บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของปัญหาที่ทำการวิจัย

การค้นพบรูปแบบที่ปรากฏบ่อย หรือ frequent pattern discovery เป็นการค้นหารูปแบบหรือแพทเทิร์นที่เกิดขึ้นซ้ำๆ ในข้อมูลขนาดใหญ่ เช่น การค้นพบจากฐานข้อมูลทรานแซคชันของห้างสรรพสินค้าสาขาบางแค พบว่าลูกค้านิยมซื้อนมผงและชาเขียว ในขณะที่การค้นพบจาก ทรานแซคชันของสาขานครราชสีมาระบุว่าลูกค้าซื้อนมผงพร้อมกับไข่ไก่ รูปแบบการซื้อสินค้าดังตัวอย่างนี้จะช่วยสะท้อนถึงพฤติกรรมของผู้บริโภคในแต่ละภูมิภาค ความรู้เช่นนี้จะเป็นประโยชน์ต่อผู้บริหารในการวางแผนจัดวางชั้นสินค้า รวมไปถึงการวางแผนจัดรายการกระตุ้นยอดขายสินค้า เช่นจากพฤติกรรมการบริโภคของลูกค้าในย่านบางแค ถ้าผู้จัดการห้างสรรพสินค้าทราบว่ากำไรต่อหน่วยของชาเขียวสูงกว่านมผงมาก การตัดสินใจลดราคานมผงให้ต่ำกว่าราคาจำหน่ายของร้านค้าทั่วไป จึงคาดได้ว่าน่าจะดึงดูดลูกค้าที่ต้องการซื้อนมผงเข้ามาซื้อของในห้างสรรพสินค้า และคาดหมายต่อไปได้ว่าลูกค้าที่ซื้อนมผงจะซื้อชาเขียวร่วมด้วย ซึ่งจะส่งผลให้ยอดขายและกำไรโดยรวมของการจำหน่ายนมผงและชาเขียวสูงขึ้น

การค้นพบความสัมพันธ์หรือความเชื่อมโยงของสินค้าที่ถูกซื้อร่วมกันบ่อยนี้ เรียกว่า การทำเหมืองเพื่อค้นหาความสัมพันธ์ (association mining) ซึ่งอาศัยขั้นตอนพื้นฐานที่สำคัญคือ การค้นพบรูปแบบที่ปรากฏบ่อย ลักษณะของรูปแบบที่ปรากฏบ่อย หรือ frequent patterns มีได้หลากหลายประเภท เช่น รูปแบบการซื้อสินค้าของลูกค้าส่วนใหญ่ในห้างสรรพสินค้า รูปแบบการเกิดแผ่นดินไหวในรอบสิบปีที่ผ่านมาของภูมิภาคเอเชียตะวันออกเฉียงใต้ รูปแบบการค้นหาข้อมูลจากเว็บเพจต่างๆของผู้ใช้อินเทอร์เน็ต รูปแบบการจัดโครงสร้างโมเลกุลในสารประกอบโปรตีน เป็นต้น

นับตั้งแต่ปีค.ศ.1993 ที่ได้มีการเริ่มเสนอแนวคิดและเทคนิคของการทำเหมืองข้อมูลประเภทการค้นพบรูปแบบที่ปรากฏบ่อยโดย Rakesh Agrawal, Tomasz Imielinski และ Arun Swami นักวิจัยของศูนย์วิจัยไอบีเอ็มที่อัลมาเดน สหรัฐอเมริกา งานวิจัยด้านนี้ก็ได้รับความนิยมอย่างกว้างขวาง เนื่องจากใช้ประโยชน์ได้กับการวิเคราะห์ข้อมูลในหลากหลายสาขา มีการพัฒนาเทคนิคต่างๆของการค้นหารูปแบบที่ปรากฏบ่อยให้มีประสิทธิภาพสูง แต่จากความก้าวหน้าของเทคโนโลยีอินเทอร์เนตทำให้รูปแบบของข้อมูลมีลักษณะเปลี่ยนแปลงไปจากข้อมูลที่มีขนาดใหญ่แต่คงตัว (static) กลายเป็นข้อมูลที่มีลักษณะพลวัต (dynamic) เปลี่ยนแปลงได้ทั้งการเพิ่มปริมาณอย่างต่อเนื่องและการเปลี่ยนรูปแบบการกระจายของข้อมูล ข้อมูลที่มีปริมาณเพิ่มได้ไม่จำกัดเช่นนี้เรียกว่า ข้อมูลสตรีม

ในช่วงเวลาทศวรรษที่ผ่านมานักวิจัยจำนวนหนึ่งในสาขาการทำเหมืองข้อมูล เริ่มให้ความสนใจกับลักษณะของข้อมูลสตรีม และพยายามปรับปรุงเทคนิคการค้นหารูปแบบที่ปรากฏบ่อยที่ใช้งาน

ได้ดีกับข้อมูลคงตัว ให้ทำงานกับข้อมูลสตรีมได้ แต่เทคนิคเหล่านั้นยังมีข้อจำกัดและใช้ได้กับเพียงบางแอพพลิเคชัน ผู้วิจัยจึงได้เสนอโครงการวิจัยนี้ขึ้นเพื่อพัฒนาเทคนิคการค้นหารูปแบบที่ปรากฏบ่อยจากข้อมูลสตรีมให้ทำงานได้ได้กับหลากหลายแอพพลิเคชัน เช่น web-page click stream, DNA stream, market basket data stream และเพื่อให้ได้ผลลัพธ์ของการค้นหารูปแบบที่ปรากฏบ่อยในเวลาที่รวดเร็วเพื่อทันต่อความต้องการใช้งาน การประมวลผลข้อมูลสตรีมจะใช้การประมาณค่าด้วยกระบวนการ stochastic ในขั้นตอนของการพัฒนาโปรแกรมต้นแบบจะใช้แนวทางการโปรแกรมเชิงฟังก์ชันด้วยภาษา Haskell (www.haskell.org) ที่มีลักษณะเด่นคือโครงสร้างของภาษาสนับสนุนการทำงานกับแพทเทิร์นและใช้การประมวลผลด้วยเทคนิค lazy evaluation ที่ช่วยให้สามารถทำงานกับข้อมูลที่ปริมาณเพิ่มขึ้นอย่างไม่มีขีดจำกัด การปรับปรุงโปรแกรมต้นแบบให้มีประสิทธิภาพสูงขึ้นผู้วิจัยใช้วิธีการโปรแกรมเชิงฟังก์ชันด้วยภาษา Erlang (www.erlang.org) เนื่องจากมีลักษณะเป็น concurrent functional language ที่สนับสนุนการประมวลผลแบบพร้อมกันบนสถาปัตยกรรม multicore

การพัฒนาเทคนิคการค้นพบรูปแบบที่ปรากฏบ่อยในข้อมูลสตรีมโดยใช้ stochastic process และการทำ rapid prototyping กับข้อมูลปริมาณไม่จำกัดด้วยวิธีการโปรแกรมเชิงฟังก์ชัน จะเป็นองค์ความรู้ใหม่ที่ช่วยพัฒนาความก้าวหน้าของงานวิจัยในสาขาการทำเหมืองข้อมูลประเภทการค้นพบรูปแบบที่ปรากฏบ่อย และการค้นพบความสัมพันธ์

1.2 วัตถุประสงค์ของโครงการ

1) เพื่อพัฒนาเทคนิคและอัลกอริทึมในการค้นพบโดยประมาณของรูปแบบที่ปรากฏบ่อยในข้อมูลสตรีม
2) พัฒนา framework ของ stochastic process ที่เหมาะสมกับการทำ association mining กับข้อมูลสตรีม
3) ออกแบบและพัฒนาโปรแกรมต้นแบบในการทำ approximate association mining ด้วยเทคนิคการโปรแกรมเชิงฟังก์ชัน

1.3 ผลงานวิจัยที่เกี่ยวข้อง

การทำเหมืองจากฐานข้อมูลทรานแซคชันขนาดใหญ่ เพื่อค้นหาความสัมพันธ์หรือความเกี่ยวข้องของสินค้าที่ถูกซื้อร่วมกันบ่อย ได้รับการเสนอแนวคิดในปี 1993 โดยทีมนักวิจัยของ IBM Almaden Research Center ที่ประกอบด้วย Rakesh Agrawal, Tomasz Imielinski และ Arun Swami โดยได้เสนออัลกอริทึมที่ต่อมาภายหลังนิยมเรียกว่า อัลกอริทึม AIS เพื่อค้นหาความสัมพันธ์และความเชื่อมโยงของสินค้าแล้วแสดงในลักษณะของกฎ ซึ่งก็คือข้อความในรูปแบบ ถ้า..แล้ว เรียกข้อความหรือกฎนี้ว่า *กฎความสัมพันธ์* (association rule) และมักจะเขียนอยู่ในรูปแบบ $X \rightarrow Y$ (s, c) แทน

ความหมายว่าเมื่อลูกค้าซื้อสินค้า X แล้วลูกค้าจะซื้อสินค้า Y ร่วมด้วย โดย X และ Y คือเซตของสินค้าหรือไอเท็มเซต ที่ไม่ใช่เซตว่างและเป็นเซตที่สมาชิกไม่ซ้ำกัน สัญลักษณ์ s แทนค่า support หรือค่าสนับสนุน ซึ่งหมายถึงสัดส่วนของจำนวนเรคคอร์ดที่ปรากฏทั้ง X และ Y ต่อจำนวนเรคคอร์ดทั้งหมดในฐานข้อมูลทรานแซคชัน ส่วนสัญลักษณ์ c แทนค่า confidence หรือค่าความเชื่อมั่นของกฎที่คำนวณได้จากสัดส่วนของจำนวนเรคคอร์ดที่ปรากฏทั้ง X และ Y ต่อจำนวนเรคคอร์ดที่ปรากฏ X เมื่อมีการกำหนด support และ confidence เป็นเกณฑ์ หรือมาตรวัด วิธีการค้นหากฎความสัมพันธ์จึงประกอบด้วยสองขั้นตอนหลัก คือ

1. ค้นหาไอเท็มเซตที่ปรากฏบ่อยทั้งหมด โดยไอเท็มเซตที่จัดเป็นเซตที่ปรากฏบ่อยจะต้องมีค่าสนับสนุนเท่ากับหรือสูงกว่าค่าสนับสนุนที่กำหนดไว้เป็นเกณฑ์ขั้นต่ำ (เรียกเกณฑ์ขั้นต่ำนี้ว่า minimum support หรือ min_sup)

2. สร้างกฎความสัมพันธ์จากไอเท็มเซตที่ปรากฏบ่อย โดยกฎนี้จะต้องมีค่าความเชื่อมั่นเท่ากับหรือสูงกว่าค่าความเชื่อมั่นที่กำหนดไว้เป็นเกณฑ์ขั้นต่ำ (เรียกว่า minimum confidence หรือ min_conf)

ตัวอย่างเช่น ถ้าฐานข้อมูลทรานแซคชันประกอบด้วย ข้อมูลการซื้อสินค้าของลูกค้า 5 ราย (รายละเอียดดังตารางที่ 1.1) การซื้อสินค้าของลูกค้าแต่ละรายจะเป็นแต่ละทรานแซคชันที่ระบุด้วยหมายเลขทรานแซคชัน หรือ TID (transaction identifier) และ Items หมายถึงรายการสินค้าที่ลูกค้าซื้อ

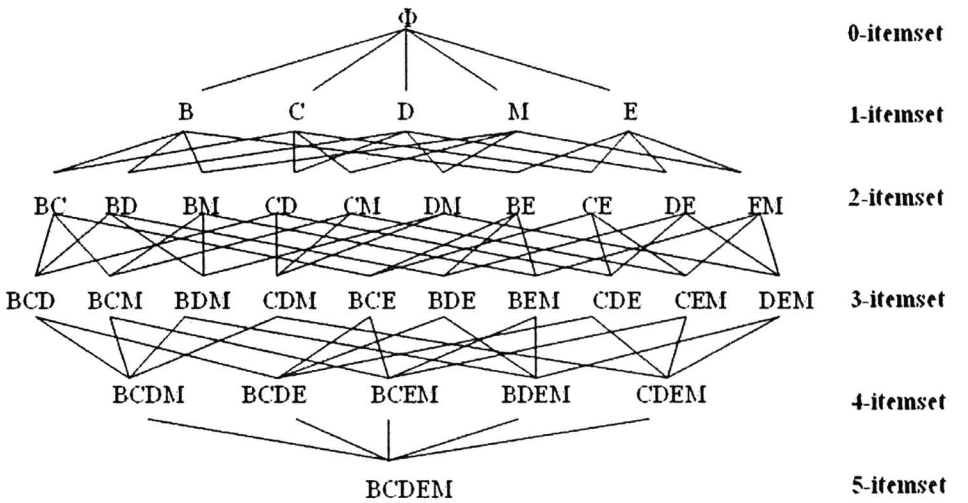
ตารางที่ 1.1 ข้อมูลตัวอย่างแสดงรายการสินค้าที่ลูกค้าซื้อจากร้านค้า

TID	Items
1	{Cereal, Milk}
2	{Beer, Cereal, Diaper, Egg}
3	{Beer, Diaper, Milk}
4	{Beer, Cereal, Diaper, Milk}
5	{Diaper, Milk}

ในฐานข้อมูลตัวอย่างมีสินค้า 5 ชนิดได้แก่ Beer, Cereal, Diaper, Egg, Milk สินค้าเหล่านี้เรียกว่าไอเท็ม กำหนดให้ในการค้นหาสินค้าที่ถูกซื้อร่วมกันบ่อย มีค่า min_sup เป็น 3/5 หรือ 60% และค่า min_conf เป็น 90%

ขั้นตอนแรกของการค้นหากฎความสัมพันธ์ จะเป็นการค้นหาไอเท็มเซตที่ปรากฏบ่อยทั้งหมด โดยจะต้องค้นหาจากไอเท็มเซตที่ไม่ใช่เซตว่างทั้งหมด 31 เซต แสดงไอเท็มเซตทั้งหมดได้ดังรูป

ที่ 1.1 (แต่ละไอเท็มในรูปใช้อักษรย่อ B, C, D, E,M แทน Beer, Cereal, Diaper, Egg, Milk ตามลำดับ และละเว้นการใช้สัญลักษณ์เซตเพื่อให้อ่านง่าย)



รูปที่ 1.1 โครงข่ายของไอเท็มเซตทั้งหมดที่ต้องพิจารณาเพื่อค้นหาไอเท็มเซตที่ปรากฏบ่อย

เมื่อนับค่า support ของแต่ละไอเท็มเซตทั้ง 31 เซตเพื่อคัดเลือกเฉพาะไอเท็มเซตที่ผ่านเกณฑ์ min_sup 60% จะประกอบด้วยไอเท็มเซต 6 เซต ต่อไปนี้

1-itemset: {Beer}, {Cereal}, {Diaper}, {Milk}

2-itemset: {Beer and Diaper}, {Diaper and Milk}

3-itemset: { }

4-itemset: { }

5-itemset: { }

ขั้นตอนที่สองของการสร้างกฎความสัมพันธ์ จะเป็นการนำข้อมูลในไอเท็มเซตที่ปรากฏบ่อยมาสร้างเป็นกฎ โดยเริ่มจาก 2-itemset, 3-itemset, ... ไปตามลำดับ แต่เนื่องจากในตัวอย่างนี้ไอเท็มเซตที่ปรากฏบ่อยตั้งแต่ 3-itemset เป็นต้นไปเป็นเซตว่าง จึงพิจารณาเฉพาะข้อมูลใน 2-itemset ที่ประกอบด้วยสองเซตคือ {Beer and Diaper} และ {Diaper and Milk} นำไอเท็มใน 2-itemset มาสร้างกฎความสัมพันธ์ในเบื้องต้นได้ 4 กฎ ดังนี้

Beer $\longrightarrow$ Diaper (confidence = 3/3 = 100%)

Diaper $\longrightarrow$ Beer (confidence = 3/4 = 75%)

Diaper $\longrightarrow$ Milk (confidence = 3/4 = 75%)

Milk $\longrightarrow$ Diaper (confidence = 3/4 = 75%)

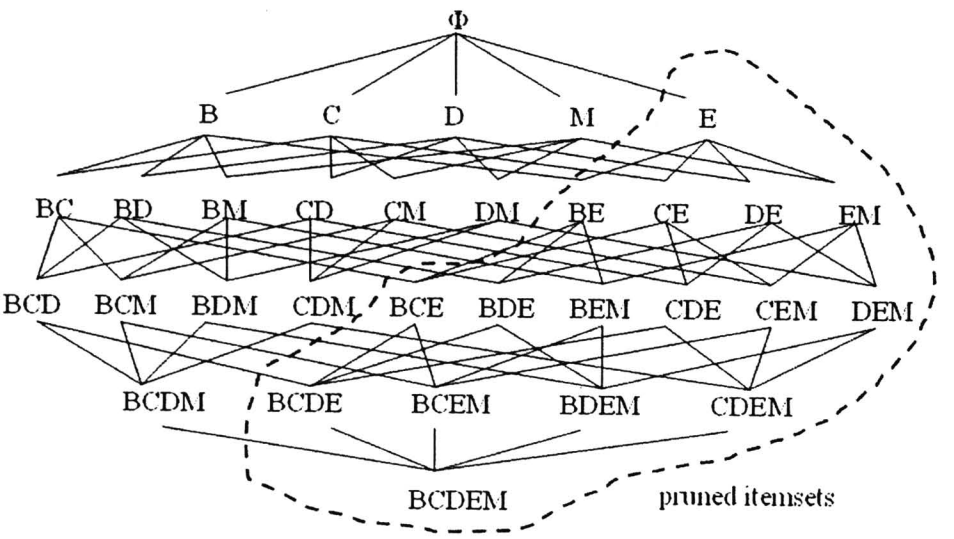
แต่จากเกณฑ์ min_conf ที่กำหนดไว้ 90% จึงทำให้ในผลลัพธ์สุดท้ายได้กฎความสัมพันธ์เพียงข้อเดียวคือ

Beer ⟶ Diaper (confidence = 3/3 = 100%)

กฎความสัมพันธ์นี้ระบุด้วยความเชื่อมั่น 100% ว่าลูกค้าของร้านค้านี้จำนวนมากถึง 60% ที่ตั้งใจมาซื้อ Beer แล้วจะซื้อ Diaper ด้วย

จากจุดเริ่มต้นของการเสนออัลกอริทึม AIS ในปีค.ศ. 1993 (Agrawal, Imielinski & Swami, 1993) ทำให้แนวคิดของการค้นหาความรู้จากฐานข้อมูลหรือการทำเหมืองข้อมูล ประเภทการค้นหาความสัมพันธ์ ได้รับความสนใจอย่างมากจากนักวิจัยในสาขาการทำเหมืองข้อมูลและการวิเคราะห์ข้อมูล ในปีต่อมา Rakesh Agrawal และ Ramakrishnan Srikant (1993; 1994) ได้ปรับปรุงอัลกอริทึม AIS ให้ทำงานได้เร็วขึ้นโดยอาศัยคุณสมบัติที่เรียกว่า Apriori property ที่ระบุว่า "เซตย่อยของไอเท็มเซตที่ปรากฏบ่อย จะต้องเป็นเซตที่ปรากฏบ่อยด้วยเช่นกัน" และเรียกชื่ออัลกอริทึมที่พัฒนาขึ้นใหม่นี้ว่า อัลกอริทึม APRIORI จุดเด่นของอัลกอริทึมนี้อยู่ที่ความสามารถในการพัฒนาความเร็วในการค้นหาไอเท็มเซตที่ปรากฏบ่อย ด้วยการละเว้นการพิจารณาไอเท็มเซตที่ปรากฏซ้ำด้วยความถี่ต่ำกว่าเกณฑ์ min_sup

ดังตัวอย่างข้อมูลทรานแซคชันในตารางที่ 1.1 ถ้ากำหนด min_sup = 2/5 หรือ 40% จะพบว่าไอเท็ม Egg (หรือ E) ปรากฏในทรานแซคชันที่ 2 เพียงทรานแซคชันเดียว จึงไม่จัดเป็นไอเท็มเซตที่ปรากฏบ่อย ด้วยคุณสมบัติ Apriori ทำให้เราสามารถตัดการพิจารณาไอเท็มเซตทุกเซตที่มี E เป็นสมาชิก จึงลดจำนวนไอเท็มเซตที่ต้องพิจารณาจาก 31 เซตลงเหลือเพียง 15 เซต ดังแสดงในรูปที่ 1.2 (ส่วนที่ล้อมรอบด้วยเส้นประ คือไอเท็มเซตที่ไม่ต้องพิจารณาเนื่องจากปรากฏบ่อยไม่ถึงเกณฑ์ min_sup)



รูปที่ 1.2 โครงข่ายของไอเท็มเซตที่ขนาดเล็กลงจากการไม่ต้องพิจารณาไอเท็ม E

นับจากความสำเร็จของการคิดค้นอัลกอริทึม APRIORI (Agrawal & Srikant, 1993; 1994) ได้มีนักวิจัยจำนวนมากใช้แนวคิดของอัลกอริทึมนี้เป็นพื้นฐาน และปรับปรุงประสิทธิภาพให้ดีขึ้นด้วยวิธีต่างๆ กัน งานวิจัยเด่นในกลุ่มนี้ประกอบด้วยงานวิจัยในปี 1995 ของ Park, Chen และ Yu ที่เสนอแนวทางของการใช้เทคนิคแฮชชิงเพื่อเพิ่มความเร็วของการค้นหาไอเท็มเซตที่ปรากฏบ่อย Han และ Fu (1995) ได้เสนอให้สร้างกฎความสัมพันธ์ที่มีโครงสร้างหลายระดับชั้นเพื่อให้มีความละเอียดมากขึ้น เช่น จากค้นพบกฎความสัมพันธ์ Beer $\rightarrow$ Diaper สามารถสืบค้นให้ละเอียดขึ้นได้ว่าเบียร์ที่ลูกค้านิยมซื้อร่วมกับผ้าอ้อมเด็กนั้นเป็นเบียร์ประเภทใด

ในกรณีของการค้นหาความสัมพันธ์จากฐานข้อมูลทรานแซคชันที่มีขนาดใหญ่มากจนกระทั่งไม่สามารถบรรจุข้อมูลทั้งหมดในหน่วยความจำหลักได้ Savasere, Omiecinski และ Navathe (1995) เสนอให้ใช้เทคนิคการแบ่งข้อมูลเป็นส่วนย่อย แล้วทยอยค้นหาความสัมพันธ์ที่ปรากฏในแต่ละส่วนย่อยนั้น Toivonen (1996) ได้ใช้แนวทางที่ต่างออกไปโดยเสนอการใช้เทคนิคการสุ่มเพื่อค้นหาความสัมพันธ์จากข้อมูลตัวแทน Cheung และคณะ (Cheung, Han, Ng, & Wong, 1996) ได้เสนอให้ใช้เทคนิค incremental หรือการทำงานกับข้อมูลครั้งละไม่มาก และเมื่ออ่านข้อมูลใหม่เพิ่มเติมจะต้องสามารถปรับกฎความสัมพันธ์ให้ถูกต้องสอดคล้องกับข้อมูลใหม่ได้ นอกจากนี้ยังมีนักวิจัยจำนวนมาก ได้แก่ Park, Chen และ Yu (1995), Agrawal และ Shafer (1996), Cheung และคณะ (Cheung et al., 1996), Zaki และคณะ (Zaki, Parthasarathy, Ogihara, & Li, 1997) ได้เสนอแนวทางการพัฒนาความเร็วในการค้นหาไอเท็มเซตที่ปรากฏบ่อย ด้วยการประมวลผลแบบขนาน

งานวิจัยที่กล่าวถึงข้างต้นล้วนแต่ใช้แนวทางของอัลกอริทึม APRIORI แต่เสริมประสิทธิภาพความเร็วด้วยเทคนิคต่างๆกัน แต่งานวิจัยของ Jiawei Han, Jian Pei และ Yiwen Yin ในปี 2000 (Han, Pei, & Yin, 2000) ได้เสนอแนวทางที่แตกต่างออกไป ด้วยการอ่านข้อมูลแล้วสร้างโครงสร้างต้นไม้เรียกว่า frequent pattern tree หรือ FP-tree จากนั้นใช้อัลกอริทึม FP-growth เพื่อค้นหาไอเท็มเซตที่ปรากฏบ่อย โดยไม่ต้องสร้างไอเท็มเซตชั่วคราวเพื่อจะตัดทิ้งภายหลังเมื่อค่าสนับสนุนของเซตต่ำกว่าเกณฑ์ค่าสนับสนุนขั้นต่ำ วิธีนี้เมื่อเปรียบเทียบกับวิธี ARIORI แล้วสามารถเพิ่มความเร็วในการค้นหาไอเท็มเซตที่ปรากฏบ่อย แนวทางการค้นหาไอท็มเซตที่ปรากฏบ่อยด้วยโครงสร้าง FP-tree ได้รับความสนใจและพัฒนาต่อเนื่องมาโดยลำดับดังปรากฏในงานวิจัยต่างๆในช่วงปี 2000 ถึงปัจจุบัน (Agrawal, Agrawal, & Prasad, 2001; Pei at al., 2001; Liu, Pan, Wang, & Han, 2002; Ghoting & Parthasarathy, 2004)

ในช่วงระยะเวลาตั้งแต่ปี 1993 ถึง 2000 เทคนิคการค้นพบกฎความสัมพันธ์และการค้นหารูปแบบที่ปรากฏบ่อยได้รับการพัฒนาอย่างต่อเนื่องให้มีประสิทธิภาพสูง และสามารถรองรับข้อมูลขนาดใหญ่ได้ แต่เมื่อเทคโนโลยีอินเทอร์เน็ตได้รับความนิยมสูงขึ้น ลักษณะของข้อมูลเปลี่ยนจาก offline เป็น online และปริมาณของข้อมูลเพิ่มขึ้นอย่างไม่มีขีดจำกัดเกิดเป็นลักษณะข้อมูลสตรีม หรือ data stream

ข้อมูลสตรีมเริ่มได้รับการนิยาม (Guha, Koudas, & Shim, 2001; Babcock et al., 2002; Gaber, Zaslavsky, & Krishnaswamy, 2005; Jiang & Gruenwald, 2006) เมื่อปี 2001 ว่าหมายถึง ข้อมูลที่เกิดขึ้นอย่างต่อเนื่อง ไม่มีขีดจำกัดในเรื่องของปริมาณและจุดสิ้นสุด ข้อมูลถูกส่งออกจากแหล่งผลิตด้วยความเร็วสูงและอาจจะมีการกระจายของข้อมูลที่ไม่คงที่ นักวิจัยได้พยายามปรับปรุงเทคนิคการค้นพบกฎความสัมพันธ์ให้ทำงานได้กับข้อมูลสตรีม เช่น ในปี 2005 Halatchev และ Gruenwald ได้ปรับปรุงเทคนิคการค้นพบกฎความสัมพันธ์ให้สามารถประเมินข้อมูลที่หายไปในข้อมูลสตรีมที่รับมาจากเซนเซอร์ที่ส่งผ่านเครือข่าย Kargupta และคณะ (Kargupta et al., 2004) ได้พัฒนาระบบ VEDAS เมื่อปี 2004 ให้สามารถตรวจจับยานพาหนะในเวลาจริง นอกจากนี้ยังมีงานวิจัยจำนวนมากในช่วงระยะเวลาปี 2004-2005 (Cai et al., 2004; Chang & Lee, 2004; Charikar, Chen, & Farah-Colton, 2004; Chi, Wang, Yu, & Muntz, 2004; Gaber, Zaslavsky, & Krishnaswamy, 2004; Ghoting & Parthasarathy, 2004; Li, Lee, & Shan, 2004; Teng, Chen, & Yu, 2004; Yu, Chong, Lu, & Zhou, 2004; Lin, Chiu, Wu, & Chen, 2005; Mao et al., 2005) ที่มุ่งพัฒนาเทคนิคในการค้นหารูปแบบความสัมพันธ์และไอเท็มเซตที่ปรากฏบ่อยในข้อมูลสตรีม อัลกอริทึมต่างๆที่ถูกเสนอนี้ บางอัลกอริทึมพัฒนาขึ้นเพื่อรองรับเฉพาะบางแอพพลิเคชัน บางอัลกอริทึมใช้ค้นหาความสัมพันธ์เฉพาะข้อมูลในบางช่วง เช่น เฉพาะข้อมูลที่เกิดขึ้นล่าสุดในสตรีม และบางอัลกอริทึมทำงานกับข้อมูลสตรีมในลักษณะ offline

1.4 แนวทางการวิจัย

โครงการวิจัยที่ผู้วิจัยเสนอขึ้นนี้ ต้องการพัฒนาเทคนิคการค้นหารูปแบบที่ปรากฏบ่อยในข้อมูลสตรีมในลักษณะ online ให้ทำงานกับข้อมูลสตรีมได้หลากหลายแอพพลิเคชัน และเพื่อให้ได้รูปแบบความสัมพันธ์ในเวลาที่รวดเร็วทันต่อความต้องการใช้งาน การค้นหาความสัมพันธ์และรูปแบบที่ปรากฏบ่อยจะมุ่งเน้นที่การค้นพบโดยประมาณ การประมาณค่าความถี่ของการปรากฏรูปแบบจะปรับปรุงจากเทคนิค Monte Carlo approximation ที่ผู้วิจัยและคณะได้นำเสนอไว้ในงานวิจัยก่อนหน้านี้ (Kerdprasop, Kerdprasop, & Sattayatham, 2006) ในการพัฒนาโปรแกรมต้นแบบจะใช้แนวทางการโปรแกรมเชิงฟังก์ชันด้วยภาษาฮาสเกล (Haskell) เนื่องจากมีความเหมาะสมในการทำ pattern matching และมีรูปแบบการประมวลผลในแบบ lazy evaluation จึงสามารถรองรับข้อมูลสตรีมที่เป็น infinite data ได้ดี นอกจากโปรแกรมในรูปแบบภาษาฮาสเกลแล้ว ผู้วิจัยยังได้พัฒนาโปรแกรมต้นแบบในลักษณะของภาษาเออแลง (Erlang) ที่เป็นภาษาเชิงฟังก์ชันเช่นเดียวกับภาษาฮาสเกลแต่วิธีการเขียนฟังก์ชันมีความยืดหยุ่นมากกว่า และมี virtual machine ที่สนับสนุนการประมวลผลบน multicore processor ทำให้โปรแกรมทำงานได้เร็วขึ้น นอกจากนี้การอำนวยความสะดวกในด้าน concurrency และ message passing mechanism จะช่วยให้การปรับปรุงโปรแกรมโปรแกรมเป็นแบบ parallel ทำได้สะดวกกว่าภาษาฮาสเกล