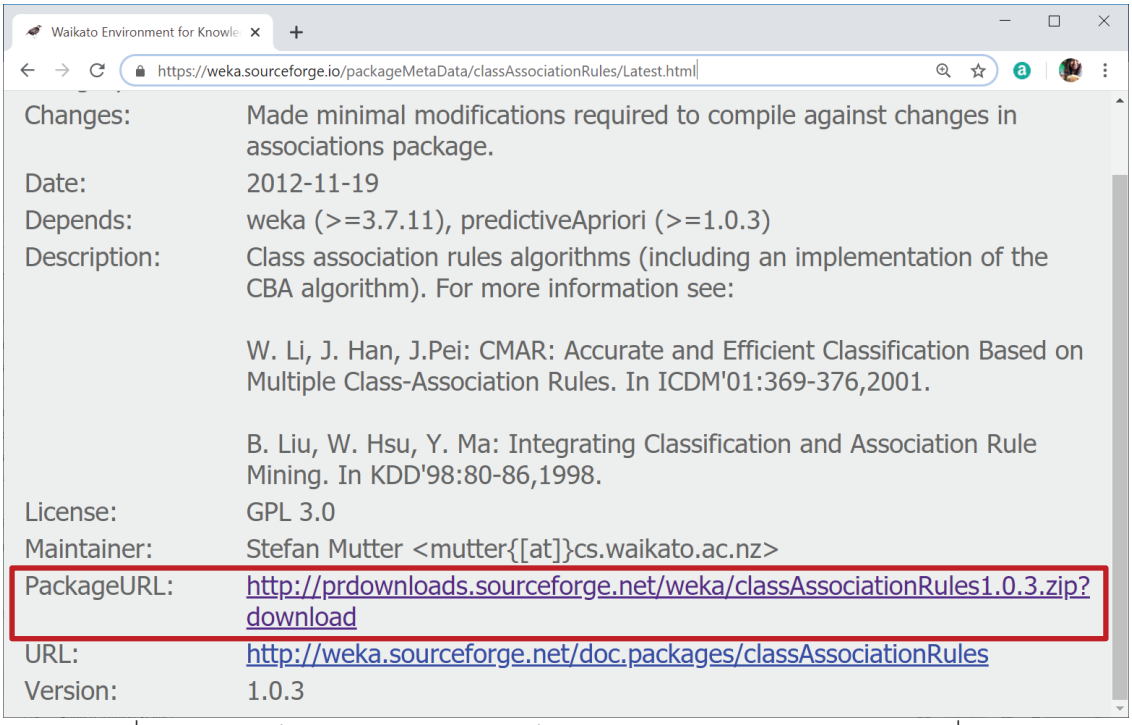
1. เข้าไปที่ลิงก์ข้างล่างจะได้หน้าเว็บดังรูปที่ ข.13 จากนั้นทำการคลิกเลือกเวอร์ชั่นที่ต้องการติดตั้ง

http://weka.sourceforge.net/packageMetaData/classAssociationRules/index.html



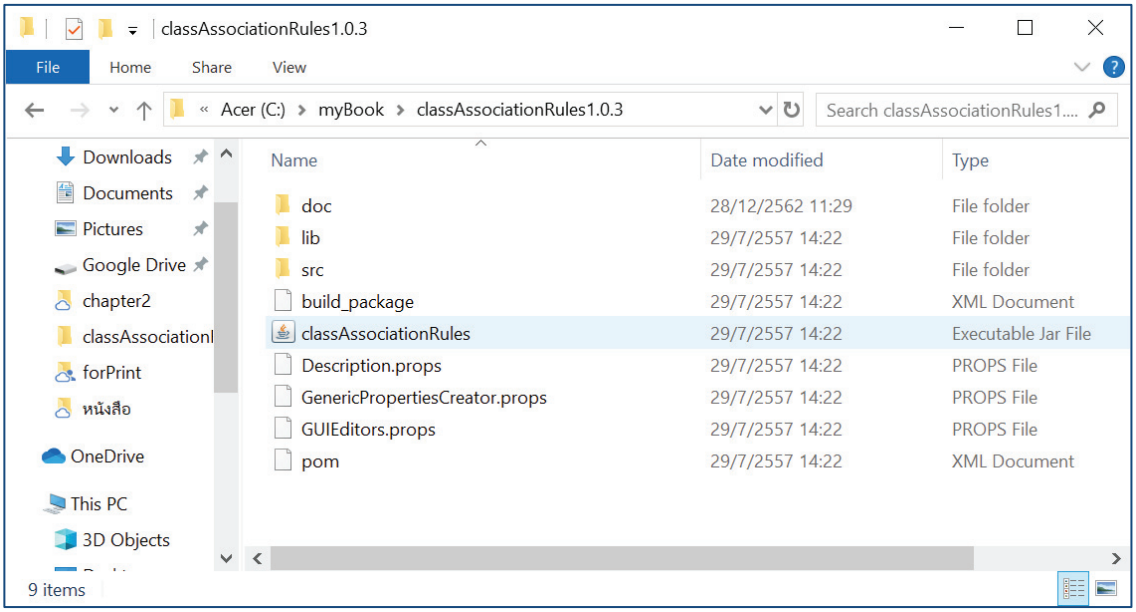
รูปที่ ข.13 หน้าเว็บสำหรับดาวน์โหลดแพ็กเกจ classAssociationRules

2. คลิกลิงก์สำหรับดาวน์โหลดตรง PackageURL: ดังรูปที่ ข.14



รูปที่ ข.14 หน้าเว็บสำหรับดาวน์โหลดแพ็กเกจ classAssociationRules เวอร์ชั่น 1.0.3

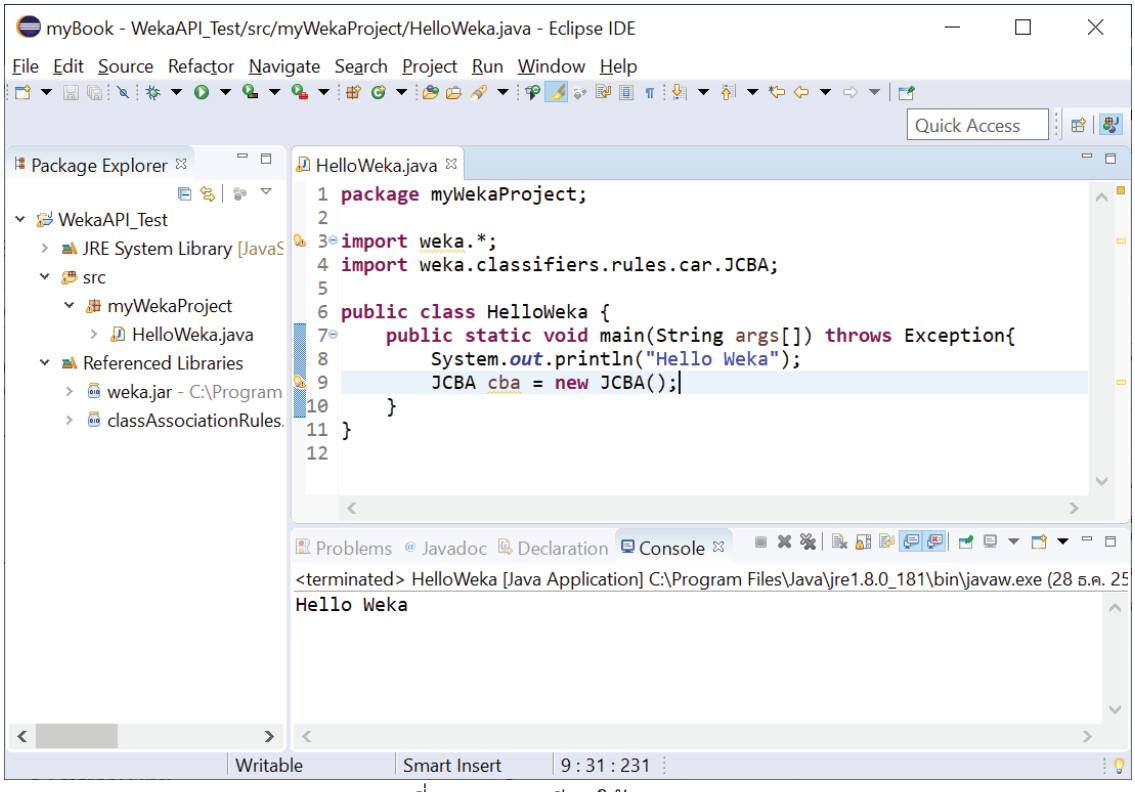
3. จากนั้นทำการแตกไฟล์ (Unzip) จะได้ไฟล์ทั้งหมดดังรูปที่ ข.15



รูปที่ ข.15 แสดงไฟล์ที่ได้จากการ unzip

4. ทำการติดตั้งแพ็กเกจ classAssociationRules ในโปรแกรม Eclipse เหมือนกับการติดตั้ง Weka API ในโปรแกรม Eclipse โดยทำการคลิกขวาที่ชื่อโปรเจค แล้วคลิกเลือก Build Path->add External Archives... จากนั้นทำการเลือกไฟล์ classAssociationRules.jar แล้วคลิกปุ่ม Open เป็นการสิ้นสุดการติดตั้งแพ็กเกจ

ทดลองเขียนโปรแกรมทดสอบ โดยใช้คำสั่ง import weka.classifiers.rules.car.JCBA; เพื่อนำเข้าคลาส JCBA และสร้างอ๊อบเจ็กต์ของคลาส JCBA ดังคำสั่งที่แสดงในรูปที่ ข.16 จากนั้นทำการประมวลผลจะได้ผลลัพธ์ดังรูป



รูปที่ ข.16 การเรียกใช้คลาส JCBA

บรรณานุกรม

Agrawal, R., Imielinski, T., and Swami, A. (1993). Mining Association Rules between Sets of Items in Large Databases. *Proceedings of the 1993 ACM SIGMOD Internatinal Conference on Mangament of Data*, USA, pp. 207-216.

Agrawal, R., and Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. *Proceedings of the 20th International Conference on Very Large Data Bases*, Santiago, Chile, pp. 487-499.

Ayres, J., Flannick, J., Gehrke, J., and Yiu, T. (2002). Sequential Pattern Mining using a Bitmap Representation. *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, pp. 429-435.

Beyer, K., and Ramakrishnan, R. (1999). Bottom-up Computation of Sparse and Iceberg CUBE. *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, Philadelphia, Pennsylvania, USA, pp. 359-370 .

Fernando, B., Fromont, E., and Tuytelaars, T. (2012). Effective Use of Frequent Itemset Mining for Image Classification. *Proceeding of European Conference on Computer Vision – ECCV 2012*, Firenze, Italy, pp. 214-227.

Fournier-Viger, P., Gomariz, A., Campos, M., and Thomas, R. (2014). Fast Vertical Mining of Sequential Patterns Using Co-occurrence Information. *Proceeding of 18th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, Tainan, Taiwan, pp. 40-52.

Fournier-Viger, P., Gomariz, A., Gueniche, T., Mwamikazi, E., and Thomas, R. (2013). TKS: Efficient Mining of Top-K Sequential Patterns. *Proceeding of International Conference on Advanced Data Mining and Applications*, China, pp. 109-120.

Fournier-Viger, P., Gueniche, T., and Tseng, V. S. (2012). Using Partially-Ordered Sequential Rules to Generate More Accurate Sequence Predictio. *Proceeding of International Conference on Advanced Data Mining and Applications*, China, pp. 431-442.

Fournier-Viger, P., Lin, J. C.-W., Gomariz, A., Gueniche, T., Soltani, A., Deng, Z., and Lam, H. T. (2016). The SPMF Open-Source Data Mining Library Version 2. *Proceeding of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Italy, pp. 36-40.

Fournier-Viger, P., Gomariz, A., Gueniche, T., Soltani, A., Wu, C.-W., and Tseng, V. (2016). SPMF: A JAVA Open-Source Pattern Mining Library. *Machine Learning Research*, 15(1), pp. 3389-3393.

Fournier-Viger, P., Nkambou, R., and Tseng, V. S.-M. (2011). RuleGrowth: Mining Sequential Rules Common to Several Sequences by Pattern-growth. *Proceedings of the 2011 ACM Symposium on Applied Computing*, TaiChung, Taiwan, pp. 956-961.

Fournier-Viger, P., Wu, C.-W., Gomariz, A., and Tseng, V. S. (2014). VMSP: Efficient Vertical Mining of Maximal Sequential Patterns. *Proceeding of Canadian Conference on Artificial Intelligence*, Canada, pp. 83-94.

Fournier-Viger, P., Wu, C.-W., and Tseng, V. S. (2013). Mining Maximal Sequential Patterns without Candidate Maintenance. *Proceeding of International Conference on Advanced Data Mining and Applications*, China, pp. 169-180.

Fournier Viger, P., Lin, C.-W., Rage, U., Koh, Y. S., and Thomas, R. (2017). A Survey of Sequential Pattern Mining. *Data Science and Pattern Recognition*, *1*(1), pp.54-75.

Fumarola, F., Lanotte, P. F., Ceci, M., and Malerba, D. (2016). CloFAST: Closed Sequential Pattern Mining Using Sparse and Vertical Id-lists. *Knowledge and Information Systems, 48*(2), pp. 429-463.

Glatz, E., Mavromatidis, S., Ager, B., and Dimitropoulos, X. (2014). Visualizing Big Network Traffic Data using Frequent Pattern Mining and Hypergraphs. *Computing, 96*(1), pp.27-38. doi: 10.1007/s00607-013-0282-8

Gomariz, A., Campos, M., Marin, R., and Goethals, B. (2013). ClaSP: An Efficient Algorithm for Mining Frequent Closed Sequences. *Proceeding of Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Australia, pp. 50-61.

Grahne, G., and Zhu, J. (2003). Efficiently Using Prefix-trees in Mining Frequent Itemsets. *Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations*, Florida, USA, pp. 1-10.

Han, J., Kamber, M., and Pei, J. (2012). 1 - Introduction Data Mining (Third Edition). *Data Mining Concepts and Techniques* (pp. 1-38). Boston: Morgan Kaufmann.

Han, J., Pei, J., and Yin, Y. (2000). Mining Frequent Patterns without Candidate Generation. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 29*(2), Texas, USA, pp.1-12, doi: 10.1145/335191.335372.

Huang, Z., Zhou, Z., He, T., and Wang, X. (2011). ACAC: Associative Classification Based on All-Confidence. *Proceedings of IEEE International Conference on Granular Computing*, Kaohsiung, Taiwan, pp.289-293.

Jamsheela, O., and Raju, G. (2015). Frequent Itemset Mining Algorithms: A Literature Survey. *Proceeding of the 2015 IEEE International Advance Computing Conference (IACC)*, Banglore, India, pp.1099-1104.

Li, W., Han, J., and Pei, J. (2001). CMAR: Accurate and Efficient Classification based on Multiple Class-association Rules. *Proceedings of the International Conference on Data Mining*, CA, USA, pp. 369-376.

Li, X., Qin, D., and Yu, C. (2008). ACCF: Associative Classification Based on Closed Frequent Itemsets. *Proceedings of the Fith International Conference on Fuzzy Systems and Knowledge Discovery*, Shandong, China, pp. 380-384.

Liu, B. (1998). Integrating Classification and Association Rule Mining. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, New York, USA, pp. 80-86.

Lucchese, C., Orlando, S., and Perego, R. (2006). Fast and Memory Efficient Mining of Frequent Closed Itemsets. *IEEE Transactions on Knowledge and Data Engineering*, *18*(1), pp. 21-36.

Mukherjee, A., Liu, B., and Glance, N. (2012). Spotting Fake Reviewer Groups in Consumer Reviews. *Proceedings of the 21st International Conference on World Wide Web*, Lyon, France. pp. 191-200.

Nasingkhun, S., and Songram, P. (2018). Predicting Stroke by Combination of Sequence Pattern Mining and Associative Classification, *Proceeding of International Conference on Information Technology (InCIT)*, Khon Kaen, Thailand, pp. 1-6.

Pasquier, N., Bastide, Y., Taouil, R., and Lakhal, L. (1999). Discovering Frequent Closed Itemsets for Association Rules. *Proceeding of International Conference on Database Theory*, Israel, pp. 398-416.

Pei, J., Han, J., and Mao, R. (2000). Closet: An Efficient Algorithm for Mining Frequent Closed Itemsets*. Proceeding of the ACM-SIGMOD International Workshop on Data Mining and Knowledge Discovery (DMKD 2000)*, Dallas, Texas, USA, pp. 21-30.

Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., Hsu, M.-C. (2004). Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach. *IEEE Transactions on Knowledge and Data Engineering, 16*(11), pp.1424-1440.

Petitjean, F., Li, T., Tatti, N., and Webb, G. I. (2016). Skopus: Mining Top-k Sequential Patterns Under Leverage. *Data Mining and Knowledge Discovery*, *30*(5), pp. 1086-1111.

Pinto, H., Han, J., Pei, J., Wang, K., Chen, Q., and Dayal, U. (2001). Multi-dimensional Sequential Pattern Mining. *Proceedings of the 10th International Conference on Information and Knowledge Management*, Atlanta, Georgia, USA, pp. 81-88.

Pramod, S., and Vyas, O. (2010). Survey on Frequent Itemset Mining Algorithms*. International Journal of Computer Applications*, *1*(15), pp. 86-91.

Shaukat Dar, K., Zaheer, S., and Nawaz, I. (2015). Association Rule Mining: An Application Perspective. *International Journal of Computer Science and Innovation*, *2015*(1), pp. 29-38.

Singh, P. K., and Husain, M. S. (2014). Methodological Study of Opinion Mining and Sentiment Analysis Techniques. *International Journal on Soft Computing*, *5*(1), pp. 11-21.

Songram, P., Boonjing, V., and Intakosum, S. (2006). Closed Multidimensional Sequential Pattern Mining. *International Journal of Knowledge Management Studies*, *2*(4), pp. 460-479.

Srikant, R., and Agrawal, R. (1996). Mining Sequential Patterns: Generalizations and Performance Improvements. *Proceeding of International Conference on Advances in Database Technology*, France, pp. 1-17.

Thabtah, F. (2007). A Review of Associative Classification Mining. *Knowledge Engineering Review*, *22*(1), pp. 37-65.

Thabtah, F. A., Cowling, P., and Peng, Y. (2004). MMAC: A New Multi-class, Multi-label Associative Classification Approach. *Proceeding of the 4th IEEE International Conference on Data Mining (ICDM'04)*, Brighton, UK, pp. 1-8.

Thabtah, F. A., Cowling, P., and Peng, Y. (2005). MCAR: Multi-class Classification Based on Association Rule Approach. *Proceedings of the 3rd IEEE Internatinal Conference on Computer Systems and Applicatins*, Cario, Ebypt, pp. 1-7.

Tzvetkov, P., Yan, X., and Han, J. (2005). TSP: Mining Top-K Closed Sequential Patterns. *Knowledge and Information Systems*, *7*(4), pp. 438-457.

Uno, T., Asai, T., Uchida, Y., and Arimura, H. (2004). LCM ver.2: Effient Mining Algorithms for Frequent/Closed/Maximal Itemsets. *Proceeding of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI 2004)*, Brighton, UK. pp. 1-11.

Wang, J., and Han, J. (2004). BIDE: Efficient Mining of Frequent Closed Sequences. *Proceeding of 20th International Conference on Data Engineering,* Boston, USA, pp. 79-81.

Wang, J., and Pei, J. (2003). Closet+: Searching for the Best Strategies for Mining Frequent Closed Itemsets. *Proceeding of the 9th ACM-SIGKDD Internation Conference on Knowledge Discovery and Data Mining (KDD 2003)*, Washington D.C., USA, pp. 236-245.

Yan, X., Han, J., and Afshar, R. (2003). CloSpan: Mining: Closed Sequential Patterns in Large Dataset. *Proceedings of the Third SIAM International Conference on Data Mining*, San Francisco, CA, USA, pp. 166-177.

Yin, X., and Han, J. (2003). CPAR: Classification Based on Predictive Association Rules. *Preceedings of the SIAM International Conference on Data Mining*, San Francisco, CA, pp. 1-5.

Zaki, M. J. (2000). Scalable Algorithms for Association Mining. *IEEE Transactions on Knowledge and Data Engineering*, *12*(3), pp. 372-390.

Zaki, M. J. (2001). SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning Journal*, *42*(1), pp.31-60.

Zaki, M. J., and hsiao, C.-J. (2002). CHARM: An Efficient Algorithm for Closed Itemset Mining. *Proceedings of the 2nd SIAM International Conference on Data Mining,* Arlington, Virginia, USA, pp. 457-474.

Zhang, H., Zhao, Y., Cao, L., Zhang, C., and Bohlscheid, H. (2009). Customer Activity Sequence Classification for Debt Prevention in Social Security. *Journal of Computer Science and Technology*, *24*(6), pp.1000-1009.

Zhenglu, Y., and Kitsuregawa, M. (2005). LAPIN-SPAM: An Improved Algorithm for Mining Sequential Pattern*. Proceeding of the 21st International Conference on Data Engineering Workshops (ICDEW'05),* Tokoyo, Japan, pp. 1-4.

พนิดา ทรงรัมย์ (2559). การสืบค้นผู้มีอิทธิพลและผู้ถูกครอบงำบนเฟสบุ๊ค. *วารสารเทคโนโลยีสารสนเทศ*, *12*(1), หน้า 1 - 10.

พัชราภรณ์ ช่วยเจริญ และ พนิดา ทรงรัมย์ (2562). การขุดค้นความสัมพันธ์หมวดหมู่ของเพจบนเฟสบุ๊ค. *วารสารเทคโนโลยีสารสนเทศ*, *15*(1), หน้า 50-59.

เอกสิทธิ์ พัชรวงศ์ศักดา (2557). *การวิเคราะห์ข้อมูลด้วยเทคนิคดาต้าไมน์นิงเบื้องต้น* (พิมพ์ครั้งที่ 1). กรุงเทพฯ: เอเชีย ดิจิตอลการพิมพ์.

ดัชนี