ระบบแนะนำภาพยนตร์แบบออนไลน์โดยใช้เทคนิคการกรองข้อมูล

ONLINE MOVIE RECOMMENDATION USING COLLABORATIVE FILTERING AND CONTENT-BASED FILTERING

นิภัทร์ วีระศิลป์

วิทวัส เพชรชาติ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

ปริญญานิพนธ์ปีการศึกษา 2550

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบแนะนำภาพยนตร์แบบออนไลน์โดยใช้เทคนิคการกรองข้อมูล

Online Movie Recommendation Using Collaborative Filtering and Content-Based Filtering

ผู้จัดทำ

1. นายนิภัทร์ วีระศิลป์ รหัสนักศึกษา 47010390

2. นายวิทวัส เพชรชาติ รหัสนักศึกษา 47010706

อาจารย์ที่ปรึกษา

(รศ. ดร. เอื้อน ปิ่นเงิน)

ระบบแนะนำภาพยนตร์แบบออนไลน์โดยใช้เทคนิคการกรองข้อมูล

นายนิภัทร์ วีระศิลป์ 47010390
นายวิทวัส เพชรชาติ 47010706
รศ. ดร. เอื้อน ปิ่นเงิน อาจารย์ที่ปรึกษา
ปีการศึกษา 2550

บทคัดย่อ

ปริญญานิพนธ์นี้นำเสนอวิธีใหม่เพื่อเพิ่มประสิทธิภาพของระบบให้คำแนะนำโดยโครงงานนี้จะนำเสนอวิธีแก้ไขปัญหาโดยการรวมเทคนิคของ Collaborative Filtering (CF) และ Content-Based Filtering (CBF) เข้าด้วยกัน โดยใช้การทำนายด้วยวิธีแบบ CBF มาปรับปรุงข้อมูลผู้ใช้และชิ้นข้อมูลที่มีอยู่ หลังจากนั้นนำเสนอการแนะนำผ่าน CF เพื่อให้ผลการแนะนำออกมาอย่างมีประสิทธิภาพมากยิ่งขึ้น โดยในการทำการทดลองนั้นเลือกใช้ข้อมูลทางด้านภาพยนตร์มาทำการทดลองผ่านทางระบบแนะนำภาพยนตร์แบบออนไลน์

จากการทดลองระบบให้การแนะนำด้วยคาต้าเซตที่ทำขึ้นพบว่าวิธีการที่นำเสนอสามารถแก้ไขปัญหาการให้เรตติ้งต่อชิ้นข้อมูล (Sparsity Problem) ปัญหาการแยกแยะเรตติ้ง (Transparency Problem) และ ปัญหาชิ้นข้อมูลที่ไม่มีการให้เรตติ้งไว้ (First-rater Problem) ลงได้ ด้วยการลดข้อเสียของวิธีที่มีอยู่ ทำให้ระบบให้การแนะนำมีประสิทธิภาพมากขึ้น

Online Movie Recommendation Using Collaborative Filtering and Content-Based Filtering

Mr. Nipatr Veerasilpa 47010390

Mr. Wittawat Bejrajati 47010706

Assoc. Prof. Dr. Ouen Pinngern Advisor

Academic Year 2007

ABSTRACT

This thesis proposes new method to improve efficiency of Recommendation System by combining two techniques, Collaborative Filtering (CF) and Content-Based Filtering (CBF). Our approach uses a content-based predictor to enhance existing user data, and then provides personalized suggestions through collaborative filtering. In the experiment we use movie data to construct recommendation via online movie recommendation system. Our method can overcome the disadvantage of CF alone or CB alone. We can solve Sparsity Problem, Transparency Problem and First-rater Problem. This can make a better Recommendation System in the future.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ได้จัดทำขึ้นเพื่อประกอบการเรียนวิชา Project ซึ่งนับว่าเป็นโอกาสอันดีที่ทำให้ข้าพเจ้าได้นำความรู้ในภาคทฤษฎีมาปฏิบัติการ เป็นการเพิ่มพูนความรู้และประสบการณ์ให้แก่ข้าพเจ้า

ข้าพเจ้าขอกราบขอบพระคุณ รศ.ดร. เอื้อน ปิ่นเงิน อาจารย์ผู้ควบคุมปริญญานิพนธ์เป็นอย่างสูงที่ให้การสนับสนุนในงานวิจัยนี้เป็นอย่างดีเสมอมา

ขอขอบพระคุณ อาจารย์ ดร. สรัญญา มณีโรจน์ อาจารย์ประจำภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ได้ให้คำแนะนำและคำปรึกษาเกี่ยวกับอัลกอรึทึมการทำงานในงานวิจัยนี้

ขอขอบคุณพี่และเพื่อนๆในห้องวิจัย Information Science (IS) Lab ของ RECCIT สำหรับคำแนะนำและกำลังใจที่ดีตลอดการทำโครงงานชิ้นนี้

ข้าพเจ้าขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูข้าพเจ้ามาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจเอาใจใส่ในทุกๆด้าน ตลอดจนครูอาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาจนข้าพเจ้ามีวันนี้ได้ ข้าพเจ้าขอรำลึกในพระคุณอันสุดประมาณ และสำหรับคุณงามความดีอันใดที่เกิดจากปริญญานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับ บิดา มารดา และครูอาจารย์ของข้าพเจ้า ข้าพเจ้าขอกราบขอบพระคุณมา ณ ที่นี้

นิภัทร์ วีระศิลป์

วิทวัส เพชรชาติ

สารบัญ

หน้า

สารบัญ(ต่อ)

หน้า

สารบัญ(ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

รูปที่ **หน้า**

สารบัญรูป(ต่อ)

รูปที่ **หน้า**

บทที่ 1
บทนำ

ปัจจุบันเป็นยุคของข้อมูลข่าวสาร ทุกหนทุกแห่งเต็มไปด้วยข้อมูลข่าวสาร ระบบการให้คำแนะนำ(Recommendation System) เป็นอีกหนึ่งเทคโนโลยี ที่ทำการกรองข้อมูลที่มีอยู่มากมายในปัจจุบันโดยแนะนำชิ้นข้อมูลที่คาดว่าผู้ใช้น่าจะสนใจโดยมีพื้นฐานมาจากการชอบหรือไม่ชอบของผู้ใช้เองโดยระบบจะทำการจับคู่ผู้ใช้กับผู้ใช้อื่นที่มีความชอบคล้ายคลึงกันและจะแนะนำชิ้นข้อมูลที่ผู้ใช้ที่คล้ายคลึงชอบ เช่น ถ้า บอย และ เจน ชอบชมภาพยนตร์คล้ายคุณในอดีตและทั้งคู่ต่างให้คะแนนความชอบภาพยนตร์เรื่อง Harry Potter สูง คุณมีแนวโน้มที่จะชอบภาพยนตร์เรื่องนี้เช่นกัน

ในปัจจุบันมีข้อมูลข่าวสารจำนวนมากไม่ว่าเราจะอยู่ส่วนไหนบนโลกเราก็จะสามารถเข้าถึงข้อมูลข่าวสารได้ในเวลาไม่นาน แต่การที่จะได้ข้อมูลที่ต้องการออกมาอย่างรวดเร็วและตรงความต้องการนั้นไม่ใช่เรื่องง่าย เนื่องจากข้อมูลที่มีอยู่นั้นกระจัดกระจายอย่างไม่เป็นระบบ ตัวอย่างเช่น ถ้าเราต้องการหาข้อมูลเกี่ยวกับ ภาพยนตร์ใน Search Engine นั้น ผลลัพธ์ที่ได้อาจไม่ตรงกับที่เราต้องการ เช่น เราต้องการหาข้อมูลเกี่ยวกับหนัง Philadelphia แทนที่ search Engine จะแสดงข้อมูลเกี่ยวกับภาพยนตร์ออกมา ลำดับแรกๆของผลการค้นหานั้นจะแสดงข้อมูลเกี่ยวกับ เมือง Philadelphia ในประเทศ สหรัฐอเมริกา ซึ่งไม่ตรงกับข้อมูลภาพยนตร์อย่างที่ผู้ใช้ต้องการ หรือบางทีผู้ใช้ต้องการชมภาพยนตร์ดีๆ เรื่องหนึ่ง แต่ไม่สามารถคิด Keyword ออกมาได้ เช่น คิดชื่อเรื่องไม่ออก ระบบก็จะไม่สามารถแสดงข้อมูลที่ผู้ใช้ต้องการได้ ระบบให้คำแนะนำ ถือเป็นตัวช่วยหนึ่ง ที่ทำให้เราเข้าถึงข้อมูลที่ต้องการได้อย่างรวดเร็วและมีประสิทธิภาพ โดยระบบให้คำแนะนำเป็นระบบที่ทำการแนะนำข้อมูลออกมาให้ตรงกับความต้องการของผู้ใช้ โดยมีพื้นฐานมาจากลักษณะการชอบของผู้ใช้เองรวมถึงผู้ใช้อื่นที่มีลักษณะการชอบใกล้เคียงกัน ปัจจุบันได้มีการนำระบบให้การแนะนำมาใช้กันอย่างแพร่หลายในการดำเนินธุรกิจแบบอิเล็กทรอนิกส์ (E-Commerce) เช่น Amazon.com ได้นำระบบให้การแนะนำในการแนะนำสินค้าแก่ผู้ใช้บริการ CDNOW.com ใช้ระบบให้การแนะนำแก่ผู้ใช้บริการในเรื่องของดนตรีและเพลง เป็นต้น

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันระบบให้การแนะนำ เป็นที่นิยมมากขึ้น โดยเทคนิคที่เป็นที่นิยมในการทำการแนะนำคือ Collaborative Filtering (CF) และ Content-Based Filtering (CBF) เทคนิค Collaborative Filtering นั้นเป็นเทคนิคที่ใช้ข้อมูลการให้ข้อมูลเรตติ้งของชิ้นข้อมูลที่ผู้ใช้เคยให้ไว้ในอดีตมาพิจารณาร่วมกับกลุ่มของผู้ใช้ที่มีลักษณะการให้เรตติ้งคล้ายคลึงกัน เพื่อใช้ในการพิจารณาหาค่าความพึงพอใจที่คาดว่าผู้ใช้จะมีต่อชิ้นข้อมูลเป้าหมายนั้น เทคนิคนี้ยังมีปัญหาในการให้คะแนน

เรตติ้งต่อชิ้นข้อมูล (Sparsity Problem) กล่าวคือ ชิ้นข้อมูลในระบบมีมากแต่ข้อมูลที่ผ่านการให้เรตติ้งนั้นมีน้อยจะทำให้แนะนำข้อมูลออกมาไม่มีประสิทธิภาพ อีกปัญหาหนึ่งคือTransparency Problem กล่าวคือ ระบบนั้นไม่สามารถแยกแยะการให้ข้อมูลเรตติ้งของชิ้นข้อมูลแต่ละชิ้นว่าผู้ใช้ให้คะแนนนั้นมาด้วยเหตุผลใด ปัญหาที่สำคัญอีกอย่างหนึ่งก็คือ เทคนิคนี้จะทำการแนะนำเฉพาะชิ้นข้อมูลที่ผ่านการให้เรตติ้งแล้วเท่านั้นสำหรับชิ้นข้อมูลที่ยังไม่ผ่านการให้เรตติ้งนั้นจะไม่มีโอกาสแสดงออกมา

อีกเทคนิคหนึ่งที่เป็นที่นิยมก็คือ Content-Based Filtering ซึ่งการวิเคราะห์นั้นดูจาก content ของข้อมูลแต่ละชิ้นที่ผู้ใช้ทำการให้ เรตติ้ง ว่ามีการสัมพันธ์กันกับโปรไฟล์ของผู้ใช้จากนั้นระบบจะทำการแนะนำชิ้นข้อมูลที่มี content ใกล้เคียงกันกับ โปรไฟล์ของผู้ใช้ ออกมาแนะนำให้กับผู้ใช้ ซึ่งเทคนิคนี้มีข้อดีคือสามารถแนะนำชิ้นข้อมูลที่ยังไม่ถูกให้เรตติ้งแต่มี content ใกล้เคียงกัน ออกมาให้กับผู้ใช้แต่ก็ยังมีปัญหาในส่วนของผลการแนะนำนั้นจะขึ้นอยู่กับโปรไฟล์ของผู้ใช้กล่าวคือชิ้นข้อมูลที่ไม่ตรงกับโปรไฟล์ของผู้ใช้จะไม่มีโอกาสที่จะถูกหยิบออกมาแนะนำให้กับผู้ใช้ถึงแม้ว่าชิ้นข้อมูลนั้นจะเป็นชิ้นข้อมูลที่ผู้ใช้สนใจก็ตาม ซึ่งเป็นความท้าทายและเป็นประเด็นสำคัญในการแก้ไขปัญหาดังกล่าวของทั้งสองเทคนิคให้มีประสิทธิภาพเพิ่มมากขึ้น นำไปสู่เทคโนโลยีและวิธีการใหม่ๆในอนาคต

1.2 วัตถุประสงค์ของการทำวิจัย

งานวิจัยนี้มุ่งเน้นที่จะศึกษาวิจัยถึงวิธีการแก้ไขปัญหาดังที่กล่าวไว้แล้วในหัวข้อที่ผ่านมาเพื่อเพิ่มประสิทธิภาพโดยการรวมเทคนิค Collaborative Filtering และ Content-Based Filtering เข้าด้วยกันโดยวิธีที่นำเสนอจะสามารถแก้ไขปัญหาความเบาบางของการให้เรตติ้ง (Spasity Problem) ด้วยการใช้ Naive Bayesian มาแก้ปัญหาด้วยการคาดคะเนค่าที่ยังไม่ได้ทำการให้ เรตติ้ง (Missing Value) จะทำให้สามารถหา ผู้ใช้ ที่มี Profile ใกล้เคียงกันได้แม่นยำมากยิ่งขึ้น

1.3 สมมุติฐานของการทำวิจัย

งานวิจัยนี้ได้ตั้งสมมุติฐานไว้ว่าการนำข้อดีของ Collaborative Filtering (CF) และ Content-Based Filtering (CBF) มารวมกัน จะสามารถนำทั้งข้อมูลการแสดงความชอบ และคุณสมบัติของชิ้นข้อมูลแต่ละชิ้นมาพิจารณาของชิ้นข้อมูลที่มีลักษณะใกล้เคียงกันกับ โปรไฟล์ของผู้ใช้ รวมทั้งพิจารณาความเห็นของผู้ใช้ที่มีลักษณะใกล้เคียงกับผู้ใช้ปัจจุบันเพื่อให้การแนะนำออกมาอย่างถูกต้องและมีประสิทธิภาพ

1.4 ขอบเขตของการวิจัย

การวิจัยนี้เลือกใช้ข้อมูลทางด้านภาพยนตร์มาทำการทดลอง ซึ่งเป็นข้อมูลการให้เรตติ้งจริงจาก ดาต้าเซต ของ Yahoo movies มาทำการทดลองโดยใช้เทคนิคทั้งสองดังที่กล่าวมาแล้วข้างต้น เพื่อให้ผลทำนายค่าความพึงพอใจของมาถูกต้องและน่าเชื่อถือโดยตรวจสอบประสิทธิภาพด้วยวิธีทั้ง 5 ได้แก่ MAE, Precision, Recall, Specificity และ Negative Predictive Value

1.5 ขั้นตอนการทำวิจัย

1.ศึกษาเทคนิคการกรองข้อมูลสำหรับระบบให้การแนะนำ

2. เลือกและศึกษาเทคนิคที่จะทำการทดลอง

2.1 Collaborative Filtering

2.2 Content-Based Filtering

3. สรุปข้อดีข้อเสียของเทคนิคที่เลือกทำการทดลอง

4. ศึกษาวิธีการรวมทั้งสองเทคนิคเข้าด้วยกัน

5. ดำเนินการทดลองเพื่อเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีพื้นฐาน

6. ประเมินและวิเคราะห์ผลการทดลอง

บทที่ 2
ทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้อง

บทนี้จะอธิบายถึงทฤษฎีพื้นฐานที่จำเป็นและสรุปถึงปัญหาที่เกิดขึ้น โดยเริ่มต้นที่การอธิบายถึงระบบให้การแนะนำ วิธีการหรือเทคนิคที่นำมาใช้ได้แก่วิธี Collaborative Filtering (CF) และวิธี Content-Based Filtering (CBF) รวมถึงการรวมสองวิธีนี้เข้าด้วยกัน

2.1 ระบบให้การแนะนำ

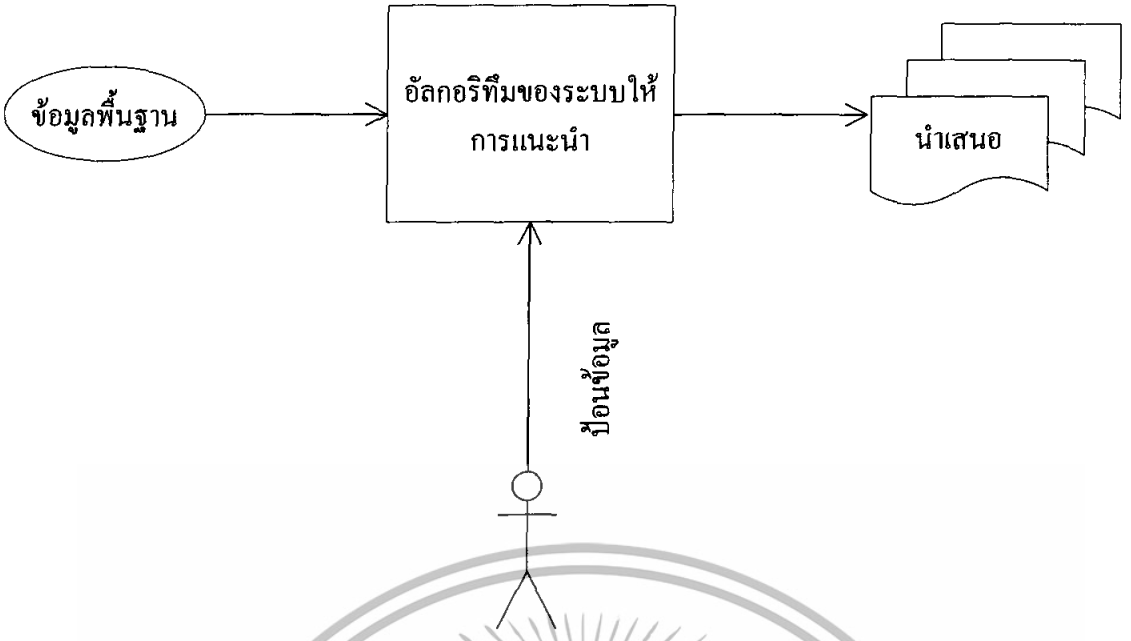
โดยทั่วไประบบให้การแนะนำประกอบไปด้วย 4 ส่วน คือ

1. ส่วนข้อมูลพื้นฐานที่ต้องใช้ในการประมวลผล เช่น โปรไฟล์ (Profile) ของผู้ใช้แต่ละคน

2. ส่วนการป้อนข้อมูล เป็นข้อมูลที่ได้มาจากการป้อนข้อมูลของผู้ใช้ เช่นการให้คะแนนเรตติ้งซึ่งมีอยู่ 2 แบบ คือ แบบชัดเจน (Explicit) และแบบไม่ชัดเจน (Implicit) เรตติ้งแบบชัดเจนจะแสดงอยู่ในรูปของจำนวนตัวเลขตามระดับความนิยมตั้งแต่ 1 ถึง 5, 1 ถึง 10 หรือระดับอื่นๆ ขึ้นอยู่กับการใช้งาน ส่วนเรตติ้งแบบไม่ชัดเจนได้มาจากพฤติกรรมการใช้งานของผู้ใช้ต่างๆ

3. ส่วนอัลกอริทึมเป็นส่วนที่สำคัญที่สุดที่ใช้ประมวลผลข้อมูลเพื่อให้การแนะนำชิ้นข้อมูลออกมา

4. ส่วนการนำเสนอคำแนะนำ มีรูปแบบอยู่ 2 รูปแบบคือ 1) Top N Recommendation โดยจะนำเสนอชิ้นข้อมูล N ชิ้นที่ตรงกับความต้องการของผู้ใช้มากที่สุด 2) Predicted Value ระบบจะทำเสนอชิ้นข้อมูลพร้อมทั้งแสดงข้อมูลเรตติ้งที่ระบบได้ทำนายเอาไว้



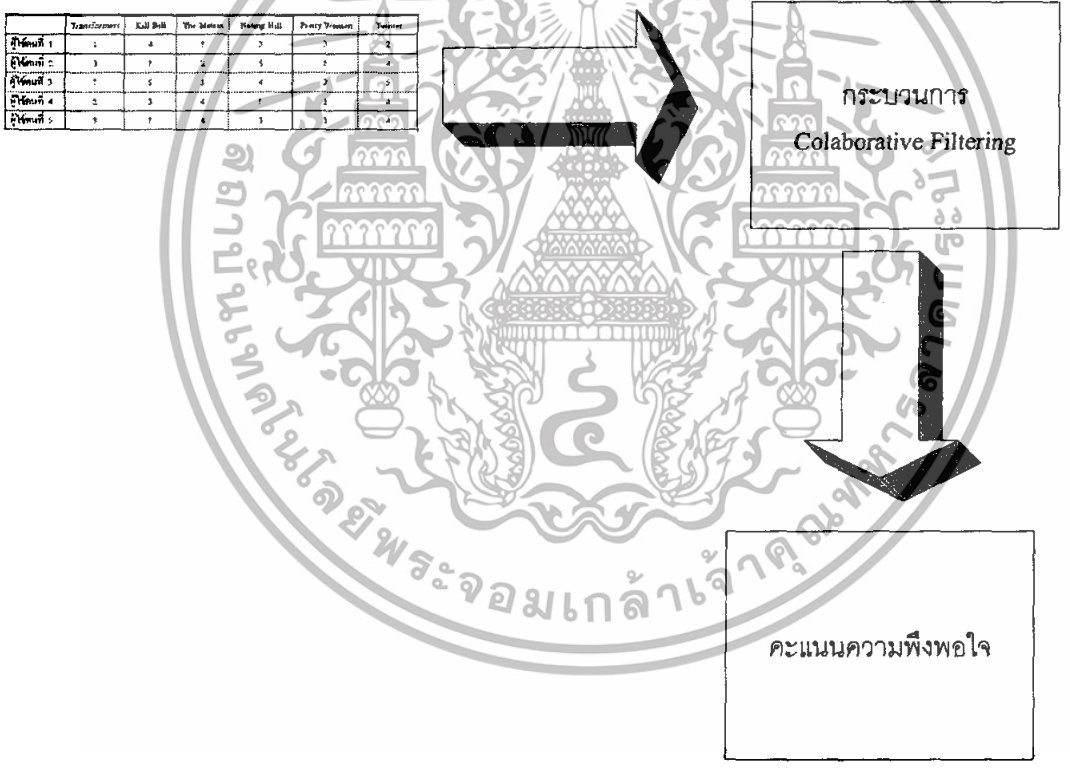
รูปที่ 2.1 รูปแสดงพื้นฐานของระบบให้การแนะนำ

จากรูปที่ 2.1 เป็นระบบให้การแนะนำชิ้นข้อมูลให้กับผู้ใช้โดยที่ระบบจะทำการนำเสนอชิ้นข้อมูลที่คาดว่าผู้ใช้จะให้ความสนใจ หรือ คาดว่าเป็นชิ้นข้อมูลที่ผู้ใช้ต้องการหรือมองหาอยู่ ซึ่งเป็นการสร้างความพึงพอใจแก่ผู้ใช้ให้ทางอ้อมอีกวิธีหนึ่ง แต่เดิมที่วิธีการที่นำมาใช้ในระบบให้การแน่นำที่เป็นที่นิยมมีอยู่ 2 วิธีด้วยกัน คือ Collaborative Filtering และ Content-Based Filtering ซึ่งวิธี CBF นั้นเป็นวิธีที่ให้ความสนใจในเนื้อหาข้อมูลเป็นหลัก โดยจะสนใจว่าชิ้นข้อมูลนั้นมีเนื้อหาตรงกับที่ผู้ใช้ต้องการหรือไม่ ซึ่งถ้าใช่ก็จะนำเสนอชิ้นข้อมูลให้ผู้ใช้ทันที แต่ถ้าไม่ตรงก็จะไม่ถูกนำเสนอออกมา ถึงแม้อาจจะเป็นชิ้นข้อมูลที่ผู้ใช้ต้องการก็ตาม ต่อมาได้มีการนำวิธี CF ที่ใช้ในการทำนายหาค่าความพึงพอใจที่ได้มาใช้ในการพิจารณาหาชิ้นข้อมูลที่จะนำเสนอให้กับผู้ใช้ สำหรับการทำนายค่าความพึงพอใจนั้น สามารถคำนวณได้จากคะแนนที่ผู้ใช้ได้เคยให้คะแนนกับชิ้นข้อมูลต่างๆซึ่งเรียกว่า เรตติ้ง ซึ่งสามารถนำไปคำนวณร่วมกับเรตติ้งของผู้ใช้ที่มีความมีความคล้ายคลึงกันได้ จากนั้นนำค่าความพึงพอใจของผู้ที่มีความคล้ายคลึงกันมาเป็นส่วนร่วมในการพิจารณาค่าความพึงพอใจของผู้ใช้แต่ละคนแตกต่างไปตามรสนิยมของผู้ใช้แต่ละคนนั้น เมื่อทำนายค่าเสร็จแล้วระบบจะทำการนำเสนอผลการแนะนำออกมา

2.2 เทคนิคในการให้คำแนะนำ

2.2.1 Collaborative Filtering

CF เป็นวิธีการหาเพื่อนบ้านที่มีลักษณะการให้คะแนนเรตติ้งใกล้เคียงกันกับผู้ใช้มาเพื่อใช้ในการทำนายเรตติ้งให้กับชิ้นข้อมูลที่ผู้ใช้ยังไม่ได้ให้เรตติ้งโดยการนำข้อมูลการให้คะแนนเรตติ้งในอดีตมาเปรียบเทียบกับกลุ่มของผู้ใช้ที่มีลักษณะการให้คะแนนเรตติ้งใกล้เคียงกันเพื่อที่จะทำนายและนำเสนอชิ้นข้อมูลที่คาดว่าผู้ใช้จะสนใจมาแนะนำให้กับผู้ใช้ตามความสนใจของผู้ใช้ซึ่งมีวิธีการดังต่อไปนี้ เริ่มต้นด้วยการหาเพื่อนบ้าน(Neighborhood) ที่มีความคิดเห็นใกล้เคียงกันซึ่งในปัจจุบันมีวิธีที่นิยมใช้ในการคำนวณหาอยู่ 2 วิธี ด้วยกันคือ Cosine Similarity และ Pearson correlation ทั้งสองวิธีนี้เป็นวิธีการหาค่าความคล้ายคลึงของทั้งคู่ ต่างกันเพียงแค่สิ่งที่นำมาคำนวณในสมการ เราหาค่าความสัมพันธ์เพื่อที่จะรู้ได้ว่าเพื่อนบ้านมีความคล้ายคลึงกับผู้ใช้มากเพียงใด จากนั้นเลือก กลุ่มผู้ใช้มาจำนวนหนึ่งเพื่อทำการทำนายชิ้นข้อมูลที่ผู้ใช้ยังไม่ได้ทำการให้เรตติ้ง



รูปที่ 2.2 รูปแสดง กระบวนการทำงานของ Collaborative Filtering

จากรูปที่ 2.2 แสดงกระบวนการทำงานของ Collaborative Filtering มี 3 ขั้นตอน คือ 1. อินพุท 2. กระบวนการอัลกอริทึม และ 3. ส่วนผลลัพธ์

ขั้นตอนที่ 1 อินพุท เป็นข้อมูลที่ใช้ในระบบอยู่ในรูปของเมตริกซ์ของคะแนนเรตติ้งของผู้ใช้ส่งเข้ามาในระบบ ดังตัวอย่าง ในตารางที่ 2.1 เป็นตัวอย่างของการให้คะแนนความชอบของภาพยนตร์ที่เกิดจากผู้ใช้ 5 คน ต่อภาพยนตร์ 6 เรื่อง ซึ่งแต่ละช่องคือคะแนนเรตติ้งที่ผู้ใช้ให้แก่ภาพยนตร์เรื่องนั้น ช่องที่ไม่มีคะแนนเรตติ้งคือ ภาพยนตร์ที่ผู้ใช้ยังไม่ได้ทำการให้เรตติ้ง

ตารางที่ 2.1 เมตริกซ์ของผู้ใช้

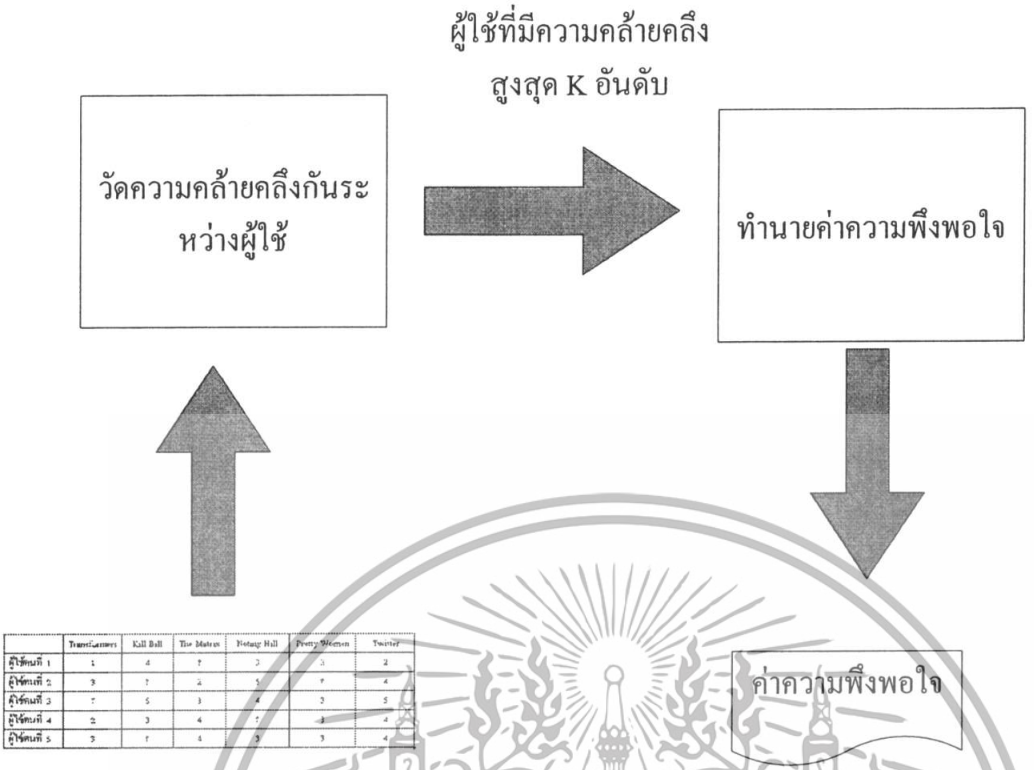
	Transformers	Kill Bill	The Matrix	Noting Hill	Pretty Women	Twister
ผู้ใช้คนที่ 1	1	4	?	3	3	2
ผู้ใช้คนที่ 2	3	?	2	5	?	4
ผู้ใช้คนที่ 3	?	5	3	4	3	5
ผู้ใช้คนที่ 4	2	3	4	?	3	4
ผู้ใช้คนที่ 5	3	?	4	3	3	4

ขั้นตอนที่ 2 เข้าสู่กระบวนการประมวลผลอัลกอริทึมเป็นขั้นตอนที่สำคัญที่สุดของเทคนิค Collaborative Filtering หลังจากที่ รับข้อมูลเข้ามาแล้วก็จะนำข้อมูลที่ได้มาคำนวณค่าพึงพอใจที่สูญหายไป (Missing Values) ซึ่งก็คือชิ้นข้อมูลที่ยังไม่ผ่านการให้เรตติ้ง

ขั้นตอนที่ 3 เป็นขั้นตอนการแสดงผลที่ได้จากการทำนายค่าความพึงพอใจที่คาดว่าผู้ใช้มีต่อชิ้นข้อมูลเป้าหมายโดยจะพิจารณาข้อมูลเป็นชิ้นๆ ไปโดยมีรูปแบบการแสดงผล 2 รูปแบบคือ Top N Recommendation และ Predicted Value

2.2.1.1 การค้นหาเพื่อนบ้านที่มีลักษณะใกล้เคียงกัน

การที่จะค้นหาเพื่อนบ้านนั้นเริ่มต้นด้วยการนำข้อมูลของผู้ใช้เป้าหมายไปเปรียบเทียบกับผู้ใช้ที่มีอยู่ในระบบเพื่อค้นหาผู้ใช้ที่มีลักษณะใกล้เคียงหรือคล้ายคลึงกับผู้ใช้เป้าหมายมากที่สุด เริ่มต้นด้วยการนำชิ้นข้อมูลที่ผู้ใช้เป้าหมายและผู้ใช้อื่นมาเปรียบเทียบต่างให้คะแนนความชอบชิ้นข้อมูลชิ้นนั้นซึ่งเรียกว่า โคเรต (Co-rated) ดังแสดงในรูป 2.4 โดยการคำนวณความคล้ายคลึงกันของผู้ใช้นั้นจะนำเฉพาะชิ้นข้อมูล ที่เป็น โคเรต กันเท่านั้น ซึ่งสามารถหาด้วยวิธี Cosine correlation และ Pearson correlation จากรูป เราจะนำเฉพาะโคเรตมาคำนวณหาค่าความคล้ายคลึงระหว่างผู้ใช้ทั้งสองโดยกำหนดให้ m เป็นจำนวนของผู้ใช้ $U = \{u_1, u_2,\ldots, u_m\}$ และ n เป็นจำนวนของชิ้นข้อมูล $I = \{i_1, i_2,\ldots, i_n\}$ โดยที่ผู้ใช้แต่ละคนแทนด้วย $u_i; i = 1,2,\ldots,m$



รูปที่ 2.3 อัลกอริทึม Collaborative Filtering

จากรูปที่ 2.3 อัลกอริทึมของ CF จะหาว่ามีชิ้นข้อมูลใดบ้างที่ผู้ใช้ t และ c ต่างให้เรตติ้งทั้งสองคน (โคเรต) จากรูปจะเห็นได้ว่ามี ชิ้นข้อมูลเพียง 4 ชิ้นเท่านั้นที่ให้โคเรตกับผู้ใช้ t และผู้ใช้ c ร่วมกัน คือ ชิ้นข้อมูล ที่ 2,3,5 และ n-1 ตามลำดับ หลังจากนั้นจะนำโคเรตจากชิ้นข้อมูลทั้ง 4 ชิ้น ดังกล่าวมาคำนวณหาความคล้ายคลึงระหว่างผู้ใช้ t และ c ตามลำดับ จากวิธีที่กล่าวข้างต้น 2 วิธี ดังนี้

1) การคำนวณหาค่าความคล้ายคลึงด้วยวิธี Cosine correlation

การคำนวณหาค่าความคล้ายคลึงด้วยวิธี Cosine correlation สามารถอธิบายได้ตามสมการที่ 2.1

$$\mathrm{RSim}(t,c)= \frac{\sum_{i \in I} R_{i,t} * R_{i,c}}{\sqrt{\sum_{i \in I} R_{i,t}^{2}} \sqrt{\sum_{i \in I} R_{i,c}^{2}}} \qquad (2.1)$$

โดยที่

RSim(t,c)	คือ	ค่าความคล้ายคลึงจากโคเรตระหว่างผู้ใช้ t และ c
$R_{i,t}$ และ $R_{i,c}$	คือ	เรตติ้งที่ผู้ใช้ t และ c มีต่อชิ้นข้อมูล i
t	คือ	ผู้ใช้เป้าหมาย
c	คือ	ผู้ใช้เปรียบเทียบ

ดังนั้นจากรูปสามารถใช้วิธี Cosine คำนวณได้ดังนี้

$$\text{RSim(t,c)} = \frac{(2*3)+(5*4)+(3*3)+(5*5)}{\sqrt{2^2+5^2+3^2+5^2}\sqrt{3^2+4^2+3^2+5^2}}$$
$$= \frac{60}{\sqrt{63}\sqrt{59}}$$
$$= 0.98$$

2) การคำนวณหาค่าความคล้ายคลึงด้วยวิธี Pearson correlation

การคำนวณหาค่าความคล้ายคลึงด้วยวิธี Pearson correlation อธิบายได้ในสมการที่ 2.2

$$\text{RSim(t,c)} = \frac{\sum_{i\in I}(R_{i,t}-\bar{R}_t)(R_{i,c}-\bar{R}_c)}{\sqrt{\sum_{i\in I}(R_{i,t}-\bar{R}_t)^2}\sqrt{\sum_{i\in I}(R_{i,c}-\bar{R}_c)^2}} \quad (2.2)$$

โดยที่

$\bar{R}_t$ และ $\bar{R}_c$ คือค่าเฉลี่ยเรตติ้งของ ผู้ใช้ t และ c

ดังนั้นจากรูป สามารถใช้วิธี Pearson correlation คำนวณได้ดังนี้

เนื่องจาก $\bar{R}_t$ คือ 3.75 $\bar{R}_c$ คือ 3.75 ดังนั้น

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

$$= \frac{(-1.75)(-0.75)+(1.25)(0.25)+(-0.75)(-0.75)+(1.25)(1.25)}{\sqrt{(-1.75)^2+(1.25)^2+(-0.75)^2+(1.25)^2}\sqrt{(-0.75)^2+(0.25)^2+(-0.75)^2+(1.25)^2}}$$

$$= \frac{1.31 + 0.31 + 056 + 1.56}{\sqrt{6.75}\sqrt{2.75}}$$

$$= 0.87$$

2.2.1.2 การทำนายค่าความพึงพอใจ

เมื่อเราได้ผู้ใช้ที่มีลักษณะคล้ายคลึงกันแล้ว จากนั้นเราจะนำผู้ใช้ที่มีความคล้ายคลึงที่สุด k คน มาทำนายค่าความพึงพอใจ ด้วยเทคนิค weighted sum ตามสมการ 2.3

$$P_{i,t} = \frac{\sum_{k \in K} (RSim(t,k) * R_{i,k})}{\sum_{k \in K} (|RSim(t,k)|} \qquad (2.3)$$

โดยที่ $P_{i,t}$ คือ ค่าความพึงพอใจที่คาดว่าชิ้นข้อมูล i มีต่อ ผู้ใช้เป้าหมาย t

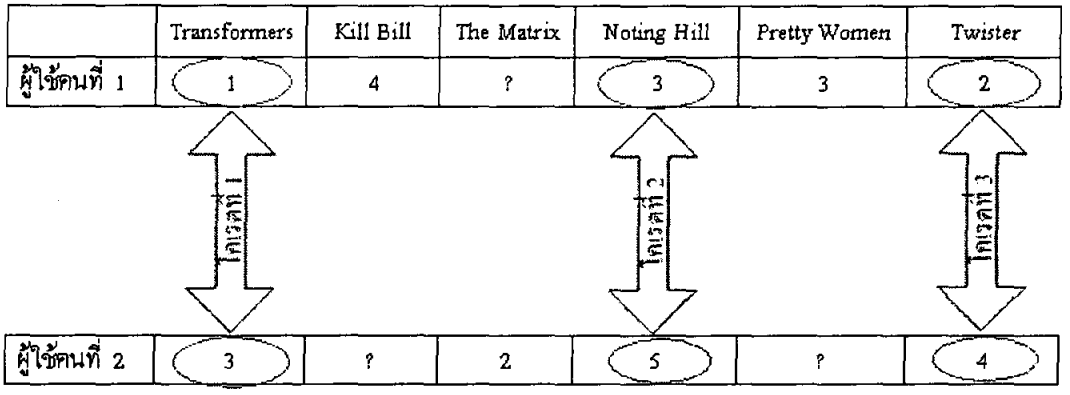
2.2.1.3 ตัวอย่างการทำงานของอัลกอริทึม Collaborative Filtering

จากตารางเมตริกซ์ในตารางที่ 2.1 ที่ผ่านมา กำหนดให้ผู้ใช้คนที่ 2 และ ชิ้นข้อมูลที่ 5 (Pretty Women) เป็นผู้ใช้เป้าหมายและชิ้นข้อมูลเป้าหมายตามลำดับ เป้าหมายของอัลกอริทึม Collaborative filtering คือทำนายว่าผู้ใช้คนที่ 2 จะมีความพึงพอใจในชิ้นข้อมูลที่ 5 เป็นเท่าใด

ตารางที่ 2.2 เป้าหมายของอัลกอริทึม Collaborative Filtering

	Transformers	Kill Bill	The Matrix	Noting Hill	Pretty Women	Twister
ผู้ใช้คนที่ 1	1	4	?	3	3	2
ผู้ใช้คนที่ 2	3	?	2	5	?	4
ผู้ใช้คนที่ 3	?	5	3	4	3	5
ผู้ใช้คนที่ 4	2	3	4	?	3	4
ผู้ใช้คนที่ 5	3	?	4	3	3	4

เริ่มต้นอัลกอริทึม CF จะทำการค้นหาผู้ใช้ที่มีความคล้ายคลึง โดยนำข้อมูลของผู้ใช้คนที่ 2 ไปเปรียบเทียบกับทุกผู้ใช้ที่ได้มีการให้เรตติ้งชิ้นข้อมูลที่ 5 ไว้ ได้แก่ ผู้ใช้คนที่ 1, 3, 4 และ 5) ดังนั้น ในขั้นตอนนี้จะทำการเปรียบเทียบกันระหว่างผู้ใช้คนที่ 2 กับ 1, 2 กับ 3, 2 กับ 4, 2 กับ 5 ตามลำดับ



รูปที่ 2.4 การคำนวณหาความคล้ายคลึงจากโคเรตระหว่างผู้ใช้คนที่ 2 และ คนที่ 1

จากรูปที่ 2.4 สามารถคำนวณหาความคล้ายคลึงด้วยวิธี pearson ได้ดังนี้

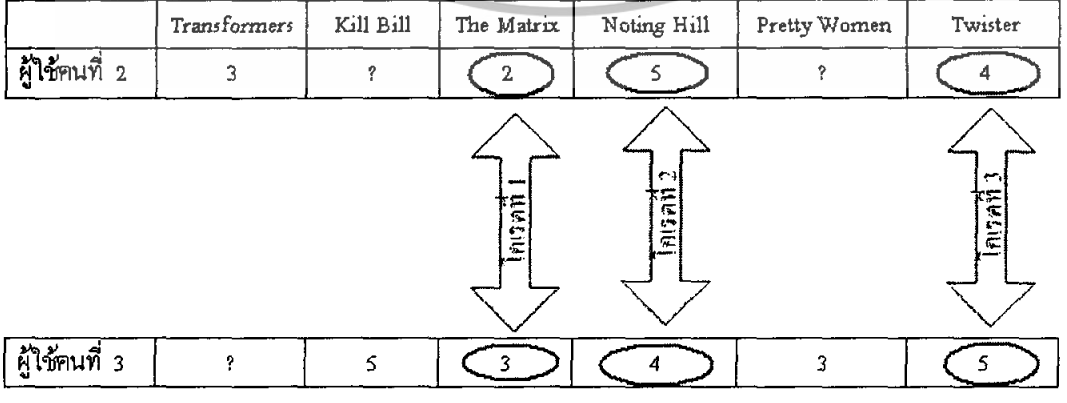
เนื่องจาก $\overline{R}_t$ คือ 4 $\overline{R}_c$ คือ 2 ดังนั้น

$$\text{RSim}(t_2,c_1) = \frac{(3-4)(1-2)+(5-4)(3-2)+(4-4)(2-2)}{\sqrt{(3-4)^2+(5-4)^2+(4-4)^2}\ \sqrt{(1-2)^2+(3-2)^2+(2-2)^2}}$$

$$= \frac{(-1)(-1)+(1)(1)+(0)(0)}{\sqrt{(-1)^2+(1)^2+(0)^2}\ \sqrt{(-1)^2+(-1)^2+(0)^2}}$$

$$= \frac{1+1+0}{\sqrt{2}\sqrt{2}}$$

$$= 0.71$$



รูปที่ 2.5 การคำนวณหาความคล้ายคลึงจากโคเรตระหว่างผู้ใช้คนที่ 2 และ คนที่ 3

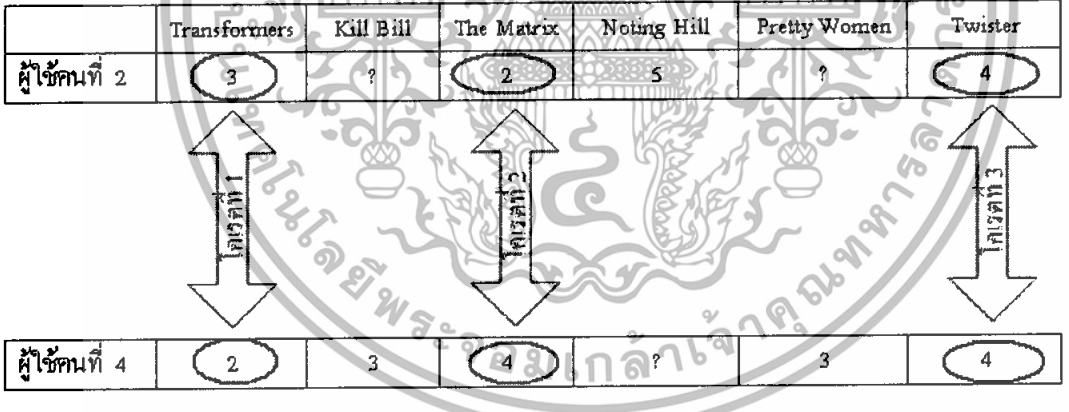
จากรูปที่ 2.5 สามารถคำนวณหาความคล้ายคลึงด้วยวิธี pearson ได้ดังนี้
เนื่องจาก $\overline{R}_t$ คือ 3.67 $\overline{R}_c$ คือ 4 ดังนั้น

$$\text{RSim}(t_2,c_3) = \frac{(2-3.67)(3-4)+(5-3.67)(4-4)+(4-3.67)(5-4)}{\sqrt{(2-3.67)^2+(5-3.67)^2+(4-3.67)^2}\ \sqrt{(3-4)^2+(4-4)^2+(5-4)^2}}$$

$$= \frac{(-1.67)(-1)+(1.33)(0)+(0.33)(1)}{\sqrt{(-1.67)^2+(1.33)^2+(0.33)^2}\ \sqrt{(-1)^2+(0)^2+(1)^2}}$$

$$= \frac{1.67+0.33}{\sqrt{4.67}\sqrt{2}}$$

$$= 0.65$$



รูปที่ 2.6 การคำนวณหาความคล้ายคลึงจากโคเรตระหว่างผู้ใช้คนที่ 2 และ คนที่ 4

จากรูปที่ 2.6 สามารถคำนวณหาความคล้ายคลึงด้วยวิธี pearson ได้ดังนี้
เนื่องจาก $\overline{R}_t$ คือ 3 $\overline{R}_c$ คือ 3.33 ดังนั้น

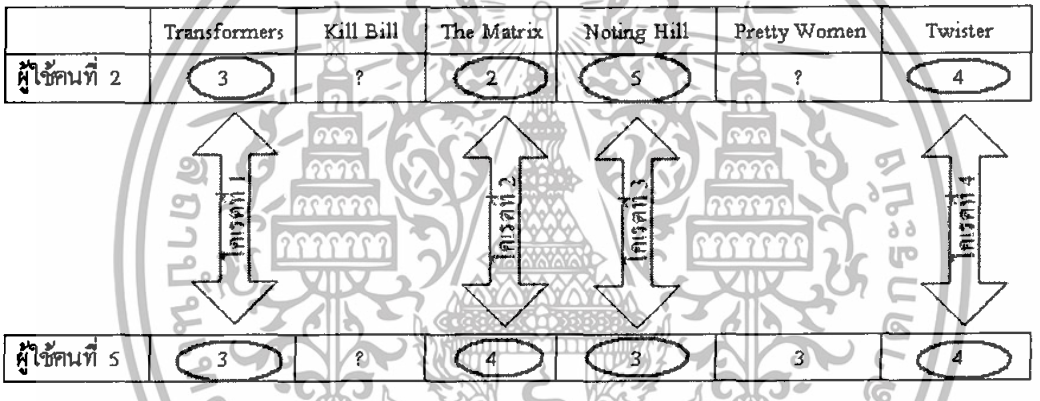
$RSim(t_2,c_4)$ =

$$\frac{(3-3)(2-3.33)+(4-3)(4-3.33)+(2-3)(4-3.33)}{\sqrt{(3-3)^2+(4-3)^2+(2-3)^2}\ \sqrt{(2-3.33)^2+(4-3.33)^2+(4-3.33)^2}}$$

$$= \frac{(0)(-1.33)+(1)(0.67)+(-1)(0.67)}{\sqrt{(0)^2+(1)^2+(-1)^2}\ \sqrt{(-1.33)^2+(0.67)^2+(0.67)^2}}$$

$$= \frac{0+0.67-0.67}{\sqrt{2}\sqrt{2.67}}$$

$$= 0$$



รูปที่ 2.7 การคำนวณหาความคล้ายคลึงจากโคเรตระหว่างผู้ใช้คนที่ 2 และ คนที่ 5

จากรูปที่ 2.7 สามารถคำนวณหาความคล้ายคลึงด้วยวิธี pearson ได้ดังนี้

เนื่องจาก $\overline{R}_t$ คือ 3.5 $\overline{R}_c$ คือ 3.5 ดังนั้น

$RSim(t_2,c_5)$ =

$$\frac{(3-3.5)(3-3.5)+(2-3.5)(4-3.5)+(5-3.5)(3-3.5)+(4-3.5)(4-3.5)}{\sqrt{(3-3.5)^2+(2-3.5)^2+(5-3.5)^2+(4-3.5)^2}\ \sqrt{(3-3.5)^2+(4-3.5)^2+(3-3.5)^2+(4-3.5)^2}}$$

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

$$= \frac{1}{\sqrt{5}\sqrt{1}}$$

$$= 0.45$$

จากการคำนวณหาค่าความคล้ายคลึงข้างต้นด้วยวิธี Pearson correlation ทำให้ได้ค่าความคล้ายคลึงจากการเปรียบเทียบทั้งหมดดังนี้

ค่าความคล้ายคลึงระหว่าง ใช้คนที่ 2 กับ 1 เท่ากับ 0.71
ค่าความคล้ายคลึงระหว่าง ใช้คนที่ 2 กับ 3 เท่ากับ 0.65
ค่าความคล้ายคลึงระหว่าง ใช้คนที่ 2 กับ 4 เท่ากับ 0
ค่าความคล้ายคลึงระหว่าง ใช้คนที่ 2 กับ 5 เท่ากับ 0.65

หลังจากนั้นจะเป็นขั้นตอนของการนำผู้ใช้ที่มีลักษณะการให้เรตติ้งคล้ายคลึงกับผู้ใช้เป้าหมายมากที่สุดจำนวน k คน มาทำนายค่าความพึงพอใจ ด้วยเทคนิค weight sum ตามสมการที่ 2.3 ในที่นี้กำหนดให้ขนาดของผู้ใช้ที่ใกล้เคียง (K) มีขนาดเท่ากับ 2 ดังนั้นจึงนำเฉพาะผู้ใช้คนที่ 1 และ 3 ซึ่งมีค่าความคล้ายคลึงกับผู้ใช้เป้าหมายมากที่สุด 2 อันดับแรกมาคำนวณ ได้ดังนี้

$$K = \{c_1, c_3\}$$

$$P_{u_2,i_5} = \frac{(RSim(t_2,c_1) * R_{u_1 c5}) + (RSim(t_2,c_3) * R_{u_3 c5})}{\left|(RSim(t_2,c_1) + (RSim(t_2,c_3)\right|}$$

แทนค่า

$$P_{u_2,i_5} = \frac{(0.71*3)+(0.65)*3)}{\left|0.71+0.65\right|}$$

$$P_{u_2,i_5} = 3$$

ดังนั้นผลลัพธ์จากการทำนายสรุปได้ว่า ผู้ใช้คนที่ 2 น่าจะมีความพึงพอใจต่อชิ้นข้อมูลชิ้นที่ 5 หรือว่าภาพยนตร์ Pretty Women เท่ากับ 3 เป็นอันเสร็จสิ้นกระบวนการ Collaborative Filtering

2.2.1.4 ปัญหาของอัลกอริทึม Collaborative Filtering

เนื่องจากอัลกอริทึม Collaborative Filtering อาศัยโคเรตในการค้นหาผู้ใช้ที่มีลักษณะใกล้เคียงกัน จึงเป็นสาเหตุสำคัญให้เกิดปัญหาสำคัญ หลายประการด้วยกัน ดังต่อไปนี้

1. ปัญหาการให้เรตติ้งต่อชิ้นข้อมูล (Sparsity Problem) เป็นปัญหาที่เกิดจากจำนวนชิ้นข้อมูลที่เพิ่มขึ้นมากจนส่งผลให้ผู้ใช้ไม่สามารถให้คะแนนได้อย่างทั่วถึง เช่นผู้ใช้อาจจะให้เรตติ้งไม่ถึง 0.1 % จากข้อมูลทั้งหมด 1 ล้านชิ้น กล่าวคือ ผู้ใช้คนหนึ่งอาจให้คะแนนเรตติ้งเพียงแค่ 1 พันเรตติ้ง หรือน้อยกว่านั้นทำให้เรตติ้งที่ผู้ใช้ให้ไว้ไม่เพียงพอสำหรับการคำนวณหาค่าความคล้ายคลึงระหว่างผู้ใช้ ดังแสดงในรูป ในกรอบสี่เหลี่ยมมน แสดงถึงการให้เรตติ้งที่ไม่ทั่วถึงของผู้ใช้ 5 คนต่อชอ้นข้อมูล 6 ชิ้น

ตารางที่ 2.3 การใส่ข้อมูลเรตติ้งไม่ทั่วถึง

	Transformers	Kill Bill	The Matrix	Noting Hill	Pretty Women	Twister
ผู้ใช้คนที่ 1	2	3	4	-	-	-
ผู้ใช้คนที่ 2	4	-	?	2	-	-
ผู้ใช้คนที่ 3	-	4	5	-	-	-
ผู้ใช้คนที่ 4	2	5	-	5	-	-
ผู้ใช้คนที่ 5	-	-	5	-	-	-

จากตารางจะเห็นว่าเกิด ปัญหา Sparsity ขึ้น โดยอัลกอริทึม Collaborative Filtering ทำนายว่าผู้ใช้คนที่ 2 จะมีความพึงพอใจต่อ ชิ้นข้อมูลที่ 3 (The Matrix) เป็นเท่าใด ในขั้นแรกอัลกอริทึมจะทำการนำ ผู้ใช้เป้าหมายไปเปรียบเทียบกับผู้ใช้ทั้งหมดที่เคยให้เรตติ้งชิ้นข้อมูลที่ 4 ไว้(ผู้ใช้คนที่ 1, 3, และ 5) เพื่อหาผู้ใช้ที่มีความคล้ายคลึงกับผู้ใช้เป้าหมายมากที่สุดแต่ปรากฏว่ามีเพียงโคเรตเดียวส่วนที่เหลือคือข้อมูลเรตติ้งที่ขาดหายไป ดังนั้นจึงไม่สามารถหาความคล้ายคลึงระหว่าง ผู้ใช้คนที่ 2 กับ 3 และ 2 กับ 5 ได้เลย เนื่องจากปัญหาการให้เรตติ้งที่ไม่ทั่วถึง

	Transformers	Kill Bill	The Matrix	Noting Hill	Pretty Women	Twister
ผู้ใช้คนที่ 1	2	3	4	-	-	-
ผู้ใช้คนที่ 2	4	-	?	2	-	-
ผู้ใช้คนที่ 3	-	4	5	-	-	-
ผู้ใช้คนที่ 4	2	5	-	5	-	-
ผู้ใช้คนที่ 5	-	-	5	-	-	-

รูปที่ 2.8 ปัญหาการให้เรตติ้งต่อชิ้นข้อมูล (Sparsity Problem)

2. ปัญหาการแยกแยะเรตติ้ง (Transparency Problem) ระบบนั้นจะไม่สามารถแยกแยะการให้เรตติ้งได้ว่าเหตุผลที่ผู้ใช้ให้คะแนนความพึงพอใจคืออะไร เช่น เมื่อผู้ใช้ 2 คน ใส่ค่าความพึงพอใจให้กับภาพยนตร์เรื่อง *Titanic* ผู้ใช้คนหนึ่งอาจจะให้คะแนน 5 ด้วยเหตุผลคือ ชอบที่ตัวเนื้อเรื่องของภาพยนตร์ ส่วนผู้ใช้อีกคนอาจจะให้ 5 คะแนน เนื่องจากชอบในตัวนักแสดงนำของเรื่อง ซึ่งเมื่อนำมาเข้าอัลกอริทึมในการทำนายภาพยนตร์ที่ยังไม่ผ่านการให้เรตติ้งแล้วจะทำให้ไม่มีประสิทธิภาพเท่าที่ควร

3. ปัญหาชิ้นข้อมูลที่ไม่มีการให้เรตติ้งไว้ (First-rater Problem) เป็นปัญหาที่เกิดจากชิ้นข้อมูลใหม่หรือชิ้นข้อมูลที่ยังไม่มีผู้ใช้คนใดเคยให้เรตติ้งมาก่อน ทำให้ชิ้นข้อมูลชิ้นนั้นไม่สามารถนำมาเปรียบเทียบกับชิ้นข้อมูลใดๆได้เลยดังนั้นจึงไม่สามารถนำมาคำนวณหาค่าความพึงพอใจได้ ตัวอย่างของปัญหาของระบบแนะนำภาพยนตร์ก็คือ ภาพยนตร์เรื่องใหม่ที่ยังไม่ได้ฉายหรือยังไม่มีผู้ใช้คนใดทำการให้เรตติ้งจะไม่มีโอกาสที่จะถูกหยิบออกมาแนะนำให้กับผู้ใช้คนใดได้เลย ซึ่งปัญหานี้สามารถแก้ไขได้ด้วยอัลกอริทึมแบบ Content-Based Filtering

	Transformers	Kill Bill	The Matrix	Noting Hill	Pretty Women	Twister
ผู้ใช้คนที่ 1	2	3	4	-	-	-
ผู้ใช้คนที่ 2	4	-	?	2	-	-
ผู้ใช้คนที่ 3	-	4	5	-	-	-
ผู้ใช้คนที่ 4	2	5	-	5	-	-
ผู้ใช้คนที่ 5	-	-	5	-	-	-

รูปที่ 2.9 ปัญหาชิ้นข้อมูลที่ไม่มีการให้เรตติ้งไว้ (First-rater Problem)

2.2.2 Content-Based Filtering

Content-Based Filtering เป็นอัลกอริทึมอีกอันหนึ่งที่เป็นที่นิยมในระบบให้คำแนะนำ ซึ่งปกติจะนิยมใช้ในการค้นคืนข้อมูล ที่ให้ความสนใจกับคุณภาพของเนื้อหาหรือคุณสมบัติของชิ้นข้อมูลโดยจะตรวจสอบว่าชิ้นข้อมูลนั้นมีเนื้อหาตรงกับโปรไฟล์ของผู้ใช้หรือไม่ถ้าคล้ายคลึงก็จะหยิบมานำเสนอทันที แต่ถ้าไม่ใช่ จะไม่สนใจถึงแม่ว่าชิ้นข้อมูลชิ้นนั้นอาจจะมีลักษณะใกล้เคียงกับที่ผู้ใช้ต้องการก็ตาม

ดังนั้นวิธีการนี้เป็นการคำนวณหาค่าความคล้ายคลึงระหว่างชิ้นข้อมูลกับโปรไฟล์ของผู้ใช้ โดยการจับคู่ชิ้นข้อมูลกับโปรไฟล์ของผู้ใช้ เพื่อค้นหาข้อมูลที่ผู้ใช้สนใจโดยใช้เวคเตอร์สเปซโมเดล แสดงข้อมูลโปรไฟล์ผู้ใช้ในรูปเมตริกซ์แต่ละแถวหมายถึงคุณสมบัติ แต่ละหลักคือ ชิ้นข้อมูล ดังนั้นในแต่ละเซลจะหมายถึงความถี่ที่ปรากฏเทอมในชิ้นข้อมูลนั้นหลังจากนั้นนำความถี่มาคำนวณหาค่าความคล้ายคลึงระหว่างชิ้นข้อมูลกับโปรไฟล์ของผู้ใช้

ที่ผ่านมาเป็นสาเหตุที่ Content-Based Filtering นั้น ไม่ประสบปัญหาการให้เรตติ้งของข้อมูลไม่ทั่วถึง (Sparsity) และปัญหาชิ้นข้อมูลใหม่ที่ยังไม่มีผู้ใช้ทำการให้เรตติ้ง (First Rater) กล่าวคือ ถ้าชิ้นข้อมูลนั้นมีความคล้ายคลึงกับโปรไฟล์ของผู้ใช้ ชิ้นข้อมูลนั้นก็มีโอกาสที่จะถูกนำเสนอออกมาถึงแม้ว่าจะไม่มีผู้ใช้คนใดเคยให้เรตติ้งก็ตาม แต่ปัญหาหลักของ Content-Based Filtering ก็คือ ชิ้นข้อมูลที่ไม่ตรงกับ โปรไฟล์ผู้ใช้จะไม่มีโอกาสที่จะถูกนำออกมานำเสนอ ถึงแม้ว่าจะเป็นชิ้นข้อมูลที่ผู้ใช้สนใจก็ตาม

2.2.3 การรวม Collaborative Filtering และ Content-Based Filtering

เนื่องด้วยทั้ง Collaborative Filtering และ Content-Based Filtering นั้นมีจุดแข็งและจุดอ่อนแตกต่างกันดังนั้นจึงเหมาะที่จะรวมสองอัลกอริทึมนี้เข้าด้วยกันจุดแข็งของ Content-Based Filtering คือไม่ประสบกับปัญหาชิ้นข้อมูลใหม่ที่ยังไม่ผู้ใช้คนใดให้เรตติ้ง(First Rater Problem) และไม่ประสบกับปัญหาความเบาบางของข้อมูล(Sparsity Problem) แต่อัลกอริทึมแบบ CBF นี้ขึ้นกับคุณสมบัติของชิ้นข้อมูลเป็นหลัก หากคุณสมบัติของชิ้นข้อมูลนั้นเป็นข้อมูลที่ผู้ใช้ไม่สนใจชิ้นข้อมูลนั้นก็จะไม่ถูกหยิบออกมานำเสนอ ส่วนวิธี Collaborative Filtering นั้นมีจุดแข็งตรงที่พิจารณาความเห็นของผู้ใช้อื่นเป็นหลักทำให้สามารถนำเสนอชิ้นข้อมูลได้หลากหลายแต่ก็ยังมีปัญหาอยู่มากดังที่ได้กล่าวไว้แล้ว ที่ผ่านมาได้มีงานวิจัยที่นำเอาข้อดีข้อเสียของทั้งสองวิธีมารวมกันเพื่อเพิ่มประสิทธิภาพให้กับระบบให้การแนะนำ

ในปี 2006 Salter และ Antonopoulos ได้นำเสนอใน Cinema Screen Recommender Agent โดยการรวมเอาทั้งสองอัลกอริทึมเข้าด้วยกัน

รูปที่ 2.10 การรวม CF กับ CBF

ในการทำงานจะแบ่งออกเป็นสองส่วน ส่วนแรกจะเป็น Collaborative Filtering ส่วนที่สองเป็น Content-Based Filtering

1) การทำงานในส่วน Collaborative Filtering

เริ่มแรกจะทำการคำนวณหาค่าความสัมพันธ์กันระหว่างผู้ใช้คนปัจจุบันกับผู้ใช้คนอื่น ด้วยวิธี pearson ได้ค่าระหว่าง -1 -1 โดย -1 คือไม่ใกล้เคียงกันเลย 1 คือ มีความใกล้เคียงกันมาก จากนั้นระบบจะทำการหาค่าเฉลี่ยของแต่ละภาพยนตร์โดย มีสูตรคือ

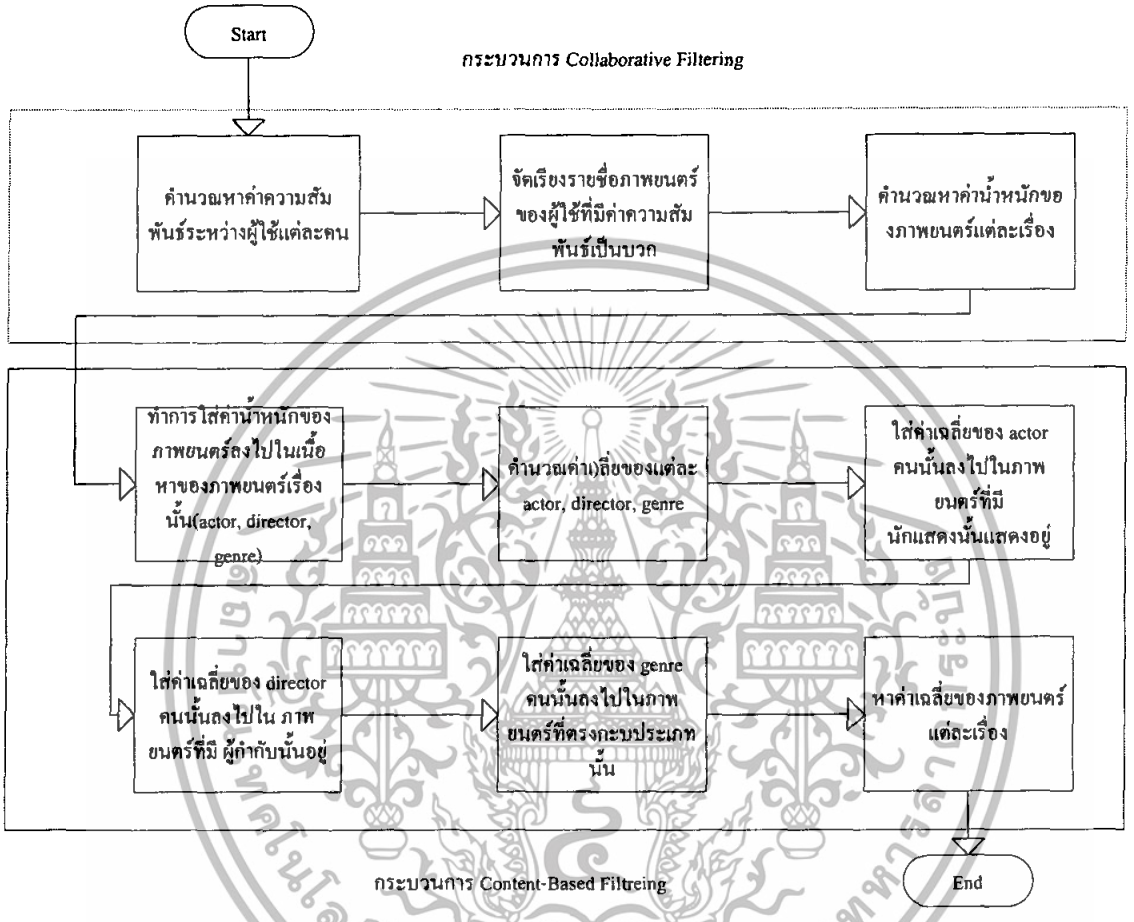
$$W_f = \frac{\sum_{p \in P} v_{p,f} xr_p}{n} \qquad (2.4)$$

โดยที่

W_f คือค่า weight ของ ภาพยนตร์เรื่อง f

$V_{p,f}$ คือ ค่าเรตติ้งที่ให้โดนผู้ใช้ p ต่อภาพยนตร์เรื่อง f

r_p คือค่าความคล้ายคลึงระหว่างผู้ใช้ p กับ ผู้ใช้ปัจจุบัน



รูปที่ 2.11 การทำงานของ CF กับ CBF

2) กระบวนการของ Content-Based Filtering

เริ่มแรกจะนำค่าน้ำหนักที่ได้ของแต่ละภาพยนตร์กระจายเข้าไปในแอตทริบิวของภาพยนตร์ ซึ่งก็คือ actor, director, genre จากนั้นระบบจะหาของแต่ละ แอตทริบิวนั้น จากนั้นนำค่าเฉลี่ยของแต่ละแอตทริบิวไปใส่ในภาพยนตร์เป้าหมายเพื่อทำนายค่าความพึงพอใจ

วิธีการนี้สามารถแก้ไขปัญหาหลักๆของทั้งสองอัลกอริทึมได้ แต่ยังมีปัญหาอยู่หลายประการด้วยกันเช่น ถ้าหากภาพยนตร์เรื่องนั้นแสดงโดยนักแสดงใหม่ที่ยังไม่เคยแสดงในเรื่องใดๆมาก่อน ค่าในส่วนของนักแสดงจะว่างทำให้ไม่สามารถคำนวณค่าความพึงพอใจในภาพยนตร์เรื่องนั้นได้และเนื่องจากเริ่มกระบวนการในส่วนของ Collaborative Filtering ซึ่งมีปัญหาสำคัญในเรื่อง Sparsity ทำให้คะแนนความคล้ายคลึงของผู้ใช้อาจคลาดเคลื่อนไปจากความเป็นจริงได้ ซึ่งเป็น

ความท้าทายและเป็นประเด็นสำคัญในการแก้ไขปัญหาดังกล่าวของทั้งสองเทคนิคให้มีประสิทธิภาพเพิ่มมากขึ้นซึ่งจะกล่าวถึงในบทถัดไป

2.3 การประเมินผล

ปัจจุบันมีวิธีวัดความถูกต้องอยู่หลายวิธีด้วยกัน สำหรับวิธีที่เป็นที่นิยมใช้กันอย่างแพร่หลายมีอยู่หลายตัวด้วยกัน

สำหรับ การวัดผลความผิดพลาดในการทำนายด้วยวิธีทางสถิติ ที่เรียกว่า Mean Absolute Error (MAE) ซึ่งวิธีนี้จะทำการวัดผลความผิดพลาดในการทำนายโดยเปรียบเทียบค่าสัมบูรณ์ของผลต่างระหว่างเรตติ้งจริงของผู้ใช้ที่เคยให้ไว้กับระบบกับเรตติ้งที่ระบบทำนายได้จากนั้นนำมาคำนวณความผิดพลาดโดยการหาค่าความผิดพลาดสัมบูรณ์เฉลี่ยหรือ MAE ของการทำนายทั้งหมด ดังสมการที่ 2.5 ค่า MAE นั้น มีค่าน้อยจะดี เพราะแสดงถึงว่าระบบสามารถทำนายได้ถูกต้องและมีประสิทธิภาพ

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

อีกเครื่องมือหนึ่งที่ใช้วัดประสิทธิภาพคือ ค่าความแม่นยำ (Precision) ค่าความระลึก (Recall) ค่า Specificity และค่า Negative Predictive Value

ค่าความแม่นยำคือ อัตราส่วนของชิ้นข้อมูลที่ถูกต้องจากชิ้นข้อมูลทั้งหมดที่แนะนำออกมา

$$\text{Precision} = \frac{|A \cap B|}{|A|} \tag{2.6}$$

ค่าความระลึก คือ อัตราส่วนระหว่างชิ้นข้อมูลที่ถูกต้องที่ถูกแนะนำออกมากับชิ้นข้อมูลที่ถูกต้องทั้งหมด

$$\text{Recall} = \frac{|A \cap B|}{|B|} \tag{2.7}$$

Specificity คือ ความน่าจะเป็นที่ข้อมูลที่ไม่ถูกต้องจะไม่ถูกแนะนำออกมา หาได้จากอัตราส่วนระหว่างชิ้นข้อมูลที่ไม่ถูกต้องที่ไม่ถูกแนะนำออกมากับชิ้นข้อมูลที่ไม่ถูกต้อง

$$\text{Specificity} = \frac{\left|\overline{A} \cap \overline{B}\right|}{\left|\overline{B}\right|} \tag{2.8}$$

Negative Predictive Value คือ ความน่าจะเป็นที่ข้อมูลที่ไม่ถูกนำเสนอจะเป็นข้อมูลที่ไม่ถูกต้อง หาได้จากอัตราส่วนระหว่างชิ้นข้อมูลที่ไม่ถูกต้องที่ไม่ถูกแนะนำออกมากับชิ้นข้อมูลที่ไม่ถูกแนะนำ

$$\text{Negative Predictive Value} = \frac{\left|\overline{A} \cap \overline{B}\right|}{\left|\overline{A}\right|} \tag{2.9}$$

โดยที่

A คือ ชิ้นข้อมูลที่ระบบแนะนำ

B คือ ชิ้นข้อมูลที่ถูกต้องทั้งหมด

$\overline{A}$ คือ ชิ้นข้อมูลที่ไม่ถูกแนะนำ

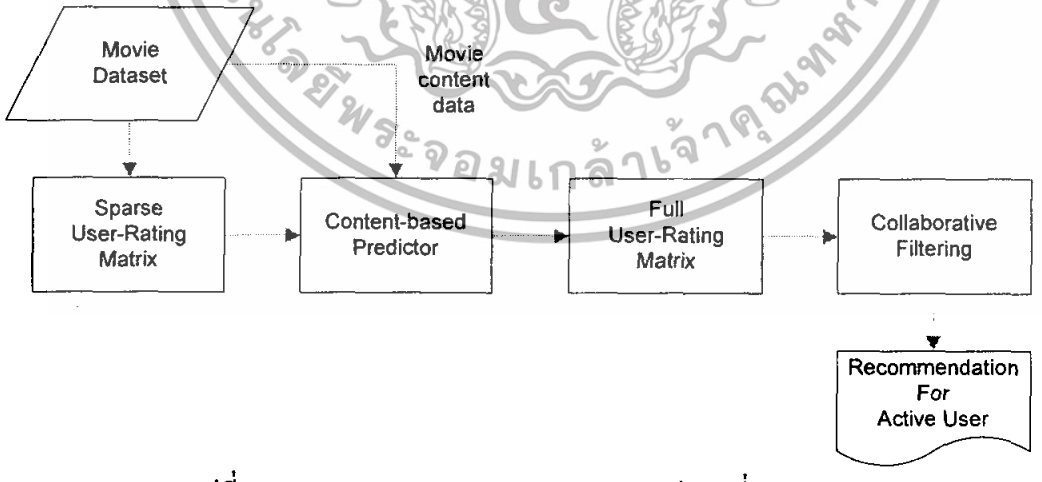
$\overline{B}$ คือ ชิ้นข้อมูลที่ไม่ถูกต้อง

บทที่ 3
การแนะนำภาพยนตร์โดยการรวม CBF กับ CF

ในบทนี้จะเสนอวิธีการในการปรับปรุงประสิทธิภาพของการแนะนำข้อมูลหรือภาพยนตร์ให้กับผู้ใช้ปัจจุบัน(Active user) โดยใช้วิธีการที่ผสมผสานระหว่างเทคนิคของ CBF เข้ากับเทคนิคของ CF ในการให้คำแนะนำ

3.1 ภาพรวมของการออกแบบวิธีการที่นำเสนอ

วิธีการที่นำเสนอนี้สามารถแบ่งออกเป็น 2 ส่วนคือ ส่วนของ Content-based predictor (Content-based Filtering หรือ CBF) และส่วนของ Collaborative Filtering (CF) โดยในส่วนของ CBF จะใช้ตัวทำนายข้อมูลแบบ Content-based หรือ Content-based predictor มาทำการปรับปรุงข้อมูลระหว่างผู้ใช้-ชิ้นงานที่มีอยู่ให้ดีขึ้น โดยแปลงเมตริกซ์ผู้ใช้-ชิ้นข้อมูลที่มีความเบาบาง (sparse) ให้กลายเป็นเมตริกซ์ผู้ใช้-ชิ้นข้อมูลเทียม (Pseudo user-rating Matrix) ซึ่งเป็นเมตริกซ์ที่มีทั้งข้อมูลเรตติ้งจริงกับข้อมูลเรตติ้งที่ได้จากการทำนายแบบ CBF ที่ไม่มีความเบาบาง (Full) จากนั้นนำข้อมูลที่ปรับปรุงแล้วมาทำงานต่อในส่วนของ CF โดยใช้เทคนิคของ CF บนข้อมูลที่ปรับปรุงแล้วในการหากลุ่มของผู้ใช้อื่นที่มีความคล้ายคลึงกับผู้ใช้ปัจจุบันสูง และทำการแนะนำรายการภาพยนตร์ที่เหมาะสมออกมาให้กับผู้ใช้ปัจจุบัน ดังรูปที่ 3.1



รูปที่ 3.1 แสดงภาพรวมของการออกแบบวิธีการที่นำเสนอ

3.2 ขั้นตอนการทำงานส่วน Content-based predictor

ประกอบไปด้วยขั้นตอนการทำงาน 3 ขั้นตอนคือ การสร้างเมตริกซ์ของผู้ใช้-ชิ้นข้อมูลจริง, ทำนายเรตติ้งด้วย Content-based predictor และการสร้างเมตริกซ์ผู้ใช้-ชิ้นข้อมูลเทียมซึ่งผสม

3.2.1 สร้างเมตริกซ์ผู้ใช้-ชิ้นข้อมูล

จากข้อมูลในฐานข้อมูลนำมาสร้างเป็นเมตริกซ์ผู้ใช้-ชิ้นข้อมูลซึ่งลักษณะเป็นตารางที่มีแถวเป็นผู้ใช้ และมีคอลัมน์เป็นชิ้นข้อมูลซึ่งก็คือชื่อเรื่องของภาพยนตร์และข้อมูลในแต่ละช่องเป็นเรตติ้งที่ผู้ใช้คนนั้นให้กับภาพยนตร์เรื่องนั้นๆ โดยเรตติ้งที่ให้มีค่าเป็นจำนวนเต็มตั้งแต่ 1-5 และเป็น null (?) ถ้าผู้ใช้ไม่เคยให้เรตติ้งกับภาพยนตร์เรื่องนั้นๆ

ตารางที่ 3.1 เมตริกซ์ผู้ใช้-ชิ้นข้อมูลที่มีความเบาบาง

	Transformers	Kill Bill	The Matrix	Nothing Hill	Pretty Women	Twister
ผู้ใช้คนที่ 1	1	4	?	3	3	2
ผู้ใช้คนที่ 2	3	?	2	5	?	4
ผู้ใช้คนที่ 3	?	5	3	4	3	?
ผู้ใช้คนที่ 4	2	3	4	?	3	4
ผู้ใช้คนที่ 5	3	?	4	3	3	4

ซึ่งเมตริกซ์ที่ได้นั้นมักจะมีความเบาบางเนื่องจากผู้ใช้แต่ละคนมีการให้เรตติ้งกับบนงานน้อยซึ่งการนำข้อมูลที่มีความเบาบางนี้ไปทำการแนะนำข้อมูลมักจะได้ข้อมูลที่มีความคลาดเคลื่อนและแม่นยำต่ำจึงต้องนำข้อมูลนี้ไปทำการปรับปรุงด้วยวิธีการดังที่จะกล่าวต่อไป

3.2.2 ทำนายเรตติ้งแบบ CBF

การทำนายค่าเรตติ้งนั้นนำข้อมูลพื้นฐานที่มี 3 ค่าด้วยกันได้แก่ คะแนนเนื้อหาของภาพยนตร์ คะแนนนักแสดง คะแนนผู้กำกับ โดยทำการแบ่งค่าเรตติ้งจาก1-5 ออกเป็น 5 คลาส คือ ค่าเรตติ้ง 1-5 โดยเรียงลำดับคะแนนจากระดับน้อยซึ่งก็คือ 1 ไปจนถึงระดับสูง ซึ่งก็คือ 5 นำข้อมูลเนื้อหาของภาพยนตร์ในฐานข้อมูลและเมตริกซ์ผู้ใช้-ชิ้นข้อมูลของแต่ละประเภทดังที่ได้กล่าวเอาไว้มาให้ classifier เรียนรู้ประวัติการให้เรตติ้งของผู้ใช้ แล้วสร้างเป็นโปรไฟล์ของผู้ใช้แต่ละคนซึ่งเก็บข้อมูลความน่าจะเป็นที่เนื้อหาต่างๆของชิ้นข้อมูลจะจัดอยู่ในคลาสต่างๆ จากนั้นนำโปรไฟล์ของผู้ใช้แต่ละคนไปทำนายเรตติ้งให้กับชิ้นข้อมูลที่ผู้ใช้นั้นๆยังไม่มีการให้เรตติ้ง โดยใช้ naïve Bayes Theorem ซึ่งมีรูปแบบของ classifier คือ

$$P_c = \left(\frac{P(A_1 = a_1 \wedge \ldots \wedge A_k = a_k \mid C = c)}{P(A_1 = a_1 \wedge \ldots \wedge A_k = a_k)} \right) \cdot P(C = c) \quad (3.1)$$

โดยที่

$P(A_1 = a_1 \wedge \ldots \wedge A_k = a_k \mid C = c)$	คือ	ค่าความน่าจะเป็นที่ชิ้นข้อมูลอยู่ในคลาส c
$P(A_1 = a_1 \wedge \ldots \wedge A_k = a_k)$	คือ	ความน่าจะเป็นของชิ้นข้อมูล
$P(C = c)$	คือ	ค่าความน่าจะเป็นของคลาส c
$A_1, \ldots, A_k$	คือ	เนื้อหาหรือแอตทริบิวต์ของชิ้นข้อมูล
$a_1, \ldots, a_k$	คือ	ค่าของเนื้อหาหรือแอตทริบิวต์ของชิ้นข้อมูล

เนื่องจากใน naïve Bayes Theorem นั้นกำหนดว่าความน่าจะเป็นของเนื้อหาหรือแอตทริบิวต์ต่างๆของชิ้นข้อมูลต้องไม่ขึ้นต่อกัน กล่าวคือผลลัพธ์หรือความน่าจะเป็นของแอตทริบิวต์ a จะไม่มีผลต่อผลลัพธ์หรือความน่าจะเป็นของแอตทริบิวต์อื่นๆ ดังนั้นจะได้เป็น Bayes rule ดังนี้

$$P_c = \left(\frac{P(C = c)}{P(A_1 = a_1 \wedge \ldots \wedge A_k = a_k)} \right) \cdot \prod_{i=1}^{k} P(A_i = a_i \mid C = c) \quad (3.2)$$

เราใช้ Bayes rule ในการเปรียบเทียบแต่ละคลาสเพื่อหาว่าสำหรับผู้ใช้ u แล้ว ชิ้นข้อมูล i ควรจะอยู่ในคลาสไหน ดังนั้นส่วนของ $P(A_1 = a_1 \wedge \ldots \wedge A_k = a_k)$ จึงสามารถละเว้นไม่ต้องนำมาคำนวณได้เพราะถือว่าเป็นค่าคงที่สำหรับผู้ใช้หรือชิ้นข้อมูลที่กำลังพิจารณาอยู่ และไม่มีส่วนสำคัญในการตัดสินใจเลือกคลาสดังนั้นค่าความน่าจะเป็นของคลาสสำหรับชิ้นงานคือ

$$P_c = P(C) \cdot \prod_{i=1}^{k} P(A_i = a_i \mid C = c) \quad (3.3)$$

โดยที่

$P(A_i = a_i \mid C = c)$ คือ ความน่าจะเป็นที่เนื้อหาหรือแอตทริบิวต์อยู่ในคลาส c

และหาความน่าจะเป็นที่เนื้อหาหรือแอตทริบิวต์ a_i $(P(A_i = a_i \mid C = c))$ อยู่ในคลาส c ได้จากวิธีในการคำนวณดังนี้

$$P(A_i = a_i \mid C = c) = \frac{count(A_i = a_i \wedge C = c)}{count(C = c)} \quad (3.4)$$

โดยที่

$count(A_i = a_i \wedge C = c)$ คือ จำนวนครั้งที่แอตทริบิวต์ a_i จัดอยู่ในคลาส c

$count(C = c)$ คือ จำนวนเรตติ้งทั้งหมดที่อยู่ในคลาส c

ตัวอย่างการคำนวณหาค่าความน่าจะเป็นจากเนื้อหาของชิ้นข้อมูล ในที่นี้เป็นเนื้อหาของภาพยนตร์ กำหนดให้ในฐานข้อมูลมีชิ้นข้อมูลของภาพยนตร์ทั้งหมด 6 ชิ้น และแต่ละชิ้นข้อมูลมีเวคเตอร์ของภาพยนตร์ในส่วนของเนื้อหา ดังนี้ movie i = {action, comedy, thriller, romantic, horror} เพราะฉะนั้นภาพยนตร์แต่ละเรื่องจะเขียนเวคเตอร์ของภาพยนตร์ในส่วนของเนื้อหาได้เป็นดังนี้

movie1 = {action, comedy} ⇒ movie1 = {1, 1, 0, 0, 0}
movie2 = {comedy, romantic} ⇒ movie2 = {0, 1, 0, 1, 0}
movie3 = {action, thriller} ⇒ movie3 = {1, 0, 1, 0, 0}
movie4 = {horror} ⇒ movie4 = {0, 0, 0, 0, 1}
movie5 = {comedy} ⇒ movie5 = {0, 1, 0, 0, 0}
movie6 = {action, romantic} ⇒ movie6 = {1, 0, 0, 1, 0}

รูปที่ 3.2 ตัวอย่างเวคเตอร์ของภาพยนตร์ในส่วนของเนื้อหาของภาพยนตร์แบบต่างๆ

สมมุติให้ผู้ใช้ A มีเวคเตอร์ระหว่างผู้ใช้-ชิ้นข้อมูลในส่วนของเนื้อหาของภาพยนตร์เป็นดังรูปที่ 3.3

	Movie1	Movie2	Movie3	Movie4	Movie5	Movie6
ผู้ใช้ A	5	5	3	1	4	?

รูปที่ 3.3 ตัวอย่างเวคเตอร์ระหว่างผู้ใช้-ชิ้นข้อมูลในส่วนของเนื้อหาของภาพยนตร์

กล่าวคือผู้ใช้ A ให้ค่าเรตติ้งกับ movie1 และ movie2 เป็น 5, ให้ค่าเรตติ้งกับ movie5 เป็น 4, ให้ค่าเรตติ้งกับ movie3 เป็น 3 และให้ค่าเรตติ้งกับ movie4 เป็น 1 และคำนวณค่าความน่าจะเป็นของคลาสเรตติ้งหนึ่งๆในส่วนของเนื้อหาของภาพยนตร์ได้ดังนี้

$$P(C=5) = \frac{2}{5}, P(C=4) = \frac{1}{5}, P(C=3) = \frac{1}{5}, P(C=2) = \frac{0}{5}, P(C=1) = \frac{1}{5}$$

รูปที่ 3.4 ตัวอย่างค่าความน่าจะเป็นของแต่ละคลาสเรตติ้งในส่วนของเนื้อหาของภาพยนตร์

การหาค่าความน่าจะเป็นของคลาสเรตติ้งในส่วนเนื้อหาของภาพยนตร์สำหรับmovie6 ของผู้ใช้ A อันดับแรกสร้างโปรไฟล์ของผู้ใช้ A โดยให้ $A_1 = action, A_2 = comedy, ..., A_5 = horror$ ตามลำดับ โดยใช้สมการที่ 3.4 เพื่อทำการสร้างโปรไฟล์ของผู้ใช้ A ซึ่งจะได้โปรไฟล์ที่เก็บข้อมูลความน่าจะเป็นที่เนื้อหาของภาพยนตร์สำหรับชิ้นข้อมูลนี้จะอยู่ในคลาสเรตติ้งหนึ่งๆสำหรับผู้ใช้ A เป็นดังรูปที่ 3.5

$$P(action|5) = \frac{1}{2}, P(action|4) = \frac{0}{1}, P(action|3) = \frac{1}{1}, P(action|2) = 0, P(action|1) = \frac{0}{1}$$

$$P(comedy|5) = \frac{2}{2}, P(comedy|4) = \frac{1}{1}, P(comedy|3) = \frac{0}{1}, P(comedy|2) = 0, P(comedy|1) = \frac{0}{1}$$

$$P(thriller|5) = \frac{0}{2}, P(thriller|4) = \frac{0}{1}, P(thriller|3) = \frac{1}{1}, P(thriller|2) = 0, P(thriller|1) = \frac{0}{1}$$

$$P(romantic|5) = \frac{1}{2}, P(romantic|4) = \frac{0}{1}, P(romantic|3) = \frac{0}{1}, P(romantic|2) = 0, P(romantic|1) = \frac{0}{1}$$

$$P(horror|5) = \frac{0}{2}, P(horror|4) = \frac{0}{1}, P(horror|3) = \frac{1}{1}, P(horror|2) = 0, P(horror|1) = \frac{1}{1}$$

รูปที่ 3.5 โปรไฟล์สำหรับผู้ใช้ A ที่เก็บค่าความน่าจะเป็นของคลาสสำหรับเนื้อหาของภาพยนตร์

พิจารณาที่ movie6 ซึ่งมีเวคเตอร์ของภาพยนตร์ในส่วนของเนื้อหาเป็น {1, 0, 0, 1, 0} และจากโปรไฟล์การให้ เรตติ้งของผู้ใช้ A จากสมการที่ 3.3 จะหาความน่าจะเป็นที่ movie6 จะอยู่ในคลาสเรตติ้งหนึ่งๆได้ ดังนี้ P(5) = (0.5)(0.5)(0.4) = 0.1, P(4) = (0)(0)(0.2) = 0, P(3) = (1)(0)(0.2) = 0, P(2) = (0)(0)(0) = 0, P(1) = (0)(0)(0.2) = 0

ในการพิจารณาเนื้อหาภาพยนตร์ผลปรากฏว่าค่าความน่าจะเป็นที่ movie6 จะอยู่ในคลาสเรตติ้งหนึ่งๆ มีค่าดังเช่นที่กล่าวไป ซึ่งค่าที่ได้นี้ยังต้องนำไปคำนวณร่วมกับค่าความน่าจะเป็นอีก 2 ค่า คือ ค่าความน่าจะเป็นในส่วนของนักแสดง และค่าความน่าจะเป็นในส่วนของผู้กำกับ ถึงจะตัดสินได้ว่า movie6 ควรจะอยู่ในคลาสเรตติ้งใด

ตัวอย่างการคำนวณหาค่าความน่าจะเป็นจากนักแสดง ในที่นี้กำหนดให้แต่ละชิ้นข้อมูลมีเวคเตอร์ของภาพยนตร์ในส่วนของนักแสดง ดังนี้ movie i = {pop, unpop} ซึ่งนักแสดงแต่ละเรื่องจะมีค่าเป็น pop หรือ unpop นั้นมีพื้นฐานมาจากภาพยนตร์ที่มีรายได้มากกว่า 100 ล้านเหรียญสหรัฐ ถ้านักแสดงที่อยู่ในภาพยนตร์เรื่องนั้นมีประวัติการแสดงในภาพยนตร์กลุ่มดังกล่าวเกิน 1 เรื่อง

อย่างน้อย 1 คน ให้ถือว่านักแสดงที่แสดงเรื่องนั้นมีค่าเป็น pop เพราะฉะนั้นภาพยนตร์แต่ละเรื่องจะเขียนเวคเตอร์ของภาพยนตร์ในส่วนของนักแสดงได้ดังนี้

movie1 = (pop) ⇨ movie1 = {1,0}

movie2 = (unpop) ⇨ movie2 = {0,1}

movie3 = (pop) ⇨ movie3 = {1,0}

movie4 = (pop) ⇨ movie4 = {1,0}

movie5 = (unpop) ⇨ movie5 = {0,1}

movie6 = (unpop) ⇨ movie6 = {0,1}

รูปที่ 3.6 ตัวอย่างเวคเตอร์ของภาพยนตร์ในส่วนของนักแสดง

สมมุติให้ผู้ใช้ A มีเวคเตอร์ระหว่างผู้ใช้-ชิ้นข้อมูลในส่วนของนักแสดงเป็นดังรูปที่ 3.7

	Movie1	Movie2	Movie3	Movie4	Movie5	Movie6
ผู้ใช้ A	4	4	3	5	1	?

รูปที่ 3.7 ตัวอย่างเวคเตอร์ระหว่างผู้ใช้-ชิ้นข้อมูลในส่วนของนักแสดง

กล่าวคือผู้ใช้ A ให้ค่าเรตติ้งกับ movie1 และ movie2 เป็น 4, ให้ค่าเรตติ้งกับ movie3 เป็น 3, ให้ค่าเรตติ้งกับ movie4 เป็น 5 และให้ค่าเรตติ้งกับ movie5 เป็น 1 และคำนวณหาค่าความน่าจะเป็นของคลาสเรตติ้งหนึ่งๆในส่วนของนักแสดงได้ดังนี้

$$P(C=5) = \frac{1}{5}, P(C=4) = \frac{2}{5}, P(C=3) = \frac{1}{5}, P(C=2) = \frac{0}{5}, P(C=1) = \frac{1}{5}$$

รูปที่ 3.8 ตัวอย่างค่าความน่าจะเป็นของแต่ละคลาสเรตติ้งในส่วนของนักแสดง

การหาค่าความน่าจะเป็นของคลาสเรตติ้งในส่วนของนักแสดงของ movie6 สำหรับผู้ใช้ A นั้น อันดับแรกสร้างโปรไฟล์ของผู้ใช้ A โดยให้ A_1 = pop และ A_2 = unpop ตามลำดับ โดยใช้สมการที่ 3.4 เพื่อทำการสร้างโปรไฟล์ของผู้ใช้ A ซึ่งจะได้โปรไฟล์ที่เก็บข้อมูลความน่าจะเป็นที่เนื้อหาของภาพยนตร์สำหรับชิ้นข้อมูลนี้จะอยู่ในคลาสเรตติ้งหนึ่งๆสำหรับผู้ใช้ A เป็นดังรูปที่ 3.9

$P(pop|5) = \frac{1}{1}$, $P(pop|4) = \frac{1}{2}$, $P(pop|3) = \frac{1}{1}$, $P(pop|2) = 0$, $P(pop|1) = \frac{0}{1}$

$P(unpop|5) = \frac{0}{1}$, $P(unpop|4) = \frac{1}{2}$, $P(unpop|3) = \frac{0}{1}$, $P(unpop|2) = 0$, $P(unpop|1) = \frac{1}{1}$

รูปที่ 3.9 โปรไฟล์สำหรับผู้ใช้ A ที่เก็บค่าความน่าจะเป็นของคลาสหนึ่งๆสำหรับนักแสดง

พิจารณาที่ *movie6* ซึ่งมีเวคเตอร์ของภาพยนตร์ในส่วนของนักแสดงเป็น {0, 1} และจากโปรไฟล์การให้เรตติ้งของผู้ใช้ A จากสมการที่ 3.3 จะหาความน่าจะเป็นในส่วนของนักแสดงที่ movie6 จะอยู่ในคลาสต่างๆได้ ดังนี้ P(5) = (0)(0.2) = 0, P(4) = (0.5)(0.4) = 0.2, P(3) = (0)(0.2) = 0, P(2) = (0)(0) = 0, P(1) = (1)(0.2) = 0.2

ผลที่ได้จากการนี้จะต้องนำไปคำนวณร่วมกับค่าที่ได้จากการคำนวณหาค่าความน่าจะเป็นอีก 2 ค่า คือ ค่าความน่าจะเป็นในส่วนของเนื้อหาของภาพยนตร์ และในส่วนของผู้กำกับ

ตัวอย่างการคำนวณหาค่าความน่าจะเป็นจากผู้กำกับ กำหนดให้แต่ละชิ้นข้อมูลมีเวคเตอร์ของภาพยนตร์ในส่วนของผู้กำกับ ดังนี้ movie i = {pop, unpop} ซึ่งผู้กำกับในแต่ละเรื่องจะมีค่าเป็น pop หรือ unpop นั้นมีพื้นฐานมาจากภาพยนตร์ที่มีรายได้มากกว่า 100 ล้านเหรียญสหรัฐ ถ้าผู้กำกับที่อยู่ในภาพยนตร์เรื่องนั้นมีประวัติการกำกับการแสดงในภาพยนตร์กลุ่มดังกล่าว เกิน 2 เรื่องอย่างน้อย 1 คน ให้ถือว่าผู้กำกับที่กำกับการแสดงเรื่องนั้นมีค่าเป็น pop เพราะฉะนั้นภาพยนตร์แต่ละเรื่องจะเขียนเวคเตอร์ของเนื้อหาได้ดังนี้

movie1 = (unpop) ⇨ movie1 = {0,1}
movie2 = (pop) ⇨ movie2 = {1,0}
movie3 = (unpop) ⇨ movie3 = {0,1}
movie4 = (unpop) ⇨ movie4 = {0,1}
movie5 = (pop) ⇨ movie5 = {1,0}
movie6 = (unpop) ⇨ movie6 = {0,1}

รูปที่ 3.10 ตัวอย่างเวคเตอร์ของภาพยนตร์ในส่วนของผู้กำกับ

สมมุติให้ผู้ใช้ A มีเวคเตอร์ระหว่างผู้ใช้-ชิ้นข้อมูลในส่วนของผู้กำกับเป็นดังรูปที่ 3.11

	Movie1	Movie2	Movie3	Movie4	Movie5	Movie6
ผู้ใช้ A	3	4	3	5	1	?

รูปที่ 3.11 ตัวอย่างเวคเตอร์ระหว่างผู้ใช้-ชิ้นข้อมูลในส่วนของผู้กำกับ

กล่าวคือผู้ใช้ A ให้ค่าเรตติ้งกับ movie1 และ movie3 เป็น 3, ให้ค่าเรตติ้งกับ movie2 เป็น 4, ให้ค่าเรตติ้งกับ movie4 เป็น 5 และให้ค่าเรตติ้งกับ movie5 เป็น 1 และคำนวณหาค่าความน่าจะเป็นของคลาสเรตติ้งต่างๆในส่วนของผู้กำกับได้ดังนี้

$$P(C=5) = \frac{1}{5}, P(C=4) = \frac{1}{5}, P(C=3) = \frac{2}{5}, P(C=2) = \frac{0}{5}, P(C=1) = \frac{1}{5}$$

รูปที่ 3.12 ตัวอย่างค่าความน่าจะเป็นของแต่ละคลาสเรตติ้งในส่วนของผู้กำกับ

การหาค่าความน่าจะเป็นของคลาสเรตติ้งในส่วนของผู้กำกับของ movie6 สำหรับผู้ใช้ A นั้น อันดับแรกสร้างโปรไฟล์ของผู้ใช้ A โดยให้ A_1 = pop และ A_2 = unpop ตามลำดับ โดยใช้สมการที่ 3.4 เพื่อทำการสร้างโปรไฟล์ของผู้ใช้ A ซึ่งจะได้โปรไฟล์ที่เก็บข้อมูลความน่าจะเป็นในส่วนของผู้กำกับที่ชี้นข้อมูลนี้จะอยู่ในคลาสเรตติ้งหนึ่งๆสำหรับผู้ใช้ A เป็นดังรูป

$$P(pop|5) = \frac{0}{1}, P(pop|4) = \frac{1}{1}, P(pop|3) = \frac{0}{2}, P(pop|2) = 0, P(pop|1) = \frac{1}{1}$$

$$P(unpop|5) = \frac{1}{1}, P(unpop|4) = \frac{0}{1}, P(unpop|3) = \frac{2}{2}, P(unpop|2) = 0, P(unpop|1) = \frac{0}{1}$$

รูปที่ 3.13 โปรไฟล์สำหรับผู้ใช้ A ที่เก็บค่าความน่าจะเป็นของคลาสหนึ่งๆสำหรับผู้กำกับ

พิจารณาที่ movie6 ซึ่งมีเวคเตอร์ของภาพยนตร์ในส่วนของผู้กำกับเป็น {0, 1} จากโปรไฟล์การให้เรตติ้งของผู้ใช้ A จากสมการที่ 3.3 จะหาความน่าจะเป็นที่ movie6 จะอยู่ในคลาสต่างๆได้ดังนี้ $P(5) = (1)(0.2) = 0.2$, $P(4) = (0)(0.2) = 0$, $P(3) = (1)(0.4) = 0.4$, $P(2) = (0)(0) = 0$, $P(1) = (0)(0.2) = 0$ ผลที่ได้คือค่าความน่าจะเป็นที่ movie6 จะอยู่ในคลาสเรตติ้งต่างๆในส่วนของผู้กำกับ

ในการที่จะหาว่า movie6 ซึ่งผู้ใช้ A ยังไม่เคยให้เรตติ้ง ควรจะมีค่าเรตติ้งเป็นเท่าไหร่ เราจะต้องทำการคำนวณหาค่าความน่าจะเป็นที่ movie6 จะอยู่ในคลาสเรตติ้งหนึ่งๆ โดยต้องทำการคำนวณค่าความน่าจะเป็นใน 3 ส่วน คือ ค่าความน่าจะเป็นในส่วนของเนื้อหาของภาพยนตร์, ค่าความน่าจะเป็นในส่วนของนักแสดง และค่าความน่าจะเป็นในส่วนของผู้กำกับ จากตัวอย่างที่กล่าวมา เราได้ค่าความน่าจะเป็นของทั้ง 3 ส่วนครบแล้ว ก็จะนำค่าความน่าจะเป็นที่ได้จากเมทริกซ์ทั้งสาม (เนื้อหาภาพยนตร์ ผู้กำกับ และนักแสดง) มาทำการหาค่าเฉลี่ยโดยใช้สูตร

$$P_S = \frac{P_g(\overline{R_g}) + P_A(\overline{R_A}) + P_D(\overline{R_D})}{\overline{R}_g + \overline{R}_A + \overline{R}_D} \tag{3.5}$$

เมื่อ

P_S คือ ความน่าจะเป็นที่ชิ้นข้อมูลจะอยู่ในคลาสเรตติ้ง S

P_g คือ ความน่าจะเป็นในส่วนของเนื้อหาภาพยนตร์ที่ชิ้นข้อมูลจะอยู่ในคลาสเรตติ้ง S

P_A คือ ความน่าจะเป็นในส่วนของนักแสดงที่ชิ้นข้อมูลจะอยู่ในคลาสเรตติ้ง S

P_D คือ ความน่าจะเป็นในส่วนของผู้กำกับที่ชิ้นข้อมูลจะอยู่ในคลาสเรตติ้ง S

$\overline{R}_g$ คือ เรตติ้งเฉลี่ยของเนื้อหาภาพยนตร์

$\overline{R}_A$ คือ เรตติ้งเฉลี่ยของนักแสดง

$\overline{R}_D$ คือ เรตติ้งเฉลี่ยของผู้กำกับ

ทำการคำนวณหาค่าความน่าจะเป็นที่ชิ้นข้อมูลจะอยู่ในคลาสเรตติ้งหนึ่งๆ ให้ครบทุกคลาสเรตติ้ง โดยใช้สมการที่ 3.5 เมื่อทราบค่าความน่าจะเป็นครบทุกคลาสเรตติ้งแล้ว ก็จะสามารถบอกได้ว่าชิ้นข้อมูลซึ่งผู้ใช้ยังไม่เคยให้เรตติ้งควรจะมีค่าเรตติ้งเป็นเท่าไหร่ โดยเลือกคลาสเรตติ้งที่มีค่าความน่าจะเป็นที่ชิ้นข้อมูลจะอยู่ในคลาสเรตติ้งนั้นที่มีค่ามากที่สุด ถ้ามีคลาสเรตติ้งมากกว่า 1 คลาสที่มีค่าความน่าจะเป็นดังกล่าวมากที่สุด ให้เลือกคลาสที่มีค่าความน่าจะเป็นในส่วนของผู้กำกับมากที่สุด, ถ้ายังเลือกไม่ได้ให้เลือกคลาสที่มีค่าความน่าจะเป็นในส่วนของเนื้อหาของภาพยนตร์มากที่สุด, ถ้ายังเลือกไม่ได้ก็ให้เลือกคลาสที่มีค่าความน่าจะเป็นในส่วนของนักแสดงมากที่สุด

จากตัวอย่างข้างต้นเมื่อนำค่าต่างๆสำหรับ movie6 ทั้งหมดที่ได้ มาคำนวณตามสมการที่ 3.5 จะได้ผลลัพธ์ คือ P(movie6|5) = 0.09, P(movie6|4) = 0.06, P(movie6|3) = 0.12, P(movie6|2) = 0 และ P(movie6|1) = 0.06 พบว่าคลาสเรตติ้ง 3 มีค่าความน่าจะเป็นสูงที่สุดดังนั้นจึงกำหนดค่าเรตติ้ง 3 ให้กับ movie6

โดยใช้วิธีดังกล่าวข้างต้นให้ทำขั้นตอนเดิมซ้ำไปเรื่อยๆจนครบทุกชิ้นข้อมูลก็จะได้เป็นเวคเตอร์ผู้ใช้-ชิ้นข้อมูลเทียมของผู้ใช้คนนั้นซึ่งจะมีทั้งเรตติ้งจริงที่ผู้ใช้ให้ และเรตติ้งที่ทำนายด้วยวิธีการ CBF ผสมผสานอยู่ด้วยกันในเวคเตอร์โดยให้

$$v_{u,i} = \begin{cases} r_{u,i} : if\ \ user\ u\ rated\ \ item\ i \\ c_{u,i} : otherwise \end{cases} \tag{3.6}$$

เมื่อ

$v_{u,i}$ คือ ค่าเรตติ้งที่ผู้ใช้ u ให้กับชิ้นข้อมูล i

$r_{u,i}$ คือ ค่าเรตติ้งจริงที่ผู้ใช้ u ให้กับชิ้นข้อมูล i

$c_{u,i}$ คือ ค่าเรตติ้งที่ทำนายได้จากขั้นตอนข้างต้น

3.2.3 สร้างเมตริกซ์ผู้ใช้-ชิ้นข้อมูลเทียม

จากเวคเตอร์ผู้ใช้-ชิ้นข้อมูลเทียมของผู้ใช้ทีละคนที่ได้มาด้วยวิธีการข้างต้น จากนั้นนำเวคเตอร์เทียมที่ได้จากผู้ใช้ทั้งหมดนำมารวมกันเป็นเมตริกซ์ผู้ใช้-ชิ้นข้อมูลเทียมเพื่อนำไปใช้ในขั้นตอนของ CF ต่อไป ดังที่ตารางตัวอย่างด้านล่างแสดงไว้ โดยค่าที่เป็นตัวหนาคือค่าที่ได้จากการทำนายด้วย CBF และค่าที่เป็นตัวบางคือค่าเรตติ้งจริงที่ผู้ใช้ให้ไว้

ตารางที่ 3.2 เมตริกซ์ผู้ใช้-ชิ้นข้อมูลเทียมที่ไม่มีความเบาบาง

	Transformers	Kill Bill	The Matrix	Nothing Hill	Pretty Women	Twister
ผู้ใช้คนที่ 1	1	4	**4**	3	3	2
ผู้ใช้คนที่ 2	3	**2**	2	5	**4**	4
ผู้ใช้คนที่ 3	**4**	5	3	4	3	**3**
ผู้ใช้คนที่ 4	2	3	4	**2**	3	4
ผู้ใช้คนที่ 5	3	**3**	4	3	3	4

จะเห็นได้ว่าหลังจากผ่านการทำนายด้วย Content-based predictor แล้วตารางที่ได้ไม่มีช่องไหนที่เป็นค่า null (?) ดังนั้นถือได้ว่าปัญหา Sparsity จึงหมดไปด้วยเช่นกัน

3.3 ขั้นตอนการทำงานส่วน Collaborative Filtering

จะใช้เทคนิค CF ที่ทำการหากลุ่มผู้ใช้ที่คล้ายคลึงด้วยการดู co-rated item เป็นหลัก ซึ่งเราแบ่งขั้นตอนการทำงานของ CF นี้ออกเป็น 4 ขั้นตอน คือ หาค่าความคล้ายคลึงของผู้ใช้ปัจจุบันกับผู้ใช้อื่น, การสร้างรายชื่อผู้ใช้ที่มีค่าความคล้ายคลึงสูงสุดกับผู้ใช้ปัจจุบัน, การทำนายค่าเรตติ้งของชิ้นข้อมูล และการแนะนำรายการของภาพยนตร์ที่น่าสนใจให้กับผู้ใช้ปัจจุบัน

3.3.1 ขั้นตอนการหาความคล้ายคลึงของผู้ใช้

จากเมตริกซ์ V หรือเมตริกซ์ผู้ใช้-ชิ้นงานเทียม(Pseudo user-rating matrix) ที่ได้จากขั้นตอนก่อนหน้าเรานำมาหาค่าความคล้ายคลึง (Similarity) ระหว่างผู้ใช้ปัจจุบัน (Active user) กับผู้ใช้อื่นๆ โดยใช้วิธีการหาค่าความคล้ายคลึงด้วยค่าสัมประสิทธิ์ความสัมพันธ์แบบเพียร์สัน หรือ Pearson correlation coefficient

$$P_{a,u} = \frac{\sum_{i=1}^{m}(v_{a,i} - \overline{v_a}) \times (v_{u,i} - \overline{v_u})}{\sqrt{\sum_{i=1}^{m}(v_{a,i} - \overline{v_a})^2} \times \sqrt{\sum_{i=1}^{m}(v_{u,i} - \overline{v_u})^2}} \quad (3.7)$$

โดยที่

$P_{a,u}$	คือ	ค่าสัมประสิทธิ์ความสัมพันธ์แบบเพียร์สัน
$v_{a,i}$	คือ	ค่าเรตติ้งในเมตริกซ์ผู้ใช้-ชิ้นข้อมูลเทียมที่ผู้ใช้ปัจจุบันให้กับชิ้นข้อมูล i
$\overline{v_a}$	คือ	ค่าเฉลี่ยของเรตติ้งของผู้ใช้ปัจจุบันในเมตริกซ์เทียม
$v_{u,i}$	คือ	ค่าเรตติ้งในเมตริกซ์ผู้ใช้-ชิ้นข้อมูลเทียมที่ผู้ใช้ u ให้กับชิ้นข้อมูล i
$\overline{v_u}$	คือ	ค่าเฉลี่ยนของเรตติ้งของผู้ใช้ u ในเมตริกซ์เทียม
m	คือ	จำนวนของชิ้นข้อมูลทั้งหมดที่มี

ใช้วิธีการเดียวกันทำการหาค่าความคล้ายคลึงสำหรับผู้ใช้ปัจจุบันกับผู้ใช้อื่น (u) ให้ครบทุกคนโดยวิธีการค่าความคล้ายคลึงของผู้ใช้นี้ ใช้วิธีคำนวณเช่นเดียวกับตัวอย่างในบทที่ 2

ตัวอย่างการคำนวณหาค่าความคล้ายคลึง จากตารางที่ 3.2 ให้ผู้ใช้ปัจจุบัน (Active user) คือ ผู้ใช้คนที่ 4 หาค่าความคล้ายคลึงระหว่างผู้ใช้คนที่ 4 กับผู้ใช้คนที่ 5 โดยใช้สมการ 3.6 โดยค่าเฉลี่ยเรตติ้งของผู้ใช้คนที่ 4 มีค่าเท่ากับ 3 และค่าเฉลี่ยเรตติ้งของผู้ใช้คนที่ 5 มีค่าเท่ากับ 3.33

$$P_{a,u} = \frac{(2-3)(3-3.33)+(3-3)(3-3.33)+(4-3)(4-3.33)+(2-3)(3-3.33)+(3-3)(3-3.33)+(4-3)(4-3.33)}{\sqrt{(2-3)^2+(3-3)^2+(4-3)^2+(2-3)^2+(3-3)^2+(4-3)^2}\sqrt{(3-3.33)^2+(3-3.33)^2+(4-3.33)^2+(3-3.33)^2+(3-3.33)^2+(4-3.33)^2}}$$

$$= \frac{(-1)(-0.33)+(0)(-0.33)+(1)(0.67)+(-1)(-0.33)+(0)(-0.33)+(1)(0.67)}{\sqrt{(-1)^2+(0)^2+(1)^2+(-1)^2+(0)^2+(1)^2}\sqrt{(-0.33)^2+(-0.33)^2+(0.67)^2+(-0.33)^2+(-0.33)^2+(0.67)^2}}$$

$$= \frac{(0.33)+(0)+(0.67)+(0.33)+(0)+(0.67)}{\sqrt{4}\sqrt{1.34}}$$

$$= 0.86$$

ดังนั้นผู้ใช้คนที่ 4 และผู้ใช้คนที่ 5 จะมีค่าความคล้ายคลึงเท่ากับ 0.86

3.3.2 การสร้างรายชื่อผู้ใช้ที่มีค่าความคล้ายคลึงสูง

เลือกผู้ใช้ที่มีค่าความคล้ายคลึงกับผู้ใช้ปัจจุบันสูงที่สุดมาจำนวน n ผู้ใช้ เรียกกลุ่มผู้ใช้นี้ว่า neighborhood ของ active user ถ้าข้อมูลในเมตริกซ์ผู้ใช้-ชิ้นข้อมูลมีความเบาบาง (Sparse) สูงทำให้

เป็นไปได้ที่จะมีผู้ใช้ซึ่งมีค่าความคล้ายคลึงสูงมากแต่มีจำนวน co-rated item ต่ำ จึงมีการกำหนดตัวถ่วงน้ำหนักค่าความสัมพันธ์ไว้ เมื่อผู้ใช้ทั้งสองมีจำนวน co-rated item ต่ำกว่า 50

$$sg_{a,u} = \begin{cases} \frac{n}{50} & : if\ 2\ users\ have\ co\text{-}rated\ item\ <\ 50 \\ 1 & : otherwise \end{cases} \tag{3.8}$$

โดยที่

n คือ จำนวน co-rated item ของผู้ใช้ทั้งสอง

จากสมการที่ 3.7 ค่าถ่วงน้ำหนักความสัมพันธ์จะมีค่าเป็น 1 ถ้าผู้ใช้ทั้งสองมีจำนวน co-rated item สูงกว่า 50 ชิ้นข้อมูล และมีค่าเป็น $\frac{n}{50}$ เมื่อผู้ใช้ทั้งสองมีจำนวน co-rated item ต่ำกว่า 50 ชิ้นข้อมูล และเนื่องจากความแม่นยำของเมตริกซ์ผู้ใช้-ชิ้นข้อมูลเทียม(Pseudo User-Item matrix) ขึ้นอยู่กับจำนวนชิ้นข้อมูลที่ผู้ใช้คนนั้นให้เรตติ้ง ซึ่งก็คือถ้าผู้ใช้ให้เรตติ้งกับชิ้นข้อมูลน้อยเมตริกซ์ผู้ใช้-ชิ้นข้อมูลเทียมที่สร้างขึ้นก็จะไม่มีความแม่นยำ กล่าวคือความแม่นยำในการทำนายข้อมูลของ Content-based predictor นั้นขึ้นอยู่กับจำนวนของชิ้นข้อมูลที่ผู้ใช้เคยให้เรตติ้ง ดังนั้นจึงมีการกำหนดค่าถ่วงน้ำหนักเพิ่มขึ้นมาอีกหนึ่งค่า คือ ค่าถ่วงน้ำหนัก Harmonic Mean (HM weighting)

$$hm_{i,j} = \frac{2m_i m_j}{m_i + m_j} \tag{3.9}$$

เมื่อ

$$m_i = \begin{cases} \frac{n_i}{50} & : if\ n_i\ <\ 50 \\ 1 & : otherwise \end{cases} \tag{3.10}$$

โดยที่

n_i คือ จำนวนของชิ้นงานที่ผู้ใช้ i เคยให้เรตติ้งไว้

จากค่าถ่วงน้ำหนักทั้งสองจากสมการ 3.7 และ 3.8 ดังที่กล่าวมา นำมารวมกันเป็นค่าถ่วงน้ำหนักความสัมพันธ์ระหว่างผู้ใช้ปัจจุบันกับผู้ใช้อื่น หรือ hybrid correlation weight ($hw_{a,u}$) ได้เป็น

$$hw_{a,u} = hm_{a,u} + sg_{a,u} \tag{3.10}$$

ตัวอย่างการคำนวณค่าถ่วงน้ำหนัก จากตารางที่ 3.1 ผู้ใช้คนที่ 4 มีจำนวนชิ้นข้อมูลที่ให้เรตติ้งเท่ากับ 5 และผู้ใช้คนที่ 5 มีจำนวนชิ้นข้อมูลที่ให้เรตติ้งเท่ากับ 5 และจากตารางได้ว่าผู้ใช้คนที่ 4 และผู้ใช้คนที่ 5 มีจำนวน co-rated item เท่ากับ 4 ดังนั้นจากสมการที่ 3.8, 3.9 และ 3.10 จะได้ว่า

$$hw_{a,u} = \frac{2 \cdot (\frac{5}{50}) \cdot (\frac{5}{50})}{\frac{5}{50} + \frac{5}{50}} + \frac{4}{50} \quad = 0.18$$

ซึ่งค่าถ่วงน้ำหนักที่ได้นี้จะถูกนำไปใช้คำนวณในขั้นตอนการทำนายข้อมูลต่อไป

สำหรับค่า 50 เป็นค่าขั้นต่ำของจำนวน co-rated item ที่ Melville [4] ได้ทำการทดลองแล้วพบว่าเป็นค่า threshold ที่จะทำให้ Content-based predictor เริ่มมีความแม่นยำในการทำนายสูง

3.3.3 ขั้นตอนการทำนายข้อมูล

ทำการรวมค่าถ่วงน้ำหนักที่กล่าวไปก่อนหน้านี้เข้าไปกับวิธีการก่อนหน้าเพื่อทำการทำนายข้อมูลสำหรับผู้ใช้ปัจจุบันกับชิ้นงาน i ได้ดังนี้

$$p_{a,i} = \bar{v}_a + \frac{\sum_{u=1}^{n} hw_{a,u} P_{a,u} (v_{u,i} - \bar{v}_u)}{\sum_{u=1}^{n} hw_{a,u} P_{a,u}} \tag{3.11}$$

โดยที่

$p_{a,i}$	คือ	ค่าที่เรตติ้งที่ทำนายให้กับชิ้นงาน i สำหรับผู้ใช้ปัจจุบัน
$v_{u,i}$	คือ	ค่าเรตติ้งจากเมตริกซ์ผู้ใช้-ชิ้นข้อมูลเทียมของผู้ใช้ u กับชิ้นงาน i
$\bar{v}_u$	คือ	ค่าเฉลี่ยของทุกชิ้นงานสำหรับผู้ใช้ u
n	คือ	จำนวนของผู้ใช้ที่มีค่าความคล้ายคลึงสูง (neighborhood size)

โดยปกติแล้วจะกำหนดค่า n หรือ neighborhood size ไว้ที่ 3, 5 หรือ 7 คือ กำหนดจำนวนผู้ใช้ที่มีค่าความคล้ายคลึงสูงที่สุดไว้ที่ 3, 5 หรือ 7 คน สำหรับการทดลองนี้กำหนดไว้ที่ 11 คน

ตัวอย่างการคำนวณในขั้นตอนการทำนายข้อมูล เพื่อทำนายค่าเรตติ้งสำหรับชิ้นข้อมูล Nothing Hill ซึ่งผู้ใช้คนที่ 4 ยังไม่เคยให้เรตติ้ง กำหนดให้ $n = 1$ กล่าวคือกำหนดให้กลุ่มของผู้ใช้ที่มีค่าความคล้ายคลึงสูงสุดมี 1 คน, ผู้ใช้คนที่ 4 เป็นผู้ใช้ปัจจุบัน (Active user), ผู้ใช้คนที่ 5 เป็นผู้ใช้ที่มีค่าความคล้ายคลึงสูงที่สุด, $hw_{a,u} = 0.18$ และ $P_{a,u} = 0.86$ (ได้จากตัวอย่างที่กล่าวไปก่อนหน้า),

ค่าเฉลี่ยของเรตติ้งของผู้ใช้ที่ 4 มีค่าเป็น $\bar{v}_a = \frac{2+3+4+2+3+4}{6} = 3$, ค่าเฉลี่ยของเรตติ้งของผู้ใช้ที่ 5 มีค่าเป็น $\bar{v}_u = \frac{3+3+4+3+3+4}{6} = 3.33$, และเมื่อทำการคำนวณด้วยสมการที่ 3.11 จะได้ว่า $p_{a,i} = 3 + \frac{(0.18)(0.86)(3-3.33)}{(0.18)(0.86)} = 2.67$

ดังนั้นชิ้นข้อมูล Nothing Hill สำหรับผู้ใช้ปัจจุบันจะได้เรตติ้งเท่ากับ 2.67 เมื่อใช้วิธีดังที่กล่าวมา และกำหนดจำนวนผู้ใช้ที่มีค่าความคล้ายคลึงสูงสุดเป็น 1 (neighborhood size = 1)

3.3.4 ขั้นตอนการแนะนำรายการภาพยนตร์

ในขั้นตอนนี้จะเกี่ยวกับวิธีการที่ใช้ในการนำเสนอข้อมูลที่ระบบทำการแนะนำให้กับผู้ใช้ปัจจุบัน โดยจะใช้หลักการแบบ *Top-N Recommendation* ในการนำรายการภาพยนตร์มานำเสนอให้กับผู้ใช้ปัจจุบัน โดยมีการทำงานเป็นขั้นตอน ดังต่อไปนี้

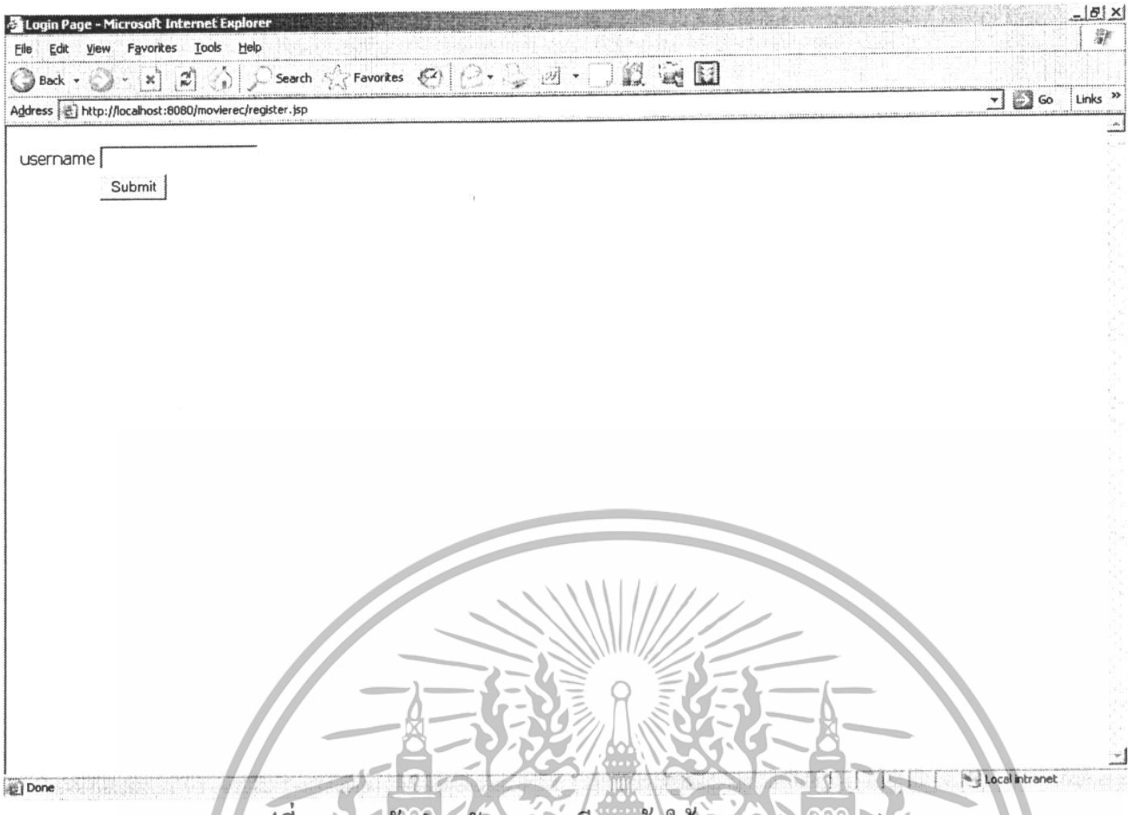
ขั้นตอนที่หนึ่ง นำภาพยนตร์มาเรียงลำดับตามค่าความพึงพอใจจากมากไปหาน้อย โดยค่าความพึงพอใจของผู้ใช้ปัจจุบันต่อภาพยนตร์เรื่องหนึ่งๆนั้นหาได้จากวิธีการในขั้นตอนที่ 3.3.3 ดังที่ได้กล่าวไปแล้ว

ขั้นตอนที่สอง เลือกภาพยนตร์ที่มีค่าความพึงพอใจสำหรับผู้ใช้ปัจจุบันสูงที่สุดจำนวน N เรื่อง มานำเสนอให้กับผู้ใช้ปัจจุบัน

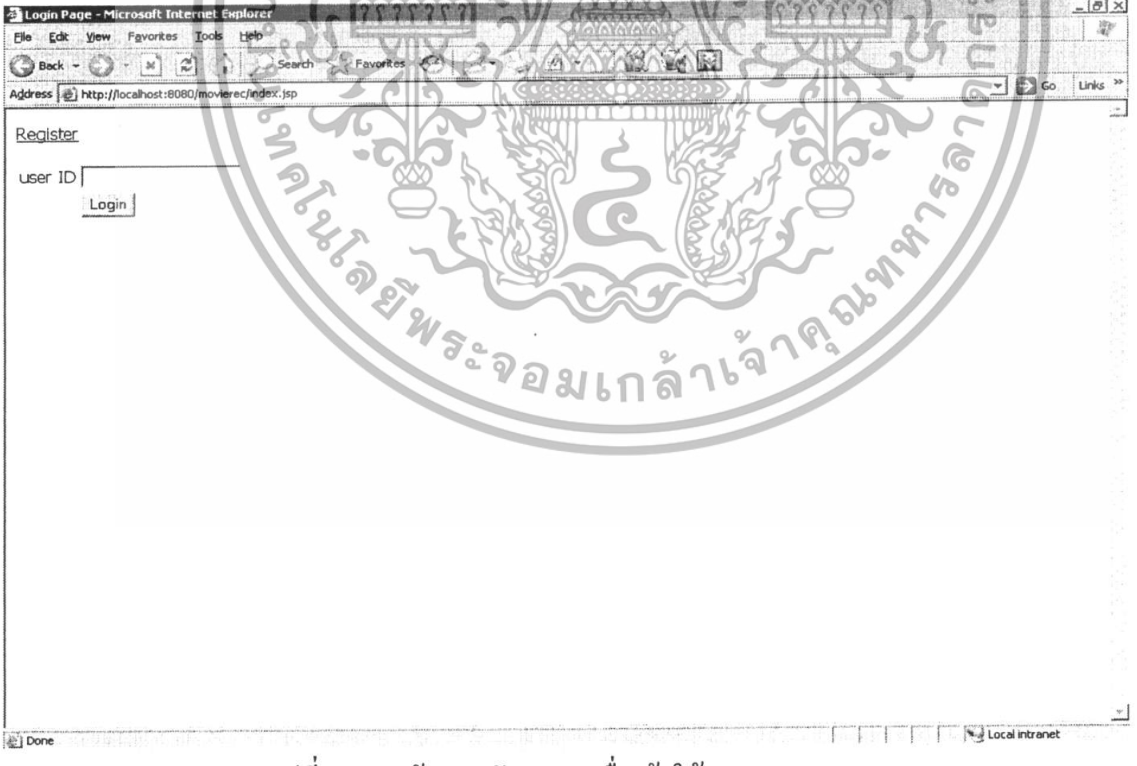
รายการของภาพยนตร์ที่ได้ก็จะนำไปเป็นคำแนะนำในเรื่องของภาพยนตร์ให้แก่ผู้ใช้ที่กำลังใช้ระบบอยู่ในขณะนั้น

3.3.5 Interface Prototype

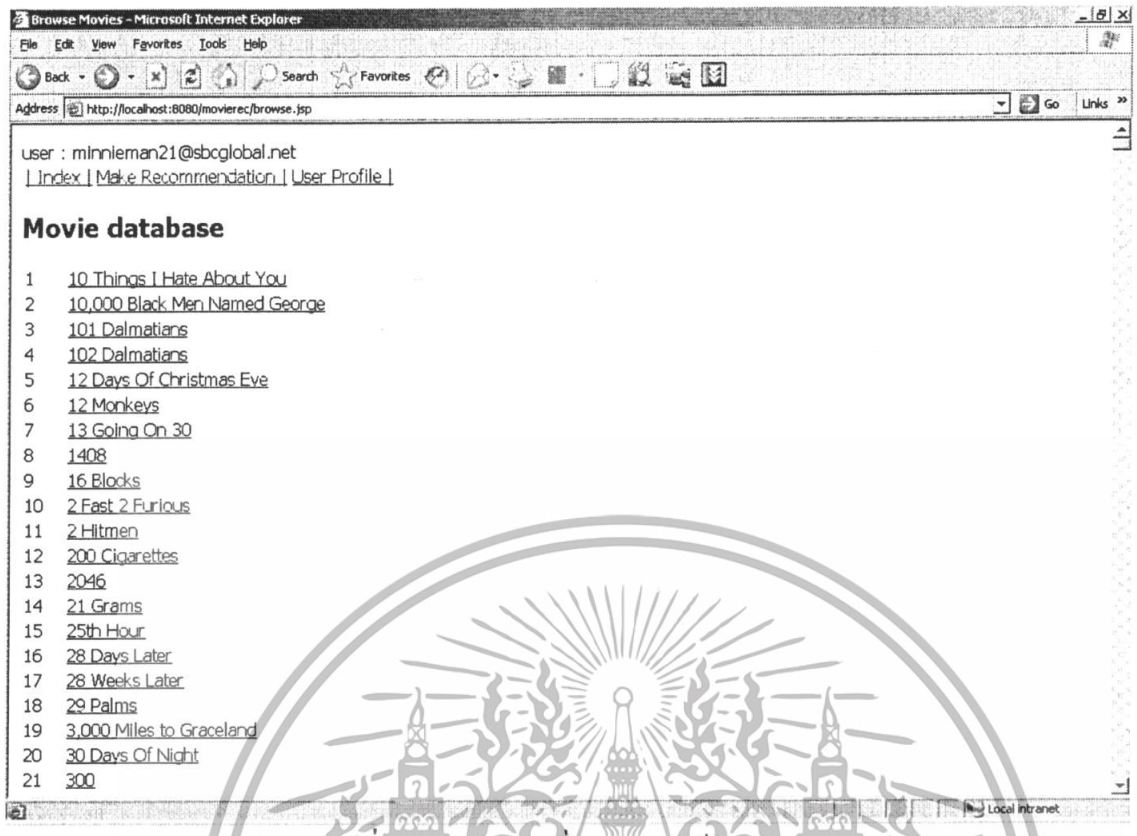
สำหรับตัวอย่างอินเทอร์เฟสของระบบให้การแนะนำเป็นดังนี้



รูปที่ 3.14 หน้าสำหรับลงทะเบียนเข้าใช้ระบบ (Register)

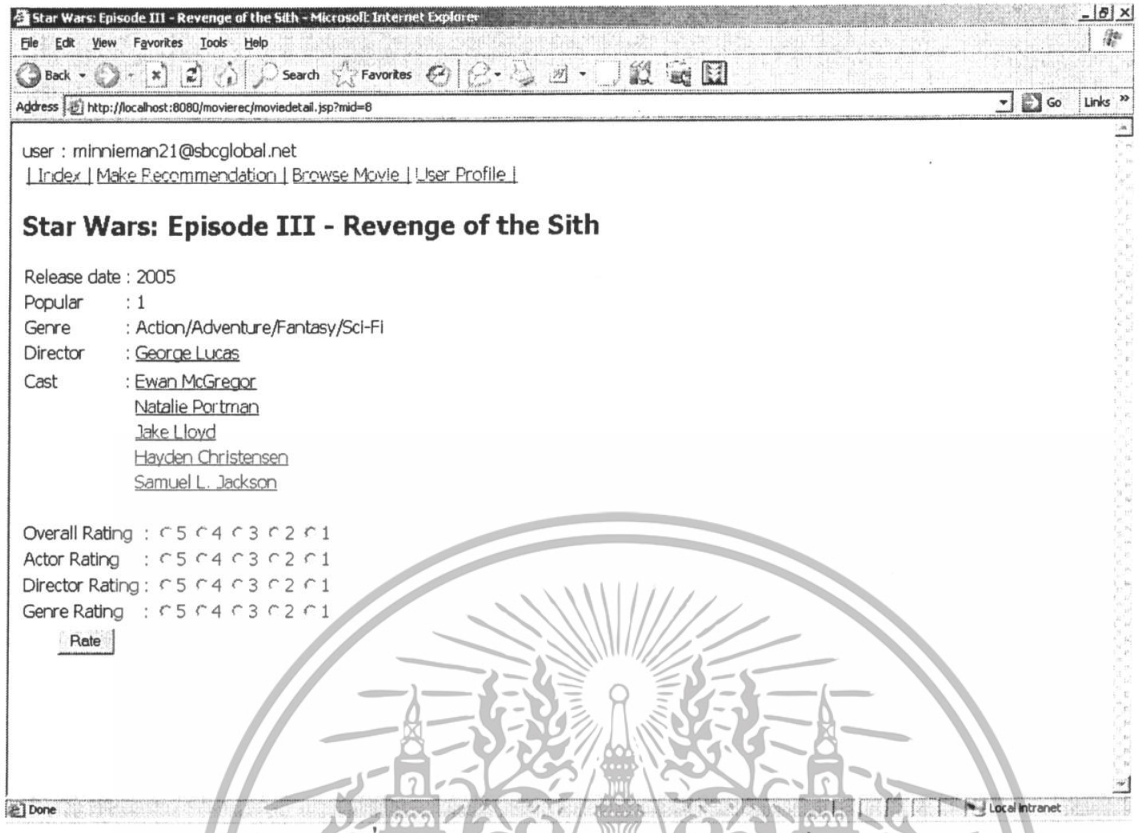


รูปที่ 3.15 หน้าสำหรับการลงชื่อเข้าใช้ระบบ (Login)



รูปที่ 3.16 แสดงรายชื่อภาพยนตร์ในระบบ

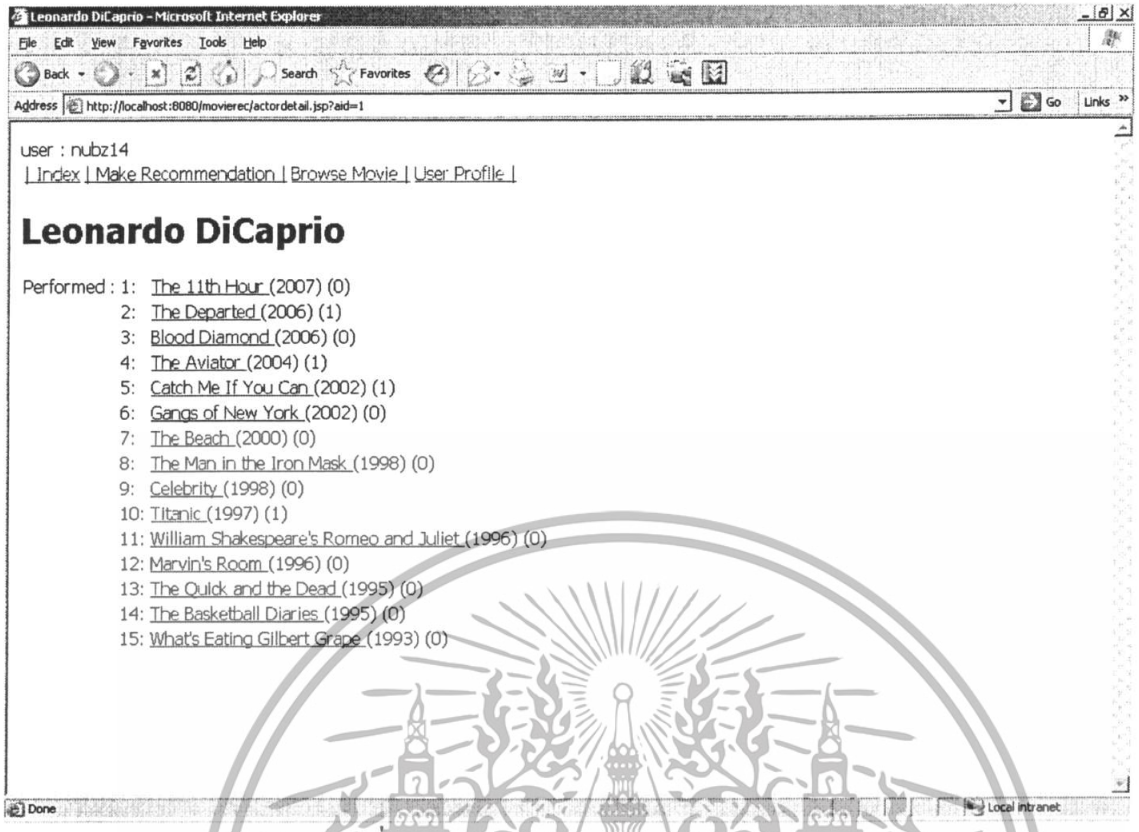
จากรูป 3.16 เป็นหน้าแรกหลังจากที่ทำการลงชื่อเข้าใช้ในระบบ โดยจะแสดงรายชื่อภาพยนตร์ในระบบ เรียงตามลำดับตัวอักษร เราสามารถดูรายละเอียดของแต่ละภาพยนตร์ได้โดยการคลิกที่ชื่อภาพยนตร์เรื่องนั้นๆ ระบบจะไปที่หน้าแสดงรายละเอียดของภาพยนตร์เรื่องนั้น



รูปที่ 3.17 แสดงรายละเอียดของภาพยนตร์

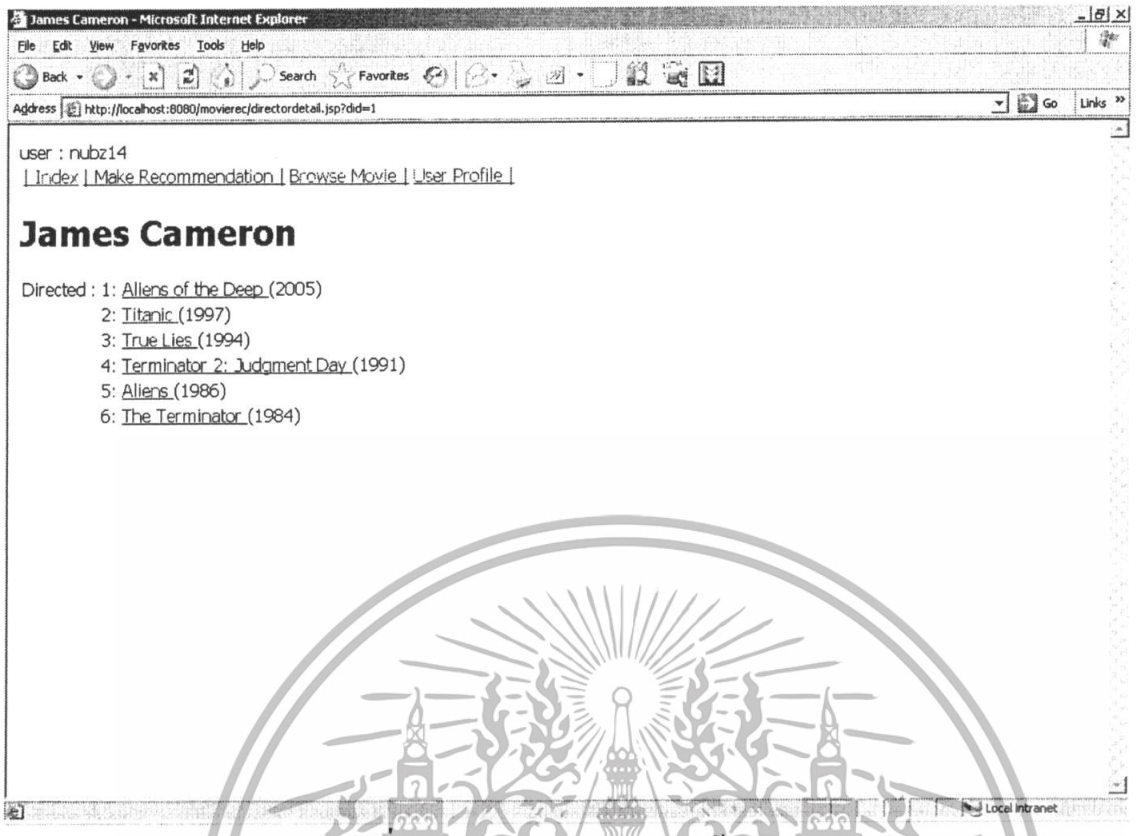
จากรูป 3.17 นั้นเป็นหน้าที่แสดงรายละเอียดของภาพยนตร์เรื่องนั้นๆ โดยจะมี ปีที่ฉาย ประเภทภาพยนตร์ ผู้กำกับ และรายชื่อนักแสดง ส่วนค่าของ Popular ที่เป็น 1 นั้นหมายความว่า ภาพยนตร์เรื่องนี้เป็นภาพยนตร์ที่ทำรายได้ ได้เกิน 100 ล้านเหรียญสหรัฐ ซึ่งในระบบถือว่า ภาพยนตร์เรื่องนี้เป็นภาพยนตร์ยอดนิยมรวมไปถึงผู้กำกับและนักแสดงที่เล่นเรื่องนี้ด้วย

ถัดลงมาจะเป็นช่องให้ผู้ใช้แต่ละคนทำการให้คะแนนความพึงพอใจที่มีต่อเรื่องนั้นๆ จำนวน 4 คะแนนด้วยกันคือ คะแนนความชอบโดยรวม (Overall Rating) คะแนนนักแสดง (Actor Rating) คะแนนผู้กำกับ (Director Rating) และคะแนนประเภทภาพยนตร์ (Genre Rating)



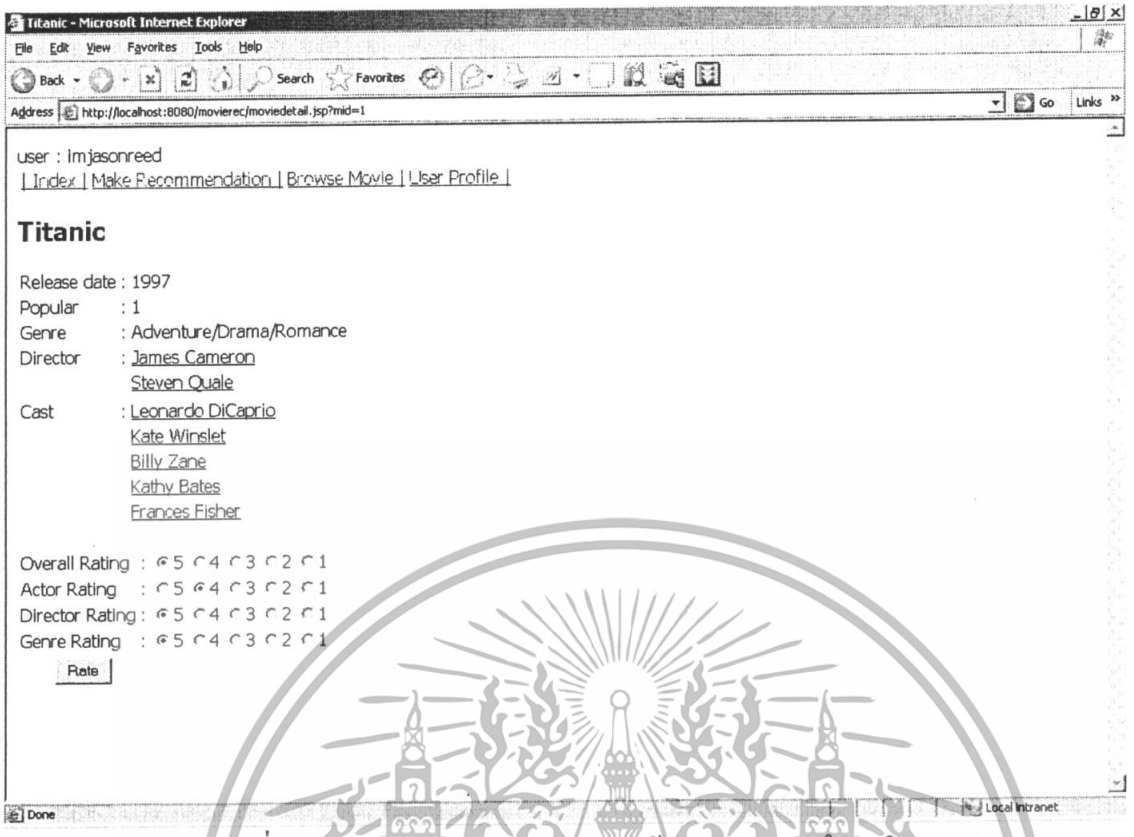
รูปที่ 3.18 แสดงรายละเอียดของนักแสดง

จากรูป 3.18 เป็นหน้าต่างแสดงรายละเอียดของนักแสดงว่าเคยแสดงในภาพยนตร์เรื่องใดบ้างเมื่อเราคลิกไปที่ชื่อของนักแสดง ในการแสดงผลนั้นระบบจะทำการแสดงภาพยนตร์ที่นักแสดงคนนั้นแสดงโดยเรียงลำดับจากปัจจุบันไปหาอดีต



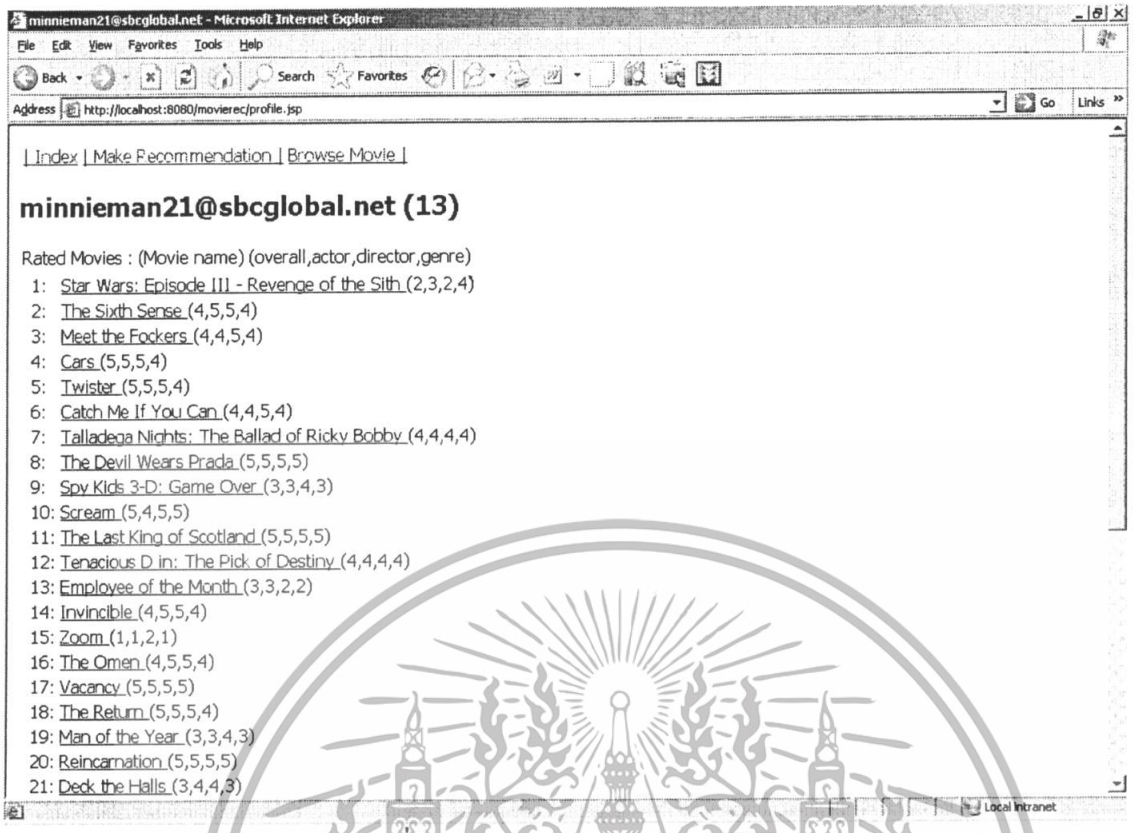
รูปที่ 3.19 แสดงรายละเอียดของผู้กำกับ

จากรูป 3.19 เป็นหน้าต่างแสดงรายละเอียดของผู้กำกับว่าเคยกำกับการแสดงในภาพยนตร์เรื่องใดบ้างเมื่อเราคลิกไปที่ชื่อของผู้กำกับ ในการแสดงผลนั้นระบบจะทำการแสดงภาพยนตร์ที่ผู้กำกับคนนั้นกำกับการแสดงโดยเรียงลำดับจากปัจจุบันไปหาอดีต



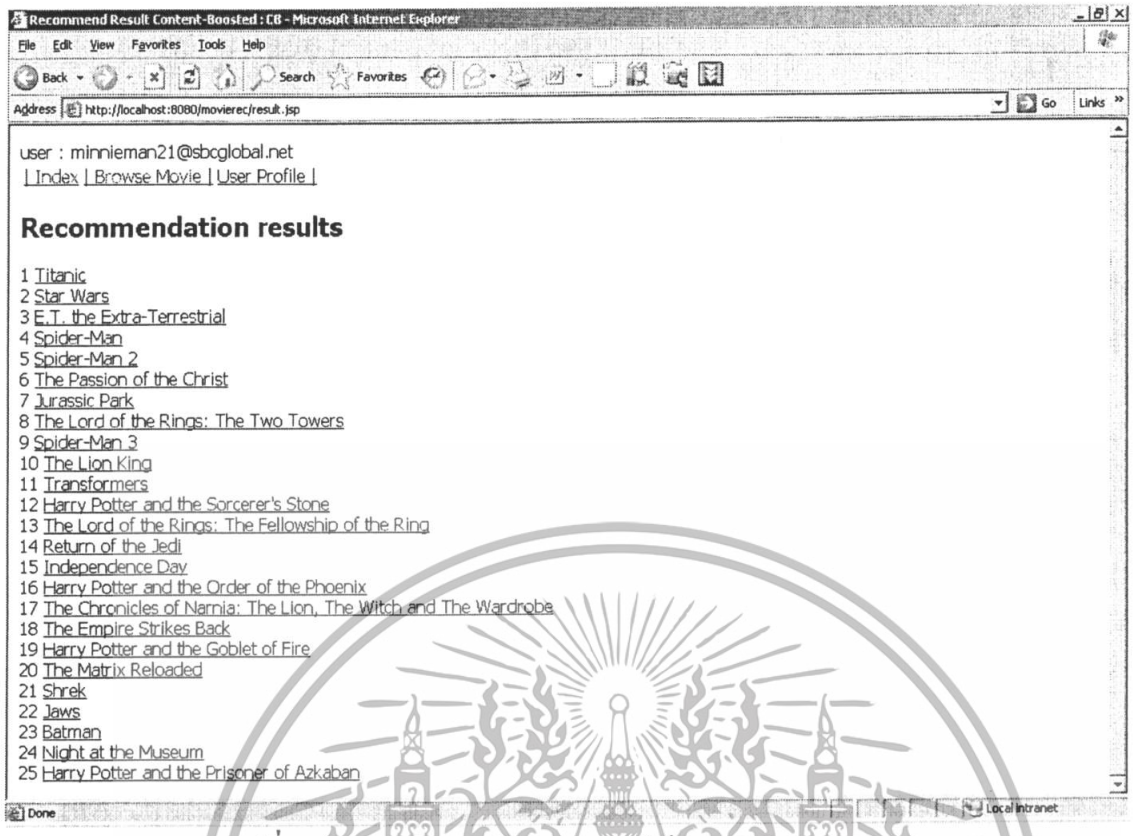
รูปที่ 3.20 แสดงรายละเอียดของการให้คะแนนความพึงพอใจ

จากรูป 3.20 เป็นหน้าต่างที่แสดงการให้คะแนนความพึงพอใจของผู้ใช้



รูปที่ 3.21 แสดงรายละเอียดของผู้ใช้

จากรูป 3.21 เป็นหน้าต่างแสดงรายละเอียดการให้คะแนนความพึงพอใจของผู้ใช้คนนั้นๆ ในภาพยนตร์ที่ผ่านการให้คะแนนแล้วโดยเรียงลำดับภาพยนตร์ตามลำดับการให้คะแนนความพึงพอใจ



รูปที่ 3.22 แสดงรายละเอียดของการให้คำแนะนำของระบบ

จากรูป 3.22 เมื่อผู้ใช้คลิกที่ Make Recommendation ระบบจะทำการแนะนำภาพยนตร์ที่คาดว่าผู้ใช้ให้ความสนใจออกมา โดยเรียงตามความสนใจจากมากไปหาน้อย

บทที่ 4

การทดลองและผลการทดลอง

งานวิจัยนี้ได้เลือกดาต้าเซตที่เหมาะสมมาทำการทดลอง และเก็บรวบรวมผลการทดลองในแต่ละครั้งพร้อมกับนำเสนอผลการทดลองทั้งหมดมาวิเคราะห์ด้วยวิธีการทางสถิติซึ่งมีรายละเอียดดังต่อไปนี้

4.1 เครื่องมือในการทดลอง

ฮาร์ดแวร์

- เครื่องคอมพิวเตอร์ Toshiba Satellite
  - Intel Pentium M 1.6 mobile centrino
  - หน่วยความจำ 1.25 GB
  - ฮาร์ดดิสก์ขนาด 40 GB

ซอฟท์แวร์

- ระบบปฏิบัติการ Windows XP
- Microsoft Excel 2003
- Microsoft Word 2003
- AppServ 2.5.9
- Apache Tomcat 6.0.14
- Mysql 5.0.45 win32
- Java SE Development Kit 6 Update 3
- Edit Plus 2

4.2 ขั้นตอนการทดลอง

ดังที่กล่าวไปแล้วในบทที่ผ่านมา วิธีการที่นำเสนอประกอบด้วยขั้นตอนการทำงานหลัก 2 ส่วน คือส่วนของ Content-Based Filtering (CBF) และส่วนของ Collaborative Filtering (CF) แต่ในการทดลองได้เพิ่มขั้นตอนในส่วนของการประเมินผลเพื่อใช้ประมวลผลจากการทดลอง ซึ่งมีรายละเอียดดังต่อไปนี้

4.2.1 กระบวนการเตรียมข้อมูล

ข้อมูลที่ใช้ในการทดลองได้นำมาจากดาต้าเซตที่เป็นที่นิยมกันทางด้านนี้โดยเฉพาะ ได้แก่

1. ดาต้าเซต จาก Yahoo Movies รวบรวมจากผู้ใช้ 200 คน และภาพยนตร์ 2,651 เรื่อง ภายใต้เงื่อนไขของดาต้าเซตเริ่มต้นมีการให้เรตติ้งภาพยนตร์จำนวน 50 เรื่อง ระดับการให้คะแนนเรตติ้งมี 5 ระดับ เรียงจากความชอบน้อยไปจนถึงความชอบมาก ได้แก่ 1,2,3,4 และ 5 เรตติ้งทั้งสิ้นรวม 10,000 เรตติ้ง

2. ข้อมูลประเภทภาพยนตร์ จำนวน 23 ประเภทจาก Internet Movie Database

กระบวนการเตรียมข้อมูลเริ่มต้นจากการกำหนดให้ภาพยนตร์ที่มีรายได้มากกว่า 100 ล้านเหรียญสหรัฐ ซึ่งมีจำนวน 379 เรื่องเป็น Popular movie และให้ถือว่า นักแสดงที่แสดงในภาพยนตร์นี้เป็น Popular actor และ ผู้กำกับที่ กำกับภาพยนตร์นั้นๆเป็น Popular director จากนั้นสุ่มเลือกผู้ใช้จำนวน 200 ราย โดยที่ผู้ใช้นั้นจะต้องมีประวัติการให้เรตติ้งไม่ต่ำกว่า 50 เรตติ้งต่อผู้ใช้แต่ละราย จากนั้นทำการเลือกเรตติ้งของผู้ใช้ ผู้ใช้ละ 50 เรตติ้ง รวมเป็น 10,000 เรตติ้ง จากนั้นเก็บข้อมูลของนักแสดงและผู้กำกับ ซึ่งมีจำนวนนักแสดงทั้งหมด 5,167 ราย ผู้กำกับ 1,811 ราย

ที่มาของดาต้าเซตนั้นมาจาก Yahoo movies และ IMDB สาเหตุที่ไม่นำดาต้าเซตที่เป็นมาตรฐานเนื่องจากข้อมูลเรตติ้งที่มีให้ในดาต้าเซตนั้นไม่ตรงกับความต้องการในการทำการทดลอง กล่าวคือ เรตติ้งที่มีให้ในดาต้าเซตทั่วๆไปนั้น มีเรตติ้งเพียงตัวเดียวคือ Overall rating ซึ่งเรตติ้งที่ใช้ในการทดลองนั้นต้องใช้ทั้งหมด 4 ตัวด้วยกันก็คือ Overall rating, Genre rating (ประเภทภาพยนตร์), Actor rating (นักแสดง), Director rating (ผู้กำกับ)

ในกระบวนการเตรียมข้อมูลเรตติ้งทางผู้จัดทำได้แบ่งออกเป็น 2 แบบจำลอง คือ แบบจำลองเรตติ้งสำหรับการสอน (Training Set) และแบบจำลองเรตติ้งสำหรับการทดสอบ (Test Set) ตามสัดส่วนที่ต้องการใช้ในการทดลองซึ่งในงานวิจัยนี้ได้จัดทำดาต้าเซตขึ้นมาโดยเอาข้อมูลเรตติ้งมาจาก Yahoo Movies และ IMDB ซึ่งดาต้าเซตที่นำมานั้นมีความเบาบางเท่ากับ 99.98 สำหรับการทดสอบนั้นจะทำการทดสอบด้วยสัดส่วน 75% ต่อ 25% หรือกล่าวในอีกนัยหนึ่งว่าเป็นการแบ่งเมตริกซ์ผู้ใช้-ชิ้นข้อมูล 100% ออกเป็นแบบจำลองสำหรับการสอน 75% แบบจำลองสำหรับการทดสอบ 25%

สำหรับฐานข้อมูลที่ใช้ในการทำการทำนายค่านั้น ประกอบไปด้วยตารางทั้งหมด 8 ตาราง ประกอบไปด้วย ACTOR, CAST, DIRECTION, DIRECTOR, GENRE, MOVIE, RATING และ USER

4.2.2 ตัวอย่างฐานข้อมูลที่ใช้ในระบบ

สำหรับฐานข้อมูลที่ใช้ในระบบนั้นมีอยู่ด้วยกัน 8 ตาราง ได้แก่ ACTOR, DIRECTOR, MOVIE, GENRE, CAST, DIRECTION, USER และ RATING

ตารางที่ 4.1 แสดงฐานข้อมูลในตาราง ACTOR

AID	ANAME
1	Leonardo DiCaprio
2	Kate Winslet
3	Billy Zane
4	Kathy Bates
5	Frances Fisher
6	Mark Hamill
7	Harrison Ford
8	Carrie Fisher
9	Peter Cushing
10	Alec Guinness

ตารางที่ 4.2 แสดงฐานข้อมูลในตาราง DIRECTOR

DID	DNAME
1	James Cameron
2	Steven Quale
3	George Lucas
4	Andrew Adamson
5	Kelly Asbury
6	Conrad Vernon
7	Steven Spielberg
8	Glenn Randall
9	Roger Christian
10	Gore Verbinski

ตารางที่ 4.3 แสดงฐานข้อมูลในตาราง MOVIE

MID	MNAME	RELEASE_DATE	POPPULAR
1	Titanic	1997	1
2	Star Wars	1977	1
3	Shrek 2	2004	1
4	E.T. the Extra-Terrestrial	1982	1
5	Star Wars: Episode I - The Phantom Menace	1999	1
6	Pirates of the Caribbean: Dead Man's Chest	2006	1
7	Spider-Man	2002	1
8	Star Wars: Episode III - Revenge of the Sith	2005	1
9	The Lord of the Rings: The Return of the King	2003	1
10	Spider-Man 2	2004	1

ตารางที่ 4.4 แสดงฐานข้อมูลในตาราง GENRE

MID	GENRE
1	Adventure
1	Drama
1	Romance
2	Action
2	Adventure
2	Family
2	Fantasy
2	Sci-Fi
3	Adventure
3	Animation
3	Comedy
3	Family
3	Fantasy

ตารางที่ 4.5 แสดงฐานข้อมูลในตาราง CAST

MID	DID
1	1
1	2
1	3
1	4
1	5
2	6
2	7
2	8
2	9
2	10
3	11
3	12
3	13
3	14
3	15

ตารางที่ 4.6 แสดงฐานข้อมูลในตาราง DIRECTION

MID	DID
1	1
1	2
2	3
3	4
3	5
3	6
4	7
4	8
5	3
5	9
6	10
7	11

ตารางที่ 4.7 แสดงฐานข้อมูลในตาราง USER

UID	UNAME
1	Tccandler
2	Kgentes
3	erving06us
4	bigdaddygamer14
5	angelunderworld1
6	Jmangell
7	dabenz24
8	arc_angel_bleed
9	bigdaddygamer14
10	Jahrune
11	Tampabayjay

ตารางที่ 4.8 แสดงฐานข้อมูลในตาราง RATING

UID	MID	O_RATING	A_RATING	D_RATING	G_RATING
1	380	5	5	5	5
1	76	4	4	4	4
1	381	5	5	5	5
1	8	3	3	1	2
1	382	4	4	4	4
1	383	5	5	5	5
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
200	44	5	5	5	5
200	193	5	5	5	5
200	213	5	5	5	5
200	16	5	5	5	5
200	333	5	5	5	5
200	2026	5	5	5	5

4.2.3 กระบวนการทดลอง

สำหรับการทดลองนั้นเริ่มต้นด้วยการนำข้อมูลดาต้าเซตเข้าอัลกอริทึม Content-Based Filtering ทำการสร้างชุดข้อมูลเทียมโดยใช้ naïve Bayes Theorem จากนั้นนำข้อมูลที่ได้เข้าสู่กระบวนการ Collaborative Filtering ทำการคำนวณหาค่าความคล้ายคลึงระหว่างผู้ใช้ และทำการทำนายค่าความพึงพอใจจากนั้นก็แสดงผลการแนะนำออกมา

4.3 การประเมินผลการวิจัย

เป็นการวิเคราะห์ข้อมูลจากการทำนายในเชิงสถิติหรือกล่าวอีกนัยหนึ่งได้ว่า เป็นการหาข้อมูลที่เป็นตัวแทนของกลุ่มข้อมูลที่ได้จากการทำนาย เนื่องจากค่าความผิดพลาดจากการทำนายซึ่งล้วนแล้วแต่เป็นค่าที่คาดเดาไม่ได้ทั้งนั้นด้วยคุณสมบัติของค่าความผิดพลาดจากการทำนายที่เปลี่ยนแปลงตลอดเวลาขึ้นอยู่กับข้อมูลที่นำมาใช้ในการทดลอง

ดังนั้นปริมาณทางสถิติหลักๆ ที่จำเป็นต้องหาค่าสำหรับการวิเคราะห์เพื่อทดสอบวิธีการที่นำเสนอ เปรียบเทียบกับวิธีการที่มีอยู่ในปัจจุบันได้แก่ ค่าเฉลี่ย (Mean) และส่วนเบี่ยงเบนมาตรฐาน (SD)

ค่าเฉลี่ย เป็นค่าความผิดพลาดสมบูรณ์เฉลี่ยจากการทดลองทั้งหมดที่ได้อธิบายไว้ในหัวข้อ 2.1.3 (สมการที่ 2.5) หรือกล่าวอีกนัยหนึ่งได้ว่า ในการทดลองแต่ละครั้งค่าความผิดพลาดสมบูรณ์คือค่าเฉลี่ย

จากสมการ 2.5 ค่าสมบูรณ์ของผลต่างระหว่างค่าเรตติ้งจริงที่ผู้ใช้เคยให้ไว้และค่าเรตติ้งที่ได้จากการทดลองจำนวน N ครั้ง และ MAE เป็นค่าความผิดพลาดสมบูรณ์เฉลี่ย

ค่าความแม่นยำ คือ อัตราส่วนของชิ้นข้อมูลที่ถูกต้องจากชิ้นข้อมูลทั้งหมดที่แนะนำออกมา

ค่าความระลึก คือ อัตราส่วนระหว่างชิ้นข้อมูลที่ถูกต้องที่ถูกแนะนำออกมากับชิ้นข้อมูลที่ถูกต้องทั้งหมด

Specificity คือ ความน่าจะเป็นที่ข้อมูลที่ไม่ถูกต้องจะไม่ถูกแนะนำออกมา หาได้จากอัตราส่วนระหว่างชิ้นข้อมูลที่ไม่ถูกต้องที่ไม่ถูกแนะนำออกมากับชิ้นข้อมูลที่ไม่ถูกต้อง

Negative Predictive Value คือ ความน่าจะเป็นที่ข้อมูลที่ไม่ถูกนำเสนอจะเป็นข้อมูลที่ไม่ถูกต้อง หาได้จากอัตราส่วนระหว่างชิ้นข้อมูลที่ไม่ถูกต้องที่ไม่ถูกแนะนำออกมากับชิ้นข้อมูลที่ไม่ถูกแนะนำ

4.4 ผลการทดลอง

งานวิจัยนี้ได้ทดลองวิธีการที่นำเสนอกับวิธีการเดิมที่ใช้อัลกอริทึม CF เพียงอย่างเดียวและอัลกอริทึม CB เพียงอย่างเดียว

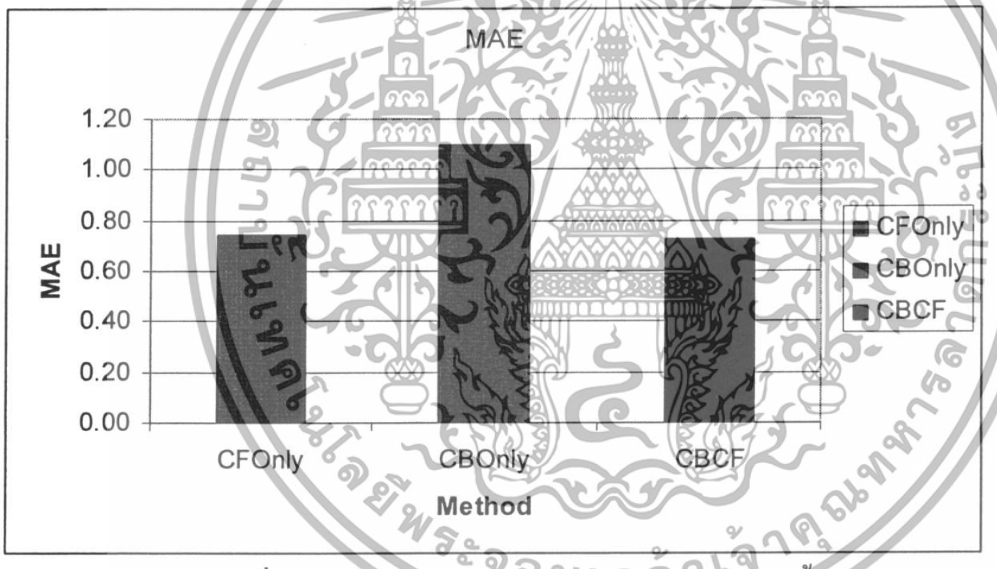
4.4.1 ผลการเปรียบเทียบค่าความผิดพลาดสมบูรณ์เฉลี่ยระหว่างวิธีที่นำเสนอ กับวิธีที่มีอยู่ในปัจจุบัน

เป็นการเปรียบเทียบวิธีที่นำเสนอกับวิธีที่มีอยู่ในปัจจุบัน (CF only และ CB only) โดยผลการทดลองที่ได้ออกมาดังแสดงในตาราง 4.9

ตารางที่ 4.9 แสดงการเปรียบเทียบค่า MAE ของทั้งสามวิธี

Method	MAE
CFOnly	0.74
CBOnly	1.10
CBCF	0.72

จากข้อมูลค่า MAE สามารถแสดงในรูปแบบกราฟได้ดังรูปที่ 4.1



รูปที่ 4.1 กราฟแสดงการเปรียบเทียบค่า MAE ของทั้งสามวิธี

จากกราฟจะเห็นได้ว่าวิธีที่นำเสนอนั้นมีค่าต่ำที่สุดคือ 0.72 แสดงว่ามีค่าผิดพลาดต่ำที่สุดมีความแม่นยำในการทำงานสูง แต่สาเหตุที่ค่า MAE ยังสูงอยู่ และไม่ต่างจากวิธีเดิมมากนัก เป็นเพราะว่า ความหนาแน่นของดาต้าเซต ซึ่งเป็นดาต้าเซตที่เก็บขึ้นมาเองนั้น มีความหนาแน่นของข้อมูลต่ำมาก

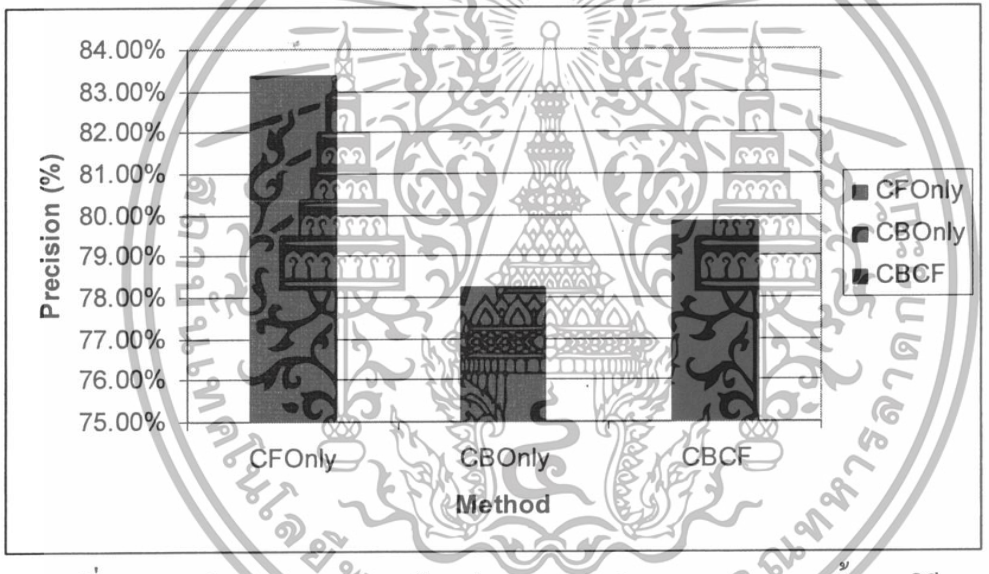
4.4.2 ผลการเปรียบเทียบค่าความแม่นยำระหว่างวิธีที่นำเสนอ กับวิธีที่มีอยู่ในปัจจุบัน

เป็นการเปรียบเทียบวิธีที่นำเสนอกับวิธีที่มีอยู่ในปัจจุบัน (CF only และ CB only) โดยผลการทดลองที่ได้ออกมาดังแสดงในตาราง 4.10

ตารางที่ 4.10 แสดงการเปรียบเทียบค่าความแม่นยำ (Precision) ของทั้งสามวิธี

Method	Precision
CFOnly	83.35%
CBOnly	78.23%
CBCF	79.82%

จากข้อมูลค่าความแม่นยำ สามารถแสดงในรูปแบบกราฟได้ดังรูปที่ 4.2



รูปที่ 4.2 กราฟแสดงการเปรียบเทียบค่าความแม่นยำ (Precision) ของทั้งสามวิธี

จากกราฟแสดงค่าความแม่นยำของการให้การแนะนำจะเห็นได้ว่าวิธีที่นำเสนอมีความแม่นยำค่อนข้างจะอยู่ในระดับสูงแต่ยังอยู่ในระดับที่ต่ำกว่าวิธี CF เพียงอย่างเดียว

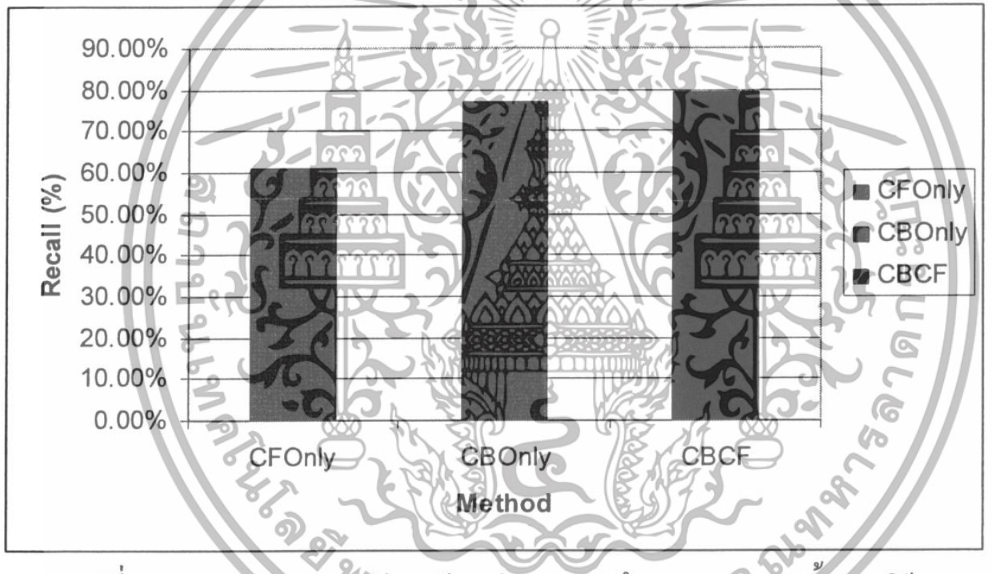
4.4.3 ผลการเปรียบเทียบค่าความระลึกระหว่างวิธีที่นำเสนอ กับวิธีที่มีอยู่ในปัจจุบัน

เป็นการเปรียบเทียบวิธีที่นำเสนอกับวิธีที่มีอยู่ในปัจจุบัน (CF only และ CB only) โดยผลการทดลองที่ได้ออกมาดังแสดงในตาราง 4.11

ตารางที่ 4.11 แสดงการเปรียบเทียบค่าความระลึก (Recall) ของทั้งสามวิธี

Method	Recall
CFOnly	61.11%
CBOnly	76.92%
CFCB	79.35%

จากข้อมูลค่าความระลึก สามารถแสดงในรูปแบบกราฟได้ดังรูปที่ 4.3



รูปที่ 4.3 กราฟแสดงการเปรียบเทียบค่าความระลึก (Recall) ของทั้งสามวิธี

จากกราฟแสดงความระลึกข้างต้นจะเห็นได้ว่าวิธีที่นำเสนอนั้นมีความระลึกสูงที่สุดแต่ไม่ต่างจากวิธี CB Only มากนัก

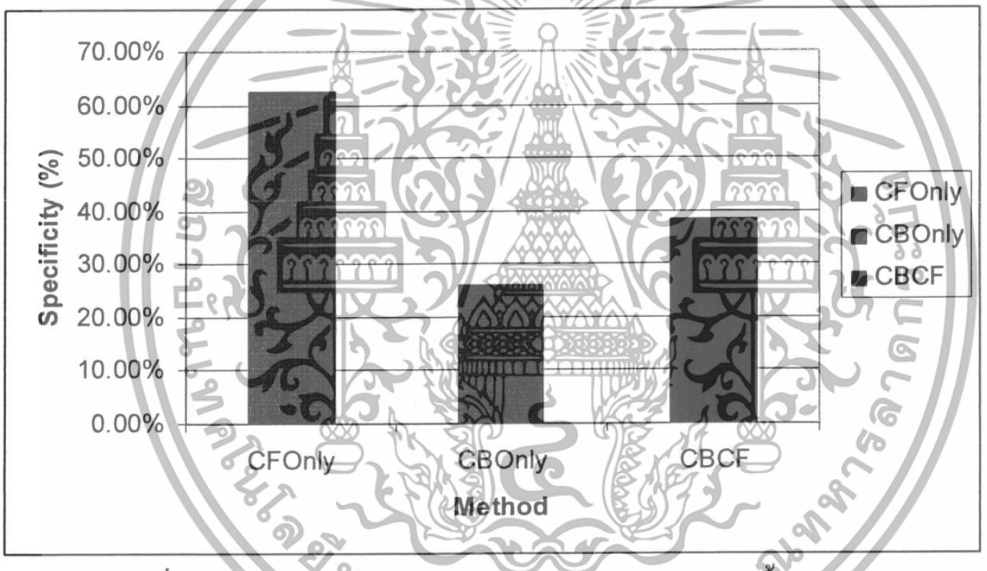
4.4.4 ผลการเปรียบเทียบSpecificity ระหว่างวิธีที่นำเสนอ กับวิธีที่มีอยู่ในปัจจุบัน

เป็นการเปรียบเทียบวิธีที่นำเสนอกับวิธีที่มีอยู่ในปัจจุบัน (CF only และ CB only) โดยผลการทดลองที่ได้ออกมาดังแสดงในตาราง 4.12

ตารางที่ 4.12 แสดงการเปรียบเทียบค่า Specificity ของทั้งสามวิธี

Method	Specificity
CFOnly	62.48%
CBOnly	25.83%
CFCB	38.32%

จากข้อมูล Specificity สามารถแสดงในรูปแบบกราฟได้ดังรูปที่ 4.4



รูปที่ 4.4 กราฟแสดงการเปรียบเทียบค่า Specificity ของทั้งสามวิธี

จากกราฟข้างต้นเป็นการแสดงค่าความน่าจะเป็นที่ข้อมูลที่ไม่ถูกต้องจะไม่ถูกแนะนำออกมาแต่ผลที่ได้นั้นยังต่ำกว่าของวิธี CF อย่างเดียวค่อนข้างมาก

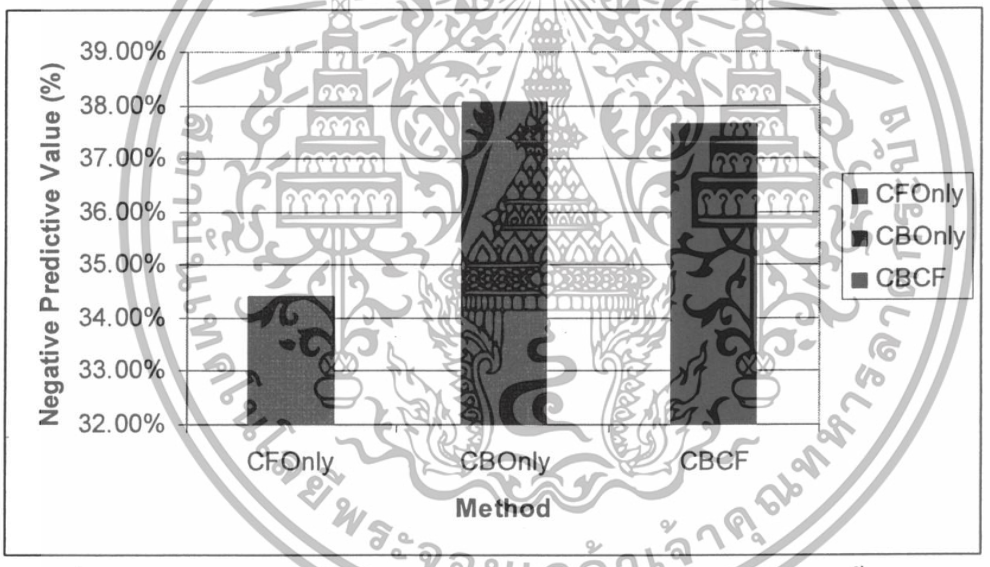
4.4.5 ผลการเปรียบเทียบ Negative Predictive Value ระหว่างวิธีที่นำเสนอ กับวิธีที่มีอยู่ในปัจจุบัน

เป็นการเปรียบเทียบวิธีที่นำเสนอกับวิธีที่มีอยู่ในปัจจุบัน (CF only และ CB only) โดยผลการทดลองที่ได้ออกมาดังแสดงในตาราง 4.13

ตารางที่ 4.13 แสดงการเปรียบเทียบค่า Negative Predictive Value ของทั้งสามวิธี

Method	Negative Predictive Value
CFOnly	62.48%
CBOnly	25.83%
CFCB	38.32%

จากข้อมูล Negative Predictive Value สามารถแสดงในรูปแบบกราฟได้ดังรูปที่ 4.5



รูปที่ 4.5 กราฟแสดงการเปรียบเทียบค่า Negative Predictive Value ของทั้งสามวิธี

จากกราฟข้างต้นเป็นการแสดงค่าความน่าจะเป็นที่ข้อมูลที่ไม่ถูกนำเสนอจะเป็นข้อมูลที่ไม่ถูกต้อง จะเห็นได้ว่า วิธีที่นำเสนอนั้นค่า Negative Predictive Value ออกมาค่อนข้างที่จะสูงแต่ยังน้อยกว่าวิธีที่ใช้ CB อย่างเดียวเล็กน้อย

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้ได้นำเสนอวิธีการเพิ่มประสิทธิภาพให้กับระบบให้คำแนะนำโดยการรวมเอาเทคนิค Content-Based Filtering (CBF) และ Collaborative Filtering (CF) โดยแก้ปัญหาการให้คะแนนเรตติ้งต่อชิ้นข้อมูลที่ไม่ทั่วถึง และปัญหาชิ้นข้อมูลที่ยังไม่ได้มีการให้เรตติ้งไว้ ซึ่งเป็นสาเหตุสำคัญที่ทำให้ผลของการทำนายค่าความพึงพอใจของอัลกอริทึมนั้นมีความผิดพลาดค่อนข้างสูงการทำการ วิธีที่นำมาประยุกต์ใช้ในงานวิจัยนี้คือการนำเอา naïve Bayes Theorem มาทำการสร้างเมตริกซ์เทียมเพื่อทำให้กระบวนการค้นหาค่าความพึงพอใจและกระบวนการหาเพื่อนบ้านทำได้ตรงกับความต้องการของผู้ใช้เป้าหมายและถูกต้องมากยิ่งขึ้น

จากการทดลองที่ผ่านมานั้น ผ่านมานั้นผ่านมาได้ด้วยดี แต่ก็อาจมีปัญหาในการทำงานบ้าง แต่สิ่งเหล่านั้นจะทำให้เราได้เรียนรู้ข้อผิดพลาดเพื่อนำไปพัฒนาศักยภาพในการทำงานต่อไป

จากการทดสอบนำเอาวิธีการที่นำเสนอเปรียบเทียบกับวิธีเดิมด้วยการวัดค่า MAE, Precision, Recall, Specificity และ Negative Predictive Value พบว่า โดยภาพรวมแล้ววิธีที่นำเสนอนั้นให้การแนะนำได้ดีกว่า วิธีเดิมที่ใช้อยู่ในปัจจุบัน แต่ก็ยังมีข้อเสียอยู่บ้าง กล่าวคือ ถึงแม้ว่าข้อมูลที่ถูกแนะนำออกมาจะมีความระลึกค่อนข้างสูงและตรงกับความต้องการของผู้ใช้แต่ข้อมูลผู้ใช้ไม่ต้องการนั้นก็ถูกแนะนำออกมาค่อนข้างสูงเช่นกัน เป็นผลมาจาก ในขั้นตอนการหาเพื่อนบ้านนั้นข้อมูลที่นำมาหานั้นเป็นข้อมูลเทียมที่ระบบสร้างขึ้น ไม่ใช่ข้อมูลจริงของผู้ใช้ทำให้ความคลาดเคลื่อนของการสร้างข้อมูลเทียมมีผลกระทบต่อการหาเพื่อนบ้านของผู้ใช้นั้นๆ

จากการทดลองนี้สามารถแก้ไขปัญหาที่กล่าวมาระบบนี้สามารถแก้ไขปัญหาการให้เรตติ้งต่อชิ้นข้อมูล (Sparsity Problem) ปัญหาการแยกแยะเรตติ้ง (Transparency Problem) และ ปัญหาชิ้นข้อมูลที่ไม่มีการให้เรตติ้งไว้ (First-rater Problem) ลงได้ และสามารถเพิ่มประสิทธิภาพให้กับระบบให้การแนะนำ

ทั้งนี้ผลการทดลองจะขึ้นอยู่กับดาต้าเซตที่นำมาทดสอบด้วย เนื่องจากดาต้าเซตที่นำมาทำการทดลองนั้น เป็นดาต้าเซตที่เก็บขึ้นเองไม่ใช่ดาต้าเซตมาตรฐานที่มีอยู่ในปัจจุบันจึงทำให้ผลการให้คำแนะนำออกมาได้ไม่ดีเท่าที่ควรเนื่องจากการเก็บข้อมูลอาจทำได้ไม่ครบถ้วนไม่ครอบคลุม

ปัญหาที่เกิดขึ้น เช่น ในช่วงแรกของการทำวิจัยนั้นยังไม่มีความรู้เกี่ยวกับ อัลกอริทึม Collaborative Filtering และ Content-Based Filtering ทำให้การดำเนินงานนั้นทำได้ค่อนข้างช้า ในช่วงของการเก็บข้อมูล เนื่องจากข้อมูลคะแนนความพึงพอใจที่ต้องการนั้นเป็นข้อมูลที่ไม่มีอยู่ในดาต้าเซตทั่วๆไป ทำให้ต้องทำการเก็บดาต้าเซตเอง ซึ่งต้องใช้เวลาค่อนข้างนานในการเก็บข้อมูล

อีกทั้งปัญหาระหว่างการ implement โปรแกรมนั้น บางครั้งการทำงานของอัลกอริทึมไม่ได้ผลอย่างที่ควรจะเป็น จึงต้องทำการปรับแก้ไปเรื่อยๆจนได้ระบบที่มีประสิทธิภาพมากที่สุดเท่าที่จะเป็นได้

ดังนั้นจึงสรุปได้ว่าวิธีการที่นำเสนอนั้นสามารถช่วยลดความผิดพลาดจากการทำนายของวิธี ที่ใช้ CF และ CB เพียงอย่างเดียวลงได้ในระดับหนึ่งและสามารถช่วยเพิ่มประสิทธิภาพให้กับอัลกอริทึมให้ผลการทำนายค่าความพึงพอใจได้ถูกต้องมากยิ่งขึ้น

5.2 ข้อเสนอแนะ

1. วิธีที่นำเสนอนั้นเป็นวิธีการซึ่งนำเอาคุณสมบัติของชิ้นข้อมูลมาพิจารณาร่วมกับอัลกอริทึม Collaborative Filtering และใช้ คะแนนความพึงพอใจมาใช้ในกระบวนการหาค่าความพึงพอใจถึง 4 ค่าด้วยกันซึ่งล้วนเป็นประโยชน์สำหรับการทำนายหาสิ่งที่ผู้ใช้สนใจเป็นอย่างดี
2. วิธีการที่นำเสนอนั้นใช้ในการแนะนำภาพยนตร์เพียงอย่างเดียวหากจะนำไปประยุกต์กับการทำนายค่าความพึงพอใจต่อสิ่งอื่นๆเช่น หนังสือ เพลง สามารถทำได้แต่ต้องทำการเปลี่ยนค่า โดเมน และ แฟคเตอร์ ที่เกี่ยวข้องกับสิ่งนั้นๆ
3. ระบบนี้ใช้ได้ดีกับภาพยนตร์ในประเทศสหรัฐอเมริกาเนื่องจากข้อมูลในดาต้าเซตนั้นส่วนใหญ่นั้นเป็นภาพยนตร์ในประเทศสหรัฐอเมริกา ระบบนี้สามารถทำได้ดีกับภาพยนตร์อื่น หากเพิ่มข้อมูลเข้าไปในดาต้าเซต
4. การทำนายความพึงพอใจจะออกมาได้ดีถ้าหากข้อมูลในดาต้าเซตมีจำนวนมากและครอบคลุมทุกคุณสมบัติ
5. ในส่วนการประเมินผล งานวิจัยนี้ใช้เพียงข้อมูลเรตติ้งที่มีอยู่แล้วในอดีต แต่หากในอนาคตความชอบอาจเปลี่ยนแปลง เช่น ที่ผ่านมาผู้ใช้อาจไม่ชอบ ชิ้นข้อมูลเหล่านั้นแต่ในอนาคตก็มีความเป็นไปได้ที่จะเริ่มชอบข้อมูลนั้น ผู้ใช้ก็สามารถเปลี่ยนแปลงคะแนนความชอบที่มีต่อชิ้นข้อมูลนั้นๆได้ เพื่อให้การแนะนำออกมาดีควรจะมีการประเมินความพึงพอใจของผู้ใช้อยู่ตลอดเวลา ว่าผู้ใช้มีความพึงพอใจต่อผลการทำนายมากน้อยเพียงใด
6. วิธีการที่นำเสนอ ถือเป็นวิธีการทำนายหาค่าความพึงพอใจของผู้ใช้ที่มีความถูกต้องสูงเมื่อเปรียบเทียบกับอัลกอริทึมแบบเดิม และเป็นพื้นฐานสำคัญที่จะนำไปพัฒนาเป็นการคำนวณแบบ Top-N Recommendation ต่อไปได้อย่างเหมาะสม

บรรณานุกรม

ชัยวัฒน์ ติรวีรชจร. 2548. "การทำนายข้อมูลโดยการรวม Content-Based Filtering with Item-Based Collaborative Filtering ด้วยกฎความสัมพันธ์." วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชา วิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

Adomavicius G. and Kwon Y. 2007. "New Recommendation Techniques for Multicriteria Rating Systems." **IEEE Computer Society May/June 2007.** : 48-55.

Chaiwat T. and Quan P. 2004. "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.

Maneeroj M. and Bhattarakosol P. 2006. "Hybrid System Based on Intelligent Neighbor Formation Algorithm." International Conference on Web Intelligence (WI 2006 Main Conference Proceeding)(WI'06).

Melville P., Mooney R. J. and Nagarajan R. 2002. "Content-Boosted Collaborative Filtering for Improved Recommendations." **The Eighteenth National Conference on Artificial Intelligence(AAAI-2002).** : 187-192.

Narisa. **Item-based Collaborative Filtering #1**. [Online]. Available : http://www.narisa.com/blog/iwat/index.php?showentry=360.

Salter J. and Antonopoulos N. 2006. "CinemaScreen Recommender Agent: Combining Collaborative and Content-Based Filtering." **IEEE Computer Society January/February 2006.** : 35-41.

ภาคผนวก ก

ดาต้าเซตภาพยนตร์

ดาต้าเซตที่นำมาใช้นั้นเป็นดาต้าเซตที่นำข้อมูลเรตติ้งและผู้ใช้มาจาก Yahoo Movies และข้อมูลภาพยนตร์มาจาก IMDB โดยมีเรตติ้งภาพยนตร์ทั้งหมด 10,000 เรตติ้งซึ่งมาจากผู้ใช้จำนวน 200 คน ภาพยนตร์จำนวน 2,651 เรื่อง

โดยข้อมูลภายในดาต้าเซตประกอบไปด้วยชื่อผู้ใช้ เรตติ้งภาพยนตร์ ชื่อภาพยนตร์ ปีที่สร้าง นักแสดง ผู้กำกับ และ คุณสมบัติของภาพยนตร์ ซึ่งคุณสมบัติของภาพยนตร์นั้นประกอบไปด้วย 22 คุณสมบัติด้วยกันคือ Action, Adventure, Animation, Biography, Comedy, Crime, Documentary, Drama, Family, Fantasy, History, Horror, Music, Musical, Mystery, Romance, Sci-Fi, Short, Sport, Thriller, War, Western