การทำนายข้อมูลโดยการรวม CONTENT-BASED FILTERING WITH ITEM-BASED COLLABORATIVE FILTERING ด้วยกฎความสัมพันธ์

DATA PREDICTION USING COMBINATION OF CONTENT–BASED FILTERING AND ITEM–BASED COLLABORATIVE FILTERING WITH ASSOCIATION RULES

ชัยวัฒน์ ติรวีรขจร

CHAIWAT TIRAWEERAKHAJOHN

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2548

ISBN 974-15-1546-4

หัวข้อวิทยานิพนธ์	การทำนายข้อมูลโดยการรวม CONTENT-BASED FILTERING WITH ITEM-BASED COLLABORATIVE FILTERING ด้วยกฎความสัมพันธ์
นักศึกษา	นายชัยวัฒน์ ติรวีรขจร
รหัสประจำตัว	43061622
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
พ.ศ.	2548
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ. ดร. เอื้อน ปิ่นเงิน

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้นำเสนอวิธีการใหม่เพื่อเพิ่มประสิทธิภาพให้กับอัลกอริทึมไอเท็มเบสคอลลาบอราทีฟฟิลเตอริ่ง ที่ผ่านมาอัลกอริทึมนี้ออกแบบมาเพื่อแก้ปัญหาเรื่องปริมาณข้อมูล โดยพิจารณาความสัมพันธ์ระหว่างชิ้นข้อมูลที่แตกต่างกันก่อน แล้วจึงนำความสัมพันธ์เหล่านี้มาคำนวณ หาความคล้ายคลึงตามข้อมูลการให้เรตติ้งของชิ้นข้อมูลในสเปซที่ลดลง อย่างไรก็ตามอัลกอริทึมนี้ยังคงประสบกับปัญหาการให้เรตติ้งต่อชิ้นข้อมูลไม่ทั่วถึงและปัญหาชิ้นข้อมูลที่ยังไม่มีการให้เรตติ้งไว้

วิธีการที่นำเสนอเพื่อแก้ไขปัญหาทั้งสองดังกล่าว ประกอบด้วยสามขั้นตอน คือ ขั้นตอนแรกใช้เทคนิคการค้นหากฎความสัมพันธ์เพื่อค้นพบความคล้ายคลึงจากความสัมพันธ์กันระหว่างคุณสมบัติ ขั้นตอนที่สองนำความคล้ายคลึงที่ได้จากขั้นตอนแรกไปคำนวณหาชิ้นข้อมูลที่คล้ายคลึงตามคุณสมบัติ ขั้นตอนที่สามรวมค่าความคล้ายคลึงตามคุณสมบัติและค่าความคล้ายคลึงตามข้อมูลเรตติ้งเข้าด้วยกัน เพื่อใช้ค้นหาชิ้นข้อมูลที่ใกล้เคียงในอัลกอริทึมไอเท็มเบสคอลลาบอราทีฟ-ฟิลเตอริ่งและให้ผลการทำนายค่าความพึงพอใจออกมาโดยอาศัยค่าความคล้ายคลึงรวม

ผลการทดลองพบว่าวิธีการที่นำเสนอสามารถช่วยให้การทำนายถูกต้องดีกว่าอัลกอริทึมไอเท็มเบสคอลลาบอราทีฟฟิลเตอริ่งแบบเดิม

สารบัญ

หน้า

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

บทที่ 1
บทนำ

ปัจจุบันวิธีการดำเนินธุรกิจแบบอิเล็กทรอนิคส์ (E-Commerce) นั้นเป็นวิธีที่ได้รับความนิยมกันอย่างแพร่หลายทั่วโลก และนับวันก็จะมีเพิ่มมากขึ้นเรื่อยๆ ส่งผลให้การแข่งขันในด้านธุรกิจค่อนข้างสูง ดังนั้นจึงได้มีการนำเอาเทคโนโลยีสมัยใหม่ เข้ามาช่วยในการดำเนินธุรกิจมากขึ้น เพราะนอกจากจะเพิ่มประสิทธิภาพในการนำเสนอชิ้นข้อมูลแล้วนั้น การสร้างความประทับใจ ความพึงพอใจแก่ผู้ใช้ในการที่จะกลับมาใช้บริการเว็บไซต์นั้นต่อ ๆ ไป ก็เป็นอีกวัตถุประสงค์หนึ่ง

ระบบให้การแนะนำ (Recommender system) [25] เป็นหนึ่งในเทคโนโลยีสมัยใหม่ที่ถูกนำมาใช้ในการแนะนำชิ้นข้อมูลต่าง ๆ ที่คาดว่าผู้ใช้น่าจะสนใจ หรืออาจจะเป็นชิ้นข้อมูลที่ผู้ใช้ต้องการ ไม่ว่าจะเป็นเว็บไซต์ใหญ่ที่มีชื่อเสียง เช่น Amazon.com ที่ใช้ระบบให้การแนะนำช่วยในการแนะนำหนังสือต่าง ๆ ให้แก่ผู้ใช้ที่เข้ามาใช้บริการเว็บไซต์ หรือเว็บไซต์อื่น ๆ เช่น เว็บไซต์ Launch.com ซึ่งมีบริการให้คำแนะนำแก่ผู้ใช้เกี่ยวกับดนตรีและเพลง หรือ MovieLen, Netflix และ MovieFinder ที่บริการให้คำแนะนำเกี่ยวกับภาพยนต์ เป็นต้น โดยเทคนิคที่ใช้ในการพัฒนาระบบส่วนใหญ่ คือ เทคนิคคอลลาบอราทีฟฟิลเตอริ่ง (Collaborative Filtering) หรือเรียกสั้น ๆ ว่า CF ซึ่งได้รับความนิยมและประสบความสำเร็จมากในการนำมาใช้งานจริง เริ่มขึ้นครั้งแรกในปี 1992 [12] ที่สถาบันวิจัย Xerox Parc โดยให้ชื่อระบบว่า "Tapestry" ซึ่งเป็นระบบที่นำเข้ามาช่วยในการจัดการอีเมลล์และข้อความในนิวส์กรุ๊ปภายในองค์กร จากนั้นก็เริ่มมีการพัฒนาระบบให้การแนะนำต่าง ๆ เพิ่มมากขึ้นจากทั้งสถาบันวิจัย หรือ บริษัทต่าง ๆ เช่น NetPerceptions, Gustos, Likeminds หรือ Web Trends เป็นต้น

1.1 ความเป็นมาและความสำคัญของปัญหา

ที่ผ่านมาเทคนิค CF ที่นิยมใช้และประสบความสำเร็จมาก คือ เทคนิคยูสเซอร์เบส CF (User-based Collaborative Filtering) ซึ่งเป็นเทคนิคที่ใช้ข้อมูลการให้เรตติ้งของชิ้นข้อมูลที่ผู้ใช้นั้นเคยให้ไว้ในอดีตมาพิจารณาร่วมกับความคิดเห็นของกลุ่มผู้ใช้ที่มีลักษณะการให้เรตติ้งคล้ายคลึงกัน เพื่อทำนายหาค่าความพึงพอใจที่คาดว่าผู้ใช้เป้าหมายจะมีต่อชิ้นข้อมูลเป้าหมายนั้นได้อย่างถูกต้อง หากแต่ปัญหาบ้างประการทำให้ประสิทธิภาพของเทคนิคนี้ลดลง [21] ได้แก่ปัญหาเรื่องปริมาณข้อมูลในระบบมีจำนวนมากเกินไป ทำให้เทคนิคนี้ต้องใช้เวลาในการประมวลผลนานมาก (Scalability problem) ด้วยปัญหาดังกล่าวนี้เอง จึงก่อให้เกิดงานวิจัยที่เกี่ยวกับการแก้ปัญหานี้ขึ้นมามากมาย ในปี ค.ศ. 1999 [1] ได้เสนอการทำดัชนีบนชิ้นข้อมูลที่คล้ายคลึงกัน เพื่อช่วยลดเวลาในการค้นหาข้อมูลลงได้ในระดับหนึ่ง ในปีเดียวกัน [19] ได้นำเทคนิคการจัดกลุ่มข้อมูลมาใช้เพื่อแบ่งแยก

ชิ้นข้อมูลในระบบออกเป็นกลุ่ม ๆ ก่อนทำการคำนวณ ในปีต่อมา [22] ได้เสนอวิธีการลดมิติของข้อมูลลงโดยใช้เทคนิค Singular Value Decomposition (SVD) ซึ่งช่วยลดเวลาในการคำนวณลงได้มากแต่ต้องสูญเสียข้อมูลบ้างส่วนไป

ต่อมาในปี ค.ศ. 2001, Sarwar [23] ได้เสนออัลกอริทึมชนิดใหม่ขึ้นมา เรียกว่า ไอเท็มเบส CF (Item-based Collaborative Filtering) เพื่อแก้ปัญหาเรื่องปริมาณข้อมูลและพบว่าอัลกอริทึมใหม่นี้มีประสิทธิภาพการใช้งานได้ดีกว่าอัลกอริทึมยูสเซอร์เบส CF และมีแนวโน้มว่าจะเป็นที่นิยมใช้กันอย่างแพร่หลายในอนาคตอันใกล้นี้ แต่อัลกอริทึมนี้ยังคงประสบกับปัญหาการให้คะแนนเรตติ้งต่อชิ้นข้อมูลในระบบได้ไม่ทั่วถึง (Sparsity problem) และปัญหาชิ้นข้อมูลที่ยังไม่มีการให้คะแนนเรตติ้งไว้ (First-rater problem) ดังนั้นทั้งสองปัญหาดังกล่าว จึงเป็นสาเหตุสำคัญที่ทำให้อัลกอริทึมไอเท็มเบส CF ขาดความน่าเชื่อถือในการทำนายหาค่าความพึงพอใจ หรือกล่าวอีกนัยหนึ่งได้ว่า ปริมาณข้อมูลการให้เรตติ้งมีน้อยเกินไปไม่เพียงพอต่อการคำนวณหากลุ่มชิ้นข้อมูลที่มีลักษณะการให้คะแนนเรตติ้งที่คล้ายคลึงกันได้ ซึ่งเป็นความท้าทายและเป็นประเด็นสำคัญในการแก้ไขปัญหาทั้งสองดังกล่าวของอัลกอริทึมนี้ให้มีประสิทธิภาพเพิ่มมากขึ้น นำไปสู่เทคโนโลยีและวิธีการใหม่ ๆ ต่อไปในอนาค

1.2 จุดมุ่งหมายและวัตถุประสงค์ของการวิจัย

งานวิจัยนี้มุ่งเน้นที่จะศึกษาวิจัยถึงวิธีการแก้ไขปัญหาดังที่กล่าวไว้ในหัวข้อที่ผ่านมาเพื่อสามารถเพิ่มประสิทธิภาพของอัลกอริทึมไอเท็มเบส CF แบบเดิมด้วยวิธีการที่นำเสนอที่นำเสนอได้อย่างอัตโนมัติ โดยวิธีการที่นำเสนอสามารถค้นหาค่าความคล้ายคลึงระหว่างชิ้นข้อมูลตามคุณสมบัติได้ ทั้งนี้เพื่อสามารถนำวิธีการที่เสนอไปรวมกับขั้นตอนการค้นหาชิ้นข้อมูลที่ใกล้ เคียงในอัลกอริทึมไอเท็มเบส CF แบบเดิม เพื่อทำให้อัลกอริทึมไอเท็มเบส CF มีความยืดหยุ่นที่จะสามารถค้นหาชิ้นข้อมูลที่ใกล้เคียงได้โดยพิจารณาจากข้อมูลการให้คะแนนเรตติ้งร่วมกับคุณสมบัติของชิ้นข้อมูลได้อย่างอัตโนมัติ ยิ่งไปกว่านั้นยังหวังว่าวิธีการที่นำเสนอในงานวิจัยนี้จะสามารถนำไปประยุกต์ใช้กับงานวิจัยในด้านอื่น ๆ ได้เป็นอย่างดีต่อไป

1.3 สมมุติฐานของการวิจัย

งานวิจัยนี้ได้ตั้งสมมุติไว้ว่าการนำข้อดีของคอนเท็นเบสฟิลเตอริ่ง (Content-Based Filtering) หรือเรียกสั้น ๆ ว่า CBF และข้อดีของ CF มารวมกัน จะสามารถนำทั้งข้อมูลการแสดงความคิดเห็นของแต่ละบุคคล (Subjective) และคุณลักษณะของชิ้นข้อมูลแต่ละชิ้น (Objective) มาพิจารณาหากลุ่มชิ้นข้อมูลที่มีลักษณะใกล้เคียงกันได้อย่างเหมาะสมและสามารถช่วยเพิ่มประสิทธิภาพให้กับอัลกอริทึมไอเท็มเบส CF ได้ โดยเฉพาะอย่างยิ่งสำหรับขั้นตอนการค้นหาชิ้น

ดาต้าเซ็ต MovieLens [14] และ EachMovie [17] โดยใช้การสุ่มเลือกข้อมูลการให้เรตติ้งจากจำนวนผู้ใช้ (ผู้ชม) สูงสุดไม่เกิน 200 รายต่อจำนวนชิ้นข้อมูลทั้งหมด (ภาพยนตร์) 200 เรื่อง และวิเคราะห์ผลการทดลองเชิงสถิติ

1.7 ข้อจำกัดของการวิจัย

งานวิจัยนี้เป็นการวิจัยเชิงทดลอง วิเคราะห์ผลการทำนายค่าความพึงพอใจ ซึ่งล้วนแล้วแต่เป็นผลการทำนายที่ไม่แน่นอน และขึ้นอยู่กับค่าพารามิเตอร์ถ่วงน้ำหนักที่เปลี่ยนแปลงตลอดเวลามีค่าไม่คงที่จึงจำเป็นต้องนำวิธีการทางสถิติมาใช้วิเคราะห์ผลการทำนาย อันได้แก่ ค่าความผิดพลาดสมบูรณ์เฉลี่ย (Mean absolute error), ส่วนเบี่ยงเบนมาตรฐาน (Standard deviation) และเปอเซ็นต์ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลง (Percent of reduced mean absolute error)

1.8 คำจำกัดความที่ใช้ในการศึกษา

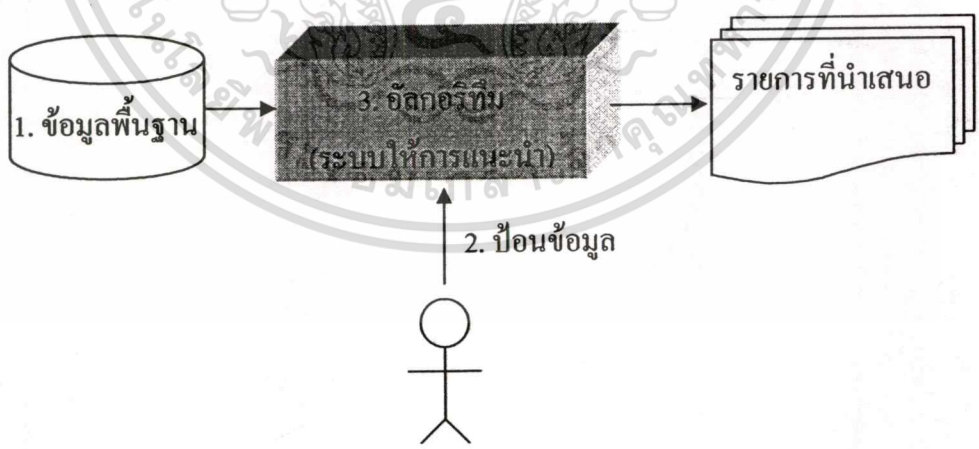
- ระบบให้การแนะนำ (Recommendation system) หมายถึง ระบบที่สามารถแนะนำชิ้นข้อมูลต่าง ๆ ที่คาดว่าผู้ใช้น่าจะสนใจ สำหรับการดำเนินธุรกิจแบบอี-คอมเมอส
- คอลลาบอราทีฟฟิลเตอริ่ง (Collaborative filtering) หมายถึง วิธีการกลั่นกรองข้อมูลจากการแสดงความคิดเห็นร่วมกัน
- คอนเท็นเบสฟิลเตอริ่ง (Content-based filtering) หมายถึง วิธีการกลั่นกรองข้อมูลจากเนื้อหาของข้อมูลทั้งหมด
- เรตติ้ง (Rating) หมายถึง ระดับความพึงพอใจที่ผู้ใช้แต่ละคนมีต่อชิ้นข้อมูลแต่ละชิ้น
- เซ็ตของไอเท็ม (Set of items) หมายถึง เซ็ตที่มีไอเท็มทั้งหมดเป็นสมาชิก ซึ่งไอเท็มในที่นี้คือ ชื่อคุณสมบัติต่างๆ ของภาพยนตร์
- ไอเท็มเซ็ต (Itemset) หมายถึง กลุ่มไอเท็มที่ประกอบด้วยไอเท็มที่เป็นสมาชิกในเซ็ตของไอเท็ม
- ทรานแซคชั่น (Transaction) หมายถึง เซ็ตที่ประกอบด้วยไอเท็มที่เป็นสมาชิกในเซ็ตของไอเท็ม
- กฎความสัมพันธ์ (Association rule) คือ การอุปนัยในรูปแบบ $X \Rightarrow Y$
- ไอเท็มเซ็ตปรากฏบ่อย (Frequent itemset) หมายถึง ไอเท็มเซ็ตที่ปรากฏร่วมกันในทรานแซคชั่นเดียวกันที่มีค่าสนับสนุนมากกว่าค่าสนับสนุนขั้นต่ำที่กำหนดไว้

บทที่ 2
ทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้อง

บทนี้จะอธิบายถึงทฤษฎีพื้นฐานที่จำเป็นและสรุปงานวิจัยที่เกี่ยวข้องกับปัญหาที่กำลังศึกษาและวิจัยอยู่ในงานวิจัยนี้ โดยเริ่มต้นที่การอธิบายถึงระบบให้การแนะนำ วิธีการหรือเทคนิคที่นำมาใช้อันได้แก่ วิธี CBF และวิธี CF รวมถึงการรวมทั้งสองวิธีนี้เข้าด้วยกัน และสุดท้ายจะกล่าวถึงเทคนิคการค้นหากฎความสัมพันธ์ที่นำมาประยุกต์ใช้กับงานวิจัยนี้

2.1 ระบบให้การแนะนำ

โดยทั่วไประบบให้การแนะนำประกอบด้วย 3 ส่วนคือ (1) ส่วนข้อมูลพื้นฐานที่จำเป็นต้องใช้ประมวลผล เช่น โปรไฟล์ (Profile) ของผู้ใช้แต่ละคน (2) ส่วนการป้อนข้อมูล เป็นข้อมูลที่ได้จากการป้อนเข้ามาของผู้ใช้ เช่น การให้คะแนนเรตติ้งซึ่งมีอยู่ 2 แบบ [7] คือ แบบชัดเจน (Explicit) และแบบไม่ชัดเจน (Implicit) เรตติ้งแบบชัดเจนจะแสดงอยู่ในรูปของจำนวนตัวเลขตามระดับความนิยมตั้งแต่ 1 ถึง 5, 1 ถึง 10 หรือระดับอื่นๆ ขึ้นอยู่กับการใช้งาน ส่วนเรตติ้งแบบไม่ชัดเจนได้มาจากพฤติกรรมการใช้งานของผู้ใช้ต่างๆ เช่น ประวัติการซื้อสินค้าหรือประวัติการเข้ามาใช้งานของผู้ใช้ในอดีตที่ผ่านมา (3) ส่วนอัลกอริทึม [7] เป็นส่วนสำคัญที่สุดที่ใช้ประมวลข้อมูลเพื่อให้การแนะนำชิ้นข้อมูลออกมา ดังแสดงไว้ในรูปที่ 2.1



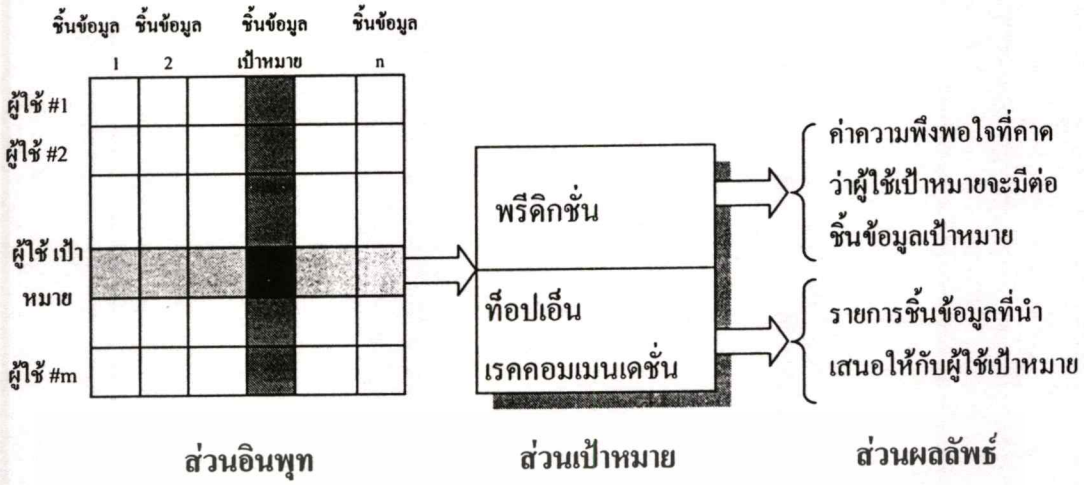
รูปที่ 2.1 สถาปัตยกรรมพื้นฐานของระบบให้การแนะนำ

จากรูปที่ 2.1 เป็นระบบช่วยในการแนะนำหรือนำเสนอชิ้นข้อมูลให้แก่ผู้ใช้ โดยระบบจะทำการวิเคราะห์และนำเสนอชิ้นข้อมูลที่คาดว่าผู้ใช้น่าจะสนใจ หรือคาดว่าจะเป็นชิ้นข้อมูลที่ผู้ใช้

คนนั้นกำลังต้องการอย่างอัตโนมัติ ซึ่งนับเป็นการอำนวยความสะดวกและสร้างความพึงพอใจให้แก่ผู้ใช้ในทางอ้อมวิธีหนึ่ง เดิมการหาข้อมูลที่ตรงตามความสนใจของผู้ใช้ทั่วไปนั้นสามารถทำได้เพียงการใช้วิธี CBF ซึ่งเป็นวิธีที่ให้ความสนใจกับคุณลักษณะของเนื้อหาข้อมูลเป็นหลัก โดยจะสนใจว่าลักษณะชิ้นข้อมูลนั้นตรงตามที่ผู้ใช้ต้องการหรือไม่ ซึ่งถ้าใช่ก็จะนำเสนอชิ้นข้อมูลนั้นทันที แต่ถ้าไม่ก็จะไม่สนใจแม้ว่าชิ้นข้อมูลนั้นจะมีลักษณะใกล้เคียงกับชิ้นข้อมูลที่ผู้ใช้ต้องการก็ตาม ซึ่งต่อมามีการนำวิธี CF ที่ใช้การทำนายหาค่าความพึงพอใจของผู้ใช้ที่คาดว่าผู้ใช้น่าจะมีต่อชิ้นข้อมูลนั้น ๆ เพื่อนำค่าความพึงพอใจที่ได้มาใช้ในการพิจารณาหาชิ้นข้อมูลที่จะนำเสนอให้กับผู้ใช้คนได้ สำหรับการทำนายหาค่าความพึงพอใจนี้ สามารถคำนวณได้จากลักษณะการให้คะแนนที่ผู้ใช้เคยให้ไว้แก่ชิ้นข้อมูลต่าง ๆ ซึ่งคะแนนเหล่านั้น เรียกว่า "เรตติ้ง" โดยเรตติ้งนี้สามารถนำไปคำนวณร่วมกับเรตติ้งจากกลุ่มผู้ใช้ที่มีความคิดเห็นคล้ายคลึงกันได้ จากนั้นจึงนำค่าความพึงพอใจที่ได้มาใช้ในการประเมินว่าสมควรที่จะนำเสนอชิ้นข้อมูลนั้นให้แก่ผู้ใช้หรือไม่ ซึ่งทำให้ลักษณะชิ้นข้อมูลที่จะนำมาเสนอแก่ผู้ใช้แต่ละคนนั้นแตกต่างกันไปตามรสนิยมของผู้ใช้แต่ละคน ทำให้ผลลัพธ์ที่ได้จากวิธี CF นี้ประสบความสำเร็จและเป็นที่นิยมใช้มากที่สุด เนื่องจากสามารถที่จะนำเสนอชิ้นข้อมูลได้ถูกต้องหรือใกล้เคียงกับความต้องการของผู้ใช้มากขึ้น

2.1.1 วิธี CF

เป็นการนำข้อมูลการให้คะแนนเรตติ้งที่ผู้ใช้นั้นเคยให้ไว้ในอดีต มาพิจารณาร่วมกับความคิดเห็นของกลุ่มผู้ใช้ที่มีลักษณะการให้คะแนนเรตติ้งคล้ายคลึงกันเพื่อทำนายหาชิ้นข้อมูลที่คาดว่าผู้ใช้คนนั้นจะสนใจและนำชิ้นข้อมูลเหล่านั้นมาแนะนำให้แก่ผู้ใช้คนนั้นได้ตามความต้องการ ด้วยหลัก การดังกล่าวนี้จึงทำให้วิธีการนี้มีชื่อเรียกอีกชื่อหนึ่งว่า โซเชียลฟิลเตอริ่ง (Social filtering) ซึ่งสามารถอธิบายหลักการได้ดังนี้ เริ่มจากการคำนวณหากลุ่มผู้ใช้ที่มีลักษณะความคิดเห็นใกล้เคียงกัน ที่เรียกว่า เพื่อนบ้าน (Neighborhood) หรือที่ใกล้เคียง วิธีทั่วไปที่ใช้ในการคำนวณหาเพื่อนบ้านนั้น มี 2 วิธี คือ Pearson correlation และ Cosine similarity ซึ่งจะแตกต่างกันในบางส่วนของสมการที่ใช้คำนวณ แต่มีจุดประสงค์เหมือนกัน คือ ต่างพยายามที่จะหากลุ่มของผู้ใช้ที่มีความคิดเห็นคล้ายคลึงกับผู้ใช้เป้าหมาย โดยอาศัยการเปรียบเทียบเรตติ้งระหว่างผู้ใช้เหล่านั้น จากนั้นจึงนำเรตติ้งของกลุ่มเพื่อนบ้านเหล่านั้นเข้ามาเป็นส่วนหนึ่งในการคำนวณที่มีอยู่ 2 แบบ ดังแสดงในรูปที่ 2.2 แบบที่ 1 คือ การคำนวณหาค่าความพึงพอใจที่คาดว่าผู้ใช้เป้าหมายจะมีต่อชิ้นข้อมูลเป้าหมาย โดยพิจารณาชิ้นข้อมูลเป็นชิ้น ๆ ไป วิธีนี้เรียกว่า การคำนวณแบบพรีดิกชั่น (Prediction) และแบบที่ 2 คือ การคำนวณหากลุ่มชิ้นข้อมูลที่คาดว่าผู้ใช้นั้นสนใจและยังไม่เคยให้เรตติ้งมาก่อน เพื่อนำรายการชิ้นข้อมูลเหล่านั้นมานำเสนอ โดยวิธีการคำนวณแบบนี้เรียกว่าการคำนวณแบบท็อปเอ็นเรคคอมเมนเดชั่น (Top-N Recommendation) [11]



รูปที่ 2.2 ขั้นตอนการทำงานของวิธีการ CF

จากรูปที่ 2.2 แสดงให้เห็นถึงขั้นตอนการทำงานของวิธีการ CF ที่ประกอบไปด้วย 3 ส่วนหลักดังต่อไปนี้

- ส่วนอินพุท (Input) เป็นข้อมูลที่ใช้ในการคำนวณที่แสดงอยู่ในรูปเมตริกซ์ของเรตติ้งที่ผู้ใช้มีต่อชิ้นข้อมูลในระบบ เรียกว่า เมตริกซ์ผู้ใช้-ชิ้นข้อมูล (User-item matrix) ดังตัวอย่างแสดงในตารางที่ 2.1 แสดงเรตติ้งที่เกิดจากผู้ใช้ 5 คนต่อชิ้นข้อมูล 6 ชิ้น โดยแต่ละแถวหมายถึงผู้ใช้ ส่วนแต่ละหลักหมายถึงชิ้นข้อมูลและแต่ละเซลในเมตริกซ์หมายถึงเรตติ้งของผู้ใช้ที่มีต่อชิ้นข้อมูลนั้น ๆ ถ้าเซลว่างหมายความว่าไม่มีข้อมูลการให้เรตติ้ง

ตารางที่ 2.1 ตัวอย่างเมตริกซ์ผู้ใช้-ชิ้นข้อมูล

	ชิ้นข้อมูล #1	ชิ้นข้อมูล #2	ชิ้นข้อมูล #3	ชิ้นข้อมูล #4	ชิ้นข้อมูล #5	ชิ้นข้อมูล #6
ผู้ใช้ #1	1	5		2	5	5
ผู้ใช้ #2	4	2		2	3	2
ผู้ใช้ #3	2	4	2	2	4	4
ผู้ใช้ #4	2	4		2	4	2
ผู้ใช้ #5	5	4		1		3

- ส่วนเป้าหมาย (Goal) หรืออัลกอริทึมประกอบด้วย 2 แบบ คือ พรีดิกชั่นและท็อปเอ็น-เรคคอมเมนเดชั่น ซึ่งในงานวิจัยเล่มนี้จะมุ่งเน้นไปที่แบบพรีดิกชั่นแบบเดียวเท่านั้นเพราะ

ถือว่า เป็นส่วนสำคัญที่สุดของวิธีการ CF ซึ่งเป็นการนำข้อมูลในส่วนแรกมาทำการคำนวณเพื่อทำนายหาค่าความพึงพอใจ ปัจจุบันมี 2 อัลกอริทึม คือ ยูสเซอร์เบส CF และไอเท็มเบส CF โดยกำหนดให้ m เป็นจำนวนของผู้ใช้ $U=\{u_1,u_2,...,u_m\}$ และ n เป็นจำนวนของชิ้นข้อมูล $I=\{i_1,i_2,...,i_n\}$ โดยที่ผู้ใช้แต่ละคนแทนด้วย $u_i; i=1,2,...,m$ จะมีกลุ่มของชิ้นข้อมูล I_{u_i} ที่ผู้ใช้แต่ละคนเคยให้เรตติ้งกับชิ้นข้อมูลเหล่านั้นไว้หรือกล่าวได้ว่า $I_{u_i} \subseteq I$ ดังนั้นไม่ว่าจะเป็นอัลกอริทึม CF แบบใดก็จะมีเป้าหมายเดียวกัน คือ พยายามที่จะทำนายหาค่าความพึงพอใจที่คาดว่าผู้ใช้เป้าหมายจะมีต่อชิ้นข้อมูลเป้าหมาย เพื่อนำเสนอชิ้นข้อมูลที่ผู้ใช้เป้าหมายยังไม่เคยให้เรตติ้งไว้หรือกล่าวอีกนัยหนึ่งได้ว่าเป็นการทำนายหาค่าความพึงพอใจที่สูญหายไป (Missing values) จากข้อมูลที่มีอยู่ดังแสดงในตารางที่ 2.2 (ในกรอบแรเงา)

ตารางที่ 2.2 แสดงเป้าหมายของวิธี CF

	ชิ้นข้อมูล #1	ชิ้นข้อมูล #2	ชิ้นข้อมูล #3	ชิ้นข้อมูล #4	ชิ้นข้อมูล #5	ชิ้นข้อมูล #6
ผู้ใช้ #1	1	5	?	2	5	5
ผู้ใช้ #2	4	2	?	2	3	2
ผู้ใช้ #3	2	4	2	2	4	4
ผู้ใช้ #4	2	4	?	2	4	2
ผู้ใช้ #5	5	4	?	1	?	3

- ส่วนผลลัพธ์ (Output) มีอยู่ 2 แบบ คือ
  - แบบพรีดิกชั่น เป็นค่าตัวเลขความพึงพอใจที่คาดว่าผู้ใช้เป้าหมายมีต่อชิ้นข้อมูลเป้าหมายโดยพิจารณาชิ้นข้อมูลเป็นชิ้นๆ ไป
  - แบบท็อปเอ็นเรคคอมเมนเดชั่น เป็นรายชื่อของ N ชิ้นข้อมูลที่คาดว่าผู้ใช้เป้าหมายสนใจที่สุด หรือกล่าวอีกนัยหนึ่งได้ว่าผลลัพธ์ของท็อปเอ็นเรคคอมเมนเดชั่นได้มาจากการนำผลลัพธ์แบบพรีดิกชั่นมาคำนวณอีกต่อหนึ่ง

ที่ผ่านมาเทคนิค CF ที่นิยมใช้กันอย่างแพร่หลายคือ ยูสเซอร์เบส CF ซึ่งประสบกับปัญหาเรื่องปริมาณข้อมูลและความเบาบางของข้อมูลการให้เรตติ้ง ดังนั้นจึงได้มีนักวิจัยจำนวนมากได้พยายามปรับปรุงแก้ไขวิธีการเพื่อแก้ไขปัญหาดังกล่าว

ปี ค.ศ. 2000, Chee [10] ได้นำเสนอ RecTree ซึ่งเป็นอัลกอริทึมที่แก้ไขปัญหาด้านเวลาในการประมวลผลของอัลกอริทึม CF และปัญหาเรื่องความเบาบางของข้อมูลเรตติ้ง โดยอาศัยหลักการจัดกลุ่มของอัลกอริทึมเคมีนที่จัดแต่ละอิลิเมนต์ (Element) ไว้เป็นกลุ่ม ด้วยการคำนวณหาระยะห่างระหว่างอิลิเมนต์กับกลุ่ม RecTree ใช้อัลกอริทึมเคมีนเพื่อจัดโครงสร้างดาต้าเซ็ตให้กลายเป็นกลุ่มของไบนารีทรี (Binary tree of clusters) โดยแบ่งแยก (Split) ดาต้าเซ็ตออกเป็นกลุ่มลูก (Child clusters) RecTree จะทำการแบ่งแยกดาต้าไปเรื่อย ๆ ถ้าขนาดของดาต้าเซ็ตนั้นมีขนาดมากกว่าขนาดของกลุ่มสูงสุด (Maximum cluster size) จนกระทั่งขนาดของลีฟโนดทั้งหมดไม่มากไปกว่าขนาดของกลุ่มสูงสุดที่กำหนดไว้

ปี ค.ศ. 2000, Sarwar [22] นำเสนอวิธีการแก้ปัญหาความเบาบางของข้อมูลเรตติ้งลง โดยใช้เทคนิค Singular Value Decomposition (SVD) ในการลดมิติของเมตริกซ์ผู้ใช้-ชิ้นข้อมูลลงซึ่งวิธีการนี้อาจทำให้สูญเสียข้อมูลบางส่วนไปในขณะทำการลดมิติข้อมูลได้และไม่เหมาะที่จะนำไปใช้งานจริงเพราะต้องใช้เวลาในการคำนวณข้อมูลค่อนข้างนานมาก

ต่อมาในปี ค.ศ. 2001, Sarwar [23] ได้คิดค้นอัลกอริทึมชนิดใหม่ เรียกว่า ไอเท็มเบส CF (Item-Based Collaborative Filtering) ขึ้นมาเพื่อแก้ปัญหาเรื่องปริมาณข้อมูลทั้งหมดที่ใช้ในการคำนวณและพบว่าอัลกอริทึมใหม่นี้มีประสิทธิภาพการใช้งานได้ดีกว่าอัลกอริทึมยูสเซอร์เบส CF ในงานวิจัยเล่มนี้ได้นำอัลกอริทึมนี้มาพัฒนาและปรับปรุงเพื่อแก้ปัญหา ในส่วนของรายละเอียดวิธีการและปัญหาของอัลกอริทึมนี้จะอธิบายในหัวข้อถัดไป

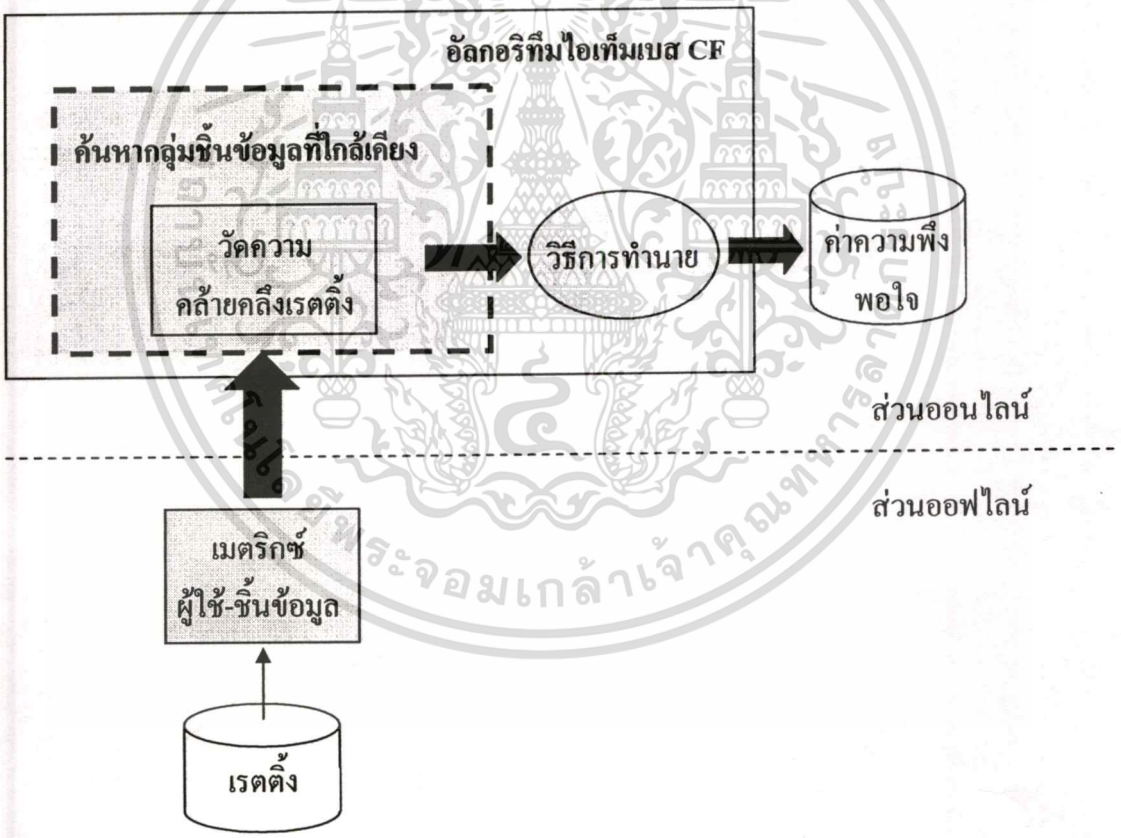
ในปี ค.ศ. 2003, Zeng [26] ได้นำเสนออัลกอริทึมใหม่สำหรับแก้ปัญหาปริมาณข้อมูลและความเบาบางของข้อมูลเรตติ้ง เรียกว่า คลาสเบส CF ที่อาศัยหลักการแบ่งแยก (Classification) เพื่อแบ่งแยกชิ้นข้อมูลออกเป็นประเภท แล้วทำการแปลงเมตริกซ์ผู้ใช้-ชิ้นข้อมูลแบบเดิมให้กลายเป็นเมตริกซ์ผู้ใช้-ชนิด (User-class matrix) ก่อนแล้วใช้เทคนิคการเลือกตัวอย่าง (Instance selection) ทำการกำจัดข้อมูลที่ไม่มีความสัมพันธ์กันออกไป เพื่อลดปริมาณข้อมูลในการค้นหาลง แต่วิธีการนี้ยังคงใช้เวลามากในการแปลงเมตริกซ์ผู้ใช้-ชิ้นข้อมูลไปเป็นเมตริกซ์ผู้ใช้-ชนิด

ล่าสุดในปี ค.ศ. 2004, Huang [15] ได้นำเทคนิคการค้นคืนอย่างสัมพันธ์มาประยุกต์ใช้สำหรับค้นหาความสัมพันธ์อย่างส่งผ่าน (Transitive associations) ระหว่างผู้ใช้กับชิ้นข้อมูลภายในเมตริกซ์ผู้ใช้-ชิ้นข้อมูล ซึ่งวิธีการเดิมของ CF จะไม่พิจารณาถึงความสัมพันธ์นี้เลย ดังนั้นด้วยวิธีการของ Huang นี้จึงสามารถนำเมตริกซ์ผู้ใช้-ชิ้นข้อมูล มาสร้างเป็นโมเดลสำหรับการค้นคืนอย่างสัมพันธ์ที่เพิ่มความสัมพันธ์อย่างส่งผ่านนี้เข้าไปด้วย ทำให้สามารถแก้ปัญหาความเบาบางของข้อมูลเรตติ้งลงได้โดยไม่สูญเสียข้อมูล

2.1.2 อัลกอริทึมไอเท็มเบส CF

อัลกอริทึมนี้ [23] อาศัยหลักการของ CF ดังที่กล่าวไว้แล้วในหัวข้อที่ 2.1.1 โดยพิจารณาถึงความสัมพันธ์ระหว่างชิ้นข้อมูลเป็นหลัก ประกอบด้วย 2 ส่วนหลัก ๆ ได้แก่ ส่วนออนไลน์ และออฟไลน์ ดังแสดงในรูปที่ 2.3 ในส่วนออฟไลน์ จะเป็นการสร้างแบบจำลองของเรตติ้งไว้ในรูปแบบเมตริกซ์ผู้ใช้-ชิ้นข้อมูลก่อนล่วงหน้า ซึ่งถือว่าเป็นข้อดีของวิธีการนี้นั่นคือ เป็นโมเดลเบส (Model based) อย่างหนึ่งที่นำแบบจำลองข้อมูลที่ได้สร้างไว้แล้วล่วงหน้าไปใช้ในการคำนวณในส่วนออนไลน์ได้ทันที ทำให้วิธีการนี้สามารถแก้ปัญหาเรื่องปริมาณข้อมูลลงไปได้ แต่มีข้อเสียคือไม่เหมาะกับระบบที่มีผู้ใช้จำนวนมากชอบเปลี่ยนแปลงแก้ไขข้อมูลการให้เรตติ้งอยู่เป็นประจำ เพราะเป็นสาเหตุทำให้ข้อมูลในส่วนออฟไลน์เป็นข้อมูลที่ไม่ได้อัพเดท

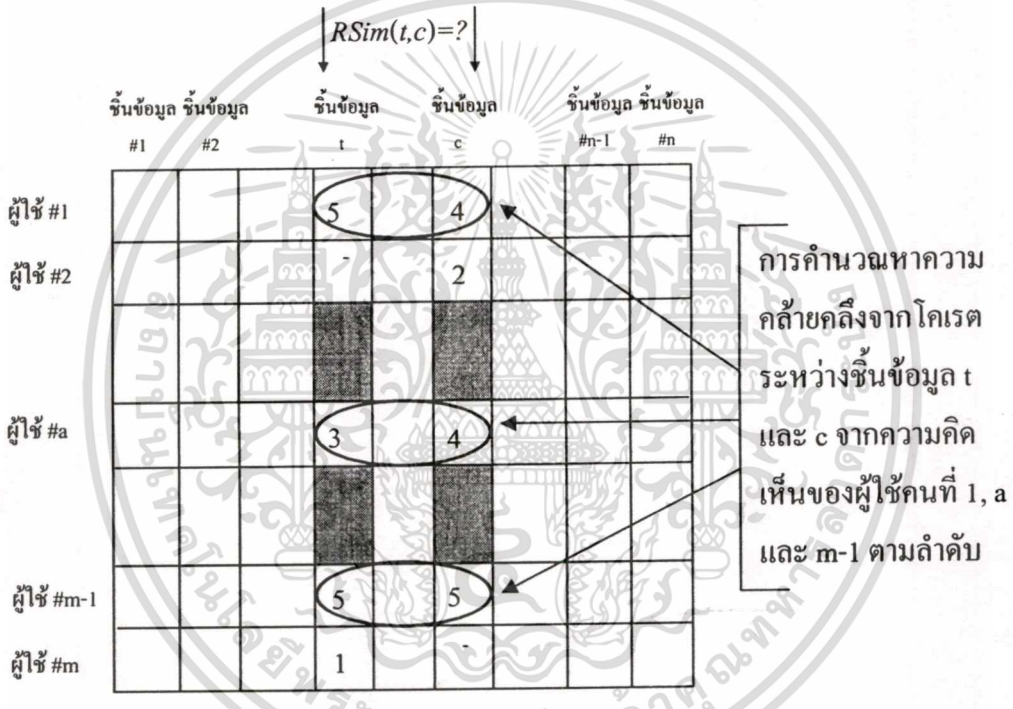
ในส่วนออนไลน์จะประกอบไปด้วย 2 ขั้นตอนหลักได้แก่ การค้นหากลุ่มชิ้นข้อมูลที่ใกล้เคียงและวิธีการทำนาย ซึ่งมีรายละเอียดดังต่อไปนี้



รูปที่ 2.3 สถาปัตยกรรมของวิธีไอเท็มเบส CF

2.1.2.1 การค้นหากลุ่มชิ้นข้อมูลที่ใกล้เคียง

ขั้นตอนแรกนี้เป็นการนำข้อมูลการให้เรตติ้งของชิ้นข้อมูลเป้าหมายไปเปรียบเทียบกับทุกชิ้นข้อมูลที่ผู้ใช้เป้าหมายเคยให้เรตติ้งเอาไว้ เพื่อค้นหากลุ่มของชิ้นข้อมูลที่มีลักษณะการให้เรตติ้งที่ใกล้เคียงหรือคล้ายคลึงกับชิ้นข้อมูลเป้าหมายมากที่สุดการค้นหาเริ่มต้นด้วยการเปรียบเทียบทีละชิ้นข้อมูลโดยอาศัยความคิดเห็นจากผู้ใช้คนอื่นๆ ที่ให้เรตติ้งกับชิ้นข้อมูลทั้งสองชิ้นนั้นร่วมกัน ที่เรียกว่า โคเรต (Co-rated) ดังแสดงในรูปที่ 2.4 และนำเฉพาะโคเรตมาคำนวณหาค่าความคล้ายคลึงระหว่างชิ้นข้อมูลทั้งสองดังกล่าว ด้วยวิธี Cosine, Pearson correlation และ Adjusted cosine



รูปที่ 2.4 การคำนวณความคล้ายคลึงจากโคเรตระหว่างชิ้นข้อมูล i และ j

จากรูปที่ 2.4 อัลกอริทึมไอเท็มเบส CF จะหาว่ามีผู้ใช้คนใดบ้างที่ให้เรตติ้งกับชิ้นข้อมูล t และ c ร่วมกัน (โคเรต) จากรูปเห็นได้ว่ามีเพียงผู้ใช้ 3 คนเท่านั้นที่ให้โคเรตกับชิ้นข้อมูล t และ c ร่วมกันได้แก่ ผู้ใช้คนที่ 1, a และ m-1 หลังจากนั้นจึงนำโคเรตจากผู้ใช้ทั้งสามคนดังกล่าวมาคำนวณหาความคล้ายคลึงระหว่างชิ้นข้อมูล t และ c ได้ 3 วิธี ดังนี้

- การคำนวณหาความคล้ายคลึงด้วยวิธี Cosine สามารถอธิบายได้ดังสมการที่ 2.1

$$RSim(t,c) = \frac{\sum_{u \in U} R_{u,t} * R_{u,c}}{\sqrt{\sum_{u \in U} R_{u,t}^{\ 2}} \sqrt{\sum_{u \in U} R_{u,c}^{\ 2}}} \quad (2.1)$$

โดยที่

$RSim(t,c)$	คือ	ค่าความคล้ายคลึงจากโคเรตระหว่างชิ้นข้อมูล t และ c
$R_{u,t}$ และ $R_{u,c}$	คือ	เรตติ้งที่ผู้ใช้ u มีต่อชิ้นข้อมูล t และ c ตามลำดับ
t	คือ	ชิ้นข้อมูลเป้าหมาย
c	คือ	ชิ้นข้อมูลเปรียบเทียบ

ดังนั้นจากรูปที่ 2.4 สามารถใช้วิธี Cosine คำนวณได้ดังนี้

$$RSim(t,c) = \frac{(5*4)+(3*4)+(5*5)}{\sqrt{5^2+3^2+5^2} * \sqrt{4^2+4^2+5^2}}$$

$$= \frac{57}{\sqrt{59} * \sqrt{57}} = 0.98$$

- การคำนวณหาความคล้ายคลึงด้วยวิธี Pearson correlation สามารถอธิบายได้ดังสมการที่ 2.2

$$RSim(t,c) = \frac{\sum_{u \in U} (R_{u,t} - \overline{R}_t)(R_{u,c} - \overline{R}_c)}{\sqrt{\sum_{u \in U} (R_{u,t} - \overline{R}_t)^2} \sqrt{\sum_{u \in U} (R_{u,c} - \overline{R}_c)^2}} \quad (2.2)$$

โดยที่

$\overline{R}_t$ และ $\overline{R}_c$ คือ ค่าเฉลี่ยเรตติ้งของชิ้นข้อมูล t และ c ตามลำดับ

ดังนั้นจากรูปที่ 2.4 สามารถใช้วิธี Pearson correlation คำนวณได้ดังนี้

เนื่องจาก $\overline{R}_t = 3.5$ และ $\overline{R}_c = 3.75$ ดังนั้น

$$RSim(t,c) = \frac{(5-3.5)(4-3.75)+(3-3.5)(4-3.75)+(5-3.5)(5-3.75)}{\sqrt{(5-3.5)^2+(3-3.5)^2+(5-3.5)^2} * \sqrt{(4-3.75)^2+(4-3.75)^2+(5-3.75)^2}}$$

$$= \frac{(1.5)(0.25)+(-0.5)(0.25)+(1.5)(1.25)}{\sqrt{(1.5)^2+(-0.5)^2+(1.5)^2} * \sqrt{(0.25)^2+(0.25)^2+(1.25)^2}}$$

$$= \frac{0.375-0.125+1.875}{\sqrt{2.25+0.25+2.25} * \sqrt{0.0625+0.0625+1.5625}} = 0.58$$

- การคำนวณหาความคล้ายคลึงด้วยวิธี Adjusted cosine สามารถอธิบายได้ดังสมการที่ 2.3

$$RSim(t,c) = \frac{\sum_{u \in U}(R_{u,t}-\overline{R}_u)(R_{u,c}-\overline{R}_u)}{\sqrt{\sum_{u \in U}(R_{u,t}-\overline{R}_u)^2}\sqrt{\sum_{u \in U}(R_{u,c}-\overline{R}_u)^2}} \quad (2.3)$$

โดยที่ $\overline{R}_u$ คือ ค่าเฉลี่ยเรตติ้งของผู้ใช้ u

ดังนั้นจากรูปที่ 2.4 สามารถใช้วิธี Adjusted cosine คำนวณได้ดังนี้

เนื่องจาก $\overline{R}_1 = 4.5$, $\overline{R}_a = 3.5$ และ $\overline{R}_{m-1} = 5$ ดังนั้น

$$RSim(t,c) = \frac{(5-4.5)(4-4.5)+(3-3.5)(4-3.5)+(5-5)(5-5)}{\sqrt{(5-4.5)^2+(3-3.5)^2+(5-5)^2} * \sqrt{(4-4.5)^2+(4-3.5)^2+(5-5)^2}}$$

$$= \frac{(0.5)(-0.5)+(-0.5)(0.5)+0}{\sqrt{(0.5)^2+(-0.5)^2+0} * \sqrt{(0.5)^2+(0.5)^2+0}}$$

$$= \frac{0.0625}{\sqrt{0.25+0.25} * \sqrt{0.25+0.25}}$$

$$= 0.125$$

2.1.2.2 การทำนายหาค่าความพึงพอใจ

ขั้นตอนสุดท้ายนำกลุ่มของชิ้นข้อมูลที่มีลักษณะการให้เรตติ้งคล้ายคลึงกับชิ้นข้อมูลเป้าหมายมากที่สุดจำนวน k ชิ้น มาทำนายหาค่าความพึงพอใจ ด้วยเทคนิค Weighted sum ตามสมการที่ 2.4

$$P_{u,i} = \frac{\sum_{k \in K} \left(RSim(t,k) * R_{u,k} \right)}{\sum_{k \in K} \left(|RSim(t,k)| \right)} \quad (2.4)$$

โดยที่ $P_{u,i}$ คือ ค่าความพึงพอใจที่คาดว่าผู้ใช้เป้าหมาย u มีต่อชิ้นข้อมูลเป้าหมาย t

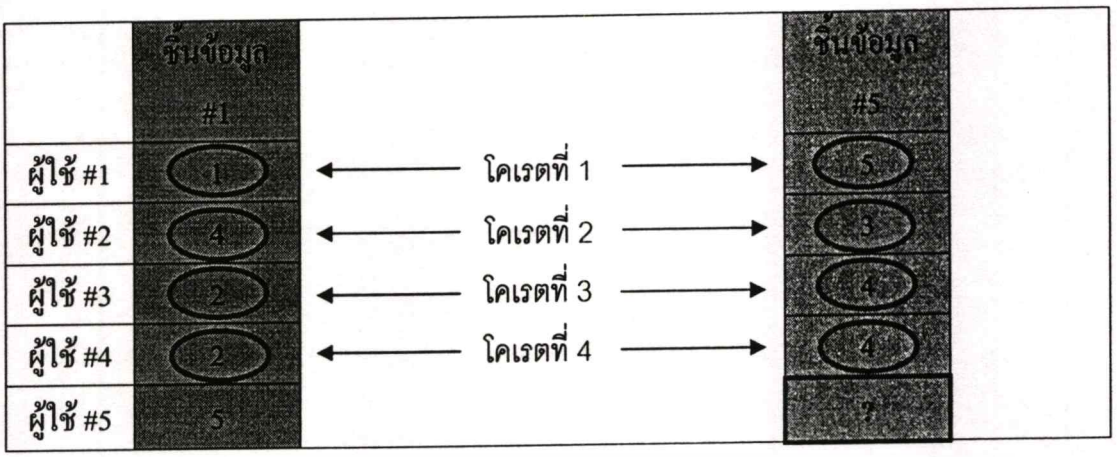
2.1.2.3 ตัวอย่างการทำงานของอัลกอริทึมไอเท็มเบส CF

จากตัวอย่างเมตริกซ์ผู้ใช้-ชิ้นข้อมูลในตารางที่ 2.1 ที่ผ่านมา กำหนดให้ผู้ใช้คนที่ 5 และชิ้นข้อมูลชิ้นที่ 5 เป็นผู้ใช้เป้าหมายและชิ้นข้อมูลเป้าหมายตามลำดับ ดังนั้นเป้าหมายของอัลกอริทึมไอเท็มเบส CF คือ ทำนายว่าผู้ใช้คนที่ 5 จะมีค่าความพึงพอใจต่อชิ้นข้อมูลที่ 5 เป็นเท่าใด ดังตารางที่ 2.3

ตารางที่ 2.3 ตัวอย่างเป้าหมายของอัลกอริทึมไอเท็มเบส CF

	ชิ้นข้อมูล #1	ชิ้นข้อมูล #2	ชิ้นข้อมูล #3	ชิ้นข้อมูล #4	ชิ้นข้อมูล #5	ชิ้นข้อมูล #6
ผู้ใช้ #1	1	5		2	5	5
ผู้ใช้ #2	4	2		2	3	2
ผู้ใช้ #3	2	4	2	2	4	4
ผู้ใช้ #4	2	4		2	4	2
ผู้ใช้ #5	5	4		1	?	3

เริ่มต้นอัลกอริทึมไอเท็มเบส CF จะทำการค้นหากลุ่มชิ้นข้อมูลที่ใกล้เคียง โดยนำชิ้นข้อมูลที่ 5 ไปเปรียบเทียบกับทุกชิ้นข้อมูลที่ผู้ใช้เป้าหมายเคยให้เรตติ้งไว้ (จากตารางที่ 2.3 ได้แก่ ชิ้นข้อมูลที่ 1,2,4 และ 6) ดังนั้นในขั้นตอนนี้จะเป็นการเปรียบเทียบระหว่างชิ้นข้อมูลที่ 5 กับ 1, 5 กับ 2, 5 กับ 4 และ 5 กับ 6 ดังแสดงในรูปที่ 2.5 – 2.8 ตามลำดับ

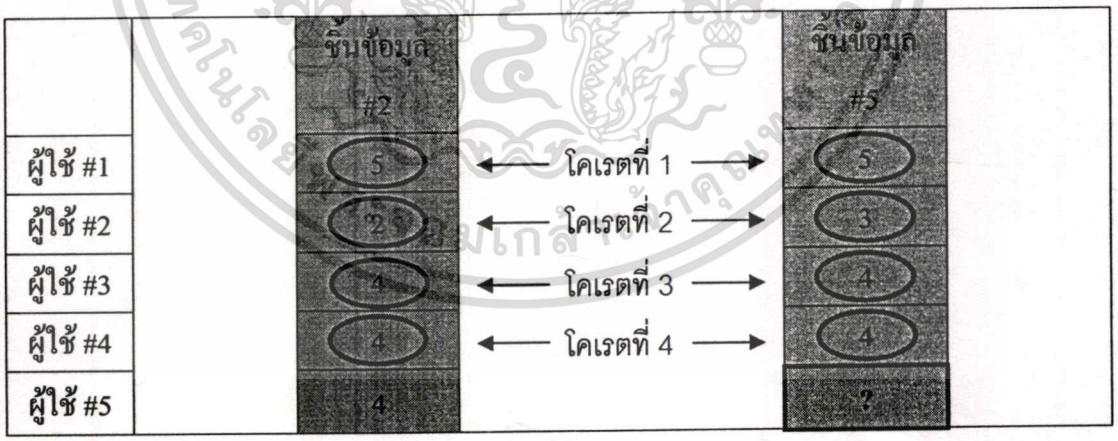


รูปที่ 2.5 การคำนวณความคล้ายคลึงจากโคเรตระหว่างชิ้นข้อมูล 5 และ 1

จากรูปที่ 2.5 สามารถคำนวณหาความคล้ายคลึงด้วยวิธี Cosine ได้ดังนี้

$$RSim(t_5, c_1) = \frac{(5*1)+(3*4)+(4*2)+(4*2)}{\sqrt{5^2+3^2+4^2+4^2} * \sqrt{1^2+4^2+2^2+2^2}}$$

$$= \frac{33}{\sqrt{66} * \sqrt{25}} = 0.81$$

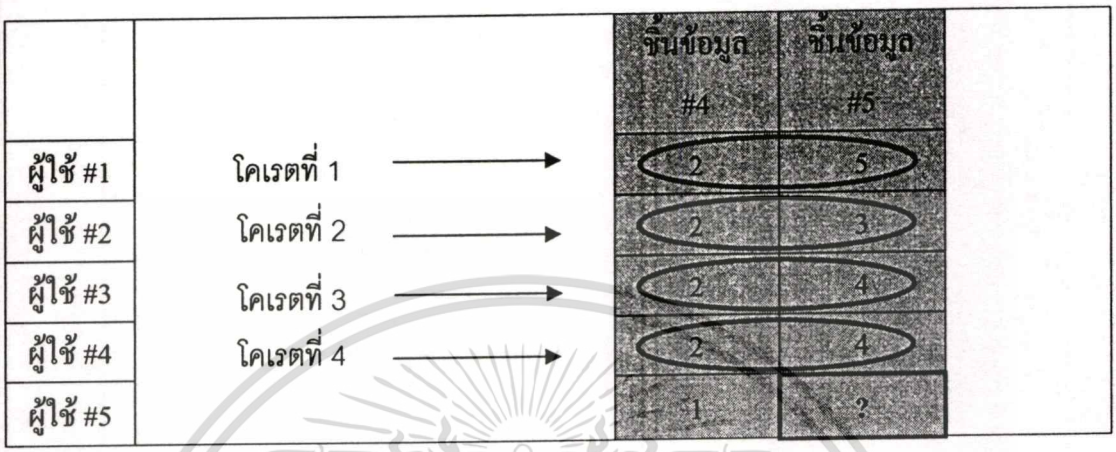


รูปที่ 2.6 การคำนวณความคล้ายคลึงจากโคเรตระหว่างชิ้นข้อมูล 5 และ 2

จากรูปที่ 2.6 สามารถคำนวณหาความคล้ายคลึงด้วยวิธี Cosine ได้ดังนี้

$$RSim(t_5, c_2) = \frac{(5*5)+(3*2)+(4*4)+(4*4)}{\sqrt{5^2+3^2+4^2+4^2} * \sqrt{5^2+2^2+4^2+4^2}}$$

$$= \frac{33}{\sqrt{66} * \sqrt{61}} = 0.52$$



รูปที่ 2.7 การคำนวณความคล้ายคลึงจากโคเรตระหว่างชิ้นข้อมูล 5 และ 4

จากรูปที่ 2.7 สามารถคำนวณหาความคล้ายคลึงด้วยวิธี Cosine ได้ดังนี้

$$RSim(t_5, c_4) = \frac{(5*2)+(3*2)+(4*2)+(4*2)}{\sqrt{5^2+3^2+4^2+4^2} * \sqrt{2^2+2^2+2^2+2^2}}$$

$$= \frac{32}{\sqrt{66} * \sqrt{16}} = 0.98$$

รูปที่ 2.8 การคำนวณความคล้ายคลึงจากโคเรตระหว่างชิ้นข้อมูล 5 และ 6

จากรูปที่ 2.8 สามารถคำนวณหาความคล้ายคลึงด้วยวิธี Cosine ได้ดังนี้

$$RSim(t_5,c_6)=\frac{(5*5)+(3*2)+(4*4)+(4*2)}{\sqrt{5^2+3^2+4^2+4^2}*\sqrt{5^2+2^2+4^2+2^2}}$$

$$=\frac{55}{\sqrt{66}*\sqrt{49}}=0.97$$

จากการคำนวณหาค่าความคล้ายคลึงด้วยวิธี Cosine ข้างต้น ทำให้ได้ค่าความคล้ายคลึงจากการเปรียบเทียบทั้งหมด ดังนี้

ค่าความคล้ายคลึงระหว่างชิ้นข้อมูลที่ 5 กับ 4 เท่ากับ 0.98

ค่าความคล้ายคลึงระหว่างชิ้นข้อมูลที่ 5 กับ 6 เท่ากับ 0.97

ค่าความคล้ายคลึงระหว่างชิ้นข้อมูลที่ 5 กับ 1 เท่ากับ 0.81

ค่าความคล้ายคลึงระหว่างชิ้นข้อมูลที่ 5 กับ 2 เท่ากับ 0.52

หลังจากนั้นในขั้นตอนสุดท้ายนำกลุ่มของชิ้นข้อมูลที่มีลักษณะการให้เรตติ้งคล้ายคลึงกับชิ้นข้อมูลเป้าหมายมากที่สุดจำนวน K ชิ้น มาทำนายหาค่าความพึงพอใจ ด้วยเทคนิค Weighted sum ตามสมการที่ 2.4 ในที่นี้กำหนดให้ขนาดของชิ้นข้อมูลที่ใกล้เคียง (K) มีขนาดเท่ากับ 2 ดังนั้นจึงนำเฉพาะชิ้นข้อมูลที่ 4 และ 6 ซึ่งมีค่าความคล้ายคลึงกับชิ้นข้อมูลเป้าหมายมากที่สุด 2 อันดับแรกมาทำนายได้ดังนี้

$$K=\{c_4,c_6\}$$

$$P_{u_5,t_5}=\frac{(RSim(t_5,c_4)*R_{u_5,c_4})+(RSim(t_5,c_6)*R_{u_5,c_6})}{|RSim(t_5,c_4)+RSim(t_5,c_6)|}$$

แทนค่า

$$P_{u_5,t_5}=\frac{(0.98*1)+(0.97*3)}{|0.98+0.97|}$$

$$P_{u_5,t_5}=1.99$$

ดังนั้นผลลัพธ์จากการทำนายสรุปได้ว่า ผู้ใช้คนที่ 5 น่าจะมีค่าความพึงพอใจหรือมีรสนิยมต่อชิ้นข้อมูลที่ 5 เท่ากับ 1.99 ถือเป็นการสิ้นสุดการทำงานของอัลกอริทึมไอเท็มเบส CF

2.1.2.4 ปัญหาของอัลกอริทึมไอเท็มเบส CF

เนื่องจากอัลกอริทึมไอเท็มเบส CF อาศัยโคเรตในการค้นหาชิ้นข้อมูลที่มีลักษณะการให้เรตติ้งที่ใกล้เคียง จึงเป็นสาเหตุสำคัญที่ทำให้เกิด 2 ปัญหา ดังต่อไปนี้

1. ปัญหาการให้เรตติ้งต่อชิ้นข้อมูลที่ไม่ทั่วถึง (Sparsity problem) เป็นปัญหาที่เกิดจากจำนวนชิ้นข้อมูลที่เพิ่มมากขึ้นจนส่งผลให้ผู้ใช้แต่ละคนไม่สามารถให้เรตติ้งต่อชิ้นข้อมูลได้ไม่ทั่วถึง เช่น ผู้ใช้อาจจะให้เรตติ้งต่อชิ้นข้อมูลทั้งหมดได้เพียง 0.1% ของชิ้นข้อมูล 1 ล้านชิ้น หมายความว่าผู้ใช้แต่ละคนให้เรตติ้งต่อชิ้นข้อมูลได้เพียง 1 พันชิ้นจากชิ้นข้อมูลทั้งหมด 1 ล้านชิ้น ทำให้เรตติ้งที่ผู้ใช้มีต่อชิ้นข้อมูลไม่เพียงพอต่อการคำนวณหาความคล้ายคลึงระหว่างชิ้นข้อมูล ดังแสดงในรูปที่ 2.9 ในกรอบเส้นประแสดงถึงการให้เรตติ้งที่ไม่ทั่วถึงของผู้ใช้ 4 คนต่อชิ้นข้อมูลทั้งหมด 8 ชิ้น

	ชิ้นข้อมูล #1	ชิ้นข้อมูล #2	ชิ้นข้อมูล #3	ชิ้นข้อมูล #4	ชิ้นข้อมูล #5	ชิ้นข้อมูล #6	ชิ้นข้อมูล #7	ชิ้นข้อมูล #8
ผู้ใช้ #1	5	4	-	4	-	-	-	-
ผู้ใช้ #2	5	4	-	-	-	-	-	-
ผู้ใช้ #3	1	-	4	-	4	-	-	-
ผู้ใช้ #4	-	5	-	-	-	-	-	-

รูปที่ 2.9 ตัวอย่างการให้เรตติ้งต่อชิ้นข้อมูลที่ไม่ทั่วถึง

จากรูปที่ 2.9 สามารถอธิบายตัวอย่างปัญหาได้ โดยใช้อัลกอริทึมไอเท็มเบส CF ทำนายว่าผู้ใช้คนที่ 3 จะมีความพึงพอใจต่อชิ้นข้อมูลที่ 4 เป็นเท่าใด ดังแสดงในรูปที่ 2.10 ขั้นแรกอัลกอริทึมนี้จะนำชิ้นข้อมูลเป้าหมาย (ชิ้นที่ 4) ไปเปรียบเทียบกับทุกชิ้นข้อมูลที่ผู้ใช้เป้าหมาย (ผู้ใช้ที่ 3) เคยให้เรตติ้งไว้ (ชิ้นข้อมูลที่ 1, 3 และ 5) เพื่อค้นหาชิ้นข้อมูลที่ใกล้เคียงที่มีลักษณะการให้เรตติ้งคล้ายคลึงกับชิ้นข้อมูลเป้าหมายที่สุด แต่ปรากฏว่ามีเพียงโคเรตเดียว (แทนด้วยวงกลมเส้นทึบ) ส่วนที่เหลือแสดงถึงข้อมูลเรตติ้งที่ขาดหายไป (แทนด้วยวงกลมเส้นประ) ดังนั้นจึงไม่สามารถคำนวณหาค่าความคล้ายคลึงระหว่างชิ้นข้อมูลที่ 4 กับ 3 และชิ้นข้อมูลที่ 4 กับ 5 ได้เลย เนื่องด้วยปัญหาการให้เรตติ้งต่อชิ้นข้อมูลที่ไม่ทั่วถึง

	ชิ้นข้อมูล #1	ชิ้นข้อมูล #2	ชิ้นข้อมูล #3	ชิ้นข้อมูล #4	ชิ้นข้อมูล #5	ชิ้นข้อมูล #6	ชิ้นข้อมูล #7	ชิ้นข้อมูล #8
ผู้ใช้ #1	5	4	-	4	-	-	-	-
ผู้ใช้ #2	5	4	-	-	-	-	-	-
ผู้ใช้ #3	1	-	4	?	4	-	-	-
ผู้ใช้ #4	-	5	-	-	-	-	-	-

รูปที่ 2.10 ตัวอย่างการค้นหาชิ้นข้อมูลที่ใกล้เคียงกับข้อมูลเรตติ้งที่ไม่ทั่วถึง

2. ปัญหาชิ้นข้อมูลที่ไม่มีการให้เรตติ้งไว้ (First-rater problem) เป็นปัญหาที่เกิดจากชิ้นข้อมูลใหม่หรือชิ้นข้อมูลที่ยังไม่มีผู้ใช้คนใดเคยให้เรตติ้งกับชิ้นข้อมูลเหล่านั้นไว้เลย ทำให้ชิ้นข้อมูลเหล่านั้นไม่สามารถนำมาเปรียบเทียบกับชิ้นข้อมูลใด ๆ ได้เลย ดังแสดงในรูปที่ 2.11 ในกรอบเส้นประแสดงถึงชิ้นข้อมูลที่ยังไม่มีผู้ใช้คนใดให้คะแนนเรตติ้งไว้เลย ดังนั้นจึงไม่สามารถนำชิ้นข้อมูลที่ 6, 7 และ 8 มาทำการคำนวณได้ จนกระทั่งมีผู้ใช้คนแรกให้คะแนนเรตติ้งกับชิ้นข้อมูลเหล่านั้นก่อน จึงเรียกปัญหานี้ว่า First-rater

	ชิ้นข้อมูล #1	ชิ้นข้อมูล #2	ชิ้นข้อมูล #3	ชิ้นข้อมูล #4	ชิ้นข้อมูล #5	ชิ้นข้อมูล #6	ชิ้นข้อมูล #7	ชิ้นข้อมูล #8
ผู้ใช้ #1	5	4	-	4	-	-	-	-
ผู้ใช้ #2	5	4	-	-	-	-	-	-
ผู้ใช้ #3	1	-	4	-	4	-	-	-
ผู้ใช้ #4	-	5	-	-	-	-	-	-

รูปที่ 2.11 ตัวอย่างปัญหาชิ้นข้อมูลที่ไม่มีการให้เรตติ้งไว้

2.1.3 การประเมินผลของวิธี CF

ในปี ค.ศ. 2004, Herlocker [13] แบ่งการประเมินผลของวิธี CF ได้หลายวิธีแต่ในงานวิจัยเล่มนี้เน้นที่การวัดคุณภาพของผลการทำนาย นั่นคือ การวัดความถูกต้อง (Accuracy) สำหรับวิธีการวัดความถูกต้องที่นิยมใช้กันอย่างแพร่หลายในระบบ CF คือ การวัดความถูกต้องทางสถิติ (Statistical accuracy metrics) เป็นการวัดผลความผิดพลาดในการทำนายด้วยวิธีการทางสถิติ ที่เรียกว่า Mean Absolute Error (MAE) ซึ่งวิธีนี้จะทำการวัดผลความผิดพลาดในการทำนายโดยเปรียบเทียบค่าสมบูรณ์ของผลต่างระหว่างเรตติ้งจริงของชิ้นข้อมูลที่ผู้ใช้เคยให้ไว้ในระบบกับเรตติ้งที่ได้จากการทำนายด้วยวิธี CF จากนั้นจึงนำค่าที่ได้มาคำนวณหาค่าความผิดพลาดสมบูรณ์เฉลี่ยหรือ MAE ของการทำนายทั้งหมดตามสมการที่ 2.5 สำหรับค่า MAE นั้น หากมีค่าน้อยจะดี เพราะแสดงให้เห็นว่าวิธี CF ที่พัฒนาขึ้นมานั้นสามารถทำนายหาค่าความพึงพอใจได้อย่างถูกต้องและมีประสิทธิภาพ

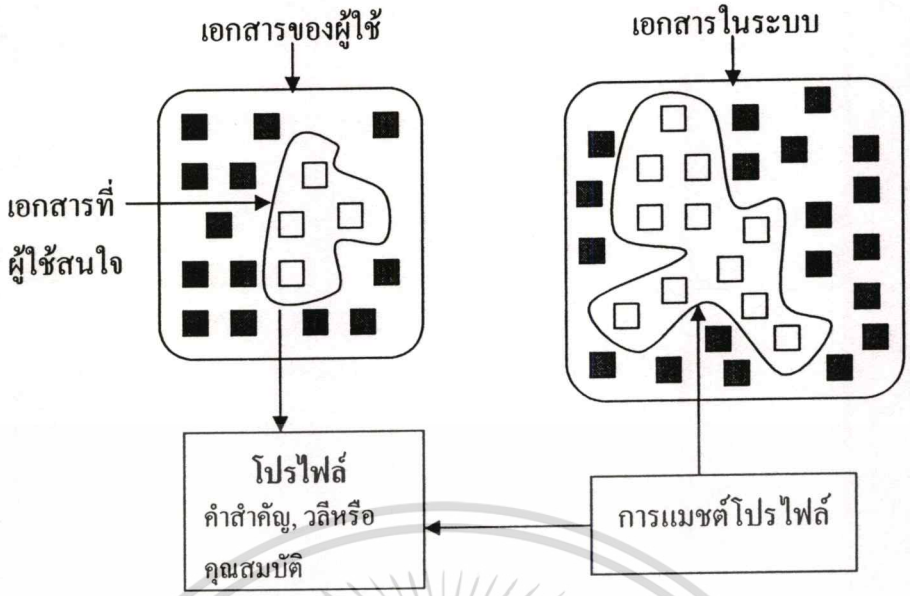
$$MAE = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N} \tag{2.5}$$

จากสมการที่ 2.5 p_i คือ ค่าเรตติ้งจริงที่ผู้ใช้เคยให้ไว้และ q_i คือ ค่าเรตติ้งที่ได้จากการทำนาย และ N คือ จำนวนคู่ของเรตติ้งทั้งหมดที่นำมาเปรียบเทียบกัน

2.1.4 วิธี CBF

เป็นวิธีที่นิยมใช้ในการค้นคืนข้อมูล [4] ที่ให้ความสนใจกับคุณภาพของเนื้อหาข้อมูลเป็นหลักโดยจะสนใจว่าลักษณะข้อมูลนั้นตรงตามโปรไฟล์ของผู้ใช้หรือไม่ ซึ่งถ้าใช่ก็จะนำเสนอข้อมูลนั้นทันที แต่ถ้าไม่ก็จะไม่สนใจแม้ว่าข้อมูลนั้นจะมีลักษณะใกล้เคียงกับข้อมูลที่ผู้ใช้ต้องการก็ตาม

ดังนั้นวิธีการนี้จะเป็นการคำนวณหาค่าความคล้ายคลึงระหว่างเอกสารกับโปรไฟล์ของผู้ใช้ ตามรูปที่ 2.12 แสดงวิธีการ CBF ที่นำเนื้อหาของข้อมูล ดังเช่น คำสำคัญ (Keywords), วลี (Phrases) หรือคุณลักษณะ (Feature) มาสร้างเป็นโปรไฟล์ของผู้ใช้แต่ละคนและใช้วิธีที่ง่ายที่สุด คือ การแมชต์โปรไฟล์ (Profile matching) เพื่อค้นหาข้อมูลที่ผู้ใช้คนนั้นสนใจ โดยใช้เวคเตอสเปซโมเดล [4] แสดงข้อมูลและโปรไฟล์ของผู้ใช้ให้อยู่ในรูปของเมตริกซ์เทอม-เอกสาร (Term-document matrix) ซึ่งแต่ละแถวในเมตริกซ์จะหมายถึงเทอมและแต่ละหลักจะหมายถึงเอกสาร ดังนั้นในแต่ละเซลของเมตริกซ์จะหมายถึงความถี่ที่ปรากฏเทอมในเอกสารนั้น หลังจากนั้นก็นำค่าความถี่ของเทอมในเอกสารเหล่านั้นมาคำนวณหาค่าความคล้ายคลึงระหว่างเอกสารในระบบกับโปรไฟล์ของผู้ใช้ เพื่อค้นหาข้อมูลที่ผู้ใช้นั้นสนใจ



รูปที่ 2.12 วิธีการ CBF

ด้วยสาเหตุนี้ทำให้วิธี CBF ไม่ประสบกับปัญหาทั้งสองในวิธี CF นั่นคือ ปัญหาการให้เรตติ้งต่อชิ้นข้อมูลได้ไม่ทั่วถึงและปัญหาชิ้นข้อมูลที่ยังไม่ได้ให้เรตติ้งไว้ ซึ่งมีงานวิจัยจำนวนมากที่นำเทคนิค CBF ไปใช้กับระบบให้การแนะนำ ดังเช่น CiteSeer [6] ซึ่งถูกพัฒนาขึ้นมาโดยสถาบันวิจัย NEC และปัจจุบันได้กลายมาเป็นเว็บไซต์ที่ใหญ่ที่สุดสำหรับรวบรวมบทความงานวิจัยทางด้านวิทยาการคอมพิวเตอร์ที่นิยมใช้กันอย่างแพร่หลาย

2.1.5 การรวม CBF กับ CF

เนื่องด้วยทั้ง CBF และ CF มีจุดแข็งและจุดอ่อนแตกต่างกันตามตารางที่ 2.4 ดังนั้นจึงเหมาะที่จะนำทั้งสองวิธีมาใช้ร่วมกัน จุดแข็งของ CBF คือ ไม่ประสบกับปัญหาชิ้นข้อมูลใหม่เนื่องด้วยชิ้นข้อมูลใหม่เหล่านั้นสามารถนำไปแมชกับโปรไฟล์ได้ตามคำสำคัญ, วลี หรือคุณลักษณะที่ผู้ใช้คนนั้นสนใจ นอกจากนั้น CBF ยังไม่ประสบกับปัญหาความเบาบางของข้อมูลเรตติ้งอีกด้วย แต่วิธี CBF นี้จะขึ้นอยู่กับเนื้อหาของชิ้นข้อมูลเป็นหลักทำให้ในส่วนของชิ้นข้อมูลที่ผู้ใช้ไม่สนใจจะไม่สามารถนำเสนอได้ ส่วนวิธี CF นั้นมีจุดแข็ง คือ พิจารณาถึงความคิดเห็นของผู้ใช้คนอื่นๆ เป็นหลักทำให้สามารถนำเสนอชิ้นข้อมูลได้หลากหลายประเภทไม่เจาะจงเฉพาะสิ่งที่ผู้ใช้คนนั้นเคยสนใจ ที่ผ่านมาได้มีงานวิจัยเกี่ยวกับการนำข้อดีของทั้งสองวิธีมาใช้ประโยชน์ร่วมกัน เพื่อนำจุดแข็งของแต่ละวิธีมาแก้จุดอ่อนให้กับอีกวิธี

ตารางที่ 2.4 ข้อดีและข้อเสียของวิธี CF และ CBF

	ข้อดี	ข้อเสีย
Collaborative Filtering (วิธี CF)	ไม่ขึ้นกับเนื้อหาแต่ขึ้นกับรสนิยมของผู้ใช้	ประสบกับปัญหา Sparsity และ First-rater
Content-based Filtering (วิธี CBF)	ไม่ประสบกับปัญหา Sparsity และ First-rater	ขึ้นกับเนื้อหาไม่สามารถใช้รสนิยมของผู้ใช้ได้

ในปี ค.ศ. 1997, Balbanovic [5] นำเสนอ Fab ซึ่งเป็นระบบแนะนำหน้าเว็บ (Web page) ที่สามารถแก้ปัญหาความเบาบางของเรตติ้งได้โดยการนำโปรไฟล์ของผู้ใช้แต่ละคนมาเก็บให้อยู่ในรูปของเวคเตอ TF-IDF เพื่อค้นหาโปรไฟล์ที่คล้ายคลึงกันทำให้สามารถนำข้อมูลโปรไฟล์เหล่านี้ไปพิจารณาร่วมกับข้อมูลการให้เรตติ้งจากผู้ใช้ได้

ต่อมาปี ค.ศ. 1999, Sawar [24] ใช้เอเจนกรองข้อมูลเรียกว่า Filterbots ซึ่งเป็นเอเจนที่ค่อยให้เรตติ้งแบบอัตโนมัติ ทำหน้าที่เหมือนผู้ใช้คนหนึ่งในระบบแต่เอเจนสามารถให้เรตติ้งกับระบบได้ตามความหมายของชิ้นข้อมูลเหล่านั้นได้

ล่าสุดในปี ค.ศ. 2002, Melville [18] นำเสนอวิธีการใช้เทคนิค Content-based predictor ในการแปลงเมตริกซ์ผู้ใช้-ชิ้นข้อมูล ให้กลายเป็นเมตริกซ์ Pseudo ผู้ใช้-ชิ้นข้อมูล และใช้เทคนิค CF ในการทำนาย โดยวิธีนี้จะพิจารณาที่แต่ละแถวของเมตริกซ์ผู้ใช้-ชิ้นข้อมูลว่ามีข้อมูลเรตติ้งหรือไม่ ถ้ามีก็ใช้ข้อมูลเรตติ้งนั้น แต่ถ้าไม่มีก็ใช้เทคนิค Content-based predictor ทำนายหาเรตติ้งตามเนื้อหาด้วยวิธีการนี้จะได้เมตริกซ์ Pseudo ผู้ใช้-ชิ้นข้อมูลที่เป็นการผสมผสานกันระหว่างข้อมูลเรตติ้งจริงกับข้อมูลเรตติ้งตามเนื้อหาออกมา

2.2 นิยามการค้นหากฎความสัมพันธ์

การค้นหากฎความสัมพันธ์ (Association rule mining) [3] ในฐานข้อมูลขนาดใหญ่ถือเป็นงานหลักงานหนึ่งในการทำเหมืองข้อมูล กฎความสัมพันธ์สามารถเขียนได้ในรูปไอเท็มเซ็ตที่เป็นเหตุไปสู่ไอเท็มเซ็ตที่เป็นผลซึ่งมีรากฐานมาจากการวิเคราะห์ทางการตลาด เช่น ลูกค้าที่ซื้อหนังสือดาต้าเบสส่วนใหญ่จะซื้อหนังสือดาต้าไมนิ่งด้วย ซึ่งสามารถเขียนกฎความสัมพันธ์ได้เป็น {ดาต้าเบส} $\Rightarrow$ {ดาต้าไมนิ่ง} เป็นต้น พื้นฐานของการค้นหากฎความสัมพันธ์ประกอบด้วยนิยามต่าง ๆ ดังต่อไปนี้

- เซ็ตของไอเท็ม (I) คือ เซ็ตที่มีไอเท็มทั้งหมดเป็นสมาชิก ซึ่งไอเท็มในที่นี้อาจเป็นชื่อสินค้าหรือชื่อใด ๆ ที่เป็นหน่วยพื้นฐานที่จะนำมาทำการเรียนรู้ โดยที่ $I = \{i_1, i_2, ...i_m\}$
- ไอเท็มเซ็ต $(Itemset)$ คือ กลุ่มไอเท็มที่ประกอบด้วยไอเท็มที่เป็นสมาชิกใน I
- ทรานแซคชั่น (T) คือ เซ็ตที่ประกอบด้วยไอเท็มที่เป็นสมาชิกใน I โดยที่ $T \subseteq I$

- ฐานข้อมูล (D) คือ เซ็ตที่มีทรานแซคชั่นทุกตัวเป็นสมาชิก โดยที่ $D=\{T_1,T_2,...,T_n\}$ หรือกล่าวอีกในหนึ่งได้ว่าทรานแซคชั่น T บรรจุไอเท็มเซ็ต X ก็ต่อเมื่อ $X \subseteq T$ เพราะฉะนั้นจึงนิยามกฎความสัมพันธ์ได้ว่า
- กฎความสัมพันธ์ คือ การอุปนัยในรูปแบบ $X \Rightarrow Y$ เมื่อ $X \subset I$, $Y \subset I$ และ $X \cap Y = \phi$

นอกจากนี้ กฎความสัมพันธ์ทุกกฎจะประกอบไปด้วยค่าสนับสนุน (Support) และค่าความเชื่อมั่น (Confidence) ซึ่งมีนิยามดังนี้

- กฎความสัมพันธ์ $X \Rightarrow Y$ มีค่าสนับสนุนเท่ากับ s ในเซ็ตฐานข้อมูล D ก็ต่อเมื่อ s% ของ ทรานแซคชั่นใน D บรรจุ $X \cup Y$ ดังแสดงในสมการที่ 2.6

$$Support(X \Rightarrow Y) = P((X \cup Y) \subseteq T) \tag{2.6}$$

- กฎความสัมพันธ์ $X \Rightarrow Y$ มีค่าความเชื่อมั่นเท่ากับ c ในเซ็ตฐานข้อมูล D ก็ต่อเมื่อ c% ของทรานแซคชั่นใน D ที่บรรจุ X แล้วบรรจุ Y ด้วย ดังแสดงในสมการที่ 2.7

$$Confidence(X \Rightarrow Y) = \frac{Support(X \cup Y)}{Support(X)} \tag{2.7}$$

ปัญหาของการค้นหากฎความสัมพันธ์เป็นปัญหาทางคณิตศาสตร์ สามารถนิยามได้ดังนี้

- การค้นหากฎความสัมพันธ์ คือ การหากฎความสัมพันธ์ทั้งหมดในทรานแซคชั่นทุกตัวของเซ็ตข้อมูลที่กำหนดให้ โดยกฎความสัมพันธ์ที่หาได้ทั้งหมดจะต้องมีค่าสนับสนุนมากกว่าค่าสนับสนุนขั้นต่ำที่กำหนดไว้และมีค่าความเชื่อมั่นมากกว่าค่าความเชื่อมั่นขั้นต่ำที่กำหนดไว้เช่นกัน

ในปี ค.ศ. 2000, Lin [16] ได้นำหลักการค้นหากฎความสัมพันธ์มาประยุกต์ใช้เป็นอัลกอริทึมสำหรับให้การแนะนำด้วยวิธี CF โดยเริ่มต้นจากการนำเมตริกซ์ผู้ใช้-ชิ้นข้อมูลมาแปลงเป็นเรตติ้งแบบไม่ชัดเจนด้วยค่า 1 และ 0 ตามลำดับ หลังจากนั้นก็แปลงให้อยู่ในรูปของทรานแซคชั่น เพื่อค้นหากฎความสัมพันธ์สำหรับผู้ใช้เป้าหมายโดยเฉพาะและนำค่าความเชื่อมั่นของแต่ละกฎที่ได้ไปจัดลำดับความสำคัญเพื่อให้การแนะนำรายการชิ้นข้อมูลออกมา

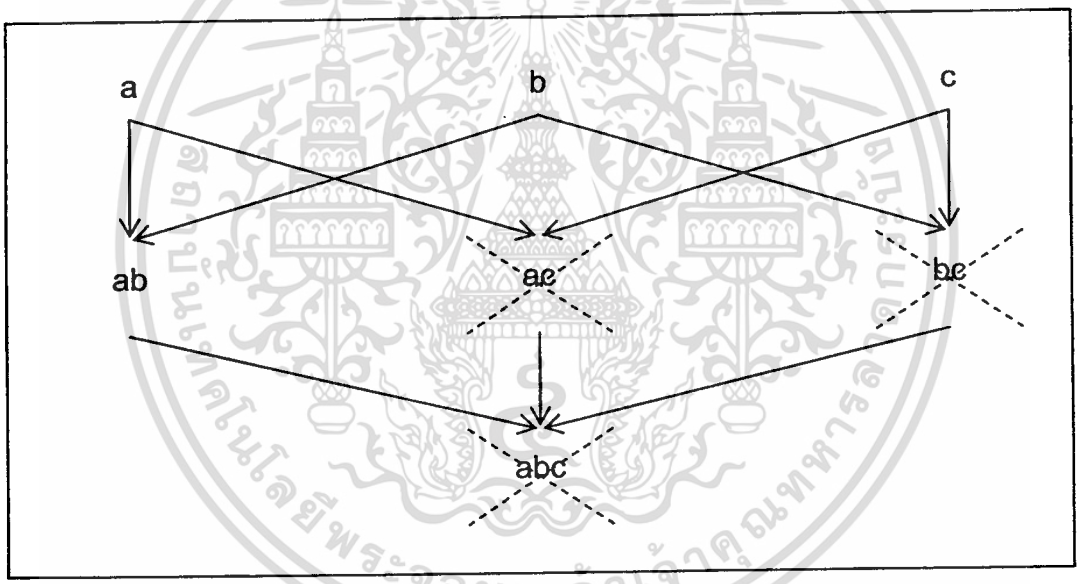
2.2.1 วิธีการค้นหากฎความสัมพันธ์

การค้นหากฎความสัมพันธ์สามารถแบ่งย่อยได้เป็นสองขั้นตอน คือ การหาไอเท็มเซ็ตที่มีค่าสนับสนุนมากกว่าค่าสนับสนุนขั้นต่ำที่กำหนดไว้ เรียกเซ็ตนี้ว่า ไอเท็มเซ็ตปรากฏบ่อย (Frequent itemset) และการนำไอเท็มเซ็ตเหล่านี้มาสร้างเป็นกฎความสัมพันธ์ต่อไป

2.2.1.1 การหาไอเท็มเซ็ตปรากฏบ่อย

การหาไอเท็มเซ็ตที่ปรากฏบ่อย เป็นปัญหาของการค้นหาในสเปซของการจัดหมู่ของไอเท็มเซ็ตทั้งหมดซึ่งสเปซในการค้นหานี้จะมีขนาดเพิ่มขึ้นเป็นเอ็กโปเนนเชียลกับจำนวนไอเท็มทั้งหมด ในไอเท็มเซ็ตซึ่งถ้าไอเท็มเซ็ตมีขนาดใหญ่ สเปซในการค้นหาจะมีขนาดใหญ่มากขึ้นหลายเท่าตัว แต่ในการค้นหานี้ไม่จำเป็นต้องไล่ค้นหาในทุกการจัดหมู่ เพราะสามารถตัดไอเท็มเซ็ตที่มีเซ็ตย่อยเป็นไอเท็มเซ็ตที่ไม่ใช่ไอเท็มเซ็ตปรากฏบ่อยออกได้ หรือกล่าวอีกนัยหนึ่งได้ว่าถ้าแจกแจงแล้วพบไอเท็มเซ็ตใดที่ไม่ใช่ไอเท็มเซ็ตปรากฏบ่อยก็ไม่จำเป็นต้องแจกแจงไอเท็มเซ็ตอื่นๆ ที่มีไอเท็มเซ็ตนี้เป็นเซ็ตย่อยอีกต่อไป

ตัวอย่างเช่นไอเท็ม a, b และ c สามารถสร้างสเปซของไอเท็มเซ็ตทั้งหมดได้ดังแสดงในรูปที่ 2.13 ซึ่งถ้ารู้ว่าไอเท็มเซ็ต {c} ไม่ใช่ไอเท็มเซ็ตปรากฏบ่อยแล้ว ก็ไม่จำเป็นที่จะต้องสร้างหรือตรวจสอบ {ac}, {bc} และ {abc} ซึ่งมี {c} เป็นเซ็ตย่อย



รูปที่ 2.13 สเปซการจัดหมู่ของสมาชิกในไอเท็มเซ็ต {a,b,c}

มีอัลกอริทึมหลายวิธีที่พยายามลดสเปซการค้นหาให้น้อยลงกว่านี้ โดยอัลกอริทึมบางวิธีได้ตัดเล็มสเปซให้เหลือเฉพาะไอเท็มเซ็ตปรากฏบ่อยแบบปิดเท่านั้น

การค้นหาไอเท็มเซ็ตแบบปิดนี้อาจทำได้ทั้งการค้นหาแบบแนวลึกก่อนและการค้นหาแบบแนวกว้างก่อน ซึ่งเมื่อทำการตรวจสอบไอเท็มเซ็ตใด ๆ จะต้องนับทรานแซคชั่นที่บรรจุไอเท็มเซ็ตนั้นการนับทรานแซคชั่นนี้อาจทำได้ทั้งแบบการไล่นับจากเซ็ตข้อมูล และการนับโดยการอินเตอเซกชั่น (Intersection) ของเซ็ตที่เก็บหมายเลขทรานแซคชั่น

อัลกอริทึมในการค้นหากฎความสัมพันธ์ได้แบ่งออกเป็นสี่ประเภทใหญ่ ๆ คือ

- การค้นหาแบบแนวกว้างก่อนที่ใช้การนับทรานแซคชั่น
- การค้นหาแบบแนวกว้างก่อนที่ใช้การอินเตอเซกชั่นของเซ็ตที่เก็บหมายเลขทรานแซคชั่น
- การค้นหาแบบแนวลึกก่อนที่ใช้การนับทรานแซคชั่น
- การค้นหาแบบแนวลึกก่อนที่ใช้การอินเตอเซกชั่นของเซ็ตที่เก็บหมายเลขทรานแซคชั่น

อัลกอริทึม Apriori [2] เป็นอัลกอริทึมพื้นฐานที่แพร่หลายและใช้ในวงกว้าง โดยทำการค้นหาแบบแนวกว้างและใช้การนับทรานแซคชั่น ซึ่งจะสร้างและตรวจสอบไอเท็มเซ็ตปรากฏบ่อยทีละชั้น เริ่มจากไอเท็มเซ็ตที่มีจำนวนสมาชิกเท่ากับหนึ่ง ถ้าไอเท็มเซ็ตใดมีค่าสนับสนุนน้อยกว่าค่าสนับสนุนขั้นต่ำที่กำหนดไว้ก็จะตัดไอเท็มเซ็ตนั้นออกไม่นำไปสร้างไอเท็มเซ็ตในชั้นถัดไป การทำงานของอัลกอริทึมจะวนอย่างนี้ไปเรื่อย ๆ จนกระทั่งไล่ไปทุกระดับชั้น หรือไม่เหลือไอเท็มเซ็ตที่จะสร้างไอเท็มเซ็ตในชั้นถัดไป

ในการนับจำนวนทรานแซคชั่นอัลกอริทึม Apriori จะไล่ทรานแซคชั่นครั้งเดียวในแต่ละระดับชั้น ในการตรวจดูว่าทรานแซคชั่นนั้นบรรจุไอเท็มเซ็ตใดบ้าง เพื่อความรวดเร็วจะเก็บไอเท็มเซ็ตในแต่ละระดับชั้นทั้งหมดไว้ในโครงสร้างต้นไม้แฮช (Hash tree) วิธีการนี้มีขั้นตอนการทำงานดังแสดงในรูปที่ 2.14

```
1)  L_1 = {Frequent 1-itemsets}
2)  For (k = 2; L_{k-1} ≠ ∅; k++) do
3)      C_k = AprioriGen(L_{k-1});
4)      For all Transaction t ∈ D do
5)          C_t = subset(C_k, t);
6)          For all candidates c ∈ C_t do
7)              c.count++
8)      end
9)          L_k = {c ∈ C_k | c.count ≥ minsup}
10) end
11) Answer = ⋃_k L_k
```

รูปที่ 2.14 การทำงานของอัลกอริทึม Apriori

จากรูปที่ 2.14 อัลกอริทึม Apriori มีขั้นตอนการทำงานดังนี้

1. อ่านข้อมูลจากฐานข้อมูลเพื่อตรวจสอบว่าข้อมูลที่มีอยู่ประกอบด้วยไอเท็มใดบ้าง พร้อมทั้งนับค่าสนับสนุนของไอเท็มเหล่านั้น จากนั้นจึงนำไอเท็มที่มีค่าสนับสนุนไม่น้อยกว่าค่าสนับสนุนขั้นต่ำที่กำหนดไว้ มาสร้างเป็นไอเท็มเซ็ตปรากฏบ่อย ขนาด 1 ไอเท็ม (Frequent 1-itemsets) แสดงในบรรทัดที่ 1

2. ในการทำงานรอบต่อ ๆ ไปซึ่งเรียกว่ารอบที่ k จะมีการสร้างไอเท็มเซ็ตที่ต้องพิจารณาที่มีขนาด k ไอเท็ม ตามขั้นตอนย่อยดังต่อไปนี้

2.1 นำไอเท็มเซ็ตปรากฏบ่อย ขนาด k-1 ไอเท็มจากการทำงานในรอบที่ k-1 มาสร้างเป็นไอเท็มเซ็ตที่ต้องพิจารณาขนาด k ไอเท็ม (Candidate k-itemsets) โดยใช้ฟังก์ชั่น AprioriGen เป็นตัวสร้าง (บรรทัดที่ 3)

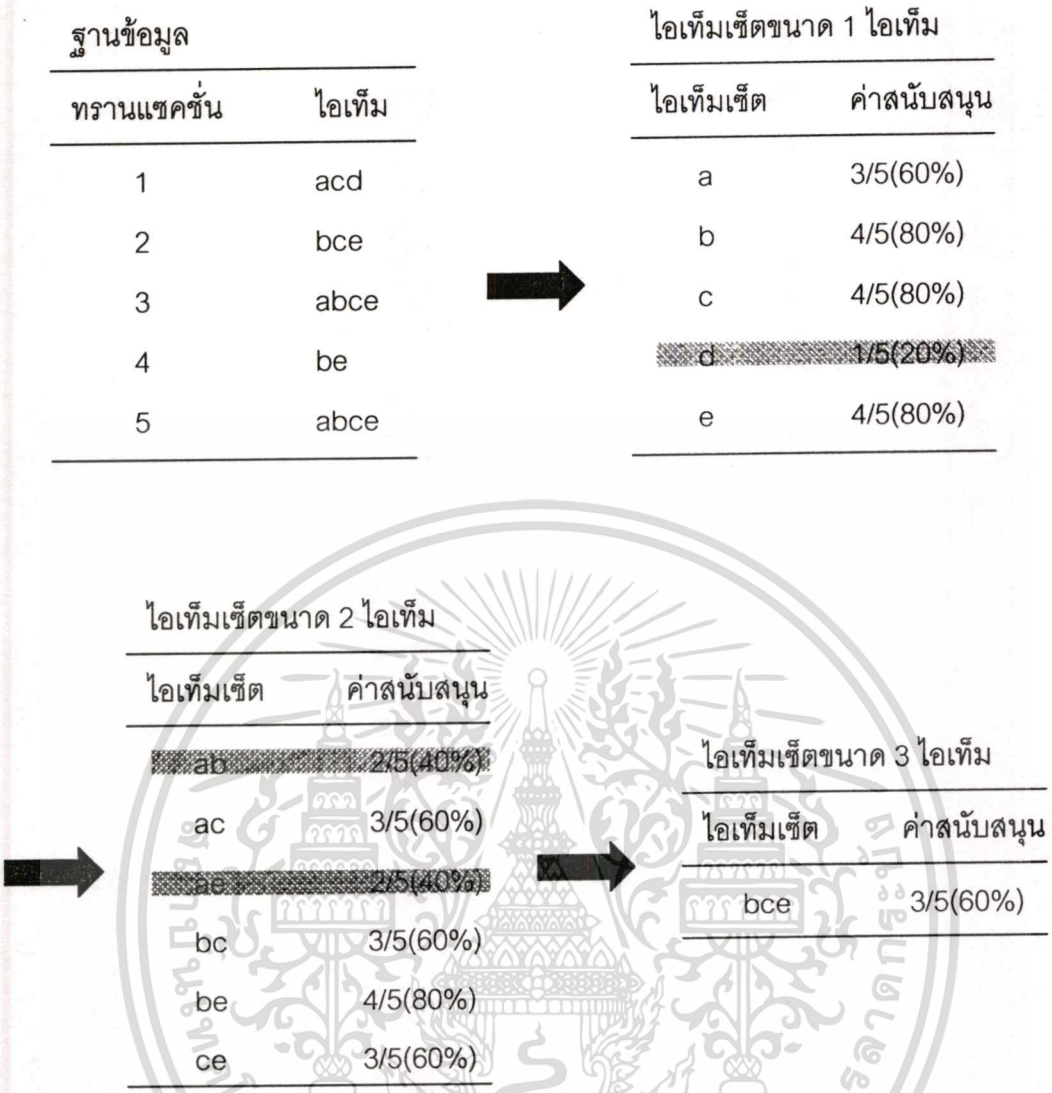
2.2 อ่านข้อมูลจากฐานข้อมูลเพื่อนับค่าสนับสนุนของไอเท็มเซ็ตที่ต้องพิจารณา โดยจะนับค่าสนับสนุนให้กับไอเท็มเซ็ตที่ต้องพิจารณาก็ต่อเมื่อ ทรานแซคชั่นที่อ่านขึ้นมาจากฐานข้อมูลมีไอเท็มเซ็ตที่กำลังพิจารณาประกอบอยู่ด้วยทั้งเซ็ต (บรรทัดที่ 5-8) จากนั้นจึงนำไอเท็มเซ็ตที่ต้องพิจารณาที่มีค่าสนับสนุนไม่น้อยกว่าค่าสนับสนุนขั้นต่ำ มาสร้างเป็นไอเท็มเซ็ตปรากฏบ่อยขนาด k ไอเท็ม ซึ่งจะนำไปใช้ในการสร้างไอเท็มเซ็ตที่ต้องพิจารณาในรอบต่อไป (บรรทัดที่ 9) การทำงานจะหยุดลงเมื่อไม่มีไอเท็มเซ็ตปรากฏบ่อยในรอบที่ k-1 เกิดขึ้น

ฟังก์ชั่น **AprioriGen** เป็นฟังก์ชั่นที่ใช้สร้างไอเท็มเซ็ตที่ต้องพิจารณาขนาด k ไอเท็ม (C_k) ที่มีขั้นตอนการทำงานดังต่อไปนี้

1. สร้างไอเท็มเซ็ตที่ต้องพิจารณาขนาด k ไอเท็ม (C_k) จากไอเท็มเซ็ตปรากฏบ่อยขนาด k-1 ไอเท็ม (L_{k-1})

2. ตัดไอเท็มเซ็ตขนาด k ไอเท็มที่ได้ ที่มีบางเซ็ตย่อยขนาด k-1 ไอเท็มของมันไม่อยู่ในเซ็ตไอเท็มปรากฏบ่อยขนาด k-1 ไอเท็ม ออกจากเซ็ตที่ได้ในขั้นตอนที่ 1

ตัวอย่างเช่น กำหนดให้ L_{k-1} = {{abc} {abd} {acd} {ace} {bcd}} จากขั้นตอนการทำงานของอัลกอริทึม Apriori ขั้นตอนที่ 1 จะได้ C_k = {{abcd} {abce} {acde}} และจากขั้นตอนที่ 2 สมาชิก {abce} และ {acde} จะถูกตัดออกจากเซ็ต C_k เพราะ {abce} มี {abe} เป็นเซ็ตย่อยที่ไม่อยู่ในเซ็ต L_{k-1} เช่นเดียวกับ {acde} ที่มี {ade} เป็นเซ็ตย่อยที่ไม่อยู่ในเซ็ต L_{k-1}



รูปที่ 2.15 ตัวอย่างการหาไอเท็มเซ็ตปรากฏบ่อยจากฐานข้อมูล

จากตัวอย่างในรูปที่ 2.15 แสดงการหาไอเท็มเซ็ตปรากฏบ่อยจากฐานข้อมูลที่ประกอบด้วย 5 ทรานแซคชั่นและกำหนดให้ค่าสนับสนุนขั้นต่ำเท่ากับ 50% เริ่มต้นจากการอ่านข้อมูลจากฐาน ข้อมูล สร้างไอเท็มเซ็ตขนาด 1 ไอเท็มพร้อมทั้งเก็บค่าสนับสนุน จากนั้นจึงลบไอเท็มเซ็ตที่มีค่าสนับสนุนน้อยกว่าค่าสนับสนุนขั้นต่ำที่กำหนดไว้ทิ้ง ในที่นี้คือ {d} หลังจากนั้นสร้างไอเท็มเซ็ตที่ต้องพิจารณาขนาด 2 ไอเท็มที่ผ่านค่าสนับสนุนขั้นต่ำ อ่านข้อมูลจากฐานข้อมูลเพื่อเก็บค่าสนับสนุนของไอเท็มเซ็ตที่ต้องพิจารณาจากนั้นจึงลบไอเท็มเซ็ตที่มีค่าสนับสนุนไม่ผ่านค่าสนับสนุนขั้นต่ำทิ้ง ในที่นี้คือ {ab} และ {ae} สุดท้ายสร้างไอเท็มเซ็ตที่ต้องพิจารณาขนาด 3 ไอเท็มจากไอเท็มเซ็ตขนาด 2 ไอเท็มที่ผ่านค่าสนับสนุนขั้นต่ำ อ่านข้อมูลจากฐานข้อมูลเพื่อเก็บค่าสนับสนุนของไอเท็มเซ็ตที่ต้องพิจารณาจากไอเท็มเซ็ตที่เหลืออยู่ไม่สามารถสร้างไอเท็มเซ็ตที่ต้องพิจารณาขนาด 4 ไอเท็มได้ต่อ การทำงานจึงจบ

ด้วยการทำงานทั้งหมดที่กล่าวมาข้างต้น จะได้ไอเท็มเซ็ตปรากฏบ่อย คือ {a}, {b}, {c}, {e}, {ac}, {bc}, {be}, {ce} และ {bce}

2.2.1.2 การหากฎความสัมพันธ์จากไอเท็มเซ็ตปรากฏบ่อย

เมื่อได้ไอเท็มเซ็ตปรากฏบ่อยมาแล้ว จำเป็นต้องหากฎความสัมพันธ์จากไอเท็มเซ็ตปรากฏบ่อย โดยกฎความสัมพันธ์ที่ได้จะต้องมีค่าความเชื่อมั่นมากกว่าค่าความเชื่อมั่นขั้นต่ำที่กำหนดไว้ การหากฎความสัมพันธ์จากไอเท็มเซ็ตปรากฏบ่อยถือเป็นปัญหาการค้นหาในสเปซของการจัดหมู่เช่นกัน โดยกฎความสัมพันธ์ทั้งหมดที่เป็นไปได้ คือ การจัดหมู่ทุกแบบของสมาชิกในไอเท็มเซ็ตปรากฏบ่อย การจัดหมู่นี้อาจจะเป็นการจัดหมู่ของกฎความสัมพันธ์ส่วนที่เป็นเหตุ หรือการจัดหมู่ของกฎความสัมพันธ์ส่วนที่เป็นผลก็ได้ ในที่นี้ให้กฎความสัมพันธ์อยู่ในรูปแบบ $(F-S)\Rightarrow S$ เมื่อ S คือ ไอเท็มเซ็ตของกฎความสัมพันธ์ส่วนที่เป็นผล และ F คือ ไอเท็มเซ็ตปรากฏบ่อย ในที่นี้ $S\subset F$ และ $S\neq\phi$ จากรูปแบบของกฎความสัมพันธ์บอกได้ว่าการค้นหาเฉพาะการแจกแจงกฎความสัมพันธ์ส่วนที่เป็นผล ก็สามารถนำไปสู่กฎความสัมพันธ์ส่วนที่เป็นเหตุและกฎความสัมพันธ์ได้

การลดขนาดของสเปซการค้นหาสามารถทำได้โดยใช้ทฤษฎีที่ว่า

ถ้า $F-S\Rightarrow S$ ไม่ใช่กฎความสัมพันธ์ที่มีค่าความเชื่อมั่นมากกว่าค่าความเชื่อมั่นขั้นต่ำที่กำหนดไว้แล้ว $F-\bar{S}\Rightarrow\bar{S}$ ก็จะไม่ใช่เช่นกัน เมื่อ $S\subseteq\bar{S}$

ซึ่งทฤษฎีนี้สามารถพิสูจน์ได้ และผลจากทฤษฎีสามารถนำมาใช้สร้างเป็นอัลกอริทึมสำหรับหากฎความสัมพันธ์จากไอเท็มเซ็ตปรากฏบ่อยได้

จากตัวอย่างการทำงานของอัลกอริทึม Apriori ในรูปที่ 2.15 สามารถนำไอเท็มเซ็ตปรากฏบ่อยที่ได้ มาค้นหากฎความสัมพันธ์ได้ดังแสดงในตารางที่ 2.5

จากตารางที่ 2.5 กฎความสัมพันธ์ทั้งหมดที่หาได้มี 14 กฎ ซึ่งสามารถอธิบายตัวอย่างกฎความสัมพันธ์ด้วยการแปลความหมายได้ดังนี้ กฎความสัมพันธ์ลำดับที่ 1 คือ $\{a\}\Rightarrow\{c\}$ มีค่าความเชื่อมั่นเท่ากับ 3/3(100%) หมายความว่า ทรานแซคชั่นทั้งหมดในฐานข้อมูลมีไอเท็ม $\{a\}$ ปรากฏอยู่ มีความเป็นไปได้ 100% ที่จะมี $\{c\}$ ปรากฏอยู่ด้วยและกฎความสัมพันธ์อีก 13 กฎที่เหลือก็สามารถอธิบายได้ในทำนองเดียวกัน

ตารางที่ 2.5 กฎความสัมพันธ์ทั้งหมดที่สร้างจากไอเท็มเซ็ตปรากฏบ่อยจากรูปที่ 2.15

ลำดับที่	กฎ	ค่าความเชื่อมั่น
1	$\{a\} \Rightarrow \{c\}$	3/3(100%)
2	$\{c\} \Rightarrow \{a\}$	3/4(75%)
3	$\{b\} \Rightarrow \{c\}$	3/4(75%)
4	$\{c\} \Rightarrow \{b\}$	3/4(75%)
5	$\{b\} \Rightarrow \{e\}$	4/4(100%)
6	$\{e\} \Rightarrow \{b\}$	4/4(100%)
7	$\{c\} \Rightarrow \{e\}$	3/4(75%)
8	$\{e\} \Rightarrow \{c\}$	3/4(75%)
9	$\{bc\} \Rightarrow \{e\}$	3/3(100%)
10	$\{e\} \Rightarrow \{bc\}$	3/4(100%)
11	$\{b\} \Rightarrow \{ce\}$	3/3(100%)
12	$\{c\} \Rightarrow \{be\}$	3/3(100%)
13	$\{be\} \Rightarrow \{c\}$	3/4(75%)
14	$\{ce\} \Rightarrow \{b\}$	3/3(100%)

จากทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้องดังที่กล่าวมา สรุปว่างานวิจัยเกี่ยวกับวิธีการหรือเทคนิคที่นำมาใช้กับระบบให้การแนะนำ ได้แก่ วิธี CF, วิธี CBF หรือการนำข้อดีของทั้งสองวิธีมาใช้ร่วมกัน เป็นที่นิยมและมีการพัฒนาปรับปรุงอย่างต่อเนื่อง ล่าสุดในปี ค.ศ. 2004, Deshpande [11] ได้นำอัลกอริทึมไอเท็มเบส CF ไปพัฒนาเป็นอัลกอริทึมให้การแนะนำ ซึ่งเป็นการพิสูจน์ว่าอัลกอริทึมไอเท็มเบส CF มีแนวโน้มที่จะนิยมใช้ในอนาคตอันใกล้นี้ ดังนั้นในบทที่จะกล่าวถึงต่อไป จะเป็นรายละเอียดของการรวมวิธี CBF เข้ากับอัลกอริทึมไอเท็มเบส CF โดยใช้กฎความสัมพันธ์เพื่อแก้ปัญหาของอัลกอริทึมไอเท็มเบส CF แบบเดิม

บทที่ 3

การทำนายข้อมูลโดยการรวม CBF กับไอเท็มเบส CF ด้วยกฎความสัมพันธ์

ในบทนี้จะเสนอวิธีการในการปรับปรุงประสิทธิภาพในการวัดความคล้ายคลึงระหว่างชิ้นข้อมูล วิธีการในงานวิจัยนี้ คือ การนำความรู้พื้นฐานเกี่ยวกับกฎความสัมพันธ์ระหว่างคุณสมบัติในแง่ความถี่ที่พบคุณสมบัติหนึ่ง (X) แล้วจะพบอีกคุณสมบัติหนึ่ง (Y) ด้วยค่าความน่าจะเป็นอย่างมีเงื่อนไข หรือกล่าวอีกนัยหนึ่งได้ว่าค่าความเชื่อมั่นของกฎความสัมพันธ์ $X \Rightarrow Y$ สามารถนำมาใช้เป็นระดับความคล้ายคลึงที่ X มีต่อ Y ได้

movie1 {action, thriller}
movie2 {action, comedy, thriller}
movie3 {action, horror, thriller}
movie4 {action, comedy}
movie5 {action, comedy}

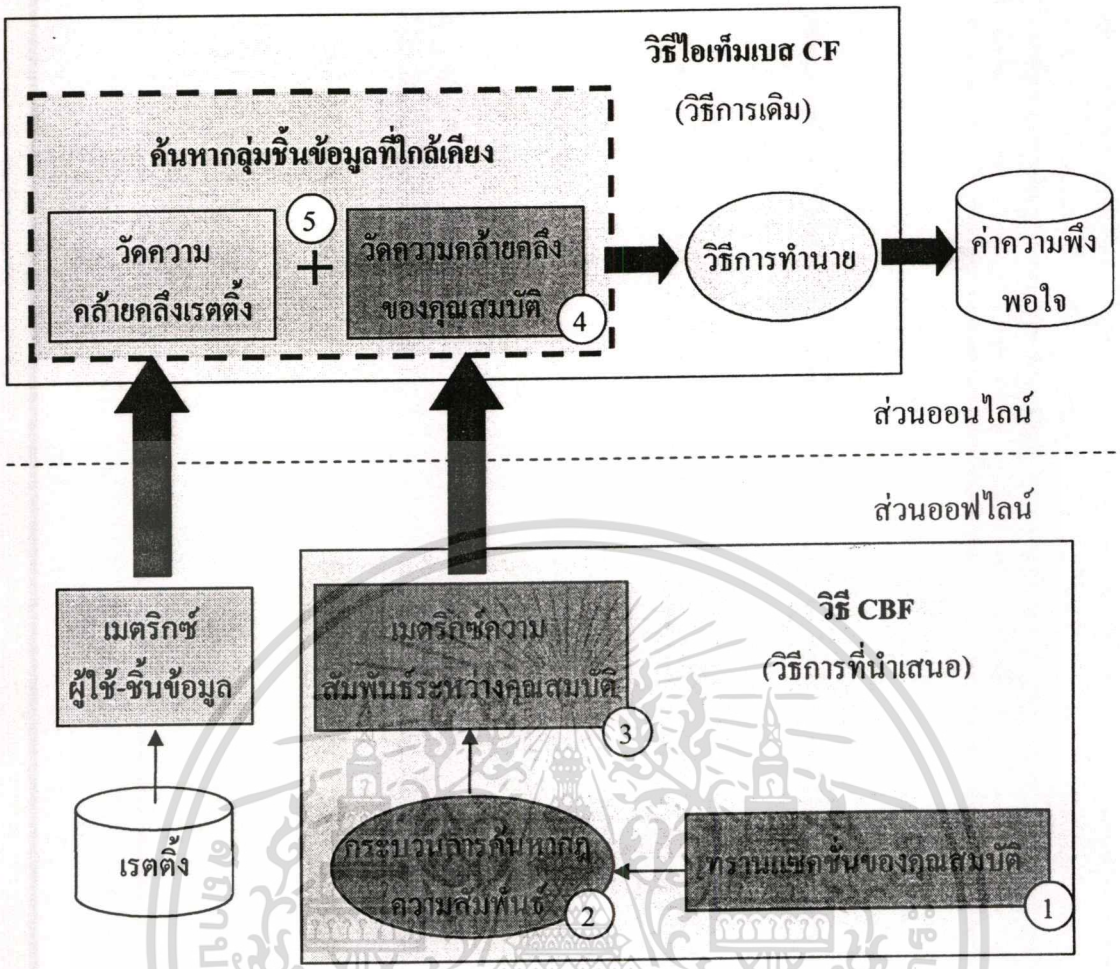
รูปที่ 3.1 แสดงตัวอย่างชิ้นข้อมูลซึ่งประกอบด้วยคุณสมบัติต่างๆ

จากรูปที่ 3.1 เห็นว่าเมื่อคำนวณค่าความคล้ายคลึงเชิงมุมระหว่าง movie1 กับ movie2 และ movie1 กับ movie3 จะมีค่าคงที่แม้ว่าจำนวนของ movie จะเพิ่มขึ้นหรือลดลง แต่ถ้าพิจารณาในแง่กฎความสัมพันธ์ ชิ้นข้อมูล movie2 น่าจะมีความคล้ายคลึงมากกว่าชิ้นข้อมูล movie3 เพราะจากชิ้นข้อมูลทั้งหมด 5 ชิ้นค้นพบว่าค่าความเชื่อมั่นของกฎความสัมพันธ์ *action* $\Rightarrow$ *comedy* มีค่าเท่ากับ 3/5 หรือ 0.6 ส่วนกฎ *action* $\Rightarrow$ *horror* มีค่าความเชื่อมั่นเพียง 1/5 หรือ 0.2 ดังนั้นงานวิจัยนี้จะใช้ค่าความเชื่อมั่นของกฎความสัมพันธ์ดังกล่าวมาปรับปรุงการวัดค่าความคล้ายคลึงระหว่างชิ้นข้อมูลให้มีความยืดหยุ่นขึ้น กล่าวคือ ชิ้นข้อมูลที่คล้ายคลึงกันไม่จำกัดอยู่เพียงการปรากฏของคุณสมบัติเดียวกันเท่านั้น

หัวข้อที่จะกล่าวถึงต่อไป เป็นรายละเอียดของขั้นตอนในการนำความรู้พื้นฐานของกฎความสัมพันธ์มาประยุกต์ใช้ในการคำนวณความคล้ายคลึงระหว่างชิ้นข้อมูลเพื่อรวมเข้ากับอัลกอริทึมไอเท็มเบส CF

3.1 ภาพรวมของการออกแบบวิธีการที่นำเสนอ

การออกแบบวิธีการที่นำเสนอจะอาศัยหลักการเดิมของวิธีไอเท็มเบส CF ดังที่ได้อธิบายไว้ในหัวข้อ 2.1.2 ที่ประกอบไปด้วย 2 ส่วนหลัก ๆ คือ ส่วนออฟไลน์ และออนไลน์ ในส่วนออฟไลน์ จะเพิ่มส่วนที่ทำการเรียนรู้ หรือเป็นการสร้างแบบจำลองในการบรรยาย (Descriptive modeling) ในที่นี้จะเป็นการค้นหากฎความสัมพันธ์ซึ่งเป็นอัลกอริทึมหลักที่ใช้ในการค้นหารูปแบบและกฎความ สัมพันธ์ที่ซ่อนอยู่ในทรานแซคชั่นของคุณสมบัติ ผลลัพธ์ที่ได้จากการเรียนรู้นี้จะถูกนำไปใช้ใน ส่วนออนไลน์ เพื่อเพิ่มการค้นหาชิ้นข้อมูลที่มีลักษณะใกล้เคียง(คล้าย) ด้วยวิธีการวัดความคล้ายคลึงของคุณสมบัติ และนำไปรวมเข้ากับขั้นตอนการค้นหาชิ้นข้อมูลที่ใกล้เคียงในอัลกอริทึมไอเท็มเบส CF ดังแสดงในรูปที่ 3.2 หรือกล่าวอีกนัยหนึ่งได้ว่า ส่วนออฟไลน์ ก็คือ วิธี CBF ที่อาศัยการประยุกต์ใช้เทคนิคการค้นหากฎความสัมพันธ์ (หรือวิธีการที่นำเสนอ) และส่วนออนไลน์ คือ การรวมวิธีการที่นำเสนอเข้ากับอัลกอริทึมไอเท็มเบส CF



รูปที่ 3.2 แสดงภาพรวมของการออกแบบวิธีการที่นำเสนอ

3.2 ขั้นตอนการทำงานในส่วนออฟไลน์

ประกอบไปด้วย 3 ส่วนหลัก ๆ ได้แก่ ทรานแซคชั่นของคุณสมบัติ, กระบวนการค้นหากฎความสัมพันธ์และเมตริกซ์ความสัมพันธ์ระหว่างคุณสมบัติ โดยมีรายละเอียดดังต่อไปนี้

3.2.1 ทรานแซคชั่นของคุณสมบัติ

เป็นส่วนแสดงผลในรูปแบบความสัมพันธ์ระหว่างชิ้นข้อมูลกับคุณสมบัติที่เกี่ยวข้อง เรียกว่า ทรานแซคชั่นของคุณสมบัติ ดังแสดงในรูปที่ 3.3 เห็นได้ชัดว่า movie1 ถึง movie5 ในรูปที่ 3.1 ได้แปลงมาเป็นทรานแซคชั่นลำดับที่ 1 ถึงลำดับที่ 5 แทน ซึ่งสามารถอธิบายด้วยการแปลความหมายได้ดังนี้ ทรานแซคชั่นลำดับที่ 1 คือ $\{action, thriller\}$ หมายความว่าทรานแซคชั่นลำดับที่ 1 ประกอบด้วยคุณสมบัติ $\{action\}$ และ $\{thriller\}$ ในทำนองเดียวกันกับทรานแซคชั่นลำดับที่ 2 ที่ประกอบด้วยคุณสมบัติ $\{action\}$, $\{comedy\}$ และ $\{thriller\}$ จนกระทั่งครบทั้ง 5 ทรานแซคชั่นถือเป็นอันสิ้นสุดและพร้อมที่จะนำไปค้นหากฎความสัมพันธ์

ทรานแซคชั่น	คุณสมบัติ
1	$\{action, thriller\}$
2	$\{action, comedy, thriller\}$
3	$\{action, horror, thriller\}$
4	$\{action, comedy\}$
5	$\{action, comedy\}$

รูปที่ 3.3 ตัวอย่างทรานแซคชั่นของคุณสมบัติ

3.2.2 กระบวนการค้นหากฎความสัมพันธ์

เป็นกระบวนการที่กระทำกับฐานข้อมูลขนาดใหญ่ซึ่งในที่นี้ คือ ทรานแซคชั่นของคุณสมบัติที่ได้มาจากขั้นตอนที่ผ่านมา เพื่อค้นหารูปแบบและความสัมพันธ์ที่ซ่อนอยู่ในทรานแซคชั่นเหล่านั้น เดิมการค้นหาไอเท็มเซ็ตปรากฏบ่อย (กล่าวไว้ในหัวข้อที่ 2.2.1.1) ตามอัลกอริทึม Apriori แบบเดิม การทำงานของอัลกอริทึมจะวนไล่ค้นหาทุกการจัดหมู่ไปเรื่อยๆ จนกระทั่งไล่ไปทุกระดับชั้น หรือไม่เหลือไอเท็มเซ็ตที่จะสร้างไอเท็มเซ็ตในชั้นถัดไปได้

ในงานวิจัยนี้ได้นำอัลกอริทึม Apriori แบบเดิม มาดัดแปลงให้เหลือการวนไล่ค้นหาเพียงสองรอบ หรือกล่าวอีกนัยหนึ่งได้ว่า "ค้นหาเพียงไอเท็มเซ็ตปรากฏบ่อยขนาด 2 ไอเท็มเท่านั้น" โดยตัดขั้นตอนการวนลูปออก (บรรทัดที่ 2 และ 10) ดังแสดงในรูปที่ 3.4 ส่วนการทำงานอื่นๆ รวมถึงขั้นตอนการหากฎความสัมพันธ์จากไอเท็มเซ็ตปรากฏบ่อยยังเหมือนเดิมทุกประการ

1) L_1 = {Frequent 1-itemsets}

2) [illegible]

3) C_k = AprioriGen(L_{k-1});

4) For all Transaction $t \in D$ do

5) C_t = subset(C_k, t);

6) For all candidates c $\in$ C_t do

7) c.count++

8) end

9) L_k = {c $\in$ C_k | c.count $\geq$ minsup}

10) [illegible]

11) Answer = $\bigcup_k L_k$

รูปที่ 3.4 การดัดแปลงอัลกอริทึม Apriori

ทรานแซคชั่นของคุณสมบัติ

ทรานแซคชั่น	คุณสมบัติ
1	{action, thriller}
2	{action, comedy, thriller}
3	{action, horror, thriller}
4	{action, comedy}
5	{action, comedy}

ไอเท็มเซ็ตขนาด 1 ไอเท็ม

ไอเท็มเซ็ต	ค่าสนับสนุน
{action}	5/5(100%)
{comedy}	3/5(60%)
{horror}	1/5(20%)
{thriller}	3/5(60%)

ไอเท็มเซ็ตขนาด 2 ไอเท็ม

ไอเท็มเซ็ต	ค่าสนับสนุน
{action, comedy}	3/5(60%)
{action, thriller}	3/5(60%)
{comedy, thriller}	1/5(20%)

รูปที่ 3.5 ตัวอย่างการหาไอเท็มเซ็ตปรากฏบ่อยขนาด 2 ไอเท็ม

จากตัวอย่างในรูปที่ 3.5 แสดงการใช้อัลกอริทึม Apriori ที่ผ่านการดัดแปลงเรียบร้อยแล้ว (รูปที่ 3.4) มาใช้ในการค้นหาไอเท็มเซ็ตปรากฏบ่อยขนาด 2 ไอเท็มจากทรานแซคชั่นของคุณสมบัติที่ประกอบด้วย 5 ทรานแซคชั่น โดยกำหนดให้ค่าสนับสนุนขั้นต่ำเท่ากับ 50% และค่าความเชื่อมั่นขั้นต่ำเท่ากับ 50% เริ่มต้นจากการอ่านข้อมูลจากทรานแซคชั่น สร้างไอเท็มเซ็ตขนาด 1 ไอเท็ม พร้อมทั้งเก็บค่าสนับสนุน จากนั้นจึงลบไอเท็มเซ็ตที่มีค่าสนับสนุนน้อยกว่าค่าสนับสนุนขั้นต่ำที่กำหนดไว้ทิ้ง ในที่นี้คือ {horror} สุดท้ายสร้างไอเท็มเซ็ตที่ต้องพิจารณาขนาด 2 ไอเท็มจากไอเท็มเซ็ตขนาด 1 ไอเท็มที่ผ่านค่าสนับสนุนขั้นต่ำ อ่านข้อมูลจากทรานแซคชั่นเพื่อเก็บค่าสนับสนุนของไอเท็มเซ็ตขนาด 2 ไอเท็มจากนั้นจึงลบไอเท็มเซ็ตที่มีค่าสนับสนุนไม่ผ่านค่าสนับสนุนขั้นต่ำทิ้ง ในที่นี้คือ {comedy, thriller} สิ้นสุดการทำงาน ด้วยการทำงานทั้งหมดที่กล่าวมาจะได้ไอเท็มเซ็ตปรากฏบ่อยขนาด 2 ไอเท็ม คือ {action, comedy} และ {action, thriller} เมื่อได้ไอเท็มเซ็ตปรากฏบ่อยขนาด 2 ไอเท็มมาแล้ว ให้นำไอเท็มเซ็ตเหล่านั้นมาหากฎความสัมพันธ์ ตามที่ได้อธิบายไว้ในหัวข้อที่ 2.2.1.2 ได้ออกมา 4 กฎความสัมพันธ์ ดังแสดงในตารางที่ 3.1

ตารางที่ 3.1 กฎความสัมพันธ์ที่สร้างจากไอเท็มเซ็ตปรากฏบ่อยขนาด 2 ไอเท็ม

กฎความสัมพันธ์	ค่าความเชื่อมั่น
$\{action\} \Rightarrow \{comedy\}$	3/5(60%)
$\{comedy\} \Rightarrow \{action\}$	3/3(100%)
$\{action\} \Rightarrow \{thriller\}$	3/5(60%)
$\{thriller\} \Rightarrow \{action\}$	3/3(100%)

จากตารางที่ 3.1 แสดงกฎความสัมพันธ์ 4 กฎพร้อมค่าความเชื่อมั่น ที่นำมาใช้เป็นระดับความคล้ายคลึงระหว่างคุณสมบัติและเก็บอยู่ในรูปของเมตริกซ์ความสัมพันธ์ระหว่างคุณสมบัติ

3.2.3 เมตริกซ์ความสัมพันธ์ระหว่างคุณสมบัติ

เป็นส่วนแสดงผลโดยนำกฎความสัมพันธ์พร้อมกับค่าความเชื่อมั่นที่หาได้จากขั้นตอนที่ผ่านมาไปเก็บไว้ในรูปแบบเมตริกซ์ความสัมพันธ์ระหว่างคุณสมบัติ ซึ่งเป็นข้อมูลที่จำเป็นต้องใช้ในส่วนออนไลน์ ดังแสดงในรูปที่ 3.6 ซึ่งสามารถอธิบายได้ว่า แถว (Row) ในเมตริกซ์หมายถึงส่วนเหตุของกฎความสัมพันธ์และหลัก (Column) ในเมตริกซ์ หมายถึง ส่วนผลของกฎความสัมพันธ์ และข้อมูลที่เก็บอยู่ในแต่ละเซลของเมตริกซ์ คือ ค่าความเชื่อมั่นของกฎความสัมพันธ์เหล่านั้น ตัวอย่างเช่น ค่าความคล้ายคลึงระหว่างคุณสมบัติ $\{action\}$ กับคุณสมบัติ $\{thriller\}$ หาได้จากเมตริกซ์แถวที่ action หลักที่ thriller ซึ่งมีค่าเท่ากับ 0.6 หรือกล่าวอีกนัยหนึ่งได้ว่า ค่าความเชื่อมั่นของกฎความสัมพันธ์ $\{action\} \Rightarrow \{thriller\}$ มีค่าเท่ากับ 0.6

	action	comedy	thriller
action	1	0.6	0.6
comedy	1	1	
thriller	1		1

รูปที่ 3.6 เมตริกซ์ความสัมพันธ์ระหว่างคุณสมบัติ

3.3 ขั้นตอนการทำงานในส่วนออนไลน์

ประกอบไปด้วย 2 ส่วนหลัก ๆ ได้แก่ กระบวนการวัดความคล้ายคลึงของคุณสมบัติและการรวมเชิงเส้น โดยมีรายละเอียดดังต่อไปนี้

3.3.1 กระบวนการวัดความคล้ายคลึงของคุณสมบัติ

การวัดความคล้ายคลึงของคุณสมบัติจะอาศัยค่าความสัมพันธ์กันระหว่างคุณสมบัติสองคุณสมบัติซึ่งเริ่มต้นขึ้นในปี ค.ศ. 2004, Chaiwat [8, 9] ในส่วนนี้จะแบ่งออกเป็น 2 หัวข้อย่อย คือ การหาค่าความคล้ายคลึงของคุณสมบัติเดียว และการนำค่าความคล้ายคลึงของคุณสมบัติเดียวมาใช้ในการคำนวณหาค่าความคล้ายคลึงระหว่างชิ้นข้อมูล

3.3.1.1 การหาค่าความคล้ายคลึงของคุณสมบัติเดียว

ค่าความคล้ายคลึงของคุณสมบัติเดียว คือ ค่าความเชื่อมั่นของกฎความสัมพันธ์ระหว่างคุณสมบัติสองคุณสมบัติแทนด้วยกฎความสัมพันธ์ $X \Rightarrow Y$ โดยที่ X และ Y คือ คุณสมบัติหากค่าความเชื่อมั่นของกฎความสัมพันธ์ระหว่างสองคุณสมบัติมีค่ามาก หมายความว่าคุณสมบัติ X มีความคล้ายคลึงกับคุณสมบัติ Y ค่อนข้างมาก หากค่าความเชื่อมั่นของกฎความสัมพันธ์ระหว่างคุณสมบัติมีค่าน้อย หมายความว่าคุณสมบัติ X มีความคล้ายคลึงกับคุณสมบัติ Y น้อย ในทางกลับกันค่าความเชื่อมั่นของกฎความสัมพันธ์ $Y \Rightarrow X$ เป็นไปได้ที่จะไม่เท่ากับค่าความเชื่อมั่นของกฎความสัมพันธ์ $X \Rightarrow Y$ ดังนั้นด้วยวิธีการที่นำเสนอนี้จึงทำให้ค่าความคล้ายคลึงระหว่างคุณสมบัติ X และ Y อาจมีค่าเท่ากันหรือไม่เท่ากันก็เป็นได้ จึงเรียกว่า ความน่าจะเป็นอย่างมีเงื่อนไขของคุณสมบัติ (Attributes conditional probability) ตัวอย่างเช่น จากรูปที่ 3.6 เห็นได้ชัดเจนว่าคุณสมบัติ action มีค่าความคล้ายคลึงกับคุณสมบัติ comedy เท่ากับ 0.6 ส่วนคุณสมบัติ comedy มีค่าความคล้ายคลึงกับคุณสมบัติ action เท่ากับ 1

ในการคำนวนค่าความคล้ายคลึงของคุณสมบัติเดียว จะอาศัยการคำนวณหาค่าความน่าจะเป็นอย่างมีเงื่อนไขของคุณสมบัติที่มีค่ามากที่สุด ในงานวิจัยนี้ใช้สูตรการคำนวณค่าความคล้ายคลึงซึ่งสามารถอธิบายในรูปสมการคณิตศาสตร์ที่แบ่งออกเป็น 2 กรณี ตามสมการที่ 3.1 สำหรับกรณีบ่งชี้และสมการที่ 3.2 สำหรับกรณีหาค่าสูงสุด

$$SASim(t_i, C) = 1, \ if \ \exists t_i \in T : t_i = c_j \tag{3.1}$$

$$= \max_{j=1..m} AttrConditionalprob(t_i, c_j) \tag{3.2}$$

โดยที่

$SASim(t_i, C)$	คือ	ค่าความคล้ายคลึงของคุณสมบัติเดียว (Single Attribute Similarity) ระหว่างคุณสมบัติ t_i กับคุณสมบัติใน C
T	คือ	ชิ้นข้อมูลเป้าหมาย (Target)
t_i	คือ	คุณสมบัติที่ i ของชิ้นข้อมูล T
C	คือ	ชิ้นข้อมูลที่นำมาเปรียบเทียบ (Compare)
c_j	คือ	คุณสมบัติที่ j ของชิ้นข้อมูล C
m	คือ	จำนวนความสัมพันธ์ทั้งหมดระหว่างคุณสมบัติ t_i กับคุณสมบัติใน C
$AttrConditionalprob(t_i, c_j)$	คือ	ความน่าจะเป็นอย่างมีเงื่อนไขของคุณสมบัติหรือค่าความเชื่อมั่นของกฎความสัมพันธ์ $t_i \Rightarrow c_j$ (เก็บไว้ในเมตริกซ์ความสัมพันธ์ระหว่างคุณสมบัติ)

จากสมการที่ 3.1 และ 3.2 สามารถอธิบายได้ว่า "ค่าความคล้ายคลึงของคุณสมบัติ t_i กับหนึ่งคุณสมบัติใน C มีค่าเท่ากับ 1 ถ้าชิ้นข้อมูล C มีคุณสมบัติ t_i หรืออีกกรณีหนึ่ง คือ ค่าความคล้ายคลึงของคุณสมบัติ t_i กับหนึ่งคุณสมบัติใน C จะมีค่าเท่ากับ ค่าความเชื่อมั่นสูงสุดของกฎความสัมพันธ์ $t_i \Rightarrow c_j$ โดยที่ j มีค่าตั้งแต่ 1 ถึง m"

3.3.1.2 การคำนวณหาค่าความคล้ายคลึงระหว่างชิ้นข้อมูล

ในหัวข้อนี้จะกล่าวถึง การวัดความคล้ายคลึงระหว่างชิ้นข้อมูลโดยอาศัยการคำนวณค่าความคล้ายคลึงของคุณสมบัติเดียว มาใช้คำนวณหาค่าความคล้ายคลึงกันระหว่างชิ้นข้อมูล

ในงานวิจัยนี้ใช้สูตรความคล้ายคลึงของคุณสมบัติเดียวกับการคำนวณค่าเฉลี่ย ซึ่งขอเรียกว่า ค่าความคล้ายคลึงของคุณสมบัติทั้งหมด (Attributes similarity)

กำหนดให้ชิ้นข้อมูลเป้าหมาย T ประกอบด้วยเซ็ตของคุณสมบัติ $T = \{t_1, t_2, ..., t_n\}$ และชิ้นข้อมูล C เป็นชิ้นข้อมูลที่นำมาเปรียบเทียบ ค่าความคล้ายคลึงของคุณสมบัติทั้งหมดคำนวณได้จากสมการที่ 3.3

$$ASim(T, C) = \frac{\sum_{t_i \in T} SASim(t_i, C)}{n} \tag{3.3}$$

โดยที่

$ASim(T,C)$	คือ	ค่าความคล้ายคลึงของคุณสมบัติทั้งหมดระหว่างชิ้นข้อมูล T และ C
T	คือ	ชิ้นข้อมูลเป้าหมาย (Target)
C	คือ	ชิ้นข้อมูลที่นำมาเปรียบเทียบ (Compare)
t_i	คือ	คุณสมบัติที่ i ของชิ้นข้อมูล T
n	คือ	จำนวนคุณสมบัติทั้งหมดใน T
$SASim(t_i,C)$	คือ	ค่าความคล้ายคลึงของคุณสมบัติเดียวจากสมการที่ 3.1และ 3.2

การคำนวณความคล้ายคลึงของคุณสมบัติทั้งหมดระหว่างชิ้นข้อมูลเขียนเป็นขั้นตอนวิธีได้ดังแสดงในรูปที่ 3.7

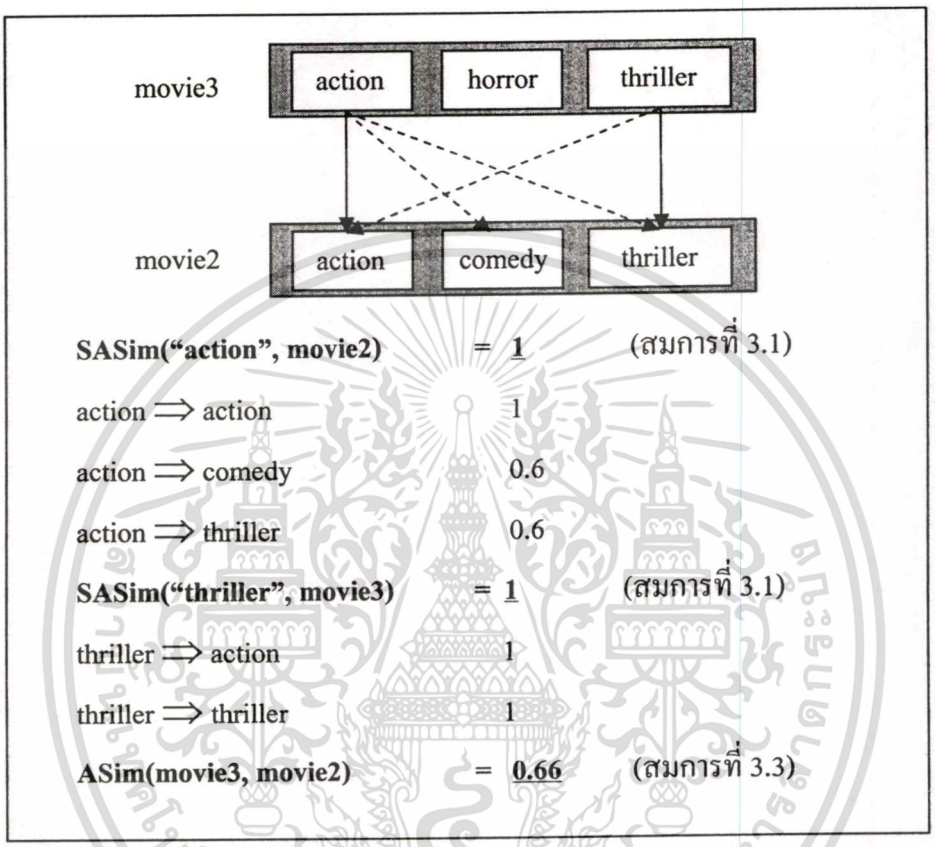
```
1) Loop สำหรับแต่ละ t_i ใน T
2)     Loop สำหรับแต่ละ c_j ใน C
3)         ค้นหาความคล้ายคลึงของคุณสมบัติเดียวระหว่าง t_i และ c_j ∈ C
4)             IF t_i = c_j
5)                 กำหนดค่าความคล้ายคลึงเท่ากับ 1 โดยใช้ สมการที่ 3.1
6)             ELSE
7)                 คำนวณหาค่าความคล้ายคลึงสูงสุดโดยใช้สมการที่ 3.2
8)     End loop
9) End loop
10) คำนวณหาค่าความคล้ายคลึงของคุณสมบัติทั้งหมดระหว่างชิ้นข้อมูลโดยใช้สมการที่ 3.3
```

รูปที่ 3.7 อัลกอริทึมคำนวณค่าความคล้ายคลึงของคุณสมบัติทั้งหมดระหว่างชิ้นข้อมูล

จากอัลกอริทึมในรูปที่ 3.7 แสดงขั้นตอนการคำนวณค่าความคล้ายคลึงของคุณสมบัติทั้งหมดระหว่างชิ้นข้อมูล เริ่มต้นพิจารณาค่าความคล้ายคลึงของคุณสมบัติเดียวทีละคุณสมบัติในชิ้นข้อมูลเป้าหมาย T เปรียบเทียบกับทุกคุณสมบัติในชิ้นข้อมูลที่นำมาเปรียบเทียบ C (บรรทัดที่ 3) คำนวณค่าความคล้ายคลึงของคุณสมบัติเดียวแบ่งออกเป็น 2 กรณี คือ กรณีบ่งชี้ (บรรทัดที่ 4-5) และกรณีหาค่าสูงสุด (บรรทัดที่ 6-7) ทำเช่นนี้จนครบทุกคุณสมบัติในชิ้นข้อมูลเป้าหมาย สุดท้าย

คำนวณหาค่าความคล้ายคลึงของคุณสมบัติทั้งหมดระหว่างชิ้นข้อมูลเป้าหมายและชิ้นข้อมูลที่นำมาเปรียบเทียบด้วยสมการที่ 3.3 (บรรทัดที่ 10)

จากขั้นตอนตามที่กล่าวไปแล้วทั้งหมด ต่อไปนี้เป็นการยกตัวอย่างการคำนวณหาค่าความคล้ายคลึงระหว่าง movie 3 กับ movie 2 ได้ดังแสดงในรูปที่ 3.8



รูปที่ 3.8 การคำนวณหาค่าความคล้ายคลึงระหว่าง movie 3 กับ movie 2

จากรูปที่ 3.8 สามารถอธิบายขั้นตอนตามอัลกอริทึมที่ 3.7 ได้ดังนี้ เริ่มด้วยการพิจารณาค่าความคล้ายคลึงของคุณสมบัติ action ใน movie3 เปรียบเทียบกับทุกคุณสมบัติใน movie2 คำนวณค่าความคล้ายคลึงของคุณสมบัติเดียวแบ่งออกเป็น 2 กรณี คือ กรณีบ่งชี้ (แสดงด้วยลูกศรทึบ) และกรณีหาค่าสูงสุด (แสดงด้วยลูกศรปะ) ดังนั้นค่าความคล้ายคลึงของคุณสมบัติ Action กับคุณสมบัติใน movie2 มีค่าเท่ากับ 1 หรือกล่าวอีกนัยหนึ่งได้ว่า $SASim("action", movie2) = 1$ วนลูปทำเช่นนี้กับคุณสมบัติ horror และ thriller แต่พบว่าคุณสมบัติ horror ไม่มีความสัมพันธ์ใดๆ กับคุณสมบัติใน movie2 เลย มีเพียงคุณสมบัติ thriller เท่านั้นที่มีความคล้ายคลึงกับคุณสมบัติใน movie2 เท่ากับ 1 หรือกล่าวอีกนัยหนึ่งได้ว่า $SASim("thriller", movie2) = 1$ สุดท้ายคำนวณหา

ค่าความคล้ายคลึงของคุณสมบัติทั้งหมดระหว่าง movie3 และ movie2 มีค่าเท่ากับ 0.66 ตามสมการที่ 3.3

3.3.2 การรวมเชิงเส้น

กระบวนการนี้เป็นการนำวิธีการที่นำเสนอไปรวมเข้ากับขั้นตอนการค้นหาชิ้นข้อมูลที่ใกล้เคียงของอัลกอริทึมไอเท็ม CF ดังแสดงไว้ในสมการที่ 3.4 ที่แสดงถึงการรวมเชิงเส้น (Linear combination) ระหว่างค่าความคล้ายคลึงที่หาได้จากวิธีการแบบเดิม (ค่าความคล้ายคลึงจากโคเรตระหว่างชิ้นข้อมูล t และ c) และวิธีการที่นำเสนอ (ค่าความคล้ายคลึงจากคุณสมบัติระหว่างชิ้นข้อมูล t และ c)

$$CSim(t,c) = \alpha \times ASim(t,c) + (1-\alpha) \times RSim(t,c) \tag{3.4}$$

โดยที่

$CSim(t,c)$	คือ	ค่าความคล้ายคลึงรวมระหว่างชิ้นข้อมูล t และ c
$ASim(t,c)$	คือ	ค่าความคล้ายคลึงของคุณสมบัติทั้งหมดระหว่างชิ้นข้อมูล t และ c
$RSim(t,c)$	คือ	ค่าความคล้ายคลึงจากโคเรตระหว่างชิ้นข้อมูล t และ c
α	คือ	ค่าพารามิเตอร์ถ่วงน้ำหนัก

สมการที่ 3.4 จะมีค่า α เป็นค่าพารามิเตอร์ถ่วงน้ำหนักว่าจะเลือกใช้ค่าความคล้ายคลึงแบบใดหรือใช้ทั้งสองแบบร่วมกันได้อย่างเหมาะสมและอัตโนมัติ ถ้าค่า α มีค่าเท่ากับ 0 หมายความว่า ใช้วิธีการแบบเดิมค้นหาชิ้นข้อมูลที่ใกล้เคียง ส่วนถ้า $\alpha \in (0,1)$ หมายความว่าใช้ทั้งสองวิธีร่วมกันเพื่อแก้ปัญหาความเบาบางของข้อมูลการให้เรตติ้ง และสุดท้ายถ้า α มีค่าเท่ากับ 1 หมายถึงใช้วิธีการที่นำเสนอเพียงอย่างเดียว

ในการปรับหาค่า α ที่เหมาะสม งานวิจัยนี้ใช้วิธีการพิจารณาจากจำนวนโคเรตและค่าความคล้ายคลึงจากโคเรตระหว่างชิ้นข้อมูล t และ c ดังแสดงในสมการที่ 3.5

$$\alpha = \frac{1}{1 + (Corated(t,c) * RSim(t,c))} \tag{3.5}$$

โดยที่

$Corated(t,c)$	คือ	จำนวนของผู้ใช้ทั้งหมดที่ให้เรตติ้งกับชิ้นข้อมูล t และ c ร่วมกัน (อธิบายไว้ในหัวข้อที่ 2.1.2.1)
$RSim(t,c)$	คือ	ค่าความคล้ายคลึงจากโคเรตระหว่างชิ้นข้อมูล t และ c (พิจารณาเฉพาะค่าบวกเท่านั้น)

ตามหลักการเดิมของอัลกอริทึมไอเท็มเบส CF ผลจากการทำนายค่าความพึงพอใจจะถูกต้องและน่าเชื่อถือมากน้อยเพียงใด ขึ้นอยู่กับจำนวนของโคเรตและค่าความคล้ายคลึงจากโคเรตเหล่านั้น (ซึ่งในที่นี้จะนำเฉพาะค่าความคล้ายคลึงจากโคเรตที่เป็นค่าบวกมาพิจารณา) ดังนั้นสมการที่ 3.5 จึงสามารถอธิบายได้ว่า ถ้าจำนวนของโคเรตและค่าความคล้ายคลึงจากโคเรตมีค่ามากพอ ก็จะถ่วงน้ำหนักให้กับค่า α มีค่าน้อยลงหรือเข้าใกล้ศูนย์ และถ้าจำนวนของโคเรตและค่าความคล้ายจากโคเรตมีค่าน้อยมาก ก็จะถ่วงน้ำหนักให้ค่า α มีค่ามากขึ้นหรือเข้าใกล้ 1

สุดท้ายเลือกกลุ่มชิ้นข้อมูลที่มีค่าความคล้ายคลึงรวมมากที่สุดมาทำการทำนายโดยใช้เทคนิค Weighted sum ตามหลักการเดิมของอัลกอริทึมไอเท็มเบส CF (อธิบายไว้ในหัวข้อที่ 2.1.2.2) ดังแสดงในสมการที่ 3.6

$$P_{u,t} = \frac{\sum_{k \in K} \left(CSim(t,k) * R_{u,k} \right)}{\sum_{k \in K} \left(|CSim(t,k)| \right)} \tag{3.6}$$

โดยที่

$CSim(t,c)$	คือ	ค่าความคล้ายคลึงรวมระหว่างชิ้นข้อมูล t และ c
K	คือ	กลุ่มชิ้นข้อมูลที่มีค่าความคล้ายคลึงรวมใกล้เคียงกับชิ้นข้อมูล t
$R_{u,k}$	คือ	เรตติ้งที่ผู้ใช้เป้าหมาย u ให้ไว้กับชิ้นข้อมูล k
$P_{u,t}$	คือ	ค่าความพึงพอใจที่คาดว่าผู้ใช้เป้าหมาย u มีต่อชิ้นข้อมูลเป้าหมาย t

เมื่อเข้าใจถึงวิธีการและหลักการของวิธีการที่นำเสนอ ในบทต่อไปจะอธิบายถึงขั้นตอนการทดลองและผลการทดลองวิธีการที่นำเสนอเปรียบเทียบกับวิธีการเดิมที่ใช้กับอัลกอริทึมไอเท็มเบส CF ด้วยวิธีการทางสถิติ

บทที่ 4

การทดลองและผลการทดลอง

งานวิจัยนี้ได้เลือกดาต้าเซ็ตที่เหมาะสมมาทำการทดลอง และเก็บรวบรวมผลการทดลองในแต่ละครั้งพร้อมกับนำผลการทดลองทั้งหมดมาวิเคราะห์ด้วยวิธีการทางสถิติซึ่งมีรายละเอียดดังต่อไปนี้

4.1 เครื่องมือในการทดลอง

ฮาร์ดแวร์

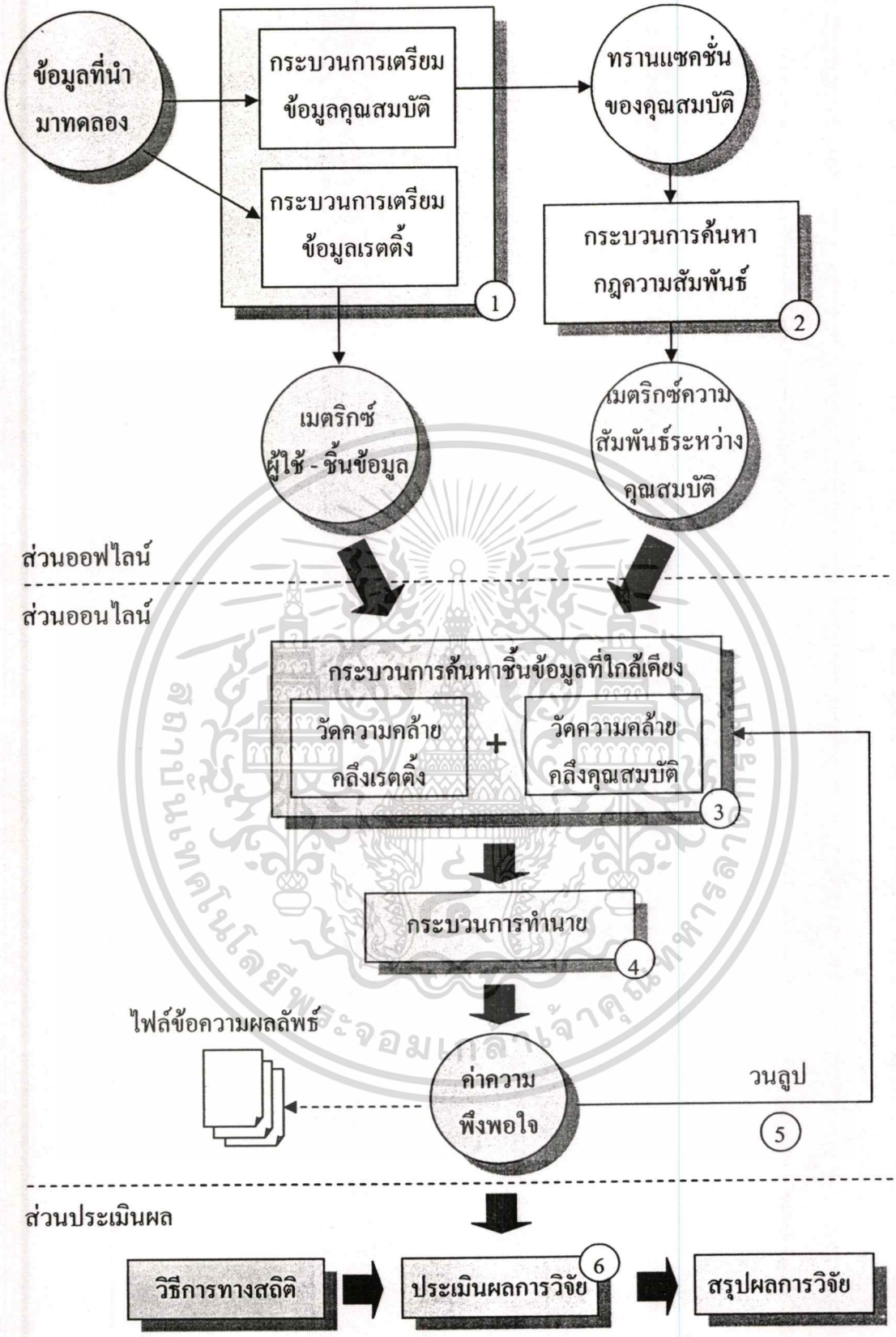
- เครื่องคอมพิวเตอร์ IBM ThinkPad
  - Intel Pentium M 1.3 mobile centrino
  - หน่วยความจำ 256 MB
  - ฮาร์ดดิสก์ขนาด 30 GB
- การ์ดเครือข่าย

ซอฟต์แวร์

- ระบบปฏิบัติการ Windows XP Professional
- ตัวแปรภาษา Java (JSDK 1.4.2)
- NetBeans IDE 3.6

4.2 ขั้นตอนการทดลอง

ดังที่กล่าวไปแล้วในบทที่ผ่านมา วิธีการที่นำเสนอประกอบด้วยขั้นตอนการทำงานหลัก 2 ส่วน คือ ส่วนออนไลน์และออฟไลน์ แต่ในการทดลองได้เพิ่มขั้นตอนในส่วนของการประเมินผลเพื่อใช้ประเมินผลจากการทดลอง ดังนั้นขั้นตอนการทดลองจึงประกอบไปด้วย 6 กระบวนการหลัก ดังแสดงในรูปที่ 4.1 ซึ่งมีรายละเอียดดังต่อไปนี้



รูปที่ 4.1 ขั้นตอนการทดลอง

4.2.1 กระบวนการเตรียมข้อมูล

กระบวนการนี้อยู่ในส่วนออฟไลน์ ข้อมูลที่ใช้ในการทดลองได้นำมาจาก 2 ดาต้าเซ็ตจริง (Natural data sets) ที่นิยมใช้กับงานวิจัยทางด้านนี้โดยเฉพาะอันได้แก่

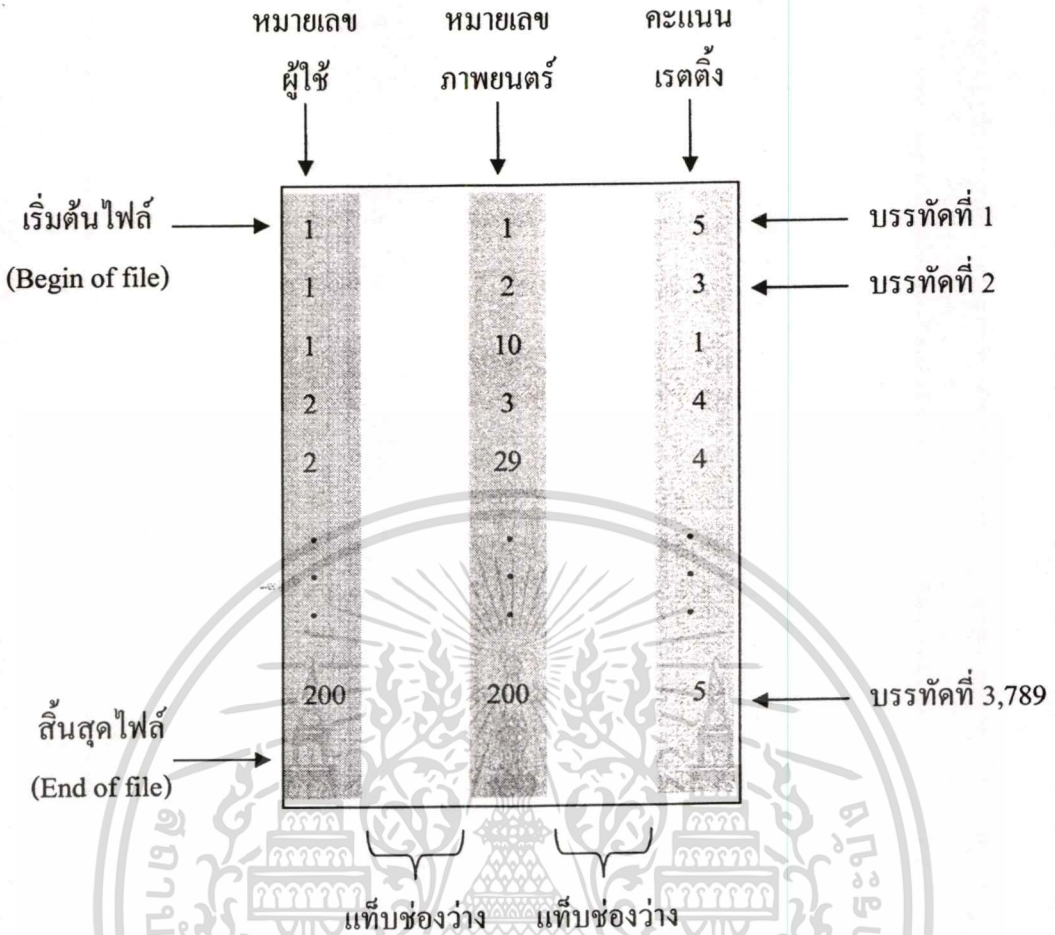
1. ดาต้าเซ็ต MovieLens [14] รวบรวมมาจากผู้ใช้ 943 รายและภาพยนตร์ 1,682 เรื่อง ประกอบด้วย 18 คุณสมบัติ ภายใต้เงื่อนไขว่าผู้ใช้แต่ละรายนั้นต้องมีการให้คะแนนเรตติ้งภาพยนตร์มากกว่า 20 เรื่องขึ้นไป ระดับการให้คะแนนเรตติ้งมี 5 ระดับ เรียงจากความชอบน้อยไปจนถึงความชอบมาก ได้แก่ 1,2,3,4 และ 5 เรตติ้งทั้งสิ้นรวม 100,000 เรตติ้ง

2. ดาต้าเซ็ต EachMovie [17] รวบรวมมาจากผู้ใช้ 72,916 รายและภาพยนตร์ 1,628 เรื่อง ประกอบด้วย 10 คุณสมบัติ ระดับการให้คะแนนเรตติ้งมี 6 ระดับ เรียงจากความชอบน้อยไปจนถึงความชอบมาก ได้แก่ 0, 0.2, 0.4, 0.6, 0.8 และ 1.0 เรตติ้งทั้งสิ้นรวม 2,811,983 เรตติ้ง

กระบวนการเตรียมข้อมูลเริ่มต้นนำแต่ละดาต้าเซ็ต MovieLens และ EachMovie มาสุ่มเลือกข้อมูลการให้เรตติ้งจากจำนวนผู้ใช้ (ผู้ชม) สูงสุดไม่เกิน 200 รายต่อจำนวนชิ้นข้อมูลทั้งหมด (ภาพยนตร์) 200 เรื่อง นอกจากนี้ภาพยนตร์ทั้งหมดจำนวน 200 เรื่อง แต่ละเรื่องยังมีค่าคุณสมบัติแตกต่างกันไป ดังนั้นกระบวนการเตรียมข้อมูลจึงแบ่งออกได้เป็น 2 ส่วน คือ กระบวนการเตรียมข้อมูลเรตติ้ง เป็นการแปลงข้อมูลการให้เรตติ้งให้อยู่ในรูปเมตริกซ์ผู้ใช้–ชิ้นข้อมูล ก่อนเข้าสู่กระบวนการค้นหาชิ้นข้อมูลที่ใกล้เคียง และกระบวนการเตรียมข้อมูลคุณสมบัติ ให้อยู่ในรูปทรานแซคชั่นของคุณสมบัติ ก่อนเข้าสู่กระบวนการค้นหากฎความสัมพันธ์ ซึ่งสามารถอธิบายได้ดังต่อไปนี้

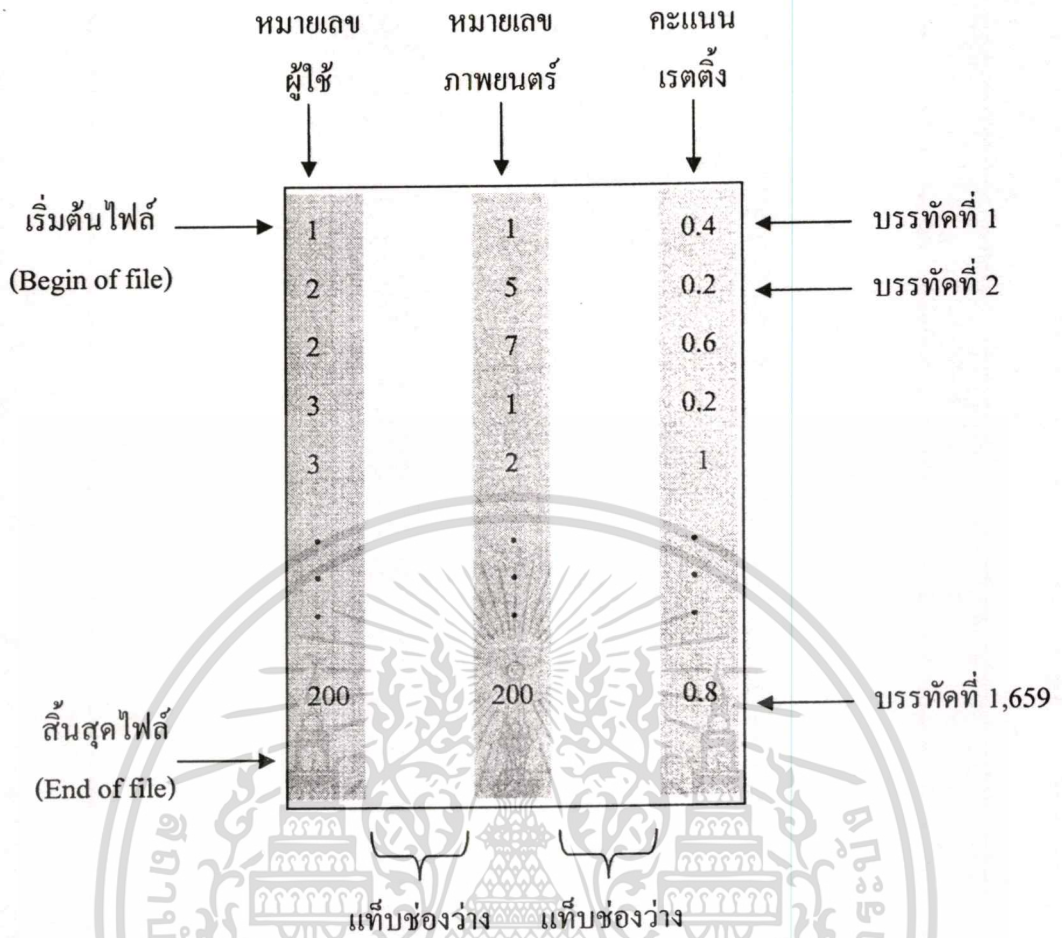
4.2.1.1 กระบวนการเตรียมข้อมูลเรตติ้ง

รูปแบบข้อมูลเรตติ้งที่ใช้ในการทดลอง จะเก็บอยู่ในรูปของไฟล์ข้อความที่ใช้ในการบรรยายถึงรายละเอียดต่างๆ ประกอบด้วย 3 ส่วน คือ หมายเลขผู้ใช้, หมายเลขภาพยนตร์และสุดท้ายเป็นระดับคะแนนเรตติ้ง โดยแต่ละส่วนจะถูกแบ่งแยกด้วยแท็บช่องว่าง ดังแสดงในรูปที่ 4.2 และ 4.3 สำหรับ MovieLens และ EachMovie ตามลำดับ



รูปที่ 4.2 รูปแบบไฟล์ข้อความเรตติ้งของดาต้าเซ็ต MovieLens

จากรูปที่ 4.2 เริ่มต้นไฟล์ที่บรรทัดที่ 1 แสดงถึงผู้ใช้คนที่ 1 ได้ให้ระดับคะแนนเรตติ้งกับภาพยนตร์หมายเลข 1 ไว้เท่ากับ 5 เช่นเดียวกับบรรทัดที่ 2 แสดงถึงผู้ใช้คนที่ 1 ได้ให้ระดับคะแนนเรตติ้งกับภาพยนตร์หมายเลข 2 ไว้เท่ากับ 3 จนกระทั่งครบเรตติ้งทั้งสิ้นรวม 3,789 เรตติ้งถือเป็นอันสิ้นสุดไฟล์



รูปที่ 4.3 รูปแบบไฟล์ข้อความเรตติ้งของคาต้าเซ็ต EachMovie

จากรูปที่ 4.3 อธิบายได้เช่นเดียวกันกับรูปที่ 4.2 ดังนี้ บรรทัดที่ 1 แสดงถึงผู้ใช้คนที่ 1 ได้ให้ระดับคะแนนเรตติ้งกับภาพยนตร์หมายเลข 1 ไว้เท่ากับ 0.4 และเรตติ้งทั้งสิ้นรวม 1,659 เรตติ้ง

นอกจากนั้นตารางที่ 4.1 ยังได้แสดงถึงการวิเคราะห์ข้อมูลเรตติ้งจากคาต้าเซ็ตทั้งสอง พบว่าปริมาณข้อมูลการให้เรตติ้งมีน้อยมากเมื่อเทียบกับปริมาณข้อมูลเรตติ้งทั้งหมดที่จะเป็นไปได้ของทั้งสองดาตาเซ็ต ดังแสดงในสมการที่ 4.1 ซึ่งเป็นการคำนวณหาค่าความหนาแน่นของข้อมูลการให้เรตติ้ง

$$Density = \frac{IU}{I * U} \tag{4.1}$$

โดยที่

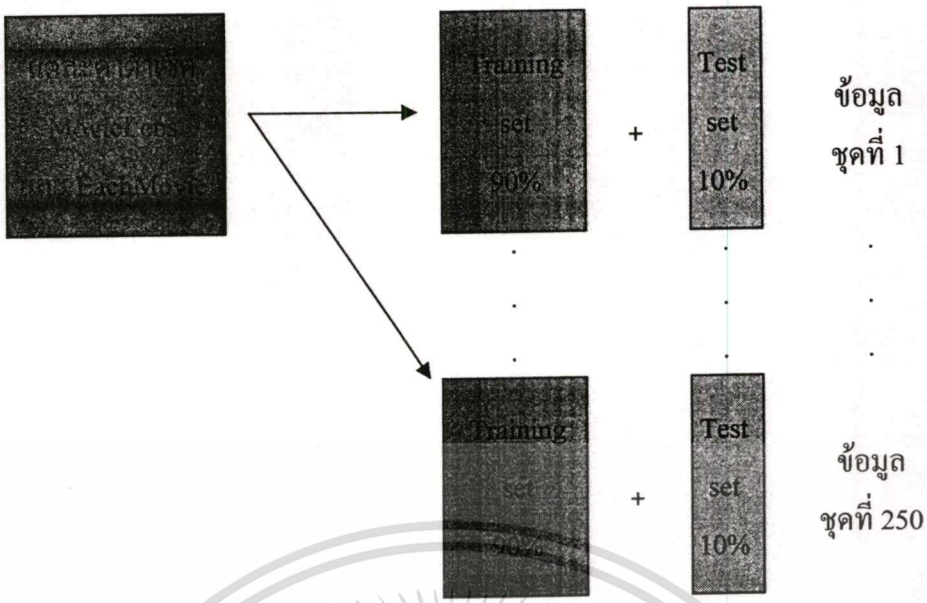
Density	คือ	ความหนาแน่นของข้อมูลเรตติ้ง
IU	คือ	จำนวนเรตติ้งทั้งหมด
I	คือ	จำนวนชิ้นข้อมูลทั้งหมด
U	คือ	จำนวนผู้ใช้ทั้งหมด

ตารางที่ 4.1 วิเคราะห์ข้อมูลเรตติ้งที่นำมาใช้ทดลอง

ข้อมูลเรตติ้งที่ใช้ทดลอง	จำนวนผู้ใช้ทั้งหมด (U)	จำนวนชิ้นข้อมูลทั้งหมด (I)	จำนวนเรตติ้งทั้งหมด (IU)	ความหนาแน่น (Density)
MovieLens	200	200	3,789	0.094
EachMovie	200	200	1,659	0.041

จากตารางที่ 4.1 เป็นการนำจำนวนเรตติ้งทั้งหมดจากทั้งสองดาต้าเซ็ต มาเปรียบเทียบกับจำนวนผู้ใช้ทั้งหมด 200 คนต่อชิ้นข้อมูลทั้งหมด 200 ชิ้น สามารถจะมีการให้เรตติ้งได้สูงสุดเป็น $200 \times 200 = 40{,}000$ เรตติ้ง แต่ในความเป็นจริงพบว่าผู้ใช้ทั้งหมดกลับไม่สามารถให้คะแนนเรตติ้งกับชิ้นข้อมูลทั้ง 200 ชิ้นได้ทั่วถึง ในกรณีของดาต้าเซ็ต MovieLens พบว่ามีความหนาแน่นของข้อมูลการให้เรตติ้งเพียง $\frac{3{,}789}{40{,}000} = 0.094$ หรือกล่าวอีกนัยหนึ่งได้ว่ามีค่าความเบาบางของข้อมูลการให้เรตติ้งเท่ากับ $1 - 0.094 = 0.906$ ในทำนองเดียวกันกับดาต้าเซ็ต EachMovie พบว่ามีความหนาแน่นของข้อมูลการให้เรตติ้งเพียง 0.041 หรือกล่าวอีกนัยหนึ่งได้ว่ามีค่าความเบาบางของข้อมูลการให้เรตติ้งสูงถึง 0.959 ด้วยเหตุผลดังกล่าวนี้ทำให้ทั้งสองข้อมูลการให้เรตติ้งนี้เหมาะสมที่จะนำมาใช้ในการทดลองเป็นอย่างยิ่ง

ในกระบวนการเตรียมข้อมูลเรตติ้ง จะเป็นการดึงข้อมูลจากไฟล์ข้อความเรตติ้ง มาสร้างเป็นแบบจำลองสำหรับเก็บข้อมูลเรตติ้งไว้ในเมโมรี หรือที่ได้กล่าวไว้แล้วในบทที่ผ่านมาเรียกว่าเมตริกซ์ผู้ใช้–ชิ้นข้อมูล โดยแบ่งออกเป็น 2 แบบจำลอง คือ แบบจำลองเรตติ้งสำหรับการสอน (Training set) และแบบจำลองเรตติ้งสำหรับการทดสอบ (Test set) ตามสัดส่วนที่ต้องการใช้ในการทดลอง ซึ่งในงานวิจัยนี้ได้นำดาต้าเซ็ต MovieLens (3,789 เรตติ้ง) และ EachMovie (1,659 เรตติ้ง) มาทำการสุ่มเลือกแบ่งแต่ละดาต้าเซ็ตออกเป็นแบบจำลองสำหรับการสอน และแบบจำลองสำหรับการทดสอบด้วยสัดส่วน 90% ต่อ 10% หรือกล่าวอีกนัยหนึ่งได้ว่าเป็นการแบ่งเมตริกซ์ผู้ใช้–ชิ้นข้อมูล 100 % ออกเป็นแบบจำลองสำหรับการสอน 90% และแบบจำลองสำหรับการทดสอบ 10% ดาต้าเซ็ตละ 250 ชุด รวมทั้งสิ้น 500 ชุด ดังแสดงในรูปที่ 4.4

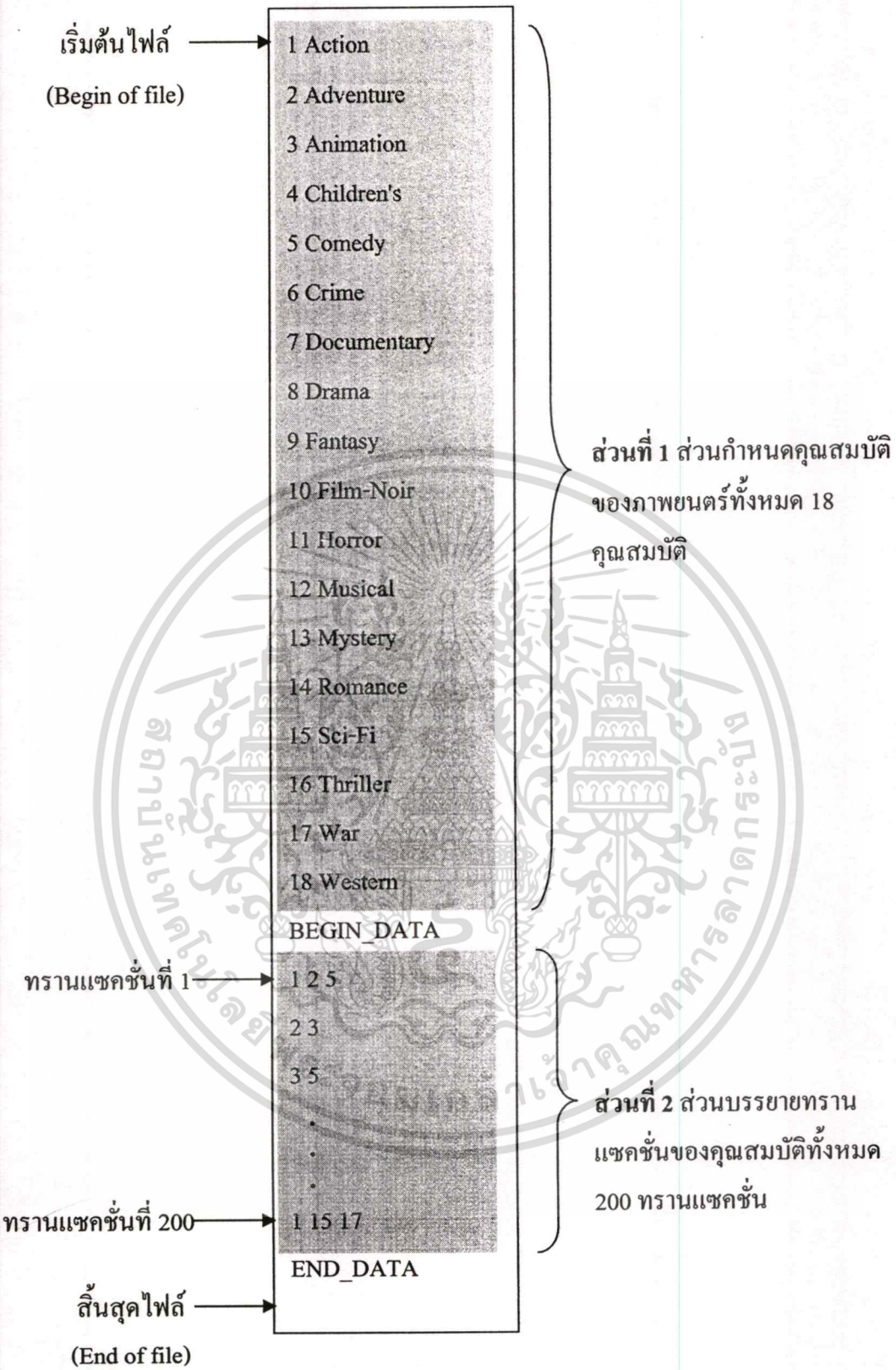


รูปที่ 4.4 แสดงการแบ่งดาต้าเซ็ตสำหรับทำการทดลอง

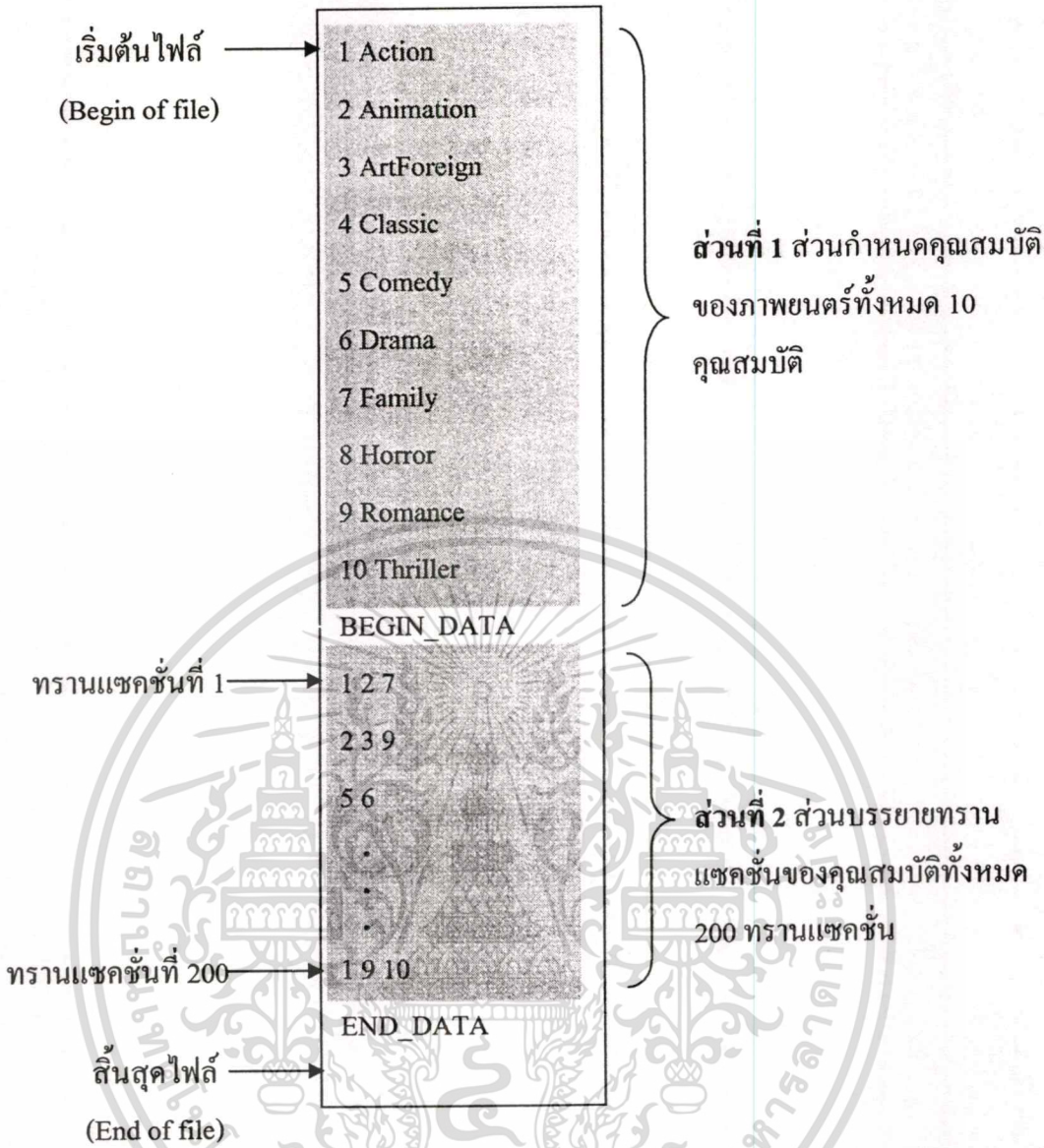
จากรูปที่ 4.4 แสดงถึงการสุ่มแบ่งแยกระหว่างเซ็ตของแบบจำลองสำหรับการสอนและเซ็ตของแบบจำลองสำหรับการทดสอบ โดยที่ในแต่ละชุดของการแบ่งเซ็ตของแบบจำลองสำหรับการสอนและเซ็ตของแบบจำลองสำหรับการทดสอบจะไม่มีสมาชิกร่วมกันเลย (Disjoint sets) แต่สมาชิกในแต่ละชุดของแบบจำลองตั้งแต่ชุดที่ 1 ถึง 250 สามารถมีสมาชิกร่วมกันได้

4.2.1.2 กระบวนการเตรียมข้อมูลคุณสมบัติ

รูปแบบข้อมูลคุณสมบัติที่ใช้ในการทดลอง จะแสดงอยู่ในรูปแบบทรานแซคชั่นของคุณสมบัติที่เก็บอยู่ในรูปของไฟล์ข้อความ ที่ใช้บรรยายถึงส่วนประกอบของทรานแซคชั่นของคุณสมบัติที่ประกอบไปด้วย 2 ส่วน คือ ส่วนกำหนดคุณสมบัติ และส่วนบรรยายทรานแซคชั่นของคุณสมบัติ โดยในการบรรยายทรานแซคชั่นของคุณสมบัติจะเริ่มต้นหลังจากคำว่า BEGIN_DATA เป็นต้นไปจนกระทั่งถึงคำว่า END_DATA ถือเป็นอันสิ้นสุดไฟล์ ดังแสดงในรูปที่ 4.5 และ 4.6 ตามลำดับ



รูปที่ 4.5 รูปแบบไฟล์ข้อความที่เก็บทรานแซคชั่นของคุณสมบัติของดาต้าเซ็ต MovieLens



รูปที่ 4.6 รูปแบบไฟล์ข้อความที่เก็บทรานแซคชั่นของคุณสมบัติของคาต้าเซ็ต EachMovie

จากรูปที่ 4.5 และ 4.6 สามารถอธิบายได้ดังนี้ ในกรณีของคาต้าเซ็ต MovieLens สำหรับภาพยนตร์ทั้ง 200 เรื่อง ในส่วนที่ 1 ประกอบด้วยคุณสมบัติ 18 คุณสมบัติตามลำดับ คือ Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War และ Western ในส่วนที่ 2 ประกอบด้วยทรานแซคชั่นของคุณสมบัติทั้งหมด 200 ทรานแซคชั่น ในกรณีของคาต้าเซ็ต EachMovie ภาพยนตร์ทั้ง 200 เรื่องประกอบด้วยคุณสมบัติ 10 คุณสมบัติตามลำดับได้ดังนี้ คือ Action, Animation, ArtForeign, Classic, Comedy, Drama, Family, Horror, Romance และ Thriller ในส่วนที่ 2 ประกอบด้วยทรานแซคชั่นของคุณสมบัติทั้งหมด 200 ทรานแซคชั่น

ในการบรรยายของส่วนที่ 2 สำหรับทั้งสองคาต้าเซ็ตจะมีความหมายเหมือนกัน คือ นำตัวเลขที่บ่งชี้ถึงแต่ละคุณสมบัติในส่วนที่ 1 มาแสดงให้อยู่ในรูปของทรานแซคชั่น ดังตัวอย่างในรูปที่ 4.6 ทรานแซคชั่นลำดับที่ 1 คือ 1 2 7 หมายความว่า ทรานแซคชั่นลำดับที่ 1 ประกอบด้วยคุณสมบัติ Action(1), Animation(2) และ Family(7) ทำนองเดียวกันกับทรานแซคชั่นลำดับที่ 2 ที่ประกอบด้วย คุณสมบัติ Animation(2), ArtForeign(3) และ Romance(9) เป็นต้น

4.2.2 กระบวนการค้นหากฎความสัมพันธ์

กระบวนการนี้อยู่ในส่วนออฟไลน์ ส่วนใหญ่ยึดขั้นตอนวิธีการตามหัวข้อที่ 3.2.2 และ 3.2.3 ซึ่งประกอบด้วย 3 ขั้นตอนหลักดังนี้

1. ขั้นตอนการค้นหาไอเท็มเซ็ตปรากฏบ่อยขนาด 2 ไอเท็ม ประกอบด้วยกระบวนการย่อยดังนี้

- อ่านข้อมูลไฟล์ข้อความทรานแซคชั่นของคุณสมบัติที่ได้มาจากขั้นตอนการเตรียมข้อมูล
- ค้นหาและจัดเก็บไอเท็มเซ็ตที่มีจำนวนสมาชิกเท่ากับหนึ่งพร้อมกับคำนวณค่าสนับสนุน และเซ็ตของหมายเลขทรานแซคชั่นที่สัมพันธ์กับไอเท็มเซ็ตนั้น
- ค้นหาไอเท็มเซ็ตปรากฏบ่อยขนาด 2 ไอเท็มพร้อมกับคำนวณค่าสนับสนุนและจัดเก็บไอเท็มเซ็ตปรากฏบ่อยขนาด 2 ไอเท็มที่หามาได้ลงในตารางแฮช (Hashtable)

2. ขั้นตอนการค้นหากฎความสัมพันธ์จากไอเท็มเซ็ตปรากฏบ่อยขนาด 2 ไอเท็ม จะนำไอเท็มเซ็ตปรากฏบ่อยขนาด 2 ไอเท็มทั้งหมดพร้อมกับค่าสนับสนุนของไอเท็มเซ็ตเหล่านั้นมาค้นหากฎความสัมพันธ์ตามหลักการในหัวข้อที่ 3.2.2 ที่ประกอบด้วย

- จำนวนกฎความสัมพันธ์ทั้งหมดของไอเท็มเซ็ตปรากฏบ่อย
- ไอเท็มเซ็ตของกฎความสัมพันธ์ส่วนที่เป็นผลในแต่ละกฎความสัมพันธ์
- ไอเท็มเซ็ตของกฎความสัมพันธ์ส่วนที่เป็นเหตุในแต่ละกฎความสัมพันธ์
- ค่าความเชื่อมั่นในแต่ละกฎความสัมพันธ์

3. ขั้นตอนการสร้างเมตริกซ์ความสัมพันธ์ระหว่างคุณสมบัติ เป็นการนำจำนวนกฎความสัมพันธ์ทั้งหมดพร้อมกับค่าความเชื่อมั่นในแต่ละกฎความสัมพันธ์มาเก็บไว้ในรูปของอะเรย์ 2 มิติขนาด 10x10 และ 18x18 สำหรับคาต้าเซ็ต EachMovie และ MovieLens ตามลำดับ

4.2.3 กระบวนการค้นหาชิ้นข้อมูลที่ใกล้เคียง

กระบวนการนี้อยู่ในส่วนออนไลน์ ขั้นตอนของกระบวนการค้นหาชิ้นข้อมูลที่ใกล้เคียงมีดังต่อไปนี้

1. กำหนดผู้ใช้เป้าหมายและชิ้นข้อมูลเป้าหมายที่ต้องการทำนายในแต่ละครั้ง โดยดึงข้อมูลในแบบจำลองสำหรับการทดสอบมาทีละข้อมูล

2. กำหนดขนาดชิ้นข้อมูลที่ใกล้เคียงตามความต้องการ (เช่น 10, 20 หรือ 30 เป็นต้น)

3. วนลูปทีละชิ้นข้อมูลที่ผู้ใช้เป้าหมายนั้นเคยให้เรตติ้งไว้

- คำนวณหาค่าความคล้ายคลึงเรตติ้งระหว่างชิ้นข้อมูลเป้าหมายกับชิ้นข้อมูลที่นำมาเปรียบเทียบ ด้วยวิธี Cosine, Pearson correlation หรือ Adjusted cosine (ดังที่กล่าวไว้ในหัวข้อที่ 2.1.2.1)
- คำนวณหาค่าความคล้ายคลึงคุณสมบัติระหว่างชิ้นข้อมูลเป้าหมายกับชิ้นข้อมูลที่นำมาเปรียบเทียบด้วยวิธีการที่นำเสนอ ตามอัลกอริทึมในรูปที่ 3.7 (ดังที่กล่าวไว้ในหัวข้อที่ 3.3.1.2)
- คำนวณหาค่าพารามิเตอร์ถ่วงน้ำหนักที่เหมาะสมให้กับค่าความคล้ายคลึงทั้งสอง ตามสมการที่ 3.5 (ดังที่กล่าวไว้ในหัวข้อที่ 3.3.2)
- คำนวณหาค่าความคล้ายคลึงรวมด้วยการรวมเชิงเส้น (ดังที่กล่าวไว้ในหัวข้อที่ 3.3.2)

4. เก็บกลุ่มชิ้นข้อมูลที่ใกล้เคียงกับชิ้นข้อมูลเป้าหมายตามลำดับจากมากไปน้อยไว้ในลิ้งลิส (Linklist) ตามขนาดที่กำหนดไว้ในขั้นตอนที่ 2 (เช่น 10, 20 หรือ 30 เป็นต้น)

4.2.4 กระบวนการทำนาย

กระบวนการนี้อยู่ในส่วนออนไลน์ เป็นการนำกลุ่มชิ้นข้อมูลที่ใกล้เคียงที่ได้จากขั้นตอนที่แล้วมาคำนวณด้วยเทคนิค Weighted sum ตามสมการที่ 3.6 (ดังที่กล่าวไว้ในหัวข้อที่ 3.3.2) เพื่อทำนายหาค่าความพึงพอใจและเก็บผลลัพธ์จากการทำนายไว้ในรูปไฟล์ข้อความ

4.2.5 วนลูป

ขั้นตอนนี้จะวนลูปทำซ้ำจนกระทั่งทำการทำนายครบทุกข้อมูลในแบบจำลองสำหรับการทดสอบ ตัวอย่างเช่น ดาต้าเซ็ต MovieLens ที่นำมาใช้มีทั้งหมด 3,789 เรตติ้งแบ่งออกเป็นแบบจำลองสำหรับการสอน 90% เท่ากับ 3,410 เรตติ้ง และแบบจำลองสำหรับการทดสอบ 10% เท่ากับ 379 เรตติ้ง ดังนั้นในขั้นตอนนี้จะทำการวนลูปเพื่อทำนายทีละเรตติ้งในแบบจำลองสำหรับการทดสอบจนกระทั่งครบทั้ง 379 เรตติ้ง ถือเป็นสิ้นสุดการวนลูป

4.2.6 การประเมินผลการวิจัย

เป็นการวิเคราะห์ข้อมูลจากการทำนายในเชิงสถิติหรือกล่าวอีกนัยหนึ่งได้ว่า เป็นการหาข้อมูลที่เป็นตัวแทนของกลุ่มข้อมูลที่ได้จากการทำนาย เนื่องจากค่าความผิดพลาดจากการทำนายซึ่งล้วนแล้วแต่เป็นค่าที่คาดเดาไม่ได้ทั้งนั้นด้วยคุณสมบัติของค่าความผิดพลาดจากการทำนายที่เปลี่ยนแปลงตลอดเวลาขึ้นอยู่กับข้อมูลที่นำมาใช้ในการทดลอง

ดังนั้นปริมาณทางสถิติหลักๆ ที่จำเป็นต้องหาค่าสำหรับวิเคราะห์เพื่อทดสอบวิธีการที่นำเสนอเปรียบเทียบกับวิธีการเดิมที่ใช้กับอัลกอริทึมไอเท็มเบส CF ได้แก่ ค่าเฉลี่ย (Mean) และ

ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation หรือ SD) หลังจากนั้นวัดประสิทธิภาพด้วยการคำนวณหาเปอร์เซ็นต์ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลง ดังแสดงในตารางที่ 4.2

ตารางที่ 4.2 สรุปวิธีการประเมินผลการวิจัย

ดาต้าเซ็ต	เปรียบเทียบเชิงสถิติ	
	ไอเท็มเบส CF	ไอเท็มเบส CF + วิธีการที่นำเสนอ
MovieLens สุ่มแบ่ง 3,789 เรตติ้งด้วยสัดส่วน 90:10% ทั้งหมด 250 ชุด	ค่าเฉลี่ย	ค่าเฉลี่ย
	ส่วนเบี่ยงเบนมาตรฐาน	ส่วนเบี่ยงเบนมาตรฐาน
		เปอร์เซ็นต์ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลง
EachMovie สุ่มแบ่ง 1,659 เรตติ้งด้วยสัดส่วน 90:10% ทั้งหมด 250 ชุด	ค่าเฉลี่ย	ค่าเฉลี่ย
	ส่วนเบี่ยงเบนมาตรฐาน	ส่วนเบี่ยงเบนมาตรฐาน
		เปอร์เซ็นต์ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลง

จากตารางที่ 4.2 สามารถอธิบายวิธีวิเคราะห์เชิงสถิติได้ดังต่อไปนี้

ค่าเฉลี่ย เป็นค่าความผิดพลาดสมบูรณ์เฉลี่ยจากการทดลองทั้งหมดที่ได้อธิบายไว้ในหัวข้อ 2.1.3 (สมการที่ 2.5) หรือกล่าวอีกนัยหนึ่งได้ว่า ในการทดลองแต่ละครั้งค่าความผิดพลาดสมบูรณ์จะมีค่าที่เป็นไปได้หลายค่าแตกต่างกัน ในทางสถิติจะถือว่าตัวแทนที่ดีที่สุดของค่าเหล่านั้น คือ ค่าเฉลี่ย ดังแสดงในสมการที่ 4.2

$$MAE = \frac{\sum_{i=1}^{N} X_i}{N} \tag{4.2}$$

จากสมการที่ 4.2 กำหนดให้ X_i แทนค่าสมบูรณ์ของผลต่างระหว่างค่าเรตติ้งจริงที่ผู้ใช้เคยให้ไว้และค่าเรตติ้งจากการทำนายที่ได้จากการทดลองจำนวน N ครั้งและ MAE เป็นค่าความผิดพลาดสมบูรณ์เฉลี่ย

<u>ส่วนเบี่ยงเบนมาตรฐาน</u> ใช้วัดการกระจายของทุกค่าความผิดพลาดสมบรูณ์ว่ามีการกระจายจากค่าเฉลี่ยมากน้อยเพียงใด เป็นค่าที่บอกให้ทราบว่าค่าความผิดพลาดสมบูรณ์ที่ได้จากการทดลอง มีค่าแตกต่างกันมากน้อยเพียงใด ตัวอย่างเช่น ถ้าส่วนเบี่ยงเบนมาตรฐานที่คำนวณได้มีค่ามาก แสดงว่าค่าความผิดพลาดสมบูรณ์ที่ได้จากการทดลอง มีค่าแตกต่างกันมากหรือมีการกระจายมาก ถ้าส่วนเบี่ยงเบนมาตรฐานที่คำนวณได้มีค่าน้อย แสดงว่าค่าความผิดพลาดสมบูรณ์ที่ได้จากการทดลอง มีค่าใกล้เคียงกันหรือมีการกระจายน้อย ดังแสดงในสมการที่ 4.5 หรือ 4.6

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(X_i - MAE)^2}{N}} \tag{4.5}$$

หรือ

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} X_i^2}{N} - MAE^2} \tag{4.6}$$

จากสมการที่ 4.5 และ 4.6 สัญลักษณ์ σ แทนส่วนเบี่ยงเบนมาตรฐานของค่าความผิดพลาดสมบูรณ์ ดังแสดงในสมการที่ 4.5 หรือ 4.6

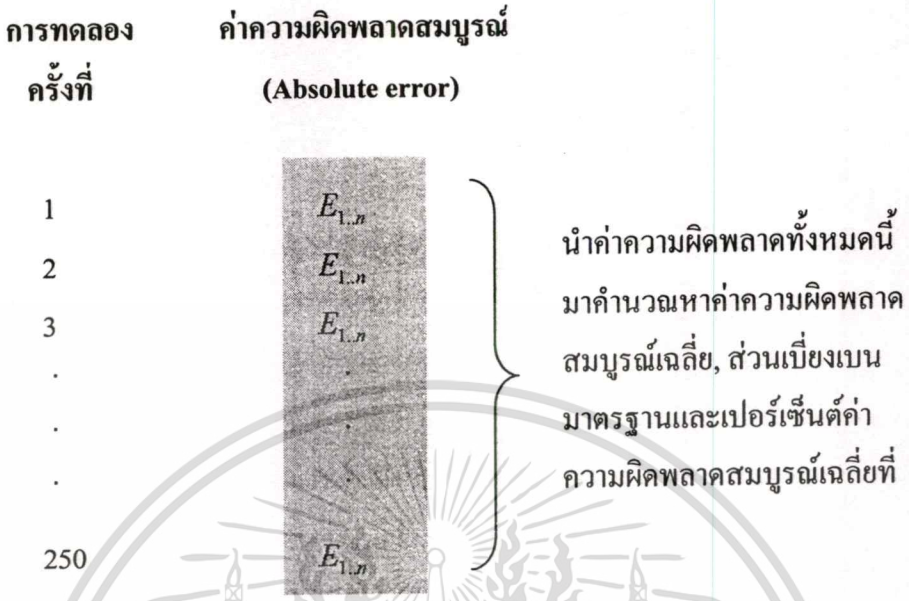
<u>เปอร์เซ็นต์ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลง</u> ใช้วิธีเปรียบเทียบผลต่างค่าความผิดพลาดสมบูรณ์เฉลี่ยจากการทดลองด้วยวิธีการเดิมและวิธีการที่นำเสนอและคำนวณออกมาเป็น % ของค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลงจากวิธีการเดิม โดยนิยามได้ตามสมการที่ 4.7

$$PercentageOfRMAE = \left[\frac{(MAE_{old} - MAE_{new})}{MAE_{old}} * 100\right] \tag{4.7}$$

โดยที่

$PercentageOfRMAE$	คือ	เปอร์เซ็นต์ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลง
MAE_{old}	คือ	ค่าความผิดพลาดสมบูรณ์เฉลี่ยของวิธีการเดิม
MAE_{new}	คือ	ค่าความผิดพลาดสมบรูณ์เฉลี่ยของวิธีการที่นำเสนอ

เพื่อให้เกิดความเข้าใจมากยิ่งขึ้นจึงขออธิบายวิธีการประเมินเพิ่มเติมได้ดังแสดงในรูปที่ 4.7



รูปที่ 4.7 แสดงวิธีการนำผลการทดลองมาประเมิน

จากรูปที่ 4.7 อธิบายได้ว่าในการทดลองแต่ละครั้งตั้งแต่ครั้งที่ 1 ถึงครั้งที่ 250 จะได้ค่าความผิดพลาดสมบูรณ์จากการทดลองแทนด้วย $E_{1..n}$ โดยที่ n คือ จำนวนรเรตติ้งทั้งหมดในชุดข้อมูลสำหรับการทดสอบ หลังจากนั้นนำค่าความผิดพลาดสมบูรณ์ทั้งหมดไปประเมินผลการทดลองด้วยวิธีการคำนวณหาค่าความผิดพลาดสมบูรณ์เฉลี่ย, ส่วนเบี่ยงเบนมาตรฐานและเปอร์เซ็นต์ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลง

สุดท้ายนำผลการประเมินมาสรุปผลการทดลองตลอดจนอธิบายถึงข้อเสนอแนะเพื่อการทำวิจัยต่อในอนาคต

4.3 ผลการทดลอง

งานวิจัยนี้ได้ทดลองวิธีการที่นำเสนอเปรียบเทียบกับวิธีการเดิมที่ใช้กับอัลกอริทึมไอเท็มเบส CF ได้แก่ Adjusted cosine, Cosine และ Pearson correlation โดยใช้อัตราส่วนระหว่างข้อมูลสำหรับการสอนและข้อมูลสำหรับการทดสอบเท่ากับ 90:10% ทั้งหมด 500 ชุดสำหรับดาต้าเซ็ต MovieLens 250 ชุดและ EachMovie 250 ชุด ตามขั้นตอนการทดลองดังที่กล่าวไว้ในหัวข้อที่ 4.2

เพื่อไม่ให้เกิดความสับสน ขอนิยามวิธีการทดลองดังต่อไปนี้

1. วิธี Pure Adjusted cosine หมายถึง การค้นหาชิ้นข้อมูลที่ใกล้เคียงด้วยวิธี Adjusted Cosine (วิธีการเดิม)
2. วิธี Pure Cosine หมายถึง การค้นหาชิ้นข้อมูลที่ใกล้เคียงด้วยวิธี Cosine (วิธีการเดิม)
3. วิธี Pure Pearson หมายถึง การค้นหาชิ้นข้อมูลที่ใกล้เคียงด้วยวิธี Pearson correlation (วิธีการเดิม)
4. วิธี Combined Adjusted cosine หมายถึง การค้นหาชิ้นข้อมูลที่ใกล้เคียงด้วยวิธี Adjusted cosine ร่วมกับวิธีการที่นำเสนอ (Adjusted cosine + วิธีการที่นำเสนอ)
5. วิธี Combined Cosine หมายถึง การค้นหาชิ้นข้อมูลที่ใกล้เคียงด้วยวิธี Cosine ร่วมกับวิธีการที่นำเสนอ (Cosine + วิธีการที่นำเสนอ)
6. วิธี Combined Pearson หมายถึง การค้นหาชิ้นข้อมูลที่ใกล้เคียงด้วยวิธี Pearson correlation ร่วมกับวิธีการที่นำเสนอ (Pearson correlation + วิธีการที่นำเสนอ)

4.3.1 ผลการเปรียบเทียบค่าความผิดพลาดสมบูรณ์เฉลี่ยและส่วนเบี่ยงเบนมาตรฐาน ระหว่างวิธี Pure Adjusted cosine กับ Combined Adjusted cosine

เป็นการเปรียบเทียบวิธี Pure Adjusted cosine กับ Combined Adjusted cosine โดยปรับเปลี่ยนขนาดของชิ้นข้อมูลที่ใกล้เคียงตั้งแต่ 10, 20, 30, 40 และ 50 ทำให้ได้ผลการทดลองออกมา ดังแสดงในตารางที่ 4.3 และ 4.4 ด้วยดาต้าเซ็ต MovieLens และ EachMovie ตามลำดับ

ตารางที่ 4.3 ผลการเปรียบเทียบวิธี Pure Adjusted cosine กับ Combined Adjusted cosine ด้วยดาต้าเซ็ต MovieLens

ดาต้าเซ็ต MovieLens	ค่าความผิดพลาดสมบูรณ์เฉลี่ย (MAE)	ส่วนเบี่ยงเบนมาตราฐาน (SD)
วิธี Pure Adjusted cosine		
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 10	0.866	0.661
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 20	0.847	0.654

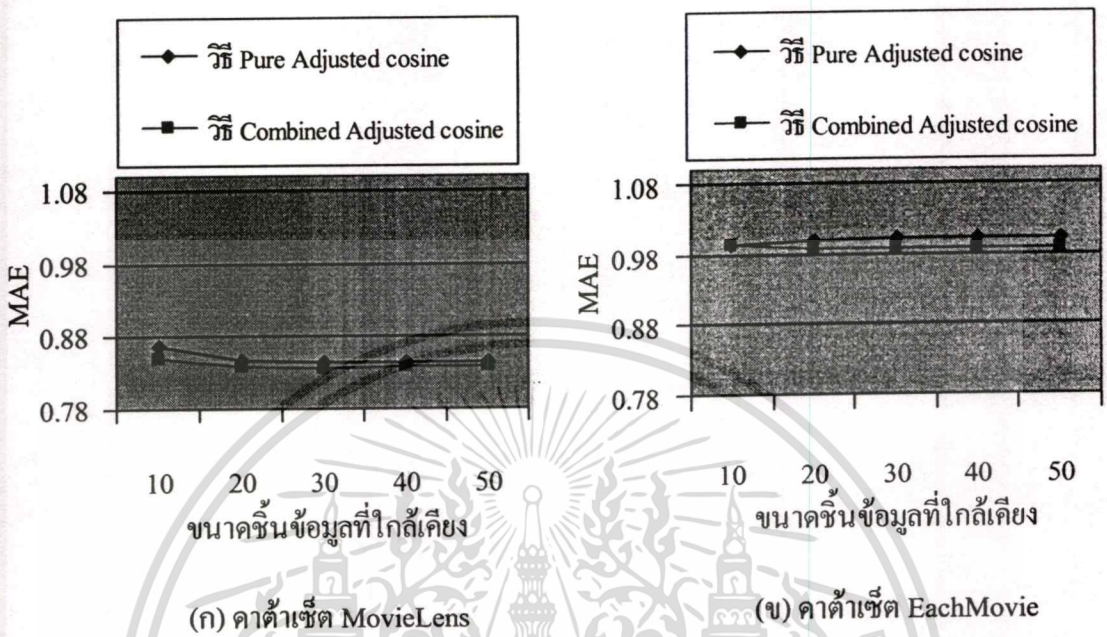
ตารางที่ 4.3 (ต่อ)

ดาต้าเซ็ต MovieLens	ค่าความผิดพลาดสมบูรณ์เฉลี่ย (MAE)	ส่วนเบี่ยงเบนมาตรฐาน (SD)
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 30	0.845	0.652
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 40	0.844	0.653
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 50	0.843	0.652
วิธี Combined Adjusted cosine		
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 10	0.850	0.667
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 20	0.840	0.652
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 30	0.837	0.644
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 40	0.838	0.639
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 50	0.837	0.640

ตารางที่ 4.4 ผลการเปรียบเทียบวิธี Pure Adjusted cosine กับ Combined Adjusted cosine ด้วยดาต้าเซ็ต EachMovie

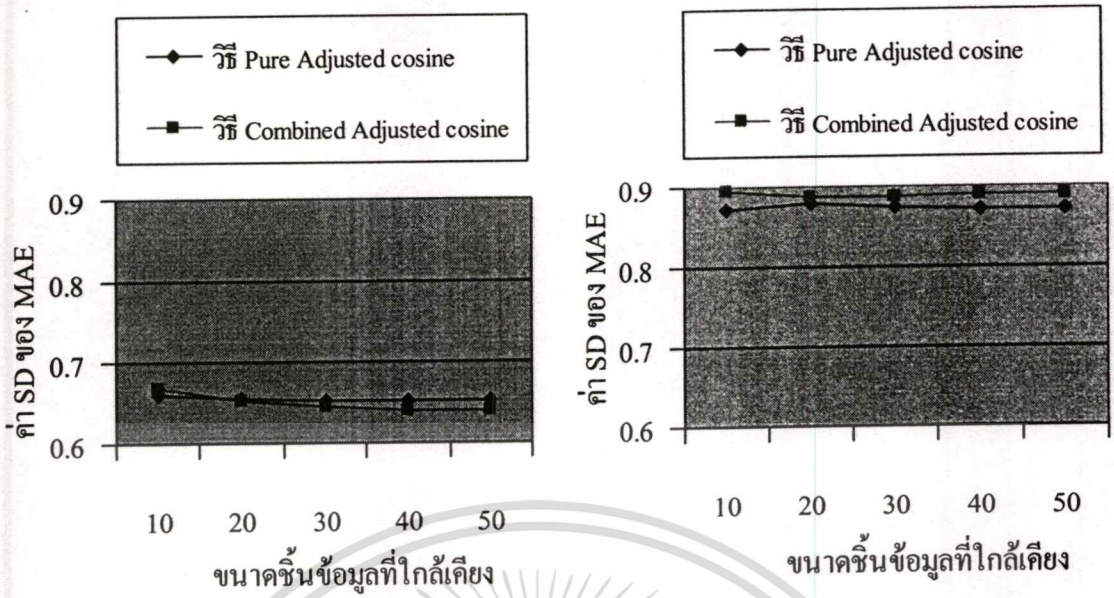
ดาต้าเซ็ต EachMovie	ค่าความผิดพลาดสมบูรณ์เฉลี่ย (MAE)	ส่วนเบี่ยงเบนมาตรฐาน (SD)
วิธี Pure Adjusted cosine		
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 10	0.994	0.870
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 20	1.000	0.878
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 30	1.002	0.874
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 40	1.002	0.871
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 50	1.002	0.871
วิธี Combined Adjusted cosine		
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 10	0.993	0.895
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 20	0.986	0.887
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 30	0.986	0.888
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 40	0.987	0.888
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 50	0.987	0.888

จากข้อมูลค่า MAE และ SD ในตารางที่ 4.3 และ 4.4 สามารถแสดงในรูปแบบกราฟได้กราฟดังรูปที่ 4.8 และ 4.9 ตามลำดับ



รูปที่ 4.8 กราฟแสดงค่า MAE ระหว่างวิธี Pure Adjusted cosine กับ Combined Adjusted cosine เมื่อขนาดชิ้นข้อมูลที่ใกล้เคียงเปลี่ยนแปลงไป

จากกราฟแสดงค่า MAE รูปที่ 4.8 พบว่าทั้งดาต้าเซ็ต MovieLens และ EachMovie วิธี Combined Adjusted cosine ให้ค่า MAE ออกมาน้อยกว่าวิธี Pure Adjusted cosine อยู่เล็กน้อยไม่ว่าขนาดชิ้นข้อมูลที่ใกล้เคียงจะเพิ่มขึ้นจาก 10, 20, 30, 40 ถึง 50 ยิ่งไปกว่านั้นเห็นได้ว่าเมื่อเพิ่มขนาดของชิ้นข้อมูลที่ใกล้เคียงตั้งแต่ 30 ชิ้นขึ้นไปค่า MAE จะค่อนข้างคงที่ สาเหตุที่กราฟค่อนข้างคงที่เพราะจำนวนชิ้นข้อมูลที่ใกล้เคียงมีจำนวนจำกัดหรือกล่าวอีกนัยหนึ่งได้ว่าทั้งดาต้าเซ็ต MovieLens และ EachMovie มีผู้ใช้ส่วนใหญ่ให้เรตติ้งต่อชิ้นข้อมูลสูงสุดไม่เกิน 30 ชิ้น นอกจากนั้นจากกราฟยังพบว่าดาต้าเซ็ต EachMovie มีค่า MAE สูงกว่าดาต้าเซ็ต MovieLens อย่างเห็นได้ชัด ทั้งนี้เพราะจากการวิเคราะห์ข้อมูลเรตติ้งในหัวข้อที่ 4.2.1.1 (ตารางที่ 4.1) พบว่าดาต้าเซ็ต MovieLens มีความหนาแน่นของข้อมูลการให้เรตติ้งมากกว่าดาต้าเซ็ต EahcMovie ค่อนข้างมาก



(ก) ดาต้าเซ็ต MovieLens (ข) ดาต้าเซ็ต EachMovie

รูปที่ 4.9 กราฟแสดงส่วนเบี่ยงเบนมาตรฐานของค่า MAE ระหว่างวิธี Pure Adjusted cosine กับ Combined Adjusted cosine เมื่อขนาดชิ้นข้อมูลที่ใกล้เคียงเปลี่ยนแปลงไป

จากกราฟแสดงส่วนเบี่ยงเบนมาตรฐานของค่า MAE รูปที่ 4.9 พบว่าทั้งดาต้าเซ็ต MovieLens และ EachMovie เมื่อขนาดชิ้นข้อมูลที่ใกล้เคียงเพิ่มขึ้นจาก 10, 20, 30, 40 ถึง 50 พบว่าวิธี Pure Adjusted cosine และ Combined Adjusted cosine มีค่าความผิดพลาดสมบูรณ์ที่ได้จากการทำนายในแต่ละครั้งแตกต่างไปจากค่า MAE อยู่ค่อนข้างใกล้เคียงกัน และยังพบว่าดาต้าเซ็ต EachMovie มีการกระจัดกระจายของค่าความผิดพลาดสมบูรณ์มากกว่าดาต้าเซ็ต MovieLens อย่างเห็นได้ชัด

4.3.2 ผลการเปรียบเทียบค่าความผิดพลาดสมบูรณ์เฉลี่ยและส่วนเบี่ยงเบนมาตรฐานระหว่างวิธี Pure Cosine กับ Combined Cosine

เป็นการเปรียบเทียบวิธี Pure Cosine กับ Combined Cosine โดยปรับเปลี่ยนขนาดของชิ้นข้อมูลที่ใกล้เคียงตั้งแต่ 10, 20, 30, 40 และ 50 ได้ผลการทดลองออกมา ดังแสดงในตารางที่ 4.5 และ 4.6 ด้วยดาต้าเซ็ต MovieLens และ EachMovie ตามลำดับ

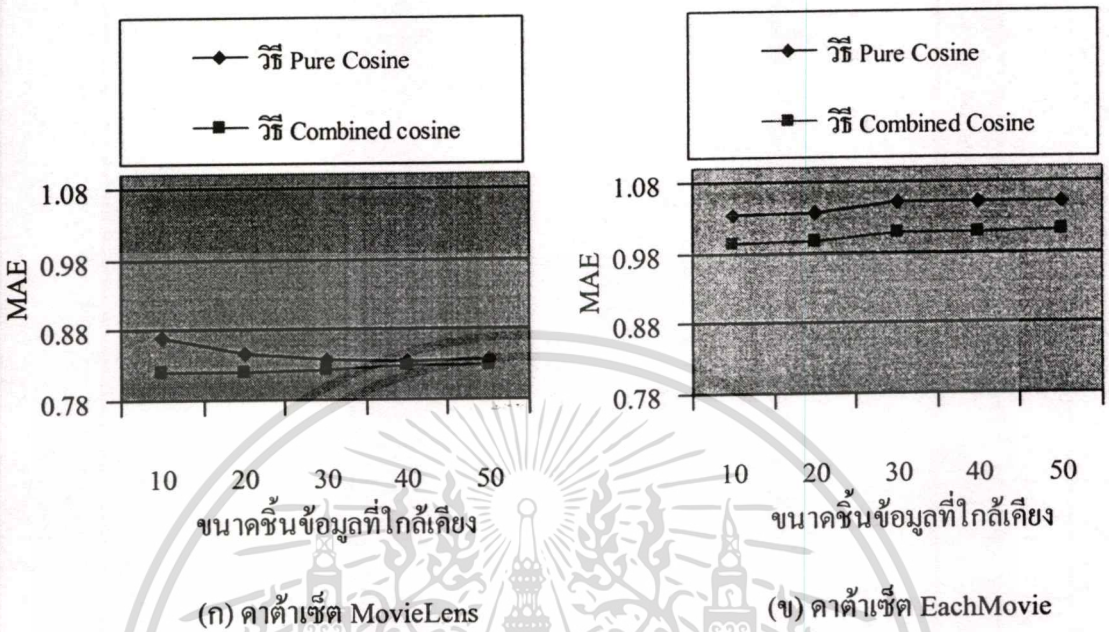
ตารางที่ 4.5 ผลการเปรียบเทียบวิธี Pure Cosine กับ Combined Cosine ด้วยดาต้าเซ็ต MovieLens

ดาต้าเซ็ต MovieLens	ค่าความผิดพลาดสมบูรณ์เฉลี่ย (MAE)	ส่วนเบี่ยงเบนมาตราฐาน (SD)
วิธี Pure Cosine		
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 10	0.869	0.671
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 20	0.846	0.654
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 30	0.837	0.646
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 40	0.835	0.643
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 50	0.837	0.643
วิธี Combined Cosine		
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 10	0.820	0.712
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 20	0.819	0.682
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 30	0.823	0.665
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 40	0.827	0.656
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 50	0.828	0.648

ตารางที่ 4.6 ผลการเปรียบเทียบวิธี Pure Cosine กับ Combined Cosine ด้วยดาต้าเซ็ต EachMovie

ดาต้าเซ็ต EachMovie	ค่าความผิดพลาดสมบูรณ์เฉลี่ย (MAE)	ส่วนเบี่ยงเบนมาตราฐาน (SD)
วิธี Pure Cosine		
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 10	1.034	0.942
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 20	1.037	0.950
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 30	1.050	0.942
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 40	1.051	0.943
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 50	1.051	0.943
วิธี Combined Cosine		
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 10	0.993	0.919
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 20	0.995	0.905
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 30	1.007	0.894
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 40	1.006	0.894
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 50	1.009	0.891

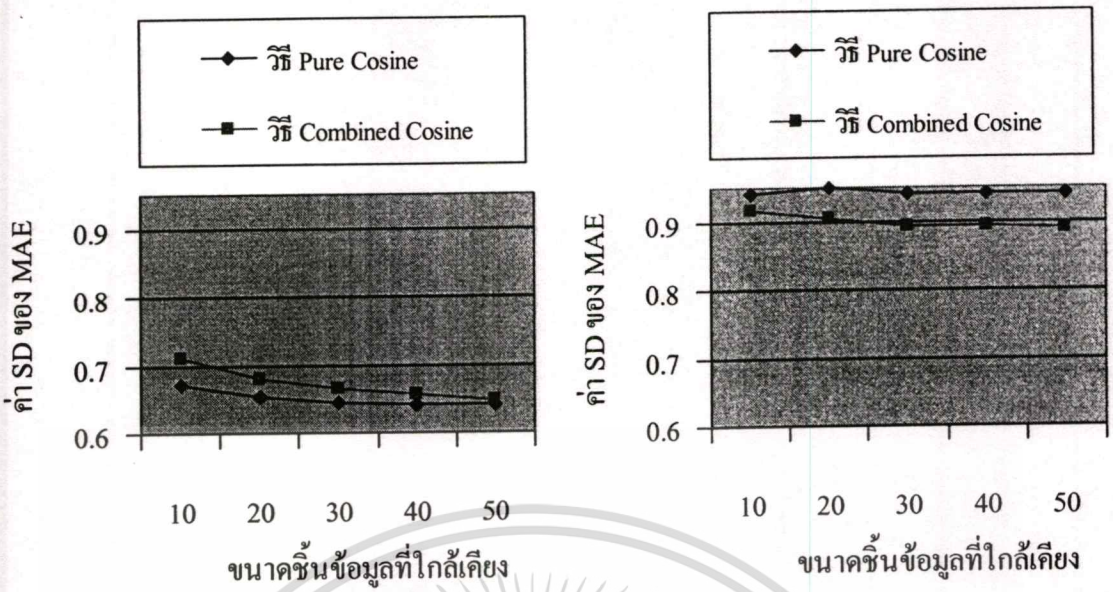
จากข้อมูลค่า MAE และ SD ในตารางที่ 4.5 และ 4.6 สามารถแสดงในรูปแบบกราฟได้กราฟดังรูปที่ 4.10 และ 4.11 ตามลำดับ



(ก) ดาต้าเซ็ต MovieLens (ข) ดาต้าเซ็ต EachMovie

รูปที่ 4.10 กราฟแสดงค่า MAE ระหว่างวิธี Pure Cosine กับ Combined Cosine เมื่อขนาดชิ้นข้อมูลที่ใกล้เคียงเปลี่ยนแปลงไป

จากกราฟแสดงค่า MAE รูปที่ 4.10 พบว่าทั้งดาต้าเซ็ต MovieLens และ EachMovie วิธี Combined Cosine ให้ค่า MAE ออกมาน้อยกว่าวิธี Pure Cosine อย่างเห็นได้ชัด โดยเฉพาะอย่างยิ่งเมื่อขนาดชิ้นข้อมูลที่ใกล้เคียงเพิ่มขึ้นจาก 10 จนถึง 30 หลังจากนั้นเมื่อขนาดของชิ้นข้อมูลที่ใกล้เคียงเพิ่มขึ้นตั้งแต่ 30 ขึ้นไปเรื่อยๆ ค่า MAE จะค่อนข้างคงที่ และยังพบว่าดาต้าเซ็ต EachMovie ยังคงมีค่า MAE สูงกว่าดาต้าเซ็ต MovieLens อย่างเห็นได้ชัด



(ก) ดาต้าเซ็ต MovieLens (ข) ดาต้าเซ็ต EachMovie

รูปที่ 4.11 กราฟแสดงส่วนเบี่ยงเบนมาตรฐานของค่า MAE ระหว่างวิธี Pure Cosine กับ Combined Cosine เมื่อขนาดชิ้นข้อมูลที่ใกล้เคียงเปลี่ยนแปลงไป

จากกราฟแสดงส่วนเบี่ยงเบนมาตรฐานของค่า MAE รูปที่ 4.11 พบว่าทั้งดาต้าเซ็ต MovieLens และ EachMovie เมื่อขนาดชิ้นข้อมูลที่ใกล้เคียงเพิ่มขึ้นเรื่อยๆ พบว่าวิธี Pure Cosine และ Combined Cosine ยังคงให้ค่าความผิดพลาดสมบูรณ์จากการทำนายในแต่ละครั้งแตกต่างไปจากค่า MAE อยู่ค่อนข้างใกล้เคียงกัน ในส่วนของดาต้าเซ็ต EachMovie ยังพบว่ามีการกระจัดกระจายของค่าความผิดพลาดสมบูรณ์มากกว่าดาต้าเซ็ต MovieLens อย่างเห็นได้ชัด

4.3.3 ผลการเปรียบเทียบค่าความผิดพลาดสมบูรณ์เฉลี่ยและส่วนเบี่ยงเบนมาตรฐานระหว่างวิธี Pure Pearson กับ Combined Pearson

เป็นการเปรียบเทียบวิธี Pure Pearson กับ Combined Pearson โดยปรับเปลี่ยนขนาดของชิ้นข้อมูลที่ใกล้เคียงตั้งแต่ 10, 20, 30, 40 และ 50 ได้ผลการทดลองออกมา ดังแสดงในตารางที่ 4.7 และ 4.8 ด้วยดาต้าเซ็ต MovieLens และ EachMovie ตามลำดับ

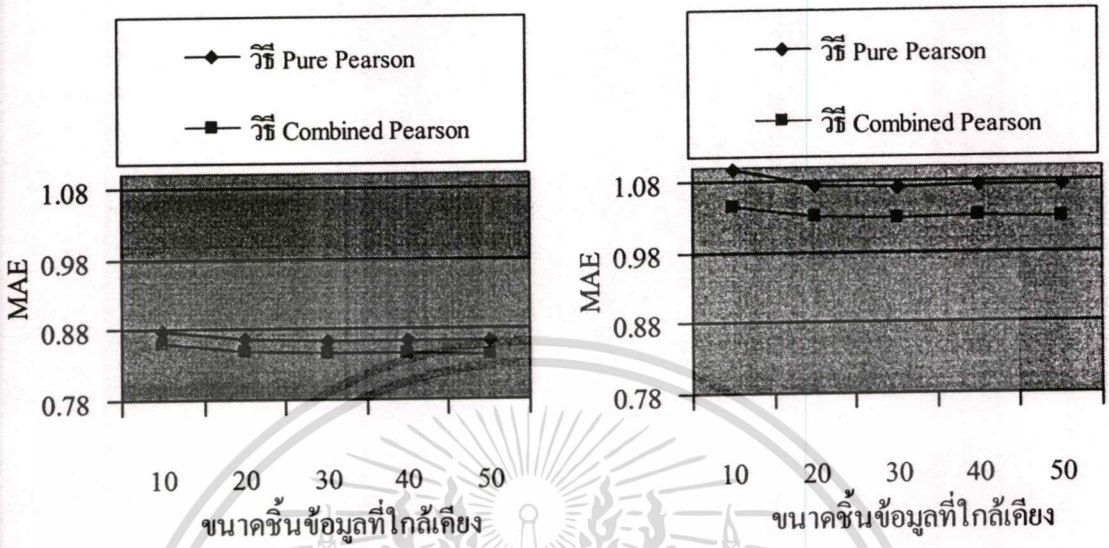
ตารางที่ 4.7 ผลการเปรียบเทียบวิธี Pure Pearson กับ Combined Pearson ด้วยดาต้าเซ็ต MovieLens

ดาต้าเซ็ต MovieLens	ค่าความผิดพลาดสมบูรณ์เฉลี่ย (MAE)	ส่วนเบี่ยงเบนมาตรฐาน (SD)
วิธี Pure Pearson		
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 10	0.877	0.671
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 20	0.867	0.665
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 30	0.865	0.665
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 40	0.864	0.664
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 50	0.864	0.664
วิธี Combined Pearson		
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 10	0.861	0.675
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 20	0.848	0.648
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 30	0.846	0.642
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 40	0.845	0.639
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 50	0.844	0.639

ตารางที่ 4.8 ผลการเปรียบเทียบวิธี Pure Pearson กับ Combined Pearson ด้วยดาต้าเซ็ต EachMovie

ดาต้าเซ็ต EachMovie	ค่าความผิดพลาดสมบูรณ์เฉลี่ย (MAE)	ส่วนเบี่ยงเบนมาตรฐาน (SD)
วิธี Pure Pearson		
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 10	1.096	0.914
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 20	1.074	0.929
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 30	1.072	0.926
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 40	1.073	0.926
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 50	1.073	0.926
วิธี Combined Pearson		
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 10	1.045	0.884
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 20	1.030	0.879
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 30	1.026	0.876
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 40	1.030	0.875
ขนาดชิ้นข้อมูลที่ใกล้เคียง = 50	1.028	0.875

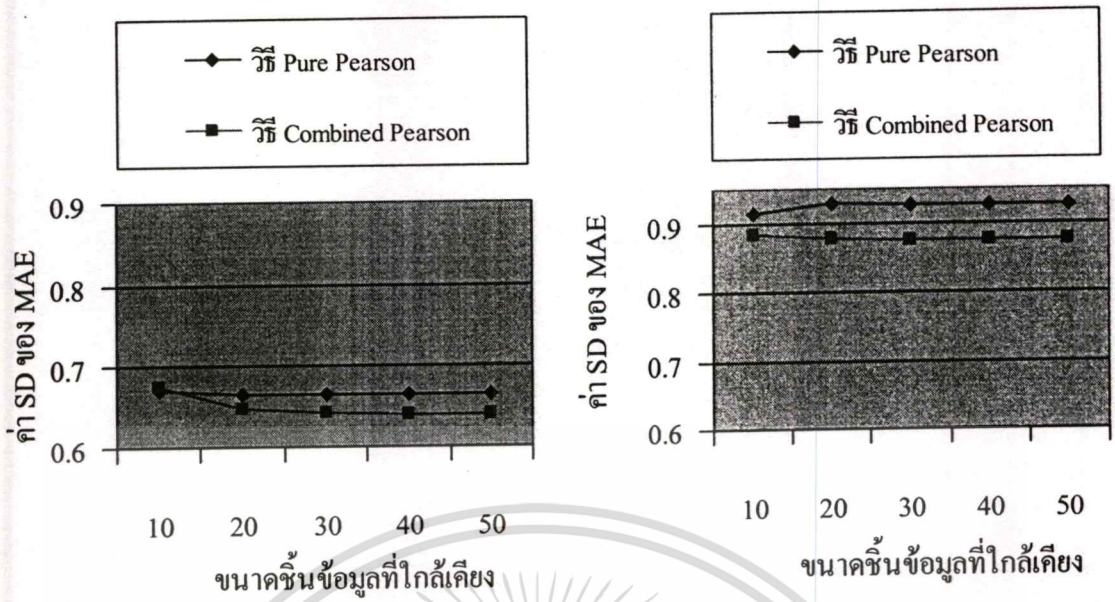
จากข้อมูลค่า MAE และ SD ในตารางที่ 4.7 และ 4.8 สามารถแสดงในรูปแบบกราฟได้กราฟดังรูปที่ 4.12 และ 4.13 ตามลำดับ



(ก) ดาต้าเซ็ต MovieLens (ข) ดาต้าเซ็ต EachMovie

รูปที่ 4.12 กราฟแสดงค่า MAE ระหว่างวิธี Pure Pearson กับ Combined Pearson เมื่อขนาดชิ้นข้อมูลที่ใกล้เคียงเปลี่ยนแปลงไป

จากกราฟแสดงค่า MAE รูปที่ 4.12 พบว่าทั้งดาต้าเซ็ต MovieLens และ EachMovie วิธี Combined Pearson ให้ค่า MAE ออกมาน้อยกว่าวิธี Pure Pearson อย่างคงที่ตลอดไม่ว่าขนาดชิ้นข้อมูลจะเพิ่มอย่างไรก็ตาม นอกจากนั้นยังพบว่าดาต้าเซ็ต EachMovie ยังคงมีค่า MAE สูงกว่าดาต้าเซ็ต MovieLens อย่างเห็นได้ชัด



รูปที่ 4.13 กราฟแสดงส่วนเบี่ยงเบนมาตรฐานของค่า MAE ระหว่างวิธี Pure Pearson กับ Combined Pearson เมื่อขนาดชิ้นข้อมูลที่ใกล้เคียงเปลี่ยนแปลงไป

จากกราฟแสดงส่วนเบี่ยงเบนมาตรฐานของค่า MAE รูปที่ 4.13 เมื่อขนาดชิ้นข้อมูลที่ใกล้เคียงเพิ่มขึ้นจาก 10 ถึง 50 พบว่าวิธี Combined Pearson มีการกระจัดกระจายของค่าความผิดพลาดสมบูรณ์จากการทำนายในแต่ละครั้งน้อยกว่าวิธี Pure Pearson แต่สำหรับดาต้าเซ็ต MovieLens ที่ขนาดชิ้นข้อมูลที่ใกล้เคียงเท่ากับ 10 พบว่า วิธี Pure Pearson มีการกระจัดกระจายของค่าความผิดพลาดสมบูรณ์ใกล้เคียงกับวิธี Combined Pearson นอกจากนั้นการทดลองนี้ยังพบว่าดาต้าเซ็ต EachMovie มีการกระจัดกระจายของค่าความผิดพลาดสมบูรณ์โดยรวมมากกว่าดาต้าเซ็ต MovieLens

4.3.4 ผลการวัดประสิทธิภาพของเปอร์เซ็นต์ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลงระหว่างวิธีการเดิมกับวิธีการที่นำเสนอ

เมื่อนำค่า MAE ที่ได้จากการทำนายด้วยวิธีการเดิม และวิธีการที่นำเสนอมาเปรียบเทียบกัน และคำนวณหาเปอร์เซ็นต์ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลงตามสมการที่ 4.7 จะได้ข้อมูลผลลัพธ์ดังแสดงในตารางที่ 4.9 และ 4.10

จากตารางที่ 4.9 และ 4.10 สามารถยกตัวอย่างการคำนวณหาเปอร์เซ็นต์ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลงได้ โดยดูที่ขนาดชิ้นข้อมูลที่ใกล้เคียงเท่ากับ 10 ด้วยวิธี Pure Adjusted cosine มีค่า MAE เท่ากับ 0.866 และวิธี Combined Adjusted cosine มีค่า MAE เท่ากับ 0.850

เพราะฉะนั้นจึงสามารถหา % ค่าเฉลี่ยความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลง เป็น [(0.866-0.850)/(0.866)]*100 = 1.85% และผลลัพธ์ที่เหลือก็สามารถคำนวณได้ในทำนองเดียวกัน

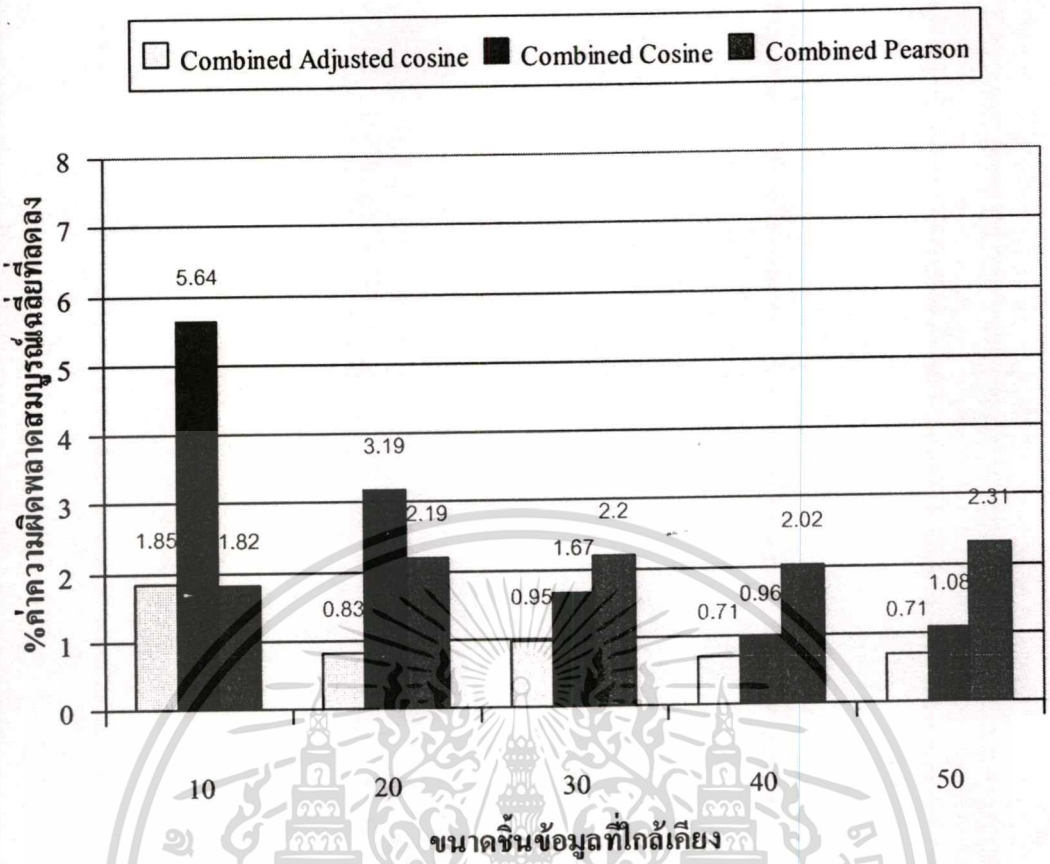
ตารางที่ 4.9 แสดงเปอร์เซ็นต์ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลงสำหรับดาต้าเซ็ต MovieLens

ขนาดชิ้นข้อมูลที่ใกล้เคียง	ค่าความผิดพลาดสมบูรณ์เฉลี่ย (MAE)		% ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลง
	Pure Adjusted cosine	Combined Adjusted cosine	
10	0.866	0.850	1.85%
20	0.847	0.840	0.83%
30	0.845	0.837	0.95%
40	0.844	0.838	0.71%
50	0.843	0.837	0.71%
ขนาดชิ้นข้อมูลที่ใกล้เคียง	**ค่าความผิดพลาดสมบูรณ์เฉลี่ย (MAE)**		**% ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลง**
	Pure Cosine	Combined Cosine	
10	0.869	0.820	5.64%
20	0.846	0.819	3.19%
30	0.837	0.823	1.67%
40	0.835	0.827	0.96%
50	0.837	0.828	1.08%
ขนาดชิ้นข้อมูลที่ใกล้เคียง	**ค่าความผิดพลาดสมบูรณ์เฉลี่ย (MAE)**		**% ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลง**
	Pure Pearson	Combined Pearson	
10	0.877	0.861	1.82%
20	0.867	0.848	2.19%
30	0.865	0.846	2.20%
40	0.864	0.845	2.20%
50	0.864	0.844	2.31%

ตารางที่ 4.10 แสดงเปอร์เซ็นต์ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลงสำหรับดาต้าเซ็ตEachMovie

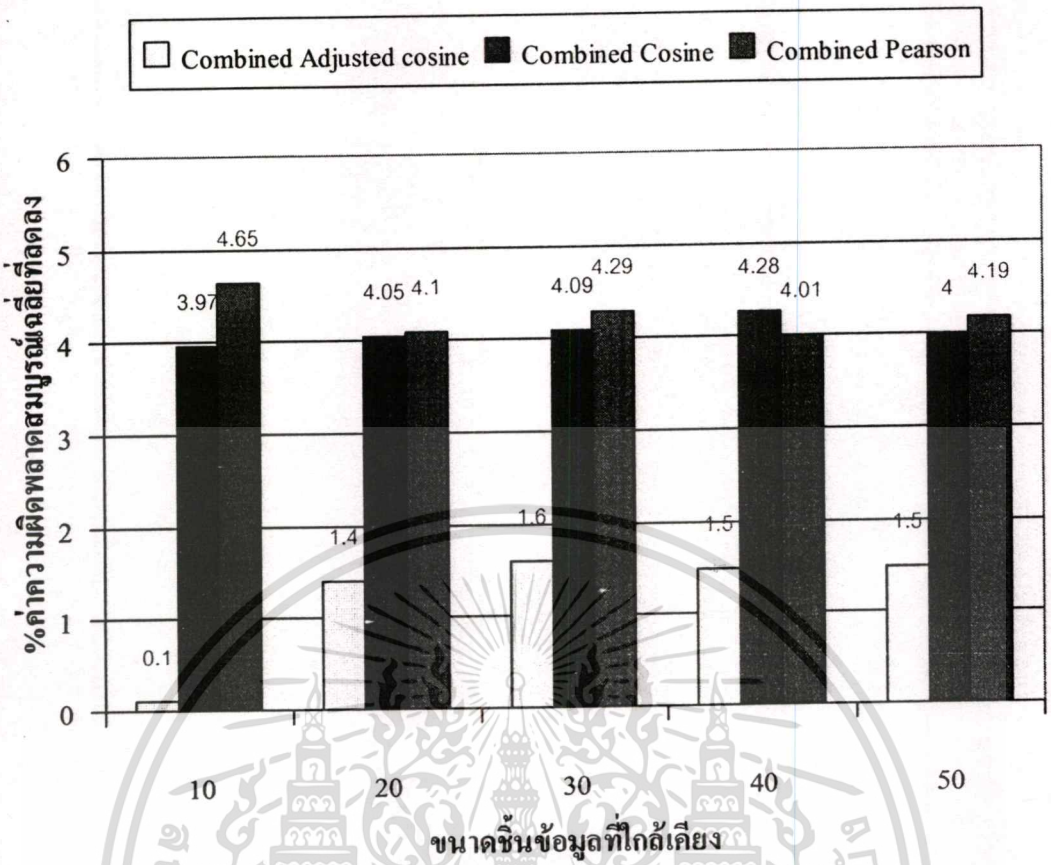
ขนาดชิ้นข้อมูลที่ใกล้เคียง	ค่าความผิดพลาดสมบูรณ์เฉลี่ย (MAE)		% ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลง
	Pure Adjusted cosine	Combined Adjusted cosine	
10	0.994	0.993	0.10%
20	1.000	0.986	1.40%
30	1.002	0.986	1.60%
40	1.002	0.987	1.50%
50	1.002	0.987	1.50%
ขนาดชิ้นข้อมูลที่ใกล้เคียง	ค่าความผิดพลาดสมบูรณ์เฉลี่ย (MAE)		% ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลง
	Pure Cosine	Combined Cosine	
10	1.034	0.993	3.97%
20	1.037	0.995	4.05%
30	1.050	1.007	4.09%
40	1.051	1.006	4.28%
50	1.051	1.009	4.00%
ขนาดชิ้นข้อมูลที่ใกล้เคียง	ค่าความผิดพลาดสมบูรณ์เฉลี่ย (MAE)		% ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลง
	Pure Pearson	Combined Pearson	
10	1.096	1.045	4.65%
20	1.074	1.030	4.10%
30	1.072	1.026	4.29%
40	1.073	1.030	4.01%
50	1.073	1.028	4.19%

จากข้อมูลเปอร์เซ็นต์ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลง ในตารางที่ 4.9 และ 4.10 สามารถแสดงในรูปแบบกราฟได้กราฟดังรูปที่ 4.14 และ 4.15 ตัวเลขที่แสดงอยู่เหนือกราฟแท่งคือเปอร์เซ็นต์ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลงของวิธีการที่นำเสนอ



รูปที่ 4.14 กราฟแสดงเปอร์เซ็นต์ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลงสำหรับดาต้าเซ็ต MovieLens

จากกราฟรูปที่ 4.14 ทดลองวิธีการที่นำเสนอกับดาต้าเซ็ต MovieLens พบว่าทั้งสามวิธี ไม่ว่าจะเป็น Combined Adjusted cosine, Combined Cosine และ Combined Pearson สามารถช่วยลดค่าความผิดพลาดสมบูรณ์เฉลี่ยลงได้ โดยเฉพาะอย่างยิ่งวิธี Combined Cosine ที่ขนาดชิ้นข้อมูลที่ใกล้เคียงมีค่าเท่ากับ 10 สามารถลดค่าความผิดพลาดสมบูรณ์เฉลี่ยลงได้สูงสุดถึง 5.64% และเมื่อขนาดชิ้นข้อมูลที่ใกล้เคียงเพิ่มขึ้นเป็น 20 วิธี Combined Cosine ยังคงสามารถลดค่าความผิดพลาดสมบูรณ์เฉลี่ยลงได้สูงถึง 3.19% และหลังจากนั้นเมื่อขนาดชิ้นข้อมูลที่ใกล้เคียงเพิ่มขึ้นเรื่อยๆ % ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลงจะน้อยลงเป็นลำดับ ในส่วนของวิธี Combined Pearson สามารถลดค่าความผิดพลาดสมบูรณ์เฉลี่ยได้สูงสุด 2.31% ที่ขนาดชิ้นข้อมูลที่ใกล้เคียงเท่ากับ 50 ส่วนวิธี Adjusted cosine สามารถลดค่าความผิดพลาดสมบูรณ์เฉลี่ยได้ค่อนข้างน้อยกว่าวิธีอื่น



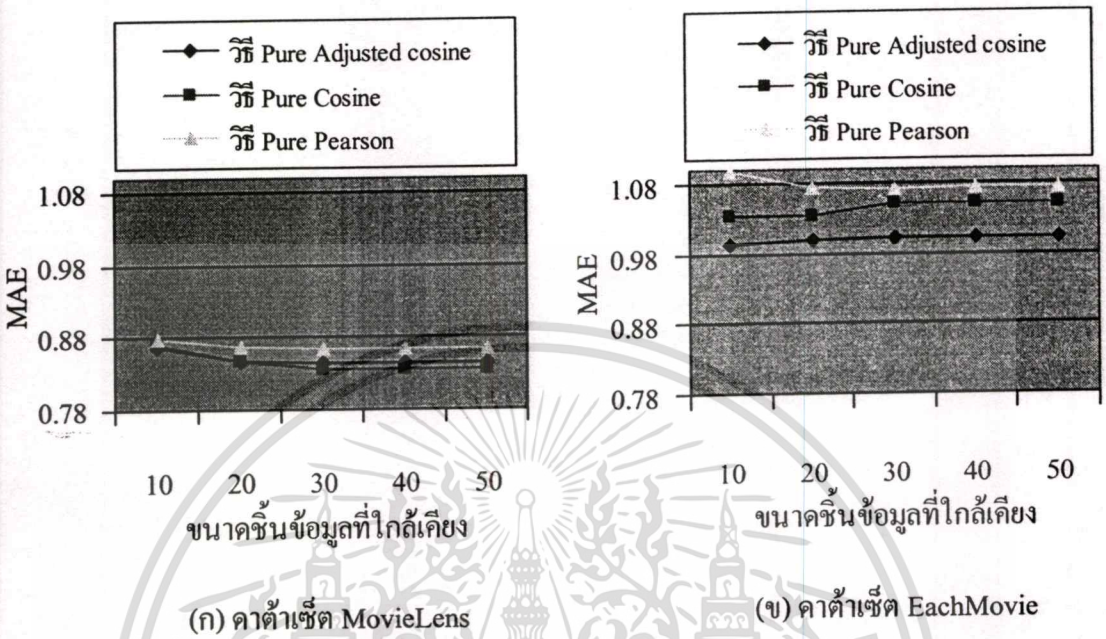
รูปที่ 4.15 กราฟแสดงเปอร์เซ็นต์ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลงสำหรับดาต้าเซ็ต EachMovie

จากกราฟรูปที่ 4.15 ทดลองวิธีการที่นำเสนอกับดาต้าเซ็ต EachMovie พบว่าทั้งสามวิธียังคงสามารถช่วยลดค่าความผิดพลาดสมบูรณ์เฉลี่ยจากวิธีการเดิมลงได้ โดยเฉพาะอย่างยิ่งทั้งวิธี Combined Pearson และ Combined Cosine สามารถลดค่าความผิดพลาดสมบูรณ์เฉลี่ยจากการทำนายลงได้สูงใกล้เคียงกัน ไม่ว่าขนาดชิ้นข้อมูลที่ใกล้เคียงจะเพิ่มขึ้นเพียงใดก็ตาม นอกจากนี้ยังเห็นได้ชัดว่าวิธี Combined Adjusted cosine ยังคงมีเปอร์เซ็นต์ค่าความผิดพลาดสมบูรณ์เฉลี่ยที่ลดลงจากการทำนายค่อนข้างต่ำกว่าวิธีอื่น

4.3.5 ผลการวิเคราะห์การวัดความคล้ายคลึงด้วยวิธี Adjusted cosine, Cosine และ Pearson correlation กับดาต้าเซ็ต MovieLens และ EachMovie

จากการทดลองที่ผ่านมาพบว่าวิธีการที่นำเสนอได้ช่วยทำให้การทำนายมีความถูกต้องมากขึ้นระดับหนึ่ง ดังนั้นจึงจำเป็นอย่างยิ่งที่จะนำวิธีการวัดความคล้ายคลึงทั้งสามวิธีมาวิเคราะห์ผลการทดลองเปรียบเทียบกัน กับดาต้าเซ็ตทั้ง 2 ชุด ในการวิเคราะห์จะเริ่มต้นด้วยการเปรียบเทียบวิธีการเดิมทั้งสามแบบได้แก่ Pure Adjusted cosine, Pure Cosine และ Pure Pearson แสดงดังรูปที่

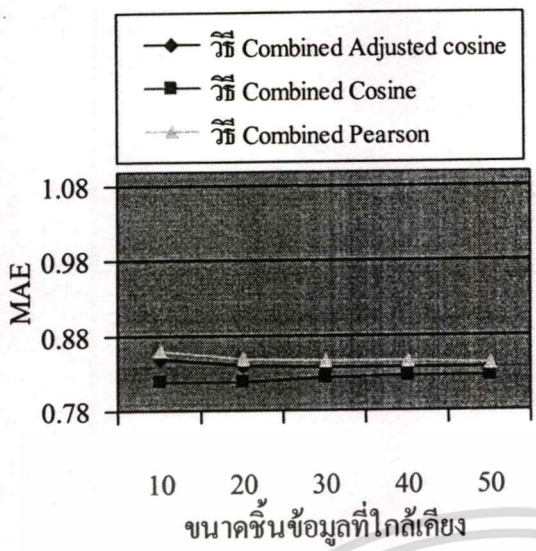
4.16 หลังจากนั้นจึงเริ่มวิเคราะห์วิธีการที่นำเสนอทั้งสามแบบได้แก่ Combined Adjusted cosine, Combined Cosine และ Combined Pearson แสดงดังรูปที่ 4.17



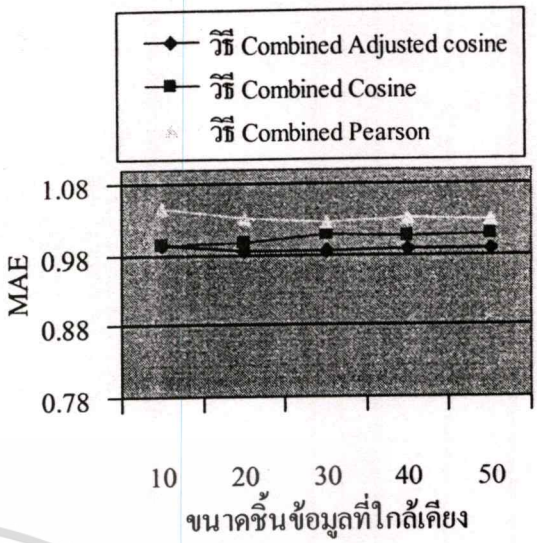
รูปที่ 4.16 กราฟแสดงการเปรียบเทียบค่า MAE ของวิธี Pure Adjusted cosine, Pure Cosine และ Pure Pearson เมื่อขนาดชิ้นข้อมูลที่ใกล้เคียงเปลี่ยนแปลงไป

จากกราฟรูปที่ 4.16 (ก) เป็นการทดสอบดาต้าเซ็ต MovieLens กับทั้งสามวิธีพบว่าค่า MAE ที่ได้มีค่าใกล้เคียงกัน โดยเฉพาะวิธี Pure Cosine ให้ค่า MAE ออกมาน้อยที่สุด เพราะข้อมูลการให้คะแนนเรตติ้งของดาต้าเซ็ต MovieLens ส่วนใหญ่ได้มาจากการแสดงความคิดเห็นในเชิงบวก หรือกล่าวอีกนัยหนึ่งได้ว่าผู้ใช้ส่วนใหญ่ให้คะแนนเรตติ้งไว้ค่อนข้างสูง ดังนั้นดาต้าเซ็ต MovieLens จึงเหมาะกับวิธี Pure Cosine ในทำนองเดียวกันพิจารณากราฟรูปที่ 4.16 (ข) เห็นได้ว่าทั้งสามวิธีให้ค่า MAE ออกมาแตกต่างกันชัดเจน โดยเฉพาะวิธี Pure Adjusted cosine ให้ค่า MAE ออกมาน้อยที่สุด เพราะข้อมูลการให้คะแนนเรตติ้งของดาต้าเซ็ต EachMovie ส่วนใหญ่ได้มาจากการแสดงความคิดเห็นทั้งเชิงบวกและเชิงลบ ดังนั้นดาต้าเซ็ต EachMovie จึงเหมาะกับวิธี Pure Adjusted cosine สุดท้ายนำวิธีการที่นำเสนอมาวิเคราะห์กับดาต้าเซ็ตทั้งสองพบว่า

จากกราฟรูปที่ 4.17 (ก) และ (ข) ไม่ว่าจะเป็นวิธี Combined Adjusted cosine, Combined Cosine และ Combined Pearson ทั้งสามวิธีให้ค่า MAE ออกมาน้อยกว่าวิธีการเดิมแต่ยังคงมีลักษณะของกราฟเหมือนเดิม นั่นคือ ดาต้าเซ็ต MovieLens ยังคงเหมาะกับวิธี Combined Cosine ส่วนดาต้าเซ็ต EachMovie ยังคงเหมาะกับวิธี Combined Adjusted cosine



(ก) ดาต้าเซ็ต MovieLens



(ข) ดาต้าเซ็ต EachMovie

รูปที่ 4.17 กราฟแสดงการเปรียบเทียบค่า MAE ของวิธี Combined Adjusted cosine, Combined Cosine และ Combined Pearson เมื่อขนาดชิ้นข้อมูลที่ใกล้เคียงเปลี่ยนแปลงไป

ดังนั้นจากการทดลองสามารถวิเคราะห์วิธีวัดความคล้ายคลึงทั้งสามวิธีกับ 2 ดาต้าเซ็ต ดังนี้

วิธี Pure/Combined Cosine ให้ค่าความคล้ายคลึงตั้งแต่ 0 ถึง 1 โดย ค่า 0 หมายถึง ข้อมูลการให้เรตติ้งจากผู้ใช้ส่วนใหญ่ไม่มีความคิดเห็นเหมือนกันเลย ส่วนค่าระหว่าง 0 ถึง 1 หมายถึง ผู้ใช้ส่วนใหญ่มีความคิดเห็นในเชิงบวกเหมือนกัน (ชอบเหมือนกัน) ดังนั้นวิธี Cosine จึงเหมาะกับข้อมูลเรตติ้งที่ได้มาจากการแสดงความคิดเห็นในเชิงบวกเหมือนกันเท่านั้น

วิธี Pure/Combined Pearson ให้ค่าความคล้ายคลึงมีค่าอยู่ระหว่าง -1 และ 1 เสมอโดยค่าติดลบหมายถึง ผู้ใช้ส่วนใหญ่มีความคิดเห็นในเชิงลบ นั่นคือ ให้เรตติ้งกับชิ้นข้อมูลหนึ่งมากแต่กลับให้เรตติ้งกับอีกชิ้นข้อมูลน้อย ส่วนค่า 0 หมายถึง ผู้ใช้ส่วนใหญ่ไม่มีความคิดเห็นเหมือนกันเลย และค่าบวกหมายถึง ผู้ใช้ส่วนใหญ่มีความคิดเห็นในเชิงบวก นั่นคือผู้ใช้ให้เรตติ้งกับชิ้นข้อมูลทั้งสองชิ้นใกล้เคียงกัน ดังนั้นวิธีนี้จึงเหมาะกับการค้นหาชิ้นข้อมูลที่ใกล้เคียงจากโคเรตของการแสดงความคิดเห็นเชิงบวกหรือโคเรตของการแสดงความคิดเห็นเชิงลบเท่านั้น อย่างไรก็ตามวิธีนี้จะนำค่าเฉลี่ยเรตติ้งของชิ้นข้อมูลแต่ละชิ้นมาพิจารณาด้วย

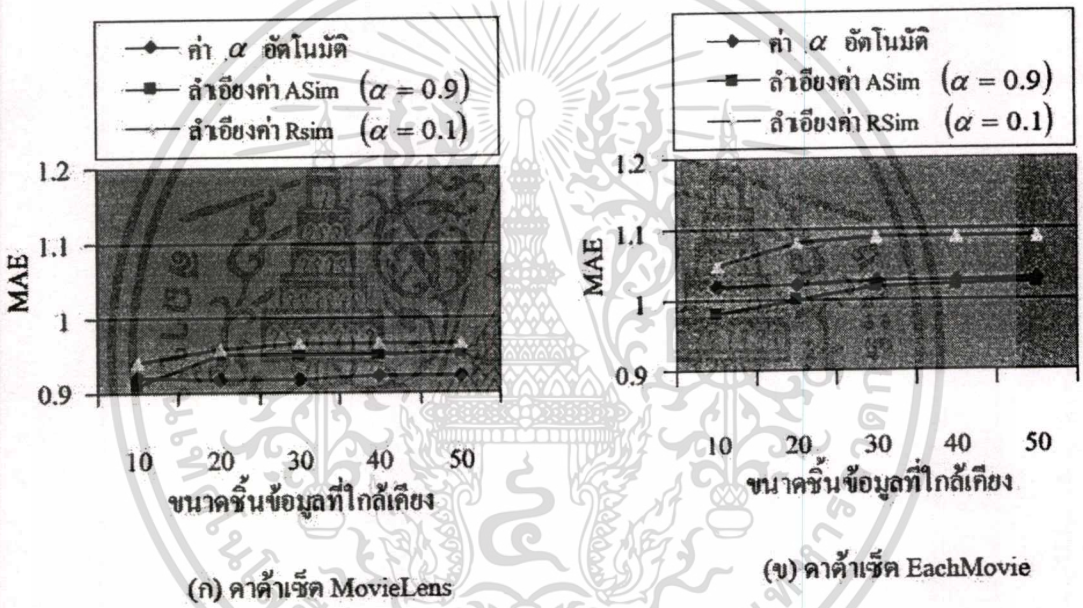
วิธี Pure/Combined Adjusted cosine ให้ค่าความคล้ายคลึงตั้งแต่ -1 ถึง 1 เช่นเดียวกับ Pure/Combined Pearson แตกต่างกันตรงที่วิธีการนี้จะนำค่าเฉลี่ยเรตติ้งของผู้ใช้แต่ละคนมาพิจารณาด้วย เพื่อแก้ปัญหาเรื่องมาตรฐานการให้ระดับคะแนนเรตติ้งที่แตกต่างกันของผู้ใช้

4.3.6 การทดสอบผลกระทบของวิธีการที่นำเสนอด้วยการสุ่มค่าเรตติ้ง

การทดลองนี้ได้นำดาต้าเซ็ต MovieLens และ EachMovie มาทำการสุ่มค่าเรตติ้งเพื่อทดสอบผลกระทบที่มีต่อวิธีการที่นำเสนอ โดยเลือกใช้วิธี Combined Adjusted cosine กับสมการที่ 3.4 (การรวมเชิงเส้นระหว่างค่า ASim และ ค่า RSim) มาพิจารณาร่วมกัน แบ่งออกได้เป็น 3 กรณี คือ

- ใช้ค่า α อย่างอัตโนมัติตามสมการที่ 3.5
- ใช้การลำเอียงค่า ASim โดยกำหนดให้ค่า α มีค่าเท่ากับ 0.9
- ใช้การลำเอียงค่า RSim โดยกำหนดให้ค่า α มีค่าเท่ากับ 0.1

ผลจากการทดลองแสดงดังรูปที่ 4.18



รูปที่ 4.18 กราฟแสดงผลกระทบของวิธี Combined Adjusted cosine เมื่อสุ่มค่าเรตติ้ง

จากกราฟรูปที่ 4.18 พบว่าทั้งสองดาต้าเซ็ตที่เกิดจากการสุ่มค่าเรตติ้ง ค่า MAE ที่ได้จะมีลักษณะคล้ายคลึงกัน นั่นคือ ทั้งสามกรณีให้ค่า MAE ต่ำที่สุดที่ขนาดชิ้นข้อมูลที่ใกล้เคียงมีค่าเท่ากับ 10 หลังจากนั้นเมื่อขนาดชิ้นข้อมูลที่ใกล้เคียงเพิ่มขึ้นจาก 10 ถึง 50 การใช้ค่า α อย่างอัตโนมัติตามสมการที่ 3.5 ค่า MAE ค่อนข้างคงที่ ส่วนการลำเอียงค่า ASim และ RSim พบว่าค่า MAE มีแนวโน้มสูงขึ้น ทั้งนี้ค่า MAE ที่ได้จากการสุ่มค่าเรตติ้งจะมีค่าสูงกว่าค่า MAE ที่ได้จากการให้เรตติ้งจริง อย่างไรก็ตามการใช้ค่า α อย่างอัตโนมัติกับวิธีการที่นำเสนอยังคงช่วยเพิ่มประสิทธิภาพของการทำนายได้ถูกต้องมากกว่าการลำเอียงค่าใดค่าหนึ่ง

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้ได้นำเสนอวิธีการเพิ่มประสิทธิภาพของอัลกอริทึมไอเท็มเบส CF เพื่อแก้ปัญหาการให้คะแนนเรตติ้งต่อชิ้นข้อมูลที่ไม่ทั่วถึงและปัญหาชิ้นข้อมูลที่ยังไม่มีการให้คะแนนเรตติ้งไว้ของอัลกอริทึมไอเท็มเบส CF แบบเดิม ซึ่งเป็นสาเหตุสำคัญที่ทำให้ผลการทำนายหาค่าความพึงพอใจของอัลกอริทึมนี้มีค่าความผิดพลาดค่อนข้างสูง จากงานวิจัยนี้ทำให้ได้วิธีการวัดความคล้ายคลึงระหว่างชิ้นข้อมูล ที่อาศัยค่าความเชื่อมั่นของกฎความสัมพันธ์มาใช้เป็นระดับความคล้ายคลึงตามคุณสมบัติของชิ้นข้อมูล เพื่อเพิ่มการค้นหาชิ้นข้อมูลที่มีลักษณะใกล้เคียง (คล้าย) และนำไปรวมกับขั้นตอนการค้นหาชิ้นข้อมูลที่ใกล้เคียงในอัลกอริทึมไอเท็มเบส CF แบบเดิม ซึ่งถือว่าเป็นขั้นตอนที่สำคัญที่สุด เพราะทำให้สามารถนำทั้งคุณสมบัติและเรตติ้งของชิ้นข้อมูลมาพิจารณาร่วมกัน เพื่อค้นหาชิ้นข้อมูลที่ใกล้เคียงได้อย่างเหมาะสม ก่อนจะนำกลุ่มชิ้นข้อมูลที่มีลักษณะใกล้เคียงกันไปทำนายหาค่าความพึงพอใจที่คาดว่าผู้ใช้เป้าหมายจะมีต่อชิ้นข้อมูลเป้าหมายได้อย่างถูกต้องมากยิ่งขึ้น

นอกจากนี้การวัดความคล้ายคลึงระหว่างชิ้นข้อมูลด้วยวิธีการที่นำเสนอถือว่าเป็นวิธี CBF อย่างหนึ่งที่มีความยืดหยุ่นมากกว่าวิธี CBF แบบเดิม ดังเช่น การวัดความคล้ายคลึงเชิงมุม เพราะการวัดระดับความคล้ายคลึงด้วยวิธีการที่นำเสนอนี้จะไม่จำกัดอยู่แค่การปรากฏของคุณสมบัติเดียวกันเท่านั้น แต่จะขึ้นอยู่กับปริมาณชิ้นข้อมูลในระบบทั้งหมดที่เพิ่มขึ้นหรือลดลงอยู่ตลอดเวลา

จากการทดสอบวิธีการที่นำเสนอเปรียบเทียบกับวิธีการเดิมของอัลกอริทึมไอเท็มเบส CF ด้วยการวัดค่า MAE, ส่วนเบี่ยงเบนมาตรฐานของค่า MAE และเปอร์เซ็นต์ของค่า MAE ที่ลดลง เทียบกับขนาดชิ้นข้อมูลที่ใกล้เคียงที่เปลี่ยนแปลงไปตั้งแต่ขนาด 10 จนถึง 50 พบว่า วิธีการที่นำเสนอสามารถช่วยเพิ่มประสิทธิภาพให้กับอัลกอริทึมไอเท็มเบส CF แบบเดิมได้ทุกกรณีของการทดสอบ ไม่ว่าขนาดชิ้นข้อมูลที่ใกล้เคียงจะเปลี่ยนแปลงไปเท่าไหร่ก็ตาม หรือกล่าวอีกนัยหนึ่งได้ว่าเมื่อนำวิธีการที่นำเสนอไปใช้ร่วมกับวิธีการเดิมทั้งหมด 3 วิธี ได้แก่ วิธี Combined Adjusted cosine, Combined Cosine และ Combined Pearson พบว่าวิธีการที่นำเสนอให้ค่า MAE ออกมาน้อยกว่าวิธีการแบบเดิม ดังเช่น Pure Adjusted cosine, Pure Cosine และ Pure Pearson โดยเฉพาะอย่างยิ่งวิธี Combined Cosine สามารถช่วยลดค่า MAE จากวิธีการเดิมลงได้สูงสุดถึง 5.64% ทั้งนี้ผลการทดลองจะขึ้นอยู่กับดาต้าเซ็ตที่นำมาทดสอบด้วย

ดังนั้นจึงสรุปได้ว่าวิธีการที่นำเสนอสามารถช่วยลดค่าความผิดพลาดจากการทำนายลงได้ในระดับหนึ่งและสามารถช่วยเพิ่มประสิทธิภาพให้กับอัลกอริทึมไอเท็มเบส CF ให้ผลการทำนายค่าความพึงพอใจได้ถูกต้องมากยิ่งขึ้น

5.2 ข้อเสนอแนะ

1. วิธีการที่นำเสนอ นำเฉพาะคุณสมบัติของชิ้นข้อมูลมาพิจารณาร่วมกับอัลกอริทึมไอเท็มเบส CF เท่านั้น แต่วิธีการที่นำเสนอนี้ยังสามารถนำข้อมูลอื่นๆ มาใช้ได้อีก ดังเช่น โปรไฟล์ หรือพฤติกรรมการใช้งานของผู้ใช้ ซึ่งล้วนเป็นประโยชน์สำหรับการทำนายหาสิ่งที่ผู้ใช้สนใจได้เป็นอย่างดี

2. วิธีการที่นำเสนอใช้การค้นหากฎความสัมพันธ์แบบระดับเดียว ทำให้ไม่สามารถจัดแบ่งประเภทชิ้นข้อมูลให้เป็นหมวดหมู่ได้ ดังนั้นวิธีการที่นำเสนอยังสามารถนำการค้นหากฎความสัมพันธ์แบบหลายระดับชั้นมาประยุกต์ใช้แทนได้ เพื่อสามารถแบ่งแยกประเภทของชิ้นข้อมูลออกเป็นหมวดหมู่และสามารถจัดการกับระบบให้การแนะนำที่มีชิ้นข้อมูลหลายประเภทได้

3. ในการค้นหากลุ่มของชิ้นข้อมูลที่ใกล้เคียงในอัลกอริทึมไอเท็มเบส CF ยังคงเป็นการคำนวณแบบเชิงเส้น ซึ่งในส่วนนี้สามารถนำวิธีอ๊อปติไมเซชั่น (Optimization) มาใช้เลือกหากลุ่มของชิ้นข้อมูลที่ดีที่สุด เพื่อนำไปสู่ระบบให้การแนะนำแบบปรับเปลี่ยนได้ (Adaptive recommender system) ต่อไป

4. ในส่วนการประเมินผล งานวิจัยนี้ใช้เพียงข้อมูลเรตติ้งที่มีอยู่แล้วในอดีต ซึ่งในโลกของความเป็นจริงผู้ใช้แต่ละคนอาจจะมีรสนิยมหรือความชอบเปลี่ยนแปลงไปตามกาลเวลา เช่น ที่ผ่านมาเคยไม่ชอบชิ้นข้อมูลเหล่านั้น แต่ต่อไปในอนาคตก็มีความเป็นไปได้ที่จะเริ่มชอบชิ้นข้อมูลเหล่านั้น ดังนั้นควรจะมีการประเมินความพอใจของผู้ใช้อยู่ตลอดเวลาผ่านทางส่วนติดต่อกับผู้ใช้ หรือที่เรียกว่า ส่วนติดต่อผู้ใช้ (Interface) เพื่อให้ผู้ใช้แต่ละคนเข้ามาใช้งานระบบและสอบถามว่าผู้ใช้พึงพอใจต่อผลการทำนายมากน้อยเพียงใด ซึ่งต้องใช้ระยะเวลาในการเก็บข้อมูลพอสมควร หลังจากนั้นนำข้อมูลที่รวบรวมได้ไปวิเคราะห์เป็นเปอร์เซ็นต์ของความพึงพอใจที่ผู้ใช้มีต่อวิธีการที่นำเสนอว่าเพิ่มขึ้นหรือลดลงมากน้อยเพียงใด

5. วิธีการที่นำเสนอ ถือเป็นวิธีการทำนายหาค่าความพึงพอใจของผู้ใช้ที่มีความถูกต้องสูงเมื่อเทียบกับอัลกอริทึมไอเท็มเบส CF แบบเดิม และเป็นพื้นฐานสำคัญที่จะนำไปพัฒนาเป็นการคำนวณแบบท็อปเอ็นเรคคอมเมนเดชั่นต่อไปได้อย่างเหมาะสม

เอกสารอ้างอิง

[1] Aggarwal C. C., Wolf J. L., and Yu P. S., "A new methods for similarity indexing for market data", In Proceedings of the 1999 ACM SIGMOD Conference, Philadelphia, PA, June 1999.

[2] Agrawal R., and Srikant R., "Fast Algorithms for Mining Association Rules", IBM Almaden Research Center Technical Report RJ9839, 1994.

[3] Agrawal R., Imielinski T., and Swami A., "Mining Association Rules Between Sets of Items in Large Databases", *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pp. 207-216, Washington, D.C., May 1993.

[4] Baeza-Yates R., and Ribeiro-Neto B., "Modern Information Retrieval", Addion-Wesley Publishing Company, ACM Press, New York, 1999.

[5] Balbanovic M., and Shoham Y., "Fab: Content-based, Collaborative Recommendation", CACM, 40(3), 66-72, March 1997.

[6] Bollacker K., et al. "A System for Automatic Personalized Tracking of Scientific Literature on the Web", In ACM Conference on Digital Libraries, 4, 1999, Berkeley, CA. Procedings, 1999.

[7] Burke R., "Hybrid Recommender Systems: Survey and Experiments", User Modeling and User-adapted Interaction, v. 12, n. 4, p. 331-370, Nov 2002.

[8] Chaiwat T., and Ouen P., "A Combination of Content-based Filtering and Item-based Collaborative Filtering Using Association Rules", The 1st ECTI Annual Conference (ECTI-CON 2004), Pattaya, Thailand, 2004.

[9] Chaiwat T., and Ouen P., "Finding Item Neighbors in Item-Based Collaborative Filtering by Adding Item Content", The Eighth International Conference on Control, Automation, Robotics and Vision (ICARCV 2004), Kunming, China, 2004.

[10] Chee S. H. S., "RecTree: A Linear Collaborative Filtering Algorithm", Master's thesis, Simon Fraser University, 2000.

[11] Deshpande M., and Karypis G., "Item-based Top-N Recommendation Algorithms", ACM Transactions on Information Systems (TOIS), Volume 22, Number 1, January 2004.

[12] Goldberg D., Nichols D., Oki B. M. and Terry D., "Using collaborative filtering to weave an information tapestry", Communications of the ACM, 35(12):61-70, 1992.

[13] Herlocker J. L., Konstan J. A., Terveen L. G., and Riedl J. T., "Evaluating Collaborative Filtering Recommender Systems", ACM Transactions on Information Systems (TOIS), Volume 22, Number 1, January 2004.

[14] Herlocker J., "MovieLens data set", GroupLens Research Project, University of Minnesota, *http://www.grouplens.org/*, 1998.

[15] Huang Z., Chen H., and Zeng D. D., "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering", ACM Transactions on Information Systems (TOIS), Volume 22, Number 1, January 2004.

[16] Lin W., Alvarez S. A., and Ruiz C., "Collaborative Recommendation via Adaptive Association Rule Mining", International Workshop on Web Mining for E-Commerce (WEBKDD'2000), 2000.

[17] McJonese P., "EachMovie collaborative filtering data set", DEC Systems Research Center., http://www.research.compaq.com/SRC/eachmovie/, 1997

[18] Melville P., Mooney R. J., and Nagarajan R., "Content-boosted collaborative filtering for improved recommendations", Eighteenth national conference on Artificial intelligence, p.187-192, July 28-August 01, Edmonton, Alberta, Canada, 2002.

[19] O'Conner M. and Herlocker J., "Clustering items for collaborative filtering", In Proceedings of the ACM SIGIR Workshop on Recommender Systems, Berkeley, CA, August 1999.

[20] Resnick P., Iacovou N., Suchak M., Bergstrom P., and Riedl J., "GroupLens: An open architecture for collaborative filtering of netnews", Computer Supported Collaborative Work Conference, 1994, Chapel Hill, North Carolina - USA. Proceedings, 1994.

[21] Sarwar B. M., Karypis G., Konstan J. A., and Riedl J. T., "Analysis of recommender algorithms for e-commerce", In proceedings of the 2nd ACM E-Commerce Conference (EC'00), Minneapolis, MN, October 2000.

[22] Sarwar B. M., Karypis G., Konstan J. A., and Riedl J. T., "Application of dimensionality reduction In recommender systems – a case study", In Proceedings of the WebKDD 2000 Workshop at the ACM-SIGKDD Conference on Knowledge Discovery in Databases (KDD'00), August 2000.

[23] Sarwar B. M., Karypis G., Konstan J., and Riedl J. T., "Item-Based Collaborative Filtering Recommendation Algorithms", In Proceedings of the 10th International WWW Conference, Hong Kong, May 2001.

[24] Sarwar B. M., Konstan J. A., Borchers A., Herlocker J., Miller B., and Riedl J. T., "Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System", Computer Support Cooperative Works 1998 Conference, Seattle, 1999.

[25] Schafer J.B., Konstan J., and Riedl J., "Recommender Systems in E-Commerce", ACM Conf. Electronic Commerce (EC-99), Denver, CO, pp. 158-166, Nov. 1999.

[26] Zeng C., Xing Chun-Xiao, and Zhou Li-Zhu., "Similarity measure and instance selection for collaborative filtering", In Proceedings of the twelfth international conference on World Wide Web, May 20-24, Budapest, Hungary, 2003.

ภาคผนวก ก
ผลงานที่ได้รับการตีพิมพ์

[1] Chaiwat T., and Ouen P., "A Combination of Content-based Filtering and Item-based Collaborative Filtering Using Association Rules", The 1st ECTI Annual Conference (ECTI-CON 2004), Pattaya, Thailand, 2004.

[2] Chaiwat T., and Ouen P., "Finding Item Neighbors in Item-Based Collaborative Filtering by Adding Item Content", The Eighth International Conference on Control, Automation, Robotics and Vision (ICARCV 2004), Kunming, China, 2004.

ประวัติผู้เขียน

ชื่อ นายชัยวัฒน์ ติรวีรขจร

เกิดวันที่ 17 กรกฎาคม 2520 จังหวัดนครราชสีมา

ที่อยู่ 243 ถ. เดชอุดม ซ. 6 ต. ในเมือง อ. เมือง จ. นครราชสีมา 30000

การศึกษา ปีการศึกษา 2540 -2542
ระดับปริญญาตรีวิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมคอมพิวเตอร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ความชำนาญเฉพาะด้าน 1) จาวาและไพตรอนโปรแกรมมิ่ง
2) ออกแบบพัฒนาระบบเชิงวัตถุและเว็บแอพพิเคชั่น
3) ควบคุมการวิเคราะห์หาสาเหตุและแก้ไขข้อผิดพลาดการทดสอบฮาร์ดดิสต์

ประสบการณ์การทำงาน

ปี พ.ศ. 2544-2546 ตำแหน่งเจ้าหน้าที่วิเคราะห์และจัดระบบสารสนเทศ โครงการฝ่ายระบบสารสนเทศ บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปี พ.ศ. 2548-ปัจจุบัน ตำแหน่งวิศวกรโปรเซส ฝ่ายแบ็คเอ็น แผนกไดรฟ์โอเปอร์เรชั่น บริษัทซีเกทประเทศไทย จำกัด (โคราช)