ขั้นตอนวิธีการ IndexNClosed สำหรับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก

AN IndexNClosed ALGORITHM FOR MINING N-most INTERESTING CLOSED ITEMSETS

พิพิธพร โพธุ์ตุ่นสง
PIPITHAPORN PHONTUSANG

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2553
KMITL-2010-SC-M-002-001

ขั้นตอนวิธีการ IndexNClosed สำหรับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก

AN IndexNClosed ALGORITHM FOR MINING N-most INTERESTING CLOSED ITEMSETS

พิพิธพร โพนตุแสง

PIPITHAPORN PHONTUSANG

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2553

KMITL – 2010 – SC – M – 002 - 001

AN IndexNClosed ALGORITHM FOR MINING N-most INTERESTING CLOSED ITEMSETS

PIPITHAPORN PHONTUSANG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE IN COMPUTER SCIENCE

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2010

KMITL – 2010 – SC – M – 002 - 001

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ ขั้นตอนวิธีการ IndexNClosed สำหรับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก

An IndexNClosed Algorithm for Mining N-most Interesting Closed Itemsets

นักศึกษา นางสาวพิพิธพร โพนตุแสง

รหัสประจำตัว 49067551

ปริญญา วิทยาศาสตรมหาบัณฑิต

สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์ รศ.ดร.วีระ บุญจริง

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.ดร.วีระ	บุญจริง	
ผศ.ดร.จีรพร	วีระพันธุ์	
ดร.พรรณทิพย์	ภัทรอินทากร	
ดร.เฉลิมศักดิ์	เลิศวงศ์เสถียร	

วัน / เดือน/ ปี ที่สอบ 23 ธันวาคม 2552 เวลา 10.00 น.

สถานที่สอบ ณ อาคารจุฬาภรณ 1 คณะวิทยาศาสตร์ ห้อง 217

คณะวิทยาศาสตร์รับรองแล้ว

(รองศาสตราจารย์ ดร.ดุษณี ธนะบริพัฒน์)

คณบดีคณะวิทยาศาสตร์

วันที่ 11 **เดือน** มค. **พ.ศ.** 53

สำนักทะเบียนและประมวลผล สจล.
วันที่ส่งเล่มวิทยานิพนธ์ฉบับสมบูรณ์
วันที่ 18 เดือน ม.ค. พ.ศ. 2553
ลงชื่อ.....

Thesis Title	An IndexNClosed Algorithm for Mining N-most Interesting Closed Itemsets
Student	Miss Pipithaporn Phontusang
Student ID.	49067551
Degree	Master of Science
Program	Computer Science
Year	2010
Thesis Advisor	Assoc. Prof. Dr. Veera Boonjing

ABSTRACT

N-most interesting closed itemsets mining was proposed to avoid a generation of redundant itemsets and a specification of an appropriate minimum support threshold. A too big threshold could give no answer whereas a too small one probably yields a large number of redundant itemsets. In addition, the determination of the optimal threshold is hard for users having no knowledge of mining queries and task-specific data. This paper adopts index array for mining N-most interesting closed itemsets and improve an efficient algorithm, called IndexNClosed. The index array is presented, which is used to discover those items that always appear together. Due to index array, items coinciding and sharing the same support are merged together and preserved as initial generators. Then generators are used in the first *N*-most Interesting Closed Itemsets mining process. The algorithm uses a Best-First Search strategy to mine closed itemsets in descending order of their supports. This leads to an efficient pruning of unnecessary itemsets. In addition, the experiments are conducted from large database to compare the performance of algorithm IndexNClosed with NCLOSED. The experimental results indicate that the proposed algorithm IndexNClosed outperforms. It uses search space less than the other.

กิตติกรรมประกาศ

วิทยานิพนธ์นี้มิอาจจะสำเร็จลุล่วงไปได้ด้วยดี หากมิได้รับคำแนะนำ คำชี้แจง ความรู้ และความเอาใจใส่จาก รศ.ดร.วีระ บุญจริง ผู้เป็นอาจารย์ที่ปรึกษา ซึ่งท่านได้สละเวลาให้กับข้าพเจ้าอย่างเต็มที่ จึงใคร่ขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ ดร.พรรณทิพย์ ภัทรอินทากร ผศ.ดร.จีรพร วีระพันธุ์ และ ดร.เฉลิมศักดิ์ เลิศวงศ์เสถียร คณะกรรมการสอบวิทยานิพนธ์ ที่กรุณาให้คำแนะนำ ตลอดจนข้อชี้แนะ จนทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลงได้

ขอขอบพระคุณบิดา มารดา ที่สนันสนุนและเป็นกำลังใจในระหว่างการศึกษาเป็นอย่างดียิ่ง

ขอขอบคุณ นางสาวพนิดา ทรงรัมย์ ผศ.รัตนา เลิศสุวรรณศรี นายขวัญชัย เล้าสุขสุวรรณ พี่ๆ และเพื่อนๆ ทุกคนที่ให้คำปรึกษา และช่วยอำนวยความสะดวกในด้านต่างๆ

สำหรับคุณงามความดีและประโยชน์อันใดที่เกิดขึ้นจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดา มารดา อาจารย์ทุกท่าน ซึ่งเป็นที่เคารพรักยิ่ง ตลอดจนญาติพี่น้อง และเพื่อนๆ ทุกคน

พิพิธพร โพนตุแสง

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญภาพ

ภาพที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การทำเหมืองข้อมูล (Data Mining) เป็นการสืบค้นหาความรู้ใหม่จากฐานข้อมูล โดยความรู้ที่ได้อาจอยู่ในรูปแบบต่างๆ เช่น กฎความสัมพันธ์ (Association Rules) รูปแบบลำดับเหตุการณ์ (Sequential Patterns) หมวดหมู่ (Cluster) เป็นต้น ได้มีงานวิจัยจำนวนมากที่ศึกษาเกี่ยวกับการหาความสัมพันธ์ของข้อมูลในรูปแบบของกฎความสัมพันธ์ โดยสืบค้นกลุ่มข้อมูลที่เกิดบ่อย (Frequent Itemsets) ซึ่งจะใช้กลุ่มข้อมูลที่เกิดบ่อยทั้งหมดในการสร้างกฎความสัมพันธ์ ทำให้มีการสร้างกลุ่มข้อมูล (Itemsets) ที่ซ้ำซ้อนจำนวนมาก และได้รูปแบบความสัมพันธ์จำนวนมาก เป็นผลให้การวิเคราะห์มีความยุ่งยาก นอกจากปัญหาเกี่ยวกับจำนวนกลุ่มข้อมูลซ้ำซ้อนที่ได้จากการสืบค้นกลุ่มข้อมูลที่เกิดบ่อย อีกปัญหาหนึ่งที่พบคือปัญหาเกี่ยวกับการกำหนดค่าสนับสนุนขั้นต่ำ ทั้งนี้เพราะในกรณีที่ผู้ใช้ไม่มีความรู้ในการกำหนดค่าสนับสนุนขั้นต่ำ หากกำหนดค่าสนับสนุนขั้นต่ำที่มีค่าสูงเกินไป อาจไม่พบกลุ่มข้อมูลใดๆ เลย หรือพบกลุ่มข้อมูลน้อยมาก แต่ถ้ากำหนดค่าสนับสนุนขั้นต่ำที่มีค่าต่ำเกินไป ก็อาจพบกลุ่มข้อมูลจำนวนมากเกินความจำเป็น

ในการลดจำนวนกลุ่มข้อมูลซ้ำซ้อน ได้มีการเสนอแนวคิดเกี่ยวกับการสืบค้นกลุ่มข้อมูลแบบปิด (Closed Itemsets) โดยใช้โอเปอร์เรเตอร์โคลสเซอร์ (Closure Operator) งานวิจัยที่นำเสนอแนวคิดดังกล่าว เช่น Nicolas Pasquired และคณะ [12] ได้เสนอขั้นตอนวิธีการ A-CLOSE หรือ CLOSE เพื่อสืบค้นกลุ่มข้อมูลแบบปิด โดยสร้างกลุ่มข้อมูลแบบปิดด้วยการหาโคลสเซอร์ของกลุ่มข้อมูลที่เกิดบ่อยที่เล็กที่สุด อย่างไรก็ดี วิธีการดังกล่าวมีการคำนวณที่ซ้ำซ้อน ทั้งนี้เพราะ กลุ่มข้อมูลที่เกิดบ่อยที่เล็กที่สุด 2 กลุ่ม อาจให้กลุ่มข้อมูลแบบปิดตัวเดียวกัน เพื่อหลีกเลี่ยงการคำนวณที่ซ้ำซ้อนของกลุ่มข้อมูลแบบปิดเดียวกัน Jiawei Han และคณะ ได้นำเสนอขั้นตอนวิธีการ CLOSET [8] และ CLOSET+ [9] โดยใช้โครงสร้าง FP-Tree (Frequent Pattern tree) ขั้นตอนวิธีการ CLOSET จะสืบค้นกลุ่มข้อมูลแบบปิดด้วยโคลสเซอร์ โดยกลุ่มข้อมูลแบบปิดถูกขยายใหญ่ขึ้นด้วยชิ้นข้อมูล (Items) ที่มีค่าสนับสนุนเดียวกัน ในการตรวจสอบว่าเป็นกลุ่มข้อมูลแบบปิดหรือไม่นั้นจะพิจารณาจากการเป็นสับเซต กลุ่มข้อมูลใดเป็นสับเซตของกลุ่มข้อมูลแบบปิดที่มีอยู่แล้ว ซึ่งมีค่าสนับสนุนเดียวกัน กลุ่มข้อมูลนั้นไม่เป็นกลุ่มข้อมูลแบบปิด ส่วนขั้นตอนวิธีการ CLOSET+ จะใช้วิธีที่แตกต่างกันในการสืบค้นชุดข้อมูลแบบกระจาย (Sparse dataset) และชุดข้อมูลแบบหนาแน่น (Dense dataset) ในการตรวจสอบว่าเป็นกลุ่มข้อมูลแบบปิดหรือไม่ ในกรณีที่เป็นชุดข้อมูลแบบหนาแน่น จะพิจารณาจากสับเซตเช่นเดียวกันกับขั้นตอนวิธี CLOSET Mohammed J. Zaki และ

คณะ [11] ได้เสนอขั้นตอนวิธีการ CHARM เพื่อสืบค้นกลุ่มข้อมูลแบบปิด โดยใช้การสืบค้นแนวลึก และใช้โครงสร้างข้อมูล IT-tree (Itemset Tidset tree) แต่ละโหนดของ IT-tree จะมีกลุ่มข้อมูลที่เกิดบ่อย และเซตลำดับรายการเปลี่ยนแปลง (Transaction id set) ทันทีที่สร้างกลุ่มข้อมูลที่เกิดบ่อย เซตลำดับรายการเปลี่ยนแปลงของกลุ่มข้อมูลที่เกิดบ่อยที่สร้างขึ้นนี้จะถูกเปรียบเทียบกับเซตลำดับรายการเปลี่ยนแปลงของกลุ่มข้อมูลอื่นๆ ที่มีโหนดแม่เดียวกัน ถ้ามีลำดับที่เท่ากันก็จะรวมโหนดทั้งสองเข้าด้วยกัน

ในการแก้ปัญหาเกี่ยวกับการกำหนดค่าสนับสนุนขั้นต่ำ ได้มีการเสนอแนวคิดในการสืบค้นกลุ่มข้อมูลที่เกิดบ่อยที่มีความยาว k สูงสุด โดยผู้ใช้ไม่จำเป็นต้องกำหนดค่าสนับสนุนขั้นต่ำ เพียงแต่ระบุจำนวนผลลัพธ์ที่ต้องการ N ลำดับ งานวิจัยที่นำเสนอแนวคิดดังกล่าว เช่น Ada Wai-chee fu และคณะ [1] ได้เสนอ 2 ขั้นตอนวิธีการ ได้แก่ Itemset-Loop และ Itemset-iLoop เพื่อสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับ โดยขั้นตอนวิธีนี้ ผู้ใช้ไม่จำเป็นต้องกำหนดค่าสนับสนุนขั้นต่ำ เพียงแต่ระบุจำนวนผลลัพธ์ที่ต้องการ k ตัว โดยที่ $k > 1$ Cheung และคณะ [2] ได้เสนอ 3 ขั้นตอนวิธีการ ได้แก่ LOOPBACK, BOLB และ BOMO เพื่อสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับ โดยประยุกต์โครงสร้าง FP-tree ซึ่งขั้นตอนวิธีทั้งสามมีประสิทธิภาพด้านเวลาสูงกว่าขั้นตอนวิธีการ Itemset-Loop และ BOMO จะสร้าง FP-tree ที่มีชิ้นข้อมูลทั้งหมดในฐานข้อมูล เพื่อหาค่าสนับสนุนตัวสุดท้าย (final threshold) ของแต่ละความยาว ส่วนขั้นตอนวิธีการ LOOPBACK จะสร้าง FP-tree และกำหนดค่าสนับสนุนตัวสุดท้ายให้เป็นค่าเริ่มต้น เพื่อเป็นค่าสนับสนุนของชิ้นข้อมูลแตกต่างตัวที่ N ที่ใหญ่ที่สุดที่ได้เรียงอันดับไว้แล้ว ขั้นตอนวิธีการ BOLB เหมือนกับขั้นตอนวิธีการ BOMO ในส่วนของการสร้าง FP-tree และประยุกต์เทคนิคจากขั้นตอนวิธีการ LOOPBACK ในกระบวนการสืบค้น อย่างไรก็ดี Ngan และคณะ[3] ได้นำเทคนิคของ COFI-tree (Co-occurrence Frequent Item Tree) มาประยุกต์ใช้เพื่อสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับ โดยไม่ต้องกำหนดค่าสนับสนุนขั้นต่ำ ในขั้นตอนวิธีดังกล่าวมีประสิทธิภาพด้านเวลา เมื่อ k มีค่าน้อย Arshad และคณะ [4] ได้เสนอ 2 ขั้นตอนวิธีการ ได้แก่ NFOLD-growth และ LOOPBACK-NFOLD-growth สำหรับการสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับ โดยใช้โครงสร้างข้อมูล SOTrieIT (Support-Ordered Trie Itemset) เพื่อค้นหากลุ่มข้อมูลที่เกิดบ่อยที่มีความยาว k = 1 และ k = 2 ซึ่งขั้นตอนวิธีดังกล่าวมีประสิทธิภาพในด้านเวลาดีกว่าขั้นตอนวิธีการ Itemset-Loop และ BOMO Songram และ Boonjing [5] ได้เสนอ ขั้นตอนวิธีการ NCLOSED เพื่อสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับ และใช้เทคนิคการค้นหาคำตอบดีที่สุด (Best-First Search Strategy) ซึ่งเริ่มจากกลุ่มข้อมูลที่มีค่าสนับสนุนสูงสุดเป็นอันดับแรก ด้วยเหตุนี้จึงได้กลุ่มข้อมูลแบบปิดที่เรียงลำดับค่าสนับสนุนจากมากไปน้อย โดยไม่จำเป็นต้องหาค่าสนับสนุนตัวสุดท้าย

งานวิจัยที่ศึกษาเกี่ยวกับการสืบค้นกลุ่มข้อมูลแบบปิดดังกล่าวข้างต้น ขั้นตอนวิธีโดยส่วนใหญ่จำเป็นต้องเก็บรูปแบบกลุ่มข้อมูลแบบปิดทุกตัว รวมทั้งกลุ่มข้อมูลแบบปิดคู่แข่งที่สืบค้นแล้ว เพื่อพิจารณาการเป็นสับเซต ดังนั้น ในกรณีที่มีการสร้างกลุ่มข้อมูลคู่แข่งเป็นจำนวนมากจำเป็นต้องใช้หน่วยความจำในการจัดการกับกลุ่มข้อมูลที่ไม่ใช่กลุ่มข้อมูลแบบปิด นอกจากนี้ ในส่วนของการสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับ ขั้นตอนวิธีที่นำเสนอโดยส่วนใหญ่จำเป็นต้องหาค่าสนับสนุนตัวสุดท้ายของแต่ละความยาว ด้วยเหตุนี้ งานวิจัยนี้จึงเสนอขั้นตอนวิธีการ IndexNClosed เพื่อสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก และสร้างกลุ่มข้อมูลแบบปิดโดยไม่เก็บกลุ่มข้อมูลแบบปิดคู่แข่ง ได้นำอินเด็กซ์อาร์เรย์ (Index Array) มาประยุกต์ใช้ในการหากลุ่มข้อมูลที่ปรากฏอยู่ด้วยกันเสมอ โดยกลุ่มข้อมูลที่เกิดขึ้นพร้อมกันบ่อยครั้งและต่างก็ใช้ค่าสนับสนุนเดียวกัน กลุ่มข้อมูลเหล่านั้นจะสามารถรวมเข้าด้วยกันได้ ซึ่งเป็นการจัดเตรียมข้อมูลเริ่มต้นก่อนนำเข้าสู่กระบวนการสืบค้น เพื่อลดเนื้อที่ในการสืบค้นข้อมูล นอกจากนี้ ยังได้ใช้เทคนิคการค้นหาคำตอบดีที่สุด เพื่อให้ได้กลุ่มข้อมูลแบบปิดที่เรียงลำดับค่าสนับสนุนจากมากไปน้อย ซึ่งเป็นการลดขั้นตอนการหาค่าสนับสนุนตัวสุดท้าย

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

เพื่อพัฒนาขั้นตอนวิธีการสำหรับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก ขั้นตอนวิธีนี้มุ่งเน้นการลดเนื้อที่ในการจัดเก็บข้อมูลเพื่อการสืบค้นโดยใช้อินเด็กซ์อาร์เรย์

1.3 ขอบเขตการวิจัย

งานวิจัยนี้เป็นการพัฒนาขั้นตอนวิธีการ โดยประยุกต์ใช้ทฤษฎีต่างๆ แล้วทำการทดลองประสิทธิภาพของขั้นตอนวิธีใหม่นี้โดยใช้ชุดข้อมูลมาตรฐาน โดยทำการทดสอบทั้งกับชุดข้อมูลแบบหนาแน่นและแบบกระจายเพื่อแสดงลักษณะข้อมูลที่เหมาะสมกับขั้นตอนวิธีใหม่นี้

1.4 ส่วนประกอบของวิทยานิพนธ์

ส่วนที่เหลือของวิทยานิพนธ์ฉบับนี้ประกอบด้วยบทต่างๆ ดังนี้

บทที่ 2 กล่าวถึงทฤษฎีบทพื้นฐานที่เกี่ยวข้อง พร้อมทั้งนำเสนองานวิจัยที่เกี่ยวข้องกับการสืบค้นรูปแบบกลุ่มข้อมูลแบบปิดและการสืบค้นกลุ่มข้อมูลที่น่าสนใจ *N* ลำดับแรก

บทที่ 3 กล่าวถึงนิยามพื้นฐานในการสืบค้นรูปแบบกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก ขั้นตอนวิธีการ IndexNClosed สำหรับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก และการวิเคราะห์ขั้นตอนวิธีการ IndexNClosed

บทที่ 4 เป็นการศึกษาประสิทธิภาพของขั้นตอนวิธีการ IndexNClosed โดยเปรียบเทียบกับขั้นตอนวิธีการ NCLOSED โดยเกณฑ์ที่ใช้ในการเปรียบเทียบ คือ จำนวนกลุ่มข้อมูลแบบปิดที่ถูกสร้างขึ้น และ เวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก โดยทั้งสองขั้นตอนวิธี

บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะเกี่ยวกับขั้นตอนวิธีการIndexNClosedสำหรับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในการทบทวนวรรณกรรมที่เกี่ยวข้องกับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับ ผู้วิจัยได้แบ่งเนื้อหาซึ่งเกี่ยวกับทฤษฎีและผลงานวิจัยที่เกี่ยวข้อง ออกเป็นสองส่วนหลัก ได้แก่ การสืบค้นรูปแบบกลุ่มข้อมูลแบบปิด และ การสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับ โดยมีรายละเอียดดังนี้

2.1 การสืบค้นรูปแบบกลุ่มข้อมูลแบบปิด

การสืบค้นกลุ่มข้อมูลแบบปิด เป็นการสืบค้นกลุ่มข้อมูลขนาดใหญ่ที่มีความถี่สูงหรือกลุ่มข้อมูลขนาดใหญ่ที่เกิดขึ้นบ่อย โดยกลุ่มข้อมูลดังกล่าวจะคลอบคลุมกลุ่มข้อมูลความถี่ขนาดย่อยที่มีค่าความถี่หรือค่าสนับสนุนเท่ากัน โดยไม่สูญเสียความถูกต้องของข้อมูล [8] [9] [10] [11] [13]

ในกรณีที่ฐานข้อมูลมีขนาดใหญ่และค่าสนับสนุนต่ำสุด (Minimum Support) มีค่าต่ำ ปัญหาหนึ่งที่พบคือ กลุ่มข้อมูลที่ได้จากการสืบค้นมักมีจำนวนมากจนเกินจำเป็น เพื่อแก้ปัญหาดังกล่าว จึงได้มีการเสนอการสืบค้นกลุ่มข้อมูลแบบปิด เพื่อลดจำนวนกลุ่มข้อมูลที่ซ้ำซ้อน โดยใช้ตัวดำเนินการโคลสเซอร์ (Closure operator) นอกจากนี้ การสืบค้นกลุ่มข้อมูลแบบปิดยังให้ผลลัพธ์ที่ดีกว่าการสืบค้นกลุ่มข้อมูล กล่าวคือ จำนวนกลุ่มข้อมูลที่ได้นั้นจะน้อยกว่าหรือเท่ากับจำนวนที่ได้จากการสืบค้นกลุ่มข้อมูล ขั้นตอนวิธีที่นำเสนอเพื่อสืบค้นกลุ่มข้อมูลแบบปิด เช่น CLOSET [8] , CLOSET+ [9] , GRG [10] , CHARM [11] , A-CLOSE [12] , TFP [13] , DCI-CLOSE [7] และ Index-CloseMiner [6] เป็นต้น ซึ่งขั้นตอนวิธีเหล่านี้ใช้นิยามพื้นฐานและคุณสมบัติเดียวกันในการสืบค้นกลุ่มข้อมูลแบบปิด เพียงแต่มีขั้นตอนที่แตกต่างกันออกไป ซึ่งเนื้อหาดังกล่าวจะกล่าวถึงในหัวข้อที่ 2.1.1 , 2.1.2 และ 2.1.3 ตามลำดับ

2.1.1 นิยามพื้นฐาน

นิยามที่ 2.1 [5]

กำหนดให้ $I = \{i_1, i_2, \ldots, i_n\}$ เป็นเซตของชิ้นข้อมูลทั้งหมดที่ปรากฏอยู่ในฐานข้อมูล $X = \{x_1, x_2, \ldots, x_k\}$ เป็นสับเซตที่ไม่ใช่เซตว่างของ I โดยที่ $k \le n$ เรียก X ว่า กลุ่มข้อมูล (Itemset)

เพื่อความสะดวก ในที่นี้ จะเขียนกลุ่มข้อมูล $X = \{x_1, x_2, \ldots, x_k\}$ แทนด้วย $x_1 x_2 \kappa\, x_k$

นิยามที่ 2.2 [9]

ความยาวของกลุ่มข้อมูล X คือ จำนวนของชิ้นข้อมูลที่อยู่ใน X กลุ่มข้อมูลที่มีความยาว k เขียนแทนด้วย k-itemset

นิยามที่ 2.3 [9]

ค่าสนับสนุน (Support) ของกลุ่มข้อมูล X คือ จำนวนของลำดับรายการเปลี่ยนแปลงที่พบ X เขียนแทน ค่าสนับสนุนของกลุ่มข้อมูล X ได้ด้วย $supp(X)$

นิยามที่ 2.4 [5]

กลุ่มข้อมูล X เป็นกลุ่มข้อมูลที่เกิดบ่อย (Frequent Itemset) ถ้าค่าสนับสนุนของ X มีค่าไม่น้อยกว่าค่าสนับสนุนขั้นต่ำ

นิยามที่ 2.5 [9]

กลุ่มข้อมูล X เป็นกลุ่มข้อมูลย่อย (Sub-itemset) ของกลุ่มข้อมูล Y ก็ต่อเมื่อ X เป็นสับเซตของ Y เรียก Y ว่า ซุปเปอร์กลุ่มข้อมูล (Super-itemset) ของ X เขียนแทนได้ด้วย $X \subseteq Y$

นิยามที่ 2.6 [12]

กำหนดให้ $C = (D, I, R)$, D เป็นเซตจำกัดของลำดับรายการเปลี่ยนแปลง (Transaction id) ทั้งหมดในฐานข้อมูล , I เป็นเซตจำกัดของชิ้นข้อมูลที่แตกต่างกัน (Distinct items) ที่ปรากฏในฐานข้อมูล และ $R \subseteq D \times I$ เป็นความสัมพันธ์ของลำดับรายการเปลี่ยนแปลงและชิ้นข้อมูล โดยแต่ละคู่อันดับ $(d,i) \in R$ แทน ลำดับรายการเปลี่ยนแปลง $d \in D$ มีชิ้นข้อมูล $i \in I$

นิยามที่ 2.7 [12]

กำหนดให้ $C = (D, I, R)$, T เป็นสับเซตที่ไม่ใช่เซตว่างของ D และ กลุ่มข้อมูล X เป็นสับเซตที่ไม่ใช่เซตว่างของ I การสืบค้นกลุ่มข้อมูลแบบปิดใช้ฟังก์ชัน f และ g ดังต่อไปนี้

$$f(T) = \{i \in I \mid \forall d \in T, (d,i) \in R\}$$

$$g(X) = \{d \in D \mid \forall i \in X, (d,i) \in R\}$$

ฟังก์ชัน f ให้กลุ่มข้อมูลที่ใหญ่ที่สุด ซึ่งถูกรวมเข้าไว้ในเซตของลำดับรายการทั้งหมดที่อยู่ใน T และ ฟังก์ชัน g ให้เซตของลำดับรายการเปลี่ยนแปลงที่มีค่าสนับสนุนของกลุ่มข้อมูล X ที่กำหนด

นิยามที่ 2.8 [4, 5, 12]

กลุ่มข้อมูล X เป็นกลุ่มข้อมูลแบบปิด ก็ต่อเมื่อ $\varsigma(X) = f(g(X)) = fog(X) = X$ คอมโพสิทฟังก์ชัน $\varsigma = fog$ เรียกว่า ตัวดำเนินการโคลสเซอร์ (Closure operator) หรือ Galois operator

กำหนดให้ $\varsigma = fog$ และ $\varsigma' = gof$ คุณสมบัติต่อไปนี้เป็นจริง สำหรับทุก X_1 , X_2 และ T_1 , T_2

$$X_1 \subseteq X_2 \rightarrow g(X_1) \supseteq g(X_2) \tag{2.1}$$

$$X_1 \subseteq \varsigma(X_1) \tag{2.2}$$

$$\varsigma(\varsigma(X_1)) = \varsigma(X_1) \tag{2.3}$$

$$X_1 \subseteq X_2 \rightarrow \varsigma(X_1) \subseteq \varsigma(X_2) \tag{2.4}$$

$$\varsigma'(\varsigma(X)) = g(X) \tag{2.5}$$

$$T_1 \subseteq T_2 \rightarrow f(T_1) \supseteq f(T_2) \tag{2.6}$$

$$T \subseteq \varsigma'(T) \tag{2.7}$$

$$\varsigma'(\varsigma'(T_1)) = \varsigma'(T_1) \tag{2.8}$$

$$T_1 \subseteq T_2 \rightarrow \varsigma'(T_1) \subseteq \varsigma'(T_2) \tag{2.9}$$

$$\varsigma'(f(T_1)) = f(T_1) \tag{2.10}$$

$$T_1 \subseteq g(X_1) \leftrightarrow X_1 \subseteq f(T_1) \tag{2.11}$$

ตัวอย่างที่ 2.1 แสดงการสืบค้นรูปแบบกลุ่มข้อมูลแบบปิด จากฐานข้อมูลดังแสดงในตารางที่ 2.1 โดยกำหนดให้ตัวอักษร A, B, C, D และ E แทนชิ้นข้อมูล

ตารางที่ 2.1 ฐานข้อมูลตัวอย่าง

ลำดับ	ชิ้นข้อมูล
1	*B D E*
2	*A C*
3	*B D E*
4	*A C E*
5	*A D E*

พิจารณา ชิ้นข้อมูล A เป็นกลุ่มข้อมูลแบบปิด ได้ดังนี้ $\varsigma(A) = f(g(A)) = f(\{2,4,5\}) = A$
แต่ชิ้นข้อมูล BE ไม่เป็นกลุ่มข้อมูลแบบปิด ทั้งนี้เพราะ $\varsigma(BE) = f(g(BE)) = f(\{1,3\}) = BDE$

นิยามที่ 2.9 [12]

กำหนดให้ ค่าสนับสนุนต่ำสุดคือ *min_supp* กลุ่มข้อมูล X เป็นกลุ่มข้อมูลแบบปิดที่เกิดบ่อย (Frequent closed itemset) ถ้า $\varsigma(X) = X$ และ $supp(X) \geq min\_supp$.

จากนิยามที่ 2.7 เราใช้ตัวดำเนินการโคลสเซอร์เพื่อหากลุ่มข้อมูลแบบปิด โดยตัวดำเนินการโคลสเซอร์จะนิยามเซตของคลาสสมมูล (Equivalent Class) ภายใต้ขอบเขตของกลุ่มข้อมูล กล่าวคือ กลุ่มข้อมูล 2 กลุ่มจะอยู่ในคลาสสมมูลเดียวกัน ก็ต่อเมื่อ กลุ่มข้อมูล 2 กลุ่มนั้นมีโคลสเซอร์เหมือนกัน [13]

จากตัวอย่างที่ 2.1 เห็นได้ว่า $\varsigma(B) = \varsigma(BE) = \varsigma(BDE) = BDE$ ดังนั้น กลุ่มข้อมูล B, BE และ BDE อยู่ในคลาสสมมูลเดียวกัน

นิยามที่ 2.10 เจเนเรเตอร์ (Generator) คือ กลุ่มข้อมูลที่เกิดบ่อยที่มีความยาวน้อยที่สุดในคลาสสมมูล

จากข้างต้น กลุ่มข้อมูล B, BE และ BDE อยู่ในคลาสสมมูลเดียวกัน และ B เป็นกลุ่มข้อมูลที่มีความยาวน้อยที่สุด ดังนั้น B เป็นเจเนเรเตอร์ของคลาสนี้

เห็นได้ว่า B, BE และ BDE ซึ่งอยู่ในคลาสสมมูลเดียวกัน มีค่าสนับสนุนเท่ากัน คือ 2 นั่นคือ ในแต่ละคลาสสมมูล จะประกอบด้วยกลุ่มข้อมูลที่มีค่าสนับสนุนเดียวกัน โดยที่กลุ่มข้อมูลแบบปิดก็คือกลุ่มข้อมูลที่ใหญ่ที่สุดในคลาสนั้น และครอบคลุมกลุ่มข้อมูลทั้งหมดที่อยู่ในคลาสเดียวกัน ในที่นี้ BDE เป็นกลุ่มข้อมูลที่ใหญ่ที่สุดในคลาส ดังนั้น BDE เป็นกลุ่มข้อมูลแบบปิด

ตัวอย่างที่ 2.2 แสดงการสืบค้นรูปแบบกลุ่มข้อมูลแบบปิด จากฐานข้อมูลดังแสดงในตารางที่ 2.1 โดยกำหนดให้ตัวอักษร A, B, C, D และ E แทนชิ้นข้อมูล และกำหนดให้ค่าสนับสนุนต่ำสุด เท่ากับ 2

คำนวณหาโคลสเซอร์ของกลุ่มข้อมูล เพื่อหากลุ่มข้อมูลแบบปิด ได้ดังนี้

$\varsigma(B) = \varsigma(BE) = \varsigma(BDE) = BDE$ ดังนั้น กลุ่มข้อมูล B , BE และ BDE อยู่ในคลาสสมมูลเดียวกัน ที่มีค่าสนับสนุนเท่ากับ 2 และ BDE เป็นกลุ่มข้อมูลแบบปิดของคลาสนี้

$\varsigma(D) = \varsigma(DE) = DE$ ดังนั้น กลุ่มข้อมูล D และ DE อยู่ในคลาสสมมูลเดียวกัน ที่มีค่าสนับสนุนเท่ากับ 3 และ DE เป็นกลุ่มข้อมูลแบบปิดของคลาสนี้

$\varsigma(AE) = AE$ ดังนั้น AE เป็นกลุ่มข้อมูลแบบปิดที่มีค่าสนับสนุนเท่ากับ 2

$\varsigma(C) = \varsigma(AC) = AC$ ดังนั้น กลุ่มข้อมูล C และ AC อยู่ในคลาสสมมูลเดียวกัน ที่มีค่าสนับสนุนเท่ากับ 2 และ AC เป็นกลุ่มข้อมูลแบบปิดของคลาสนี้

$\varsigma(E) = E$ ดังนั้น E เป็นกลุ่มข้อมูลแบบปิดที่มีค่าสนับสนุนเท่ากับ 4

$\varsigma(A) = A$ ดังนั้น A เป็นกลุ่มข้อมูลแบบปิดที่มีค่าสนับสนุนเท่ากับ 3

ผลลัพธ์ที่ได้จากการสืบค้นรูปแบบกลุ่มข้อมูลแบบปิดจากฐานข้อมูลในตารางที่ 2.1 มีทั้งหมด 6 รูปแบบ ซึ่งเขียนแสดงในรูปของ (กลุ่มข้อมูล : ค่าสนับสนุน) ได้ดังนี้

$$\{(A:3),(E:4),(AC:2),(AE:2),(DE:3),(BDE:2)\}$$

ส่วนผลลัพธ์ที่ได้จากการสืบค้นรูปแบบกลุ่มข้อมูลมีทั้งหมด 11 รูปแบบ คือ

$$\{(A:3),(B:2),(C:2),(D:3),(E:4),(AC:2),(AE:2),(BD:2),(BE:2),(DE:3),(BDE:2)\}$$

เห็นได้ว่า จำนวนรูปแบบที่ได้จากการสืบค้นรูปแบบกลุ่มข้อมูลแบบปิด น้อยกว่า จำนวนรูปแบบที่ได้จากการสืบค้นกลุ่มข้อมูล นั่นคือ การสืบค้นรูปแบบกลุ่มข้อมูลแบบปิดสามารถครอบคลุมกลุ่มข้อมูลย่อยได้ โดยที่ข้อมูลไม่สูญหาย

2.1.2 คุณสมบัติของรูปแบบกลุ่มข้อมูลแบบปิด

ขั้นตอนวิธีที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิดอาศัยคุณสมบัติต่อไปนี้ในการสืบค้นกลุ่มข้อมูลปิด [11]

ข้อที่ 1 กลุ่มข้อมูลย่อยทุกตัวของกลุ่มข้อมูลเป็นกลุ่มข้อมูล

ข้อที่ 2 ซุปเปอร์กลุ่มข้อมูลทุกตัวที่ไม่ใช่กลุ่มข้อมูลจะไม่ใช่กลุ่มข้อมูล

ข้อที่ 3 เซตของกลุ่มข้อมูลขนาดใหญ่เท่ากับเซตของกลุ่มข้อมูลแบบปิดขนาดใหญ่

ข้อที่ 4 ค่าสนับสนุนของกลุ่มข้อมูล I ใดๆ เท่ากับค่าสนับสนุนของกลุ่มข้อมูลแบบปิดขนาดเล็กที่สุดที่บรรจุกลุ่มข้อมูล I

จากคุณสมบัติข้อที่ 1 , 3 และ 4 เห็นได้ว่า เซตของกลุ่มข้อมูลแบบปิดและค่าสนับสนุนของกลุ่มข้อมูลแบบปิด สามารถสร้างเซตของกลุ่มข้อมูลและค่าสนับสนุนของกลุ่มข้อมูลทั้งหมดได้ ซึ่งแสดงให้เห็นว่า การสืบค้นกลุ่มข้อมูลแบบปิดครอบคลุมการสืบค้นกลุ่มข้อมูล

2.1.3 การพัฒนาขั้นตอนวิธีสืบค้นรูปแบบกลุ่มข้อมูลแบบปิด

ในปัจจุบันได้มีการศึกษาเกี่ยวกับการสืบค้นกลุ่มข้อมูลแบบปิดอย่างกว้างขวาง เริ่มโดย Nicolas Pasquired และคณะ ได้เสนอขั้นตอนวิธีการ A-CLOSE หรือ CLOSE [12] โดยสร้างกลุ่มข้อมูลแบบปิดด้วยการหาโคลสเซอร์ของเจเนเรเตอร์ ในขั้นตอนแรกจะระบุเจเนเรเตอร์ และตัดตัวที่ไม่ใช่เจเนเรเตอร์ของแต่ละคลาสสมมูลออกไป จากนั้นจะคำนวณหาโคลสเซอร์ของทุกกลุ่มข้อมูลที่มีขนาดเล็กที่ค้นพบก่อนหน้านั้น วิธีการนี้อาจเกิดความซ้ำซ้อนในการคำนวณหาโคลสเซอร์ เพราะว่าในแต่ละคลาสสมมูลอาจมีกลุ่มข้อมูลขนาดเล็กมากกว่า 1 ตัว ทำให้เสียเวลาในการคำนวณหาโคลสเซอร์และค้นหาจำนวนสับเซตที่มีจำนวนมาก นอกจากนี้ วิธีการดังกล่าวอาจมีการคำนวณที่ซ้ำซ้อน ทั้งนี้เพราะเจเนเรเตอร์ 2 ตัว อาจให้กลุ่มข้อมูลแบบปิดตัวเดียวกัน และจำเป็นต้องเก็บกลุ่มข้อมูลที่เกิดบ่อยที่สืบค้นแล้ว เพื่อระบุเจเนเรเตอร์

Jiawei Han และคณะ ได้นำเสนอขั้นตอนวิธีการ CLOSET [8] และ CLOSET+ [9] โดยใช้โครงสร้าง FP-Tree (Frequent Pattern tree) ขั้นตอนวิธีการ CLOSET จะสืบค้นกลุ่มข้อมูลแบบปิดด้วยโคลสเซอร์ โดยกลุ่มข้อมูลแบบปิดถูกขยายใหญ่ขึ้นด้วยชิ้นข้อมูลที่มีค่าสนับสนุนเดียวกัน ในการตรวจสอบว่าเป็นกลุ่มข้อมูลแบบปิดหรือไม่นั้นจะพิจารณาจากการเป็นสับเซต กลุ่มข้อมูลใดเป็นสับเซตของกลุ่มข้อมูลแบบปิดที่มีอยู่แล้ว ซึ่งมีค่าสนับสนุนเดียวกัน กลุ่มข้อมูลนั้นไม่เป็นกลุ่มข้อมูลแบบปิด ส่วนขั้นตอนวิธีการ CLOSET+ จะใช้วิธีที่แตกต่างกันในการสืบค้นชุดข้อมูลแบบกระจาย (Sparse dataset) และชุดข้อมูลแบบหนาแน่น (Dense dataset) ในการตรวจสอบว่าเป็น

กลุ่มข้อมูลแบบปิดหรือไม่ ในกรณีที่เป็นชุดข้อมูลแบบหนาแน่น จะพิจารณาจากสับเซต เช่นเดียวกันกับขั้นตอนวิธีการ CLOSET

LiLi และคณะ [10] ได้เสนอขั้นตอนวิธีการ GRG โดยใช้กราฟเพื่อแสดงความสัมพันธ์ระหว่างกลุ่มข้อมูล และใช้บิตเวกเตอร์แทนกลุ่มข้อมูล ขั้นตอนวิธีดังกล่าวสามารถลดจำนวนรอบในการอ่านฐานข้อมูลและยังหลีกเลี่ยงการสร้างกลุ่มข้อมูลแบบปิดคู่แข่ง อย่างไรก็ตาม การแทนกลุ่มข้อมูลด้วยบิตเวกเตอร์นั้นอาจทำให้เสียเวลาในการแทนค่า ในกรณีที่กลุ่มข้อมูลมีขนาดใหญ่

Mohammed J. Zaki และคณะ [11] ได้เสนอขั้นตอนวิธีการ CHARM เพื่อสืบค้นกลุ่มข้อมูลแบบปิด โดยใช้การสืบค้นแนวลึก และใช้โครงสร้างข้อมูล IT-tree (Itemset Tidset tree) แต่ละโหนดของ IT-tree จะมีกลุ่มข้อมูลที่เกิดบ่อย และเซตลำดับรายการเปลี่ยนแปลง (Transaction id set) ทันทีที่สร้างกลุ่มข้อมูลที่เกิดบ่อย เซตลำดับรายการเปลี่ยนแปลงของกลุ่มข้อมูลที่เกิดบ่อยที่สร้างขึ้นนี้จะถูกเปรียบเทียบกับเซตลำดับรายการเปลี่ยนแปลงของกลุ่มข้อมูลอื่นๆ ที่มีโหนดแม่เดียวกัน ถ้ามีลำดับที่เท่ากันก็จะรวมโหนดทั้งสองเข้าด้วยกัน

Petre Tzvetkov และคณะ [13] ได้เสนอขั้นตอนวิธีการ TFP โดยพยายามที่จะลดพื้นที่ในการเก็บกลุ่มข้อมูลแบบปิด โดยเก็บอยู่ในรูปของต้นไม้กลุ่มข้อมูลเติมหน้า และใช้ดัชนีแฮช 2 ระดับเพื่อช่วยลดพื้นที่ในการค้นหา

Claudio Lucchese และคณะ [7] ได้เสนอขั้นตอนวิธีการ DCI-CLOSE ซึ่งเป็นขั้นตอนวิธีที่พัฒนามาจากขั้นตอนวิธีการ CHARM ขั้นตอนวิธีการ DCI-CLOSE จะสืบค้นกลุ่มข้อมูลแบบปิดโดยไม่เก็บกลุ่มข้อมูลแบบปิดคู่แข่ง ขั้นตอนวิธีดังกล่าวสามารถค้นหากลุ่มข้อมูลแบบปิดและตัดกลุ่มข้อมูลที่นำไปสู่กลุ่มข้อมูลแบบปิดตัวเดียวกันออกไป โดยไม่จำเป็นต้องเก็บกลุ่มข้อมูลแบบปิดนั้นลงในหน่วยความจำ ที่อาศัยการสืบค้นแนวลึกและการเรียกซ้ำในการสืบค้นกลุ่มข้อมูลแบบปิด

Wei Song และคณะ [6] ได้เสนอขั้นตอนวิธีการ Index-CloseMiner ซึ่งมีประสิทธิภาพสูงโดยเฉพาะอย่างยิ่งในการสืบค้นชุดข้อมูลแบบหนาแน่น (Dense dataset) ขั้นตอนวิธี Index-CloseMiner ได้พัฒนาปรับปรุงจากขั้นตอนวิธีการ DCI-CLOSE [7] ในการสืบค้นกลุ่มข้อมูลแบบปิดเพื่อลดจำนวนความซ้ำซ้อนและลดพื้นที่ในการค้นหา โดยนำเทคนิคที่น่าสนใจ 3 ข้อมาใช้ดังนี้คือ

(1) เสนออินเด็กซ์อาร์เรย์ (Index array) เพื่อหากลุ่มข้อมูลที่ปรากฏอยู่ด้วยกันเสมอ

(2) บนพื้นฐานของอินเด็กซ์อาร์เรย์ สามารถระบุกลุ่มข้อมูลแบบปิดได้โดยตรง โดยกลุ่มข้อมูลที่เกิดขึ้นพร้อมกันและใช้ค่าสนับสนุนเดียวกัน จะถูกรวมเข้าด้วยกันและเก็บเป็นกลุ่มข้อมูลเริ่มต้น

(3) เสนอ รีดิวพรี-เซต (reduced pre-set) และ รีดิวโพส-เซต (reduce post-set) ซึ่งได้พิสูจน์ให้เห็นว่า กลุ่มข้อมูลที่เกินจำเป็นใน พรี-เซต (pre-set) และ โพส-เซต (post-set) ที่ใช้ในขั้นตอนวิธีการ DCI-CLOSE ได้ถูกลบทิ้ง จึงหลีกเลี่ยงการดำเนินการที่เกินจำเป็น

2.2 การสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับแรก

ในการสืบค้นกลุ่มข้อมูลที่เกิดบ่อย มักพบปัญหาเกี่ยวกับการกำหนดค่าสนับสนุนขั้นต่ำ ทั้งนี้เพราะในกรณีที่ผู้ใช้ไม่มีความรู้ในการกำหนดค่าสนับสนุนขั้นต่ำ หากกำหนดค่าสนับสนุนขั้นต่ำที่มีค่าสูงเกินไป อาจไม่พบกลุ่มข้อมูลใดๆ เลย หรือพบกลุ่มข้อมูลน้อยมาก หากกำหนดค่าสนับสนุนขั้นต่ำที่มีค่าต่ำเกินไป อาจพบกลุ่มข้อมูลจำนวนมากเกินความจำเป็น [1][2][3][4] เพื่อแก้ปัญหาดังกล่าวข้างต้นได้มีการเสนอแนวคิดในการสืบค้นกลุ่มข้อมูลที่เกิดบ่อยที่มีความยาว k สูงสุด โดยผู้ใช้ไม่จำเป็นต้องกำหนดค่าสนับสนุนขั้นต่ำ เพียงแต่ระบุจำนวนผลลัพธ์ที่ต้องการ N ลำดับ ในปัจจุบันได้มีการพัฒนาขั้นตอนวิธีสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับ มากมายหลายขั้นตอนวิธี ขั้นตอนวิธีเหล่านี้ใช้นิยามพื้นฐานและคุณสมบัติเดียวกันในการสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับ เพียงแต่มีขั้นตอนที่แตกต่างกันออกไป ซึ่งเนื้อหาดังกล่าวจะกล่าวถึงในหัวข้อที่ 2.2.1 , 2.2.2 และ 2.2.3 ตามลำดับ

2.2.1 นิยามพื้นฐาน

นิยามที่ 2.11 [2,3,4,5] กลุ่มข้อมูลที่น่าสนใจ N ลำดับแรก ที่มีความยาว k (N-most interesting k-itemsets) คือ เซตของกลุ่มข้อมูลที่น่าสนใจ N ลำดับแรกที่มีความยาว k โดยจะเรียงจากความยาว k น้อยไปมาก

นิยามที่ 2.12 [2,3,4,5] กลุ่มข้อมูลที่น่าสนใจ N ลำดับแรก (N-most interesting itemsets) คือ การรวมกันของกลุ่มข้อมูลแบบปิด k สำหรับ $1 \leq k \leq k_{max}$ โดยที่ k_{max} คือความยาวสูงสุดของกลุ่มข้อมูลที่ต้องการ

ตัวอย่างที่ 2.3 แสดงการสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับแรก จากฐานข้อมูลดังแสดงในตารางที่ 2.2 โดยกำหนดให้ตัวอักษร A, B, C, D และ E แทนชิ้นข้อมูล และกำหนดให้ $N = 3$, $k_{max} = 3$ โดยที่ k_{max} คือ ความยาวสูงสุดของกลุ่มข้อมูลที่มีความยาว k และกำหนดค่าสนับสนุนต่ำสุดเท่ากับ 2

ตารางที่ 2.2 ฐานข้อมูลตัวอย่างที่มีรายการเกิดบ่อยแบบเรียงลำดับ

TID	Items	Sorted Frequent Items
001	*A, B, C, D*	*C, D, A, B*
002	*B, C, D, E*	*C, D, B, E*
003	*A, C, D*	*C, D, A*
004	*E , F*	*E, F*

จากตารางที่ 2.2 เมื่อสืบค้นกลุ่มข้อมูลทั้งหมด และเรียงลำดับของกลุ่มข้อมูลในรูปแบบ (กลุ่มข้อมูล : ค่าสนับสนุน) ได้ผลลัพธ์ทั้งหมด 12 รูปแบบ ดังนี้

$$\left\{ \begin{array}{l} (C:3),(D:3),(A:2),(B:2),(E:2),(CD:3),(CA:2),(CB:2),(DA:2),(DB:2), \\ (CDA:2),(CDB:2) \end{array} \right\}$$

กลุ่มข้อมูลที่มีค่าสนับสนุนต่ำกว่าค่าสนับสนุนขั้นต่ำ นั่นคือ กลุ่มข้อมูลที่มีค่าสนับสนุนต่ำกว่า 2 จะถูกตัดออกไป โดยไม่เก็บไว้ จากนั้นจะพิจารณากลุ่มข้อมูลที่เหลือ เนื่องจากกำหนดให้ $N=3$ และ $k_{max}=3$ ด้วยเหตุนี้ จึงพิจารณากลุ่มข้อมูลที่น่าสนใจ 3 ลำดับแรก ของกลุ่มข้อมูลสูงสุดเท่ากับ 3 ผลลัพธ์ที่ได้เป็นดังนี้

กรณี 3 ลำดับ ของ 1-itemset พบกลุ่มข้อมูลทั้งหมด 4 รูปแบบ ดังนี้

$$\{(C:3),(D:3),(A:2),(B:2),(E:2)\}$$

เห็นได้ว่า ผลลัพธ์ที่ได้นั้นมากกว่าที่เราได้กำหนดไว้ (3 ลำดับ) เนื่องด้วย ค่าสนับสนุนของแต่ละกลุ่มข้อมูลมีค่าเท่ากัน ฉะนั้น จึงถือว่าเป็นคำตอบทั้งหมด

กรณี 3 ลำดับของ 2-itemset พบกลุ่มข้อมูลทั้งหมด 5 รูปแบบ ดังนี้

$$\{(CD:3),(CA:2),(CB:2),(DA:2),(DB:2)\}$$

กรณี 3 ลำดับของ 3-itemset พบกลุ่มข้อมูลทั้งหมด 2 รูปแบบ ดังนี้

$$\{(CDA:2),(CDB:2)\}$$

เห็นได้ว่า การสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับแรก ตั้งแต่ $1 < k < k_{max}$ ให้ผลลัพธ์ตามจำนวนลำดับที่เราได้กำหนดไว้ของแต่ละกลุ่มข้อมูลที่เราต้องการสูงสุดได้

2.2.2 การพัฒนาขั้นตอนวิธีสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับ

Ada Wai-chee และคณะ[1] ได้เสนอขั้นตอนวิธีการ Itemset-Loop และ Itemset-iLoop [1] เพื่อสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับแรก โดยขั้นตอนวิธีนี้ ผู้ใช้ไม่จำเป็นต้องกำหนดค่าสนับสนุนขั้นต่ำ เพียงแต่ระบุจำนวนผลลัพธ์ที่ต้องการ k ตัว โดยที่ $k > 1$

Y-LCheng และ A. Fu [2] ได้นำเสนอ 3 ขั้นตอนวิธีการ ได้แก่ LOOPBACK , BOLB และ BOMO เพื่อสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับแรก โดยประยุกต์โครงสร้าง FP-tree ซึ่งขั้นตอนวิธีทั้งสามมีประสิทธิภาพด้านเวลาสูงกว่าขั้นตอนวิธีการ Itemset-Loop ขั้นตอนวิธีการ BOMO จะสร้าง FP-tree ที่มีกลุ่มข้อมูลทั้งหมดในฐานข้อมูล เพื่อหาค่าสนับสนุนตัวสุดท้าย (final threshold) ของแต่ละความยาว ส่วนขั้นตอนวิธีการ LOOPBACK จะสร้าง FP-tree และกำหนดค่าสนับสนุนตัวสุดท้าย ให้เป็นค่าเริ่มต้นเพื่อเป็นค่าสนับสนุนของกลุ่มข้อมูลที่เกิดบ่อยความยาว 1 ขั้นตอนวิธีการ BOLB เหมือนกับขั้นตอนวิธีการ BOMO ในส่วนของการสร้าง FP-tree และประยุกต์เทคนิคจากขั้นตอนวิธีการ LOOPBACK ในกระบวนการสืบค้น

S-C. Ngan และคณะ [3] ได้นำเทคนิคของ COFI-tree (Co-occurrence Frequent Item Tree) มาประยุกต์ใช้เพื่อสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับแรก โดยไม่ต้องกำหนดค่าสนับสนุนขั้นต่ำ ขั้นตอนวิธีดังกล่าวมีประสิทธิภาพด้านเวลา เมื่อ k มีค่าน้อย

M.Arshad และ M.Ayyaz [4] ได้เสนอ 2 ขั้นตอนวิธีการ ได้แก่ NFOLD-growth และ LOOPBACK- NFOLD-growth สำหรับการสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับแรก โดยใช้โครงสร้างข้อมูล SOTrieIT (Support-Ordered Trie Itemset) เพื่อค้นหากลุ่มข้อมูลที่เกิดบ่อยที่มีความยาว $k = 1$ และ $k = 2$ ซึ่งขั้นตอนวิธีดังกล่าวมีประสิทธิภาพในด้านเวลาดีกว่าขั้นตอนวิธีการ Itemset-Loop และ BOMO

Songram และ Boonjing ได้เสนอ ขั้นตอนวิธีการ NCLOSED [5] เพื่อสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก และใช้เทคนิคการค้นหาคำตอบดีที่สุด (Best-First Search Strategy) ซึ่งเริ่มจากกลุ่มข้อมูลที่มีค่าสนับสนุนสูงสุดเป็นอันดับแรก ด้วยเหตุนี้จึงได้กลุ่มข้อมูลแบบปิดที่เรียงลำดับค่าสนับสนุนจากมากไปน้อย โดยไม่จำเป็นต้องหาค่าสนับสนุนตัวสุดท้าย

บทที่ 3
ขั้นตอนวิธีการ IndexNClosed

จากบทที่ 2 ได้แสดงการคำนวณโคลสเซอร์ของกลุ่มข้อมูลเพื่อหากลุ่มข้อมูลแบบปิด และจำนวนรูปแบบที่ได้จากการสืบค้นกลุ่มข้อมูลแบบปิดจะน้อยกว่าหรือเท่ากับจำนวนรูปแบบที่ได้จากการสืบค้นกลุ่มข้อมูล โดยที่ข้อมูลไม่สูญหาย นอกจากนี้ เราใช้การสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับ เพื่อหลีกเลี่ยงปัญหาการกำหนดค่าสนับสนุนขั้นต่ำ ในบทที่ 3 นี้ จะกล่าวถึงขั้นตอนวิธี IndexNClosed ซึ่งเป็นการผสมผสานระหว่างการสืบค้นรูปแบบกลุ่มข้อมูลแบบปิดกับการสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับ โดยเนื้อหาในบทนี้แบ่งออกเป็น 3 ส่วน ได้แก่ นิยามพื้นฐานในการสืบค้นรูปแบบกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับ การพัฒนาขั้นตอนวิธี IndexNClosed สำหรับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับ และการวิเคราะห์ขั้นตอนวิธี IndexNClosed

3.1 นิยามพื้นฐาน

กำหนดให้ $D = \{d_1, d_2, \ldots, d_m\}$ เป็นเซตของลำดับรายการเปลี่ยนแปลง (Transaction id) ทั้งหมดในฐานข้อมูล

$I = \{i_1, i_2, \ldots, i_n\}$ เป็นเซตของชิ้นข้อมูลทั้งหมดที่ปรากฏอยู่ในฐานข้อมูล

$T = \{t_1, t_2, \ldots, t_h\}$ เป็นสับเซตที่ไม่ใช่เซตว่างของ D โดยที่ $h \le m$

$X = \{x_1, x_2, \ldots, x_k\}$ เป็นสับเซตที่ไม่ใช่เซตว่างของ I โดยที่ $k \le n$

เรียก X ว่า กลุ่มข้อมูล (Itemset) ในที่นี้ จะเขียนกลุ่มข้อมูล $X = \{x_1, x_2, \ldots, x_k\}$ แทนด้วย $x_1 x_2 \kappa x_k$

จากนิยามที่กล่าวแล้วในบทที่ 2 เราใช้ฟังก์ชัน f และ g ต่อไปนี้ ในการสร้างกลุ่มข้อมูลแบบปิด

$$f(T) = \{i \in I \mid \forall d \in T, (d,i) \in R\}$$

$$g(X) = \{d \in D \mid \forall i \in X, (d,i) \in R\}$$

นิยามที่ 3.1 [5]

กลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก ที่มีความยาว k (N-most interesting k-closed itemsets) คือ เซตของกลุ่มข้อมูลแบบปิดที่มีความยาว k ซึ่งมีค่าสนับสนุนมากกว่าหรือเท่ากับ s โดยที่ s คือ ค่าสนับสนุนของกลุ่มข้อมูลแบบปิดที่มีความยาว k ตัวที่ N ในรายการที่ได้เรียงลำดับค่าสนับสนุนจากมากไปน้อยของกลุ่มข้อมูลแบบปิดที่มีความยาว k

นิยามที่ 3.2 [5]

กลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก (N-most interesting closed itemsets) เป็นการรวมกันของกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก ที่มีความยาว k สำหรับ $1 \leq k \leq k_{max}$ โดยที่ k_{max} คือความยาวสูงสุดของกลุ่มข้อมูลแบบปิดที่ต้องการ

3.2 ขั้นตอนวิธีการ IndexNClosed

ขั้นตอนวิธีการ IndexNClosed ถูกพัฒนาขึ้นเพื่อลดพื้นที่ในการค้นหา สำหรับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก เนื่องจากขั้นตอนวิธีการ NCLOSED [5] ไม่มีการจัดเตรียมเจเนเรเตอร์เริ่มต้นสำหรับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก ซึ่งอาจจำเป็นต้องใช้พื้นที่ในการค้นหามาก ดังนั้น เพื่อแก้ปัญหาดังกล่าว เราจึงเสนอขั้นตอนวิธีการ IndexNClosed เพื่อสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก และสร้างกลุ่มข้อมูลแบบปิดโดยไม่เก็บกลุ่มข้อมูลแบบปิดคู่แข่ง นอกจากนี้ได้นำอินเด็กซ์อาร์เรย์ (Index Array) มาประยุกต์ใช้ในการหากลุ่มข้อมูลที่ปรากฏอยู่ด้วยกันเสมอ โดยกลุ่มข้อมูลที่เกิดขึ้นพร้อมกันบ่อยครั้งและต่างก็ใช้ค่าสนับสนุนเดียวกัน กลุ่มข้อมูลเหล่านั้นจะสามารถรวมเข้าด้วยกันได้ ซึ่งเป็นการจัดเตรียมเจเนเรเตอร์เริ่มต้นก่อนนำเข้าสู่กระบวนการสืบค้น เพื่อลดเนื้อที่ในการสืบค้นข้อมูล และได้ใช้เทคนิคการค้นหาคำตอบดีที่สุด เพื่อให้ได้กลุ่มข้อมูลแบบปิดที่เรียงลำดับค่าสนับสนุนจากมากไปน้อย ซึ่งเป็นการลดขั้นตอนในการหาค่าสนับสนุนสุดท้าย

สำหรับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก โดยนำอินเด็กซ์อาร์เรย์ จากขั้นตอนวิธีการ Index-CloseMiner [6] มาประยุกต์ใช้ โดยขั้นตอนวิธีการ IndexNClosed ที่นำเสนอนี้ประกอบด้วย 2 ขั้นตอนหลัก คือ

1) สร้างอินเด็กซ์อาร์เรย์ และจัดเก็บลงในคิว (Queue) เพื่อใช้เป็นเจเนเรเตอร์เริ่มต้นสำหรับการสืบค้นกลุ่มข้อมูลในขั้นตอนต่อไป

2) ขั้นตอนการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก

3.2.1 การสร้างอินเด็กซ์อาร์เรย์

นิยามต่างๆ ที่เกี่ยวข้องกับการสร้างอินเด็กซ์อาร์เรย์ มีดังต่อไปนี้

นิยามที่ 3.3 [6]

อินเด็กซ์อาร์เรย์ คือ เซตของคู่อันดับ (item, subsume) เมื่อ item คือ ชิ้นข้อมูล และ $subsume(item) = \{ j \in I \mid j \neq item \wedge g(item) \subseteq g(j) \}$ สำหรับแต่ละสมาชิกของอินเด็กซ์อาร์เรย์ เราเรียก $subsume(item)$ ว่า *subsume index*

ตารางที่ 3.1 และ ตารางที่ 3.2 แสดงฐานข้อมูลตัวอย่างและลำดับรายการเปลี่ยนแปลงตามลำดับ เพื่อความสะดวก ในที่นี้ จึงเขียนกลุ่มข้อมูล $\{A, B, C\}$ แทนด้วย ABC และ เซตของลำดับการเปลี่ยนแปลง {2, 4, 5} เขียนแทนด้วย 245

ตารางที่ 3.1 ฐานข้อมูลตัวอย่าง

TID	Items
1	*A B C D E F*
2	*A D E*
3	*B C E F*
4	*B C D E F*
5	*B C F*

ตารางที่ 3.2 ลำดับรายการเปลี่ยนแปลง

Items	TID
A	1 2
B	1 3 4 5
C	1 3 4 5
D	1 2 4
E	1 2 3 4
F	1 3 4 5

ตัวอย่างที่ 3.1 แสดงการหาอินเด็กซ์อาร์เรย์ของแต่ละชิ้นข้อมูล จากตารางข้างต้น ได้ดังนี้

ในการหาอินเด็กซ์อาร์เรย์ ของแต่ละชิ้นข้อมูล จะพิจารณาจากลำดับรายการเปลี่ยนแปลงของอินเด็กซ์ของชิ้นข้อมูลนั้นๆ ว่าเกิดขึ้นที่ชิ้นข้อมูลใด ตัวอย่างเช่น ชิ้นข้อมูล A มี ลำดับรายการเปลี่ยนแปลง คือ 1 2 เห็นได้ว่า ลำดับรายการเปลี่ยนแปลง 1 2 นี้ เกิดขึ้นที่ชิ้นข้อมูล D และ E ดังนั้น D E จึงเป็นซับซูมอินเด็กซ์ของ A สำหรับชิ้นข้อมูลตัวอื่นๆ ที่เหลือ สามารถพิจารณาได้ในทำนองเดียวกัน ดังนั้น อินเด็กซ์อาร์เรย์ของฐานข้อมูลตัวอย่างในตารางที่ 3.1 ประกอบด้วยสมาชิกดังนี้ (A, DE) (B, CF) (C, BF) (D, E) $(E, \varnothing)$ และ (F, BC)

เห็นได้ว่าซับซูมอินเด็กซ์ของ B C และ F มีกลุ่มข้อมูลที่เกิดขึ้นเหมือนกันและมีค่าสนับสนุนเดียวกันซึ่งสามารถรวมเข้าด้วยกันได้

ทฤษฎีบทที่ 3.1 [6]

ถ้าชิ้นข้อมูล $j \in subsume(i)$ และ $supp(i) = supp(j)$ แล้ว $i \cup subsume(i) = j \cup subsume(j)$

ทฤษฎีบทที่ 3.2 [6]

กำหนดให้ X เป็นกลุ่มข้อมูล จะได้ว่า $X \cup subsume(X)$ เป็นกลุ่มข้อมูลแบบปิด, และ $supp(X \cup subsume(X)) = supp(X)$

เมื่อได้ซับซูมอินเด็กซ์ของแต่ละชิ้นข้อมูลแล้ว จะพิจารณาต่อไปว่า ซับซูมอินเด็กซ์ใดที่เกิดขึ้นเหมือนกันและมีค่าสนับสนุนเท่ากัน ซับซูมอินเด็กซ์นั้น จะถูกรวมเข้าด้วยกัน (ดังทฤษฎีบทที่ 3.1) เห็นได้ว่าซับซูมอินเด็กซ์ของ B คือ CF, ซับซูมอินเด็กซ์ของ C คือ BF และ ซับซูมอินเด็กซ์ของ F คือ BC นอกจากนี้ $supp(B) = supp(C) = supp(F)$ ดังนั้น กลุ่มข้อมูลของซับซูมอินเด็กซ์ B , C และ F จึงถูกรวมเข้าด้วยกันและเลือกเอากลุ่มข้อมูลใดข้อมูลหนึ่งมาเป็นเจเนเรเตอร์เริ่มต้นซึ่งในที่นี้เลือก BCF สำหรับ ซับซูมอินเด็กซ์อื่นๆ ที่เหลือ สามารถพิจารณาได้ในทำนองเดียวกัน จากทฤษฎีบทที่ 3.1 และ ทฤษฎีบทที่ 3.2 จะได้เจเนเรเตอร์เริ่มต้น (Initial generators) ที่มีค่าสนับสนุนดังที่แสดงไว้หลังเครื่องหมาย : ดังนี้

$$E:4 \ , \ DE:3 \ , \ BCF:4 \ , \ ADE:2$$

หลังจากที่ได้กลุ่มข้อมูลเริ่มต้นแล้ว ก็จะนำเจเนเรเตอร์เริ่มต้นนี้ไปสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก ต่อไป

3.2.2 การสร้างกลุ่มข้อมูลแบบปิด

จากที่กล่าวข้างต้น เราคำนวณโคลสเซอร์ของกลุ่มข้อมูลเพื่อหากลุ่มข้อมูลแบบปิด อย่างไรก็ดี เพื่อหลีกเลี่ยงการคำนวณซ้ำซ้อนของกลุ่มข้อมูลปิดเดียวกัน ขั้นตอนวิธีการ IndexNClosed ที่นำเสนอนี้ สร้างกลุ่มข้อมูลแบบปิดโดยไม่เก็บกลุ่มข้อมูลแบบปิดคู่แข่ง ด้วยการคำนวณโคลสเซอร์ของกลุ่มข้อมูล

ตัวอย่างที่ 3.2 จากตารางที่ 3.1 และ ตารางที่ 3.2 ที่แสดงตัวอย่างฐานข้อมูลและลำดับรายการเปลี่ยนแปลง เซตของลำดับรายการเปลี่ยนแปลงที่มี กลุ่มข้อมูล EF คือ $g(\{EF\}) = \{1, 3, 4\}$ และ เซตของลำดับรายการเปลี่ยนแปลงที่มีชิ้นข้อมูล C คือ $g(\{C\}) = \{1, 3, 4, 5\}$ เห็นได้ว่า $g(\{EF\}) \subseteq g(\{C\})$ ดังนั้น $C \in \varsigma(\{EF\})$

เราสามารถคำนวณโคลสเซอร์ของกลุ่มข้อมูล X ด้วยการรวมชิ้นข้อมูล ทุกชิ้นที่มีซูเปอร์เซตของเซตลำดับรายการเปลี่ยนแปลงของ X จากวิธีการนี้โคลสเซอร์เดียวกันอาจถูกคำนวณ 2 ครั้ง จากกลุ่มข้อมูล 2 กลุ่มที่แตกต่างกัน ดังนั้น กลุ่มข้อมูลแบบปิดกลุ่มเดียวกันจะถูกสร้างขึ้น 2 ครั้ง ตัวอย่างเช่น กลุ่มข้อมูลแบบปิด $BCDEF$ ถูกคำนวณ 2 ครั้ง จาก BD และ DF เพื่อหลีกเลี่ยงปัญหาดังกล่าว จึงจำเป็นต้องตรวจการซ้ำของกลุ่มข้อมูล โดยกลุ่มข้อมูลที่ไม่ซ้ำจะถูกคำนวณเพื่อหากลุ่มข้อมูลแบบปิด ขณะที่กลุ่มข้อมูลที่ซ้ำจะถูกตัดทิ้งไป

นิยามที่ 3.4 [7]

กำหนดให้ $X = Y \cup \{i\}$ เป็นกลุ่มข้อมูล โดยที่ Y เป็นกลุ่มข้อมูลแบบปิด $i \in I$ และ $i \notin Y$ กล่าวได้ว่า X เป็นเจเนเรเตอร์ที่ไม่ซ้ำ ก็ต่อเมื่อ $\varsigma(X) = X$

นิยามที่ 3.5 [7]

กำหนดให้ $X = Y \cup \{i\}$ เป็นกลุ่มข้อมูล โดยที่ Y เป็นกลุ่มข้อมูลแบบปิด , $i \in I$ และ $i \notin Y$ เซตของโพส-ไอเทม (post-items) ของ X นิยามได้ดังนี้

$$post-items(X) = \{r \mid r \in I, r \notin X, and\ i \ _{\mathrm{p}}\ r\}$$

ตัวดำเนินการ : $i \ _{\mathrm{p}}\ r$ หมายถึง i ปรากฏก่อน r ในรายการของชิ้นข้อมูลที่ถูกเรียงอันดับแล้ว ซึ่งรายการของชิ้นข้อมูลที่ถูกเรียงแล้วนั้นประกอบด้วยชิ้นข้อมูลที่แตกต่างกันที่ถูกเรียงอันดับตามค่าสนับสนุนจากมากไปน้อย

ทฤษฎีบทที่ 3.3 [7]

กำหนดให้ $X = Y \cup \{i\}$ เป็นกลุ่มข้อมูล โดยที่ Y เป็นกลุ่มข้อมูลแบบปิด , $i \in I$ และ $i \notin Y$ ถ้า $\exists r \in post-items(X)$ ที่ทำให้ $g(X) \subseteq g(\{r\})$ แล้ว X ไม่เป็นเจเนเรเตอร์ที่ไม่ซ้ำ

จากทฤษฎีบทที่ 3.3 เราสามารถตรวจได้ว่า กลุ่มข้อมูล X เป็นเจเนเรเตอร์ที่ไม่ซ้ำ หรือไม่ โดยตรวจเซตลำดับรายการเปลี่ยนแปลงของ X ด้วย เซตลำดับรายการเปลี่ยนแปลงของ *post-items(X)* เนื่องจากชิ้นข้อมูลที่แตกต่างกันนั้นได้ถูกเรียงอันดับตามค่าสนับสนุนจากมากไปน้อย ดังนั้น จึงไม่จำเป็นต้องตรวจ *post-items(X)* บางตัว ในกรณีที่ค่าสนับสนุนน้อยกว่าค่าสนับสนุนของ X

นิยามที่ 3.6 [7]

กำหนดให้ $X = Y \cup \{i\}$ เป็นกลุ่มข้อมูล โดยที่ Y เป็นกลุ่มข้อมูลแบบปิด , $i \in I$ และ $i \notin Y$ เซตของพรี-ไอเทม (pre-items) ของ X นิยามได้ดังนี้

$$pre-items(X) = \{j \mid j \in I, j \notin X, and\ j \text{ p } i\}$$

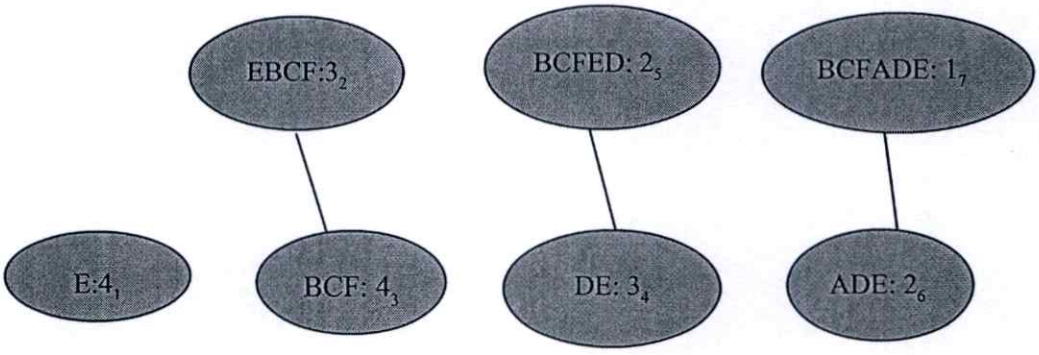
ทฤษฎีบทที่ 3.4 [7]

กำหนดให้ $X = Y \cup \{i\}$ เป็นเจเนเรเตอร์ที่ไม่ซ้ำ โดยที่ Y เป็นกลุ่มข้อมูลแบบปิด , $i \in I$ และ $i \notin Y$ ถ้า $j \in pre-items(X)$ และ $g(X) \subseteq g(j)$ แล้ว $j \in \varsigma(X)$

ทฤษฎีบทที่ 3.4 แสดงให้เห็นว่า เฉพาะ *pre-items(X)* เท่านั้นที่ถูกพิจารณาเพื่อหาโคลสเซอร์ของ X

ตัวอย่างที่ 3.3 จากตารางที่ 3.1 และ ตารางที่ 3.2 ที่แสดงฐานข้อมูลตัวอย่างและลำดับรายการเปลี่ยนแปลง ตามลำดับ หากต้องการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ 3 ลำดับแรก $N = 3$ และ $k_{max} = 3$ โดยที่ k_{max} คือ ความยาวสูงสุดของกลุ่มข้อมูลที่มีความยาว k โดยใช้ผลลัพธ์ที่ได้จากตัวอย่างที่ 3.3 ได้ว่า เจเนเรเตอร์เริ่มต้นสำหรับขั้นตอนวิธี IndexNClosed เมื่อเรียงลำดับค่าสนับสนุนจากมากไปน้อย เป็นดังนี้

$$E:4\ ,\ BCF:4\ ,\ DE:3\ ,\ ADE:2$$



ชิ้นข้อมูล *E* : 4 ถูกพิจารณากลุ่มข้อมูลแบบปิดเป็นอันดับแรก ตรวจสอบการสร้างชิ้นข้อมูลซ้ำซ้อนจาก *post-item* คือ *BCF* : 4, *DE* : 3, *ADE* : 2 เมื่อตรวจสอบแล้ว พบว่า ไม่เป็นชิ้นข้อมูลที่ซ้ำซ้อน เพราะฉะนั้น *E* : 4 จะถูกนำไปขยายเพื่อหาโคลสเซอร์ ซึ่งการหาโคลสเซอร์สามารถหาได้จาก pre- item ในที่นี้ ไม่มี pre- item ของ *E* : 4 ดังนั้น *E* : 4 จึงเป็นกลุ่มข้อมูลแบบปิดตัวแรก และค่าสนับสนุนจะถูกกำหนดให้มีค่าเท่ากับ 0

BCF : 4 ถูกพิจารณาเป็นชิ้นข้อมูลแบบปิดเป็นอันดับถัดไป ชิ้นข้อมูล *BCF* : 4 มี *pre- item* คือ *E* : 4 และ *post- item* คือ *DE* : 3 , *ADE* : 2 ตรวจสอบชิ้นข้อมูล *BCF* : 4 พบว่าเป็นชิ้นข้อมูลที่ไม่ถูกสร้างซ้ำซ้อนจาก *post- item* ดังนั้น *BCF* : 4 จึงถูกขยายเพื่อหาโคลสเซอร์ โคลสเซอร์ของ *BCF* : 4 สามารถสร้างได้จาก *pre- item* คือ *E* : 4 และขยายได้เป็น *EBCF* : 3 เมื่อตรวจสอบ *EBCF* : 3 พบว่า เป็นชิ้นข้อมูลที่ไม่ถูกสร้างซ้ำซ้อน ดังนั้น *EBCF* : 3 เป็นกลุ่มข้อมูลแบบปิด และ *BCF* : 4 ก็เป็นกลุ่มข้อมูลแบบปิดเช่นกัน เพราะไม่สามารถขยายต่อไปได้

DE : 3 ถูกพิจารณาเป็นชิ้นข้อมูลแบบปิดตัวถัดไป *pre- item* ของ *DE* : 3 คือ *E* : 4 , *BCF* : 4 *post- item* คือ *ADE* : 2 เมื่อตรวจสอบชิ้นข้อมูล *DE* : 3 พบว่า เป็นชิ้นข้อมูลที่ไม่ถูกสร้างซ้ำซ้อนจาก *post- item* ดังนั้น *DE* : 3 ถูกขยายเพื่อหาโคลสเซอร์ จาก *pre- item* คือ *E* : 4 , *BCF* : 4 ซึ่งในที่นี้ จะเลือกขยาย *BCF* : 4 ก่อน คือ *BCFDE* : 2 ตรวจสอบชิ้นข้อมูล *BCFDE* : 2 เป็นชิ้นข้อมูลที่ไม่ถูกสร้างซ้ำซ้อน ดังนั้น *BCFDE* : 2 เป็นกลุ่มข้อมูลแบบปิด

ADE : 2 ถูกพิจารณาเป็นชิ้นข้อมูลแบบปิด ตัวถัดไป *pre- item* ของ ADE : 2 ได้แก่ *E* : 4, *BCF* : 4 และ *DE* : 3 ซึ่งไม่มี *post- item* ดังนั้น *ADE* : 2 จึงถูกขยายได้เป็น *BCFADE* : 1 ซึ่งชิ้นข้อมูลทั้งสองเป็นกลุ่มข้อมูลแบบปิด

เมื่อสืบค้นกลุ่มข้อมูลแบบปิดเรียบร้อยแล้ว ผลลัพธ์ที่ได้เป็นดังนี้

1-itemset คือ $E:4$

2-itemset คือ $ED:3$

3-itemset คือ $BCF:4$ และ $ADE:2$

เห็นได้ว่า จากวิธีการข้างต้น เราสามารถลดพื้นที่การค้นหาในการหากลุ่มข้อมูลแบบปิด

3.2.3 ขั้นตอนวิธีการ IndexNClosed

ในหัวข้อนี้จะแสดงขั้นตอนทั้งหมดในการทำงานของ IndexNClosed ดังภาพที่ 3.1 – 3.5 รหัสเทียม (psudocode) เพื่อแสดงขั้นตอนการทำงานของขั้นตอนวิธีการ IndexNClosed ดังภาพที่ 3.1

Algorithm: IndexNCLOSED (D, N, k_{max})

Input : A database D, the upper bound of the length k_{max} and the desired number of k-itemsets N.

Output : N-most interesting closed itemsets NCI.

Method:

1. scan database D and find set of transaction ids in each item
2. Call **IndexClosed** subroutine find index array to construct initial generator
3. sort distinct items in descending order of their supports
4. insert the distinct items in queue q
5. S[1] = S[2] =… = S[k_{max}] = 0
6. C[1] = C[2] =… = C[k_{max}] = 0
7. N-mine(q)

ภาพที่ 3.1 รหัสเทียมขั้นตอนวิธีการ IndexNClosed

ฐานข้อมูล D ถูกอ่านครั้งแรกในทันที ในขั้นตอนที่ 1 จะเรียกซับรูทีนอินเด็กซ์โคลส (IndexClosed Subroutine) ดังภาพที่ 3.2 เพื่อหาอินเด็กซ์อาร์เรย์ของแต่ละชิ้นข้อมูล (บรรทัดที่ 30) จากนั้นชิ้นข้อมูลที่แตกต่างจะถูกเรียงเป็นอันดับที่แน่นอนด้วยโครงสร้างบิทเมป (Bitmap) (บรรทัดที่ 32) นำชิ้นข้อมูลที่ได้เรียงอันดับรายการเปลี่ยนแปลงเรียบร้อยแล้ว มาสร้างอินเด็กซ์

อาร์เรย์ โดยพิจารณาจากลำดับรายการเปลี่ยนแปลงที่เกิดของชิ้นข้อมูล แล้วนำไปพิจารณากับแต่ละชิ้นข้อมูลทั้งหมดว่า มีลำดับรายการเปลี่ยนแปลงใดที่เกิดเหมือนกันกับชิ้นข้อมูลที่เป็นอินเด็กซ์ ก็จะถือเป็นซับซูมอินเด็กซ์ของชิ้นข้อมูลนั้น และเก็บไว้เป็นเจเนเรเตอร์เริ่มต้นในคิว (บรรทัด 33-36)

ขั้นตอนที่ 1

Subroutine: IndexClosed // ซับรูทีนหาอินเด็กซ์อาร์เรย์

Method:

30. **for** each element *index[j]* of index array **do**
31. *index[j].item* = a_j
32. Represent the database ***D*** with bitmap
33. **for** each element *index[j]* of index array **do**
34. *index[j].subsume*=$\varnothing$
35. *Candidate* = $\cap_{t \in g(index[j].item)} t$;
36. Store the item corresponding to i_k^{th} position in candidate to *index[j].subsume* (excluding *index[j].subsume*) ,if the value of the i_k^{th} bit in candidate is set
37. **for** each item *i* in *index[j].subsume* **do**
38. **if** *supp*(*index[j].item*)==*supp(i)* **then**
39. Delete *index[k] with index[k].item== i ;*

ภาพที่ 3.2 รูทีนย่อย IndexClosed

หลังจากผ่านขั้นตอนในขั้นตอนที่ 1 ดังภาพที่ 3.2 แล้ว ให้ส่งค่าเจเนเรเตอร์เริ่มต้นที่เก็บไว้ในคิวไปยังขั้นตอนที่ 2 ดังภาพที่ 3.3 จากนั้น จะนำเจเนเรเตอร์เริ่มต้นจากคิวที่สร้างเก็บไว้ในขั้นตอนที่ 1 มาสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ *N* ลำดับแรก

ในส่วนของขั้นตอนที่ 2 ดังภาพที่ 3.3 จะตรวจสอบคิวว่าเป็นคิวว่างหรือไม่ ถ้าว่างให้หยุดสืบค้น (บรรทัดที่ 8) แต่ถ้าไม่ว่าง ให้ดึงกลุ่มข้อมูลออกมาเช็คว่ามีการสร้างกลุ่มข้อมูลนี้ซ้ำ (Duplicate) หรือไม่ โดยเรียกไปที่ซับรูทีน Generator_check(*X*) ดังภาพที่ 3.4 เมื่อตรวจเรียบร้อยแล้ว กรณีที่ไม่ใช่กลุ่มข้อมูลซ้ำ จะนำกลุ่มข้อมูลดังกล่าวนี้ไปสร้างโคลสเซอร์ โดยเรียกไปที่ซับรูทีน Closure_calculate(*X*) ดังภาพที่ 3.5 ทำขั้นตอนนี้จนกระทั่งได้กลุ่มข้อมูลแบบปิดที่น่าสนใจ *N* ลำดับแรก ตามที่เรากำหนดไว้ใน *N* ที่เราต้องการ

ขั้นตอนที่ 2

Subroutine: N-mine (*NCI*, *q*) // ซับรูทีนสำหรับการสืบค้นกลุ่มข้อมูลที่น่าสนใจ

Input: set of *N*-most interesting closed itemsets *NCI*,queue *q*

Output: set of *N*-most interesting closed itemsets *NCI*.

Method:

```
7.  while(1)
8.      if queue is empty then break
9.      X = de-queue()
10.     if Generator_check(X)
11.         Y = Closure_calculate(X)
12.         if supp(Y) ≥ S[Y.len] then
13.             NCI = NCI ∪ Y
14.         C[Y.len] =C[Y.len]+1
15.         if C[Y.len] ==N then
16.             S[Y.len] = supp(Y)
17.         if Y.len < k_max then
18.             P = {Y∪i_1, Y∪i_2, ..., Y∪i_n} ; i ∈ pre-items(Y)  //Extend Y
19.             while(j<n)
20.                 en-queue(Y∪i_j)
```

ภาพที่ 3.3 รูทีนย่อย N-mine

Subroutine: Generator_check(*X*) //ตรวจสอบเจเนอร์เรเตอร์ซ้ำซ้อนหรือไม่

Input: An itemset *X*

Output: If *X* is a non-duplicate generator, 1 is returned. Otherwise, 0 is returned.

Method:

```
21. while(supp(X)=supp(l_i)) ; l_i ∈ post-items(X)
22.     if(g(X) ⊆ g(l_i))
23.         return 0
24. return 1
```

ภาพที่ 3.4 รูทีนย่อยตรวจสอบตัวสร้าง เจเนเรเตอร์

Subroutine: Closure_calculate(X) // คำนวณโคสลเซอร์

Input: A itemset X

Output: A Closure of X, Y

Method:

25. $Y = X$
26. while($supp(X) = supp(j_j)$) ; $j_j \in pre\text{-}items(X)$
27. if($g(X) \subseteq g(j_j)$)
28. $Y = Y \cup j_j$
29. return Y

ภาพที่ 3.5 รูทีนย่อยคำนวณโคลสเซอร์

3.3 วิเคราะห์ขั้นตอนวิธี การIndexNClosed

ดังที่ได้กล่าวแล้วในหัวข้อ 3.2 ขั้นตอนวิธีการ IndexNClosed ที่นำเสนอนี้ประกอบด้วย 2 ขั้นตอนหลัก คือ 1) การสร้างอินเด็กซ์อาร์เรย์ และ 2) ขั้นตอนวิธีการ IndexNClosed สำหรับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก ในหัวข้อนี้จะกล่าวถึงการวิเคราะห์ความซับซ้อนด้านเวลา สำหรับขั้นตอนวิธีการ IndexNClosed เนื้อหาในหัวข้อนี้จึงแบ่งออกเป็นสองส่วน ได้แก่ การวิเคราะห์ความซับซ้อนด้านเวลาของขั้นตอนการสร้างอินเด็กซ์อาร์เรย์และการวิเคราะห์ความซับซ้อนด้านเวลาของขั้นตอนวิธีการ IndexNClosed โดยมีรายละเอียดดังนี้

3.3.1 ความซับซ้อนด้านเวลาของขั้นตอนการสร้างอินเด็กซ์อาร์เรย์

เมื่อพิจารณาการสร้างอินเด็กซ์อาร์เรย์ในรูปแบบของกราฟ กราฟดังกล่าวจะประกอบด้วย N โหนด และในการประมวลผลเพื่อสร้างอินเด็กซ์อาร์เรย์ จะกำหนดให้ประมวลผลเรียงตามจำนวนชิ้นข้อมูลที่แตกต่างกัน (Distinct item) ในฐานข้อมูล ด้วยคำสั่ง

For each element index[j] of index array do index[j].item = a_j

คำสั่งนี้มีความซับซ้อนด้านเวลาเท่ากับ $O(n)$ และเมื่อพิจารณาขั้นตอนการหาอินเด็กซ์อาร์เรย์ ดังแสดงในภาพที่ 3.2 สามารถวิเคราะห์ความซับซ้อนด้านเวลาของขั้นตอนได้ดังนี้

for each element *index[j]* of index array **do**

34. *index[j].subsume*=$\varnothing$

35. *Candidate* = $\cap_{t \in g(index[j].item)} t$;

36. Store the item corresponding to i_k^{th} position in candidate to *index[j].subsume* (excluding *index[j].subsume*) ,if the value of the i_k^{th} bit in candidate is set

ในขั้นตอนนี้ ชิ้นข้อมูลทุกชิ้นที่แตกต่างกัน ตั้งแต่รอบ $k=0$ ถึง $k<n$ จะทำการหาอินเด็กซ์อาร์เรย์ โดยเริ่มต้นให้ทุกชิ้นข้อมูลในอินเด็กซ์อาร์เรย์มีซับซูมอินเด็กซ์เป็นเซตว่าง จากนั้น จะเลือกชิ้นข้อมูลอินเด็กซ์มาคำนวณเพื่อหาซับซูมอินเด็กซ์ตั้งแต่รอบที่ $i = 0$ ถึง $i < n$ โดยนำเซตลำดับรายการเปลี่ยนแปลงที่เกิดขึ้นนั้นมาเปรียบเทียบกัน เพื่อหาว่ามีลำดับรายการเปลี่ยนแปลงใดที่เหมือนกัน ซึ่งก็คือการนำชิ้นข้อมูลเหล่านั้นมาอินเตอร์เซคชั่น (Intersection) กัน โดยการเปรียบเทียบเซตลำดับรายการเปลี่ยนแปลงจะเริ่มตั้งแต่รอบที่ $j = 0$ ถึง $j < n$ เมื่อได้ชิ้นข้อมูลอินเด็กซ์แล้วก็จะเก็บลงไว้ในคิวเพื่อใช้เป็นเจเนเรเตอร์เริ่มต้น ด้วยเหตุนี้ คำสั่งการสร้างอินเด็กซ์อาร์เรย์จึงมีความซับซ้อนด้านเวลาเท่ากับ $O(n^3)$

3.3.2 ความซับซ้อนด้านเวลาของขั้นตอนวิธีการ IndexNClosed

ขั้นตอนวิธีการ IndexNClosed เริ่มต้นโดยการสร้างอินเด็กซ์อาร์เรย์ก่อนเป็นอันดับแรก เพื่อจัดเตรียมเจเนเรเตอร์เริ่มต้น ดังที่กล่าวแล้วในหัวข้อ 3.3.1 ความซับซ้อนด้านเวลาของการหาอินเด็กซ์อาร์เรย์เท่ากับ $O(n^3)$ เมื่อได้เจเนเรเตอร์เริ่มต้นเก็บไว้ในคิวเรียบร้อยแล้ว จะนำเจเนเรเตอร์เริ่มต้นที่เตรียมไว้ในคิวมาสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับ โดยดึงชิ้นข้อมูล (de-queue) ที่มีค่าสนับสนุนสูงสุดออกจากคิว เพื่อขยายกลุ่มข้อมูลตามที่เรากำหนดไว้ Nk_{max} เมื่อ N คือจำนวนกลุ่มข้อมูลแบบปิดที่น่าสนใจ N จำนวนที่ต้องการ และ k_{max} คือ ขอบเขตบนของความยาวกลุ่มข้อมูลที่ต้องการ ซึ่งจำนวนกลุ่มข้อมูลใหม่ที่ถูกขยายจากกลุ่มข้อมูลแบบปิดนั้น มีขนาดใหญ่ที่สุดเท่ากับจำนวนชิ้นข้อมูลในคิว นั่นคือ เท่ากับ $|I|$ ดังนั้น จำนวนกลุ่มข้อมูลที่ถูกใส่ในคิวจึงเท่ากับ $Nk_{max}(|I|-1)$ กรณีที่แย่ที่สุดด้านเวลาสำหรับการดึงออกและการใส่กลุ่มข้อมูลในคิว คือ $O(Nk_{max} log(Nk_{max}(|I|-1)))$ และ $O(Nk_{max}(|I|-1) log(Nk_{max}(|I|-1)))$ ทั้งนี้เนื่องจากในการดึงออกและการใส่กลุ่มข้อมูลในคิว ใช้เวลาแบบคงที่ หลังจากที่ดึงกลุ่มข้อมูลที่มีค่าสนับสนุนสูงสุดออกจากคิว ชิ้นข้อมูลจะถูกตรวจเพื่อหากลุ่มข้อมูลซ้ำซ้อน โดยกรณีที่แย่สุดด้านเวลาของการตรวจกลุ่มข้อมูลที่ซ้ำซ้อนเกิดขึ้นเมื่อชิ้นข้อมูล X ไม่ถูกสร้างเป็นกลุ่มข้อมูลซ้ำซ้อน ซึ่งในกรณีดังกล่าว

เซตของลำดับรายการเปลี่ยนแปลงของ *post-item* ทั้งหมดจะถูกตรวจด้วย $g(X)$ เซตของลำดับรายการเปลี่ยนแปลงทุกตัวของ *post-item* จะถูกเปรียบเทียบเพื่อรวมเข้าไว้ใน $g(X)$ ซึ่งจำนวนสูงสุดของลำดับรายการเปลี่ยนแปลงของแต่ละ *post-item* มีจำนวนเท่ากับจำนวนลำดับรายการเปลี่ยนแปลงในคิว ซึ่งเท่ากับ $|Q|$ และจำนวนของ *post-item* มีขนาดใหญ่สุด คือ $|I|$ ดังนั้น กรณีที่แย่สุดด้านเวลาของการตรวจกลุ่มข้อมูลซ้ำซ้อน ในขั้นตอนนี้ คือ $O(|Q|(|I|-1))$ เมื่อเสร็จสิ้นการตรวจกลุ่มข้อมูลซ้ำซ้อนแล้ว ชิ้นข้อมูล X จะถูกคำนวณเพื่อหาโคลสเซอร์ กรณีที่แย่สุดด้านเวลาของขั้นตอนการหาโคลสเซอร์เกิดขึ้น เมื่อ $\forall j \in pre-items$ ที่ทำให้ $g(X) \subseteq g(j)$ จำนวนสูงที่สุดของ *pre-items* คือ $|I|$ ทุกเซตลำดับรายการเปลี่ยนแปลงของ *pre-item* จะถูกเปรียบเทียบเพื่อรวมเข้าไว้ใน $g(X)$ นอกจากนี้แล้ว เซตลำดับรายการเปลี่ยนแปลงของ *pre-item* แต่ละตัว มีขนาดใหญ่ที่สุดคือเท่ากันกับจำนวนลำดับรายการเปลี่ยนแปลงในคิว นั่นคือ $|Q|$ ดังนั้น กรณีแย่ที่สุดใช้เวลาหาโคลสเซอร์ในขั้นตอนนี้ เท่ากับ $O(|Q|(|I|-1))$ อย่างไรก็ดี ในขั้นตอนวิธีการ IndexNClosed นั้น ชิ้นข้อมูลในคิวที่ถูกใช้เป็น *post-item* เพื่อตรวจความซ้ำซ้อนจะไม่ถูกใช้เป็น *pre-item* เพื่อคำนวณโคลสเซอร์ในทำนองเดียวกัน ชิ้นข้อมูลในคิวที่ถูกใช้เป็น *pre-item* เพื่อคำนวณโคลสเซอร์ จะไม่ถูกใช้เป็น *post-item* เพื่อตรวจความซ้ำซ้อน ด้วยเหตุนี้ กรณีแย่ที่สุดใช้เวลาตรวจความซ้ำซ้อนและการคำนวณโคลสเซอร์ในขั้นตอนนี้คือ $O(|Q|(|I|-1))$ [5]

กล่าวโดยสรุป คือ ขั้นตอนวิธีการ IndexNClosed เพื่อสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก นั้นมีความซับซ้อนด้านเวลาเท่ากับ

$$O(n^3 + Nk_{max}\log(Nk_{max}(|I|-1)) + Nk_{max}(|I|-1)\log(Nk_{max}(|I|-1)))$$

บทที่ 4
การวัดประสิทธิภาพ

จากที่กล่าวในบทที่ 3 ขั้นตอนวิธีการ IndexNClosed สำหรับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก ประกอบด้วยสองขั้นตอนสำคัญ ได้แก่ (1) การสร้างอินเด็กซ์อาร์เรย์ และ (2) การสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก เห็นได้ว่า ในขั้นตอนที่ 2 นั้น ได้จากการนำผลลัพธ์จากขั้นตอนที่ 1 มาสืบค้นกลุ่มข้อมูล ในบทนี้จะนำเสนอการทดลองและผลการทดลองเพื่อศึกษาประสิทธิภาพของขั้นตอนวิธีการ IndexNClosed โดยเปรียบเทียบกับขั้นตอนวิธีการ NCLOSED [5] ซึ่งในที่นี้ ประสิทธิภาพที่สนใจศึกษา ได้แก่ ประสิทธิภาพในการลดพื้นที่การค้นหากลุ่มข้อมูลแบบปิด และประสิทธิภาพด้านเวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับ

4.1 การทดลอง

4.1.1 ลักษณะของฐานข้อมูลที่นำมาทดลอง

ชุดข้อมูลที่ใช้ทดลอง ได้แก่ ชุดข้อมูลแบบหนาแน่น (Dense datasets) และชุดข้อมูลแบบกระจาย (Sparse dataset) โดยชุดข้อมูลหนาแน่น ได้แก่ Chess, Mushroom และ Connect ซึ่งชุดข้อมูล Chess และ Connect เป็นแบบหนาแน่นมาก และลักษณะของชุดข้อมูลทั้ง 2 ชนิดนี้ได้จากข้อมูลการเล่นเกม ขณะที่ชุดข้อมูล Mushroom เป็นแบบหนาแน่นน้อย ลักษณะของชุดข้อมูล Mushroom ประกอบด้วยลักษณะของเห็ดหลากหลายชนิด ในส่วนชุดข้อมูลแบบกระจายที่ใช้ในการทดลอง ได้แก่ T10I4D8K และ Gazelle โดยชุดข้อมูล T10I4D8K เป็นข้อมูลสังเคราะห์ที่สร้างจาก IBM ส่วนชุดข้อมูล Gazella ได้จากการคลิกข้อมูลจากเว็บ Gazelle.com

ลักษณะของชุดข้อมูลที่ใช้ในการทดลองแสดงไว้ในตารางที่ 4.1 และฐานข้อมูลที่ใช้ในการทดลองสามารถดาวน์โหลดได้จาก http://fimi.cs.helsinki.fi/

ตารางที่ 4.1 ลักษณะของชุดข้อมูลที่ใช้ในการทดลอง

Datasets	#Items	#Transaction	Avg.Length
ชุดข้อมูลหนาแน่น(Dense Dataset)			
Chess	75	3,196	37
Mushroom	119	8,124	23
Connect	129	65,557	43
ชุดข้อมูลแบบกระจาย (Sparse Dataset)			
T10I4D8K	862	8000	10
Gazelle	333	10000	2.5

4.1.2 โปรแกรมที่ใช้ในการทดลอง

โปรแกรมที่ใช้ในการทดสอบกับชุดข้อมูล คือ IndexNClosed กับ NCLOSED[5] ซึ่งทั้งสองโปรแกรมเป็นการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก

4.1.3 เครื่องมือที่ใช้ในการวิจัย

เครื่องมือที่ใช้ในการทดสอบ มีคุณสมบัติดังต่อไปนี้

หน่วยประมวลผลกลาง (CPU) : Core 2 Duo 1.6 GHz
หน่วยความจำหลัก (RAM) : 2 GB DDR2
หน่วยความจำสำรอง (Hard Disk) : 120 GB
ระบบปฏิบัติการ (OS) : Window XP Professional ServicePack3
โปรแกรมที่ใช้ในการพัฒนา : Visual Studio C++ โดยใช้ g++ Compiler

4.1.4 การออกแบบการทดลองและเหตุผล

งานวิจัยนี้สนใจศึกษาประสิทธิภาพของขั้นตอนวิธีการ IndexNClosed ในการลดพื้นที่การค้นหากลุ่มข้อมูลแบบปิด โดยพิจารณาจากจำนวนกลุ่มข้อมูลแบบปิดที่ถูกสร้างขึ้น และศึกษาประสิทธิภาพด้านเวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก โดยเปรียบเทียบประสิทธิภาพดังกล่าวกับขั้นตอนวิธีการ NCLOSED ด้วยเหตุนี้ การทดลองจึงแบ่งออกเป็น 2 ส่วน ดังนี้

ส่วนที่ 1 ทำการทดลองชุดข้อมูลหนาแน่น Chess, Mushroom และ Connect กับโปรแกรม IndexNClosed และ NCLOSED โดยกำหนดค่า k_{max} มีค่าคงที่เท่ากับ 4 และกำหนดค่า N = 10, 50, 100, 150, 200, 250, 300, 350, 400 และ 500

ส่วนที่ 2 ทำการทดลองชุดข้อมูลกระจาย T10I4D8K และ Gazelle กับโปรแกรม IndexNClosed และ NCLOSED โดยกำหนดค่า k_{max} มีค่าคงที่เท่ากับ 4 และกำหนดค่า N = 10, 50, 100, 150, 200, 250, 300, 350, 400 และ 500

4.2 ผลการทดลอง

ผลการทดลองโดยใช้โปรแกรม IndexNClosed และ NCLOSED กับชุดข้อมูลหนาแน่นและชุดข้อมูลกระจาย แบ่งออกเป็น 2 ส่วน ดังนี้

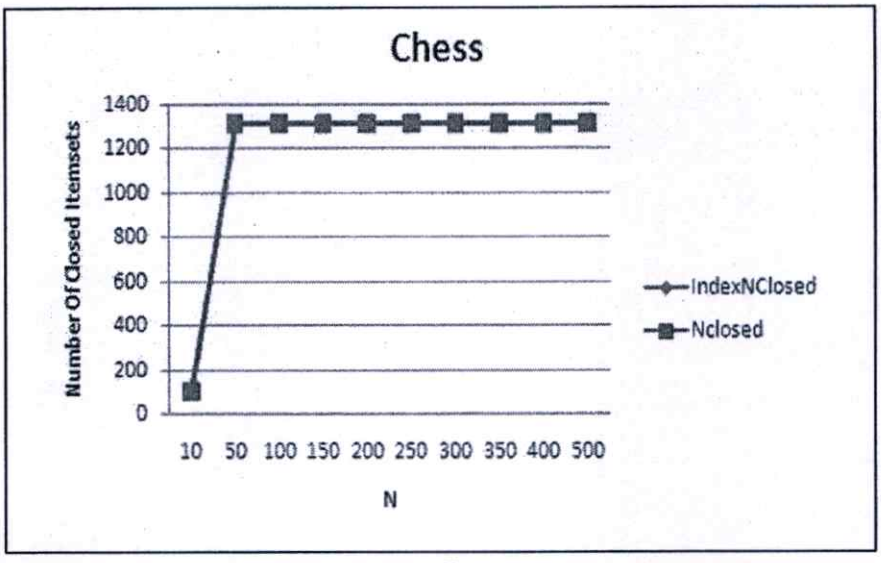
1. เปรียบเทียบจำนวนกลุ่มข้อมูลแบบปิดที่ถูกสร้างขึ้น และ เวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก โดยขั้นตอนวิธีการ IndexNClosed และ NCLOSED กรณีชุดข้อมูลแบบหนาแน่น ได้แก่ Chess, Mushroom และ Connect

2. เปรียบเทียบจำนวนกลุ่มข้อมูลแบบปิดที่ถูกสร้างขึ้น และเวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก โดยขั้นตอนวิธีการ IndexNClosed และ NCLOSED กรณีชุดข้อมูลแบบกระจาย ได้แก่ T10I4D8K และ Gazelle

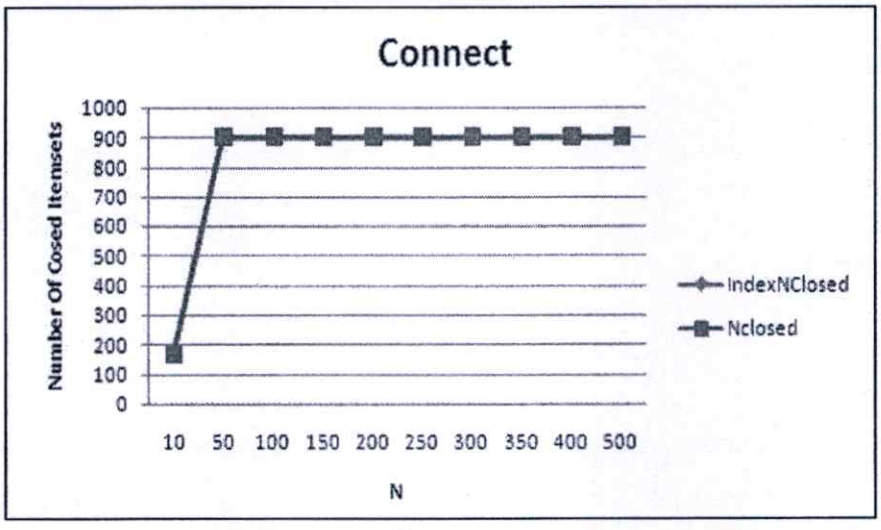
3. เปรียบเทียบจำนวนการสร้างเจเนเรเตอร์เริ่มต้นก่อนการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก

4.2.1 จำนวนกลุ่มข้อมูลแบบปิดที่ถูกสร้างขึ้นและเวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก กรณีชุดข้อมูลแบบหนาแน่น

ภาพที่ 4.1 แสดงจำนวนกลุ่มข้อมูลแบบปิดที่ถูกสร้างขึ้นโดยขั้นตอนวิธีการ IndexNClosed และ NCLOSED เมื่อทดลองกับชุดข้อมูลแบบหนาแน่น โดยกำหนด k_{max} = 4 และ N มีค่าระหว่าง 10 ถึง 500 เห็นได้ว่า ในชุดข้อมูลแบบหนาแน่นมาก ซึ่งในที่นี้ ได้แก่ ชุดข้อมูล Chess และ Connect ประสิทธิภาพในการสร้างกลุ่มข้อมูลแบบปิดของทั้งสองขั้นตอนวิธีไม่แตกต่างกัน สำหรับทุกค่า N ดังภาพที่ 4.1 (a) และ 4.1 (b) สำหรับชุดข้อมูล Mushroom ซึ่งเป็นชุดข้อมูลแบบหนาแน่นน้อย ขั้นตอนวิธีการ IndexNClosed สามารถลดพื้นที่ในการสร้างกลุ่มข้อมูลแบบปิดได้ดีกว่าขั้นตอนวิธีการ NCLOSED สำหรับทุกค่า N ดังภาพที่ 4.1 (C)

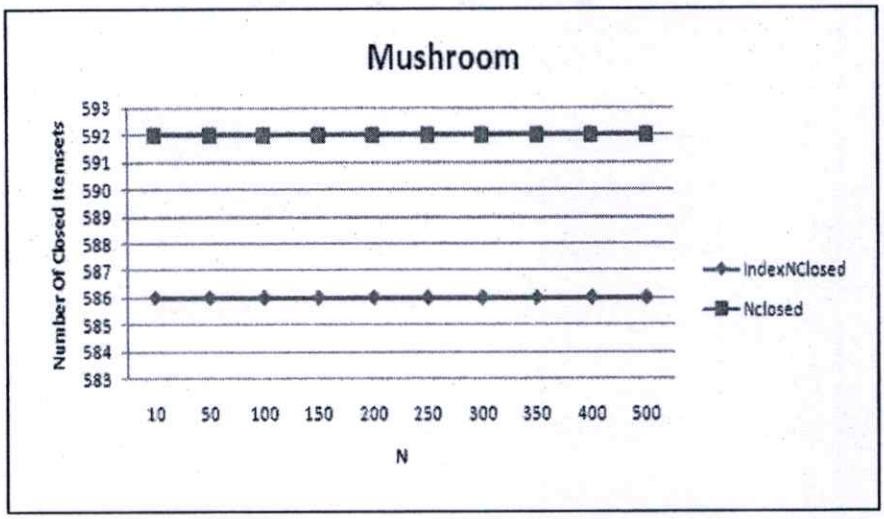


(a) ชุดข้อมูล Chess



(b) ชุดข้อมูล Connect

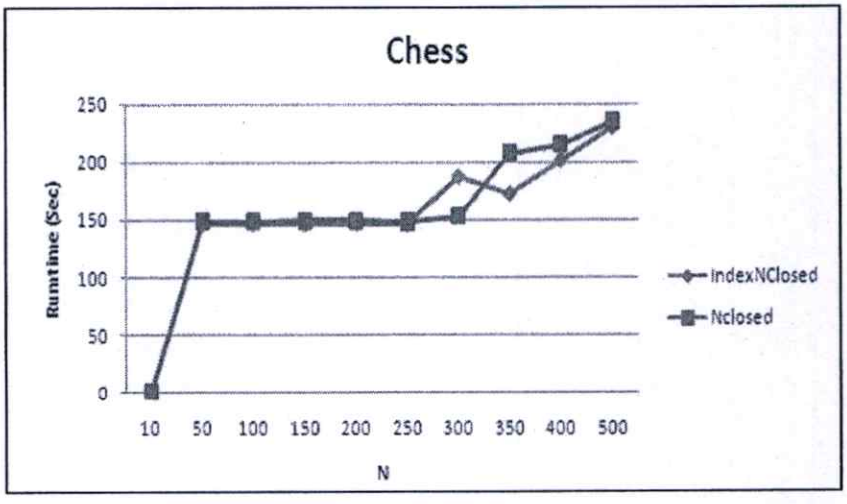
ภาพที่ 4.1 แสดงจำนวนกลุ่มข้อมูลแบบปิดที่ถูกสร้างขึ้น ข้อมูลแบบหนาแน่น



(c) ชุดข้อมูล Mushrom

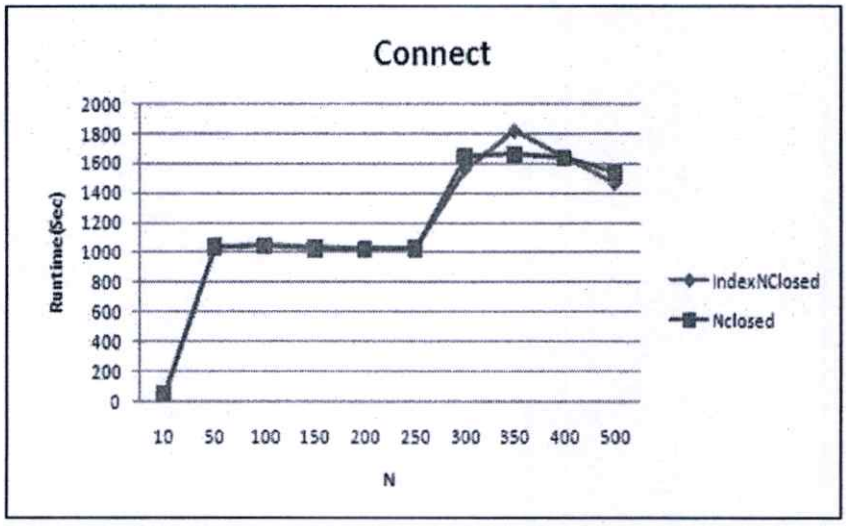
ภาพที่ 4.1 จำนวนกลุ่มข้อมูลแบบปิดที่ถูกสร้างขึ้น ข้อมูลแบบหนาแน่น

ภาพที่ 4.2 แสดงเวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก โดยขั้นตอนวิธีการ IndexNClosed และ NCLOSED เมื่อทดลองกับชุดข้อมูลแบบหนาแน่น โดยกำหนด $k_{max} = 4$ และ N มีค่าระหว่าง 10 ถึง 500 เห็นได้ว่า ประสิทธิภาพด้านเวลาที่ใช้ในการสืบค้นของทั้งสองขั้นตอนวิธีการ ไม่แตกต่างกัน จากภาพที่ 4.1 (a) , 4.1 (b) และ 4.1 (c) จะเห็นว่า เวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก โดยขั้นตอนวิธีการ IndexNClosed และ NCLOSED ใกล้เคียงกัน

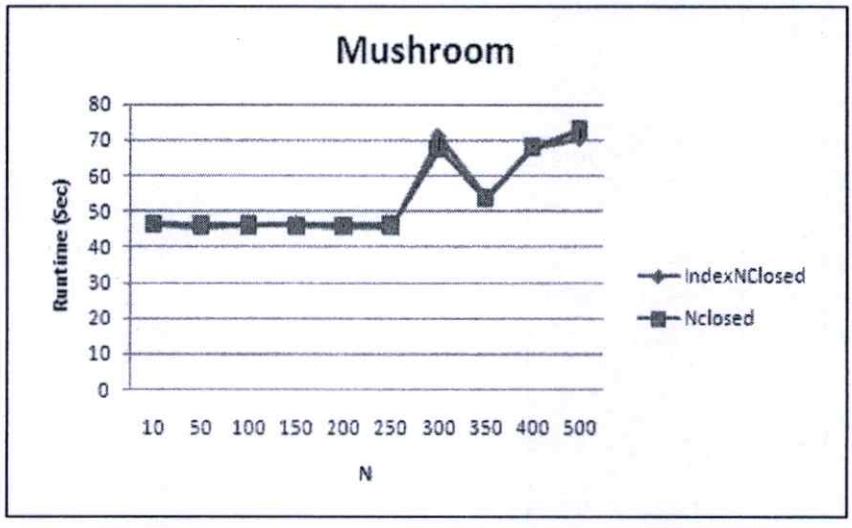


(a) ชุดข้อมูล Chess

ภาพที่ 4.2 เวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก ข้อมูลแบบหนาแน่น



(b) ชุดข้อมูล Connect



(c) ชุดข้อมูล Mushrom

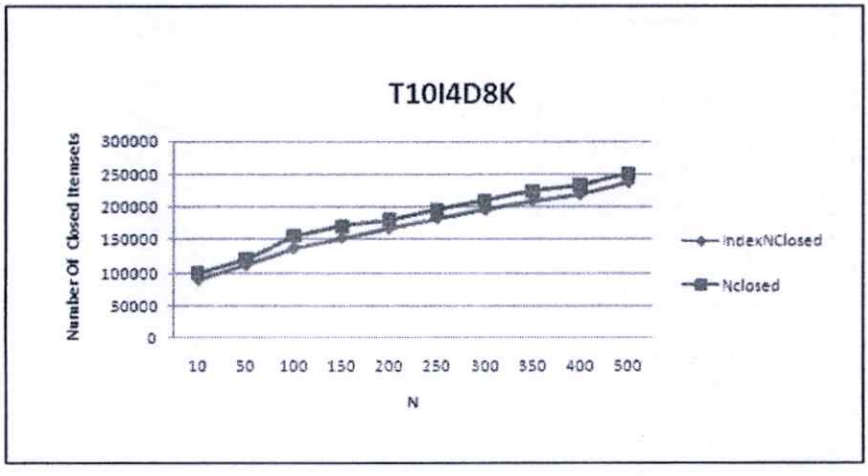
ภาพที่ 4.2 เวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิด N ลำดับแรก ข้อมูลแบบหนาแน่น

4.2.2 จำนวนกลุ่มข้อมูลแบบปิดที่ถูกสร้างขึ้นและเวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก กรณีชุดข้อมูลแบบกระจาย

ตารางที่ 4.2 แสดงจำนวนกลุ่มข้อมูลแบบปิดที่ถูกสร้างขึ้นของขั้นตอน IndexNClosed และ NCLOSED ของชุดข้อมูลแบบกระจาย

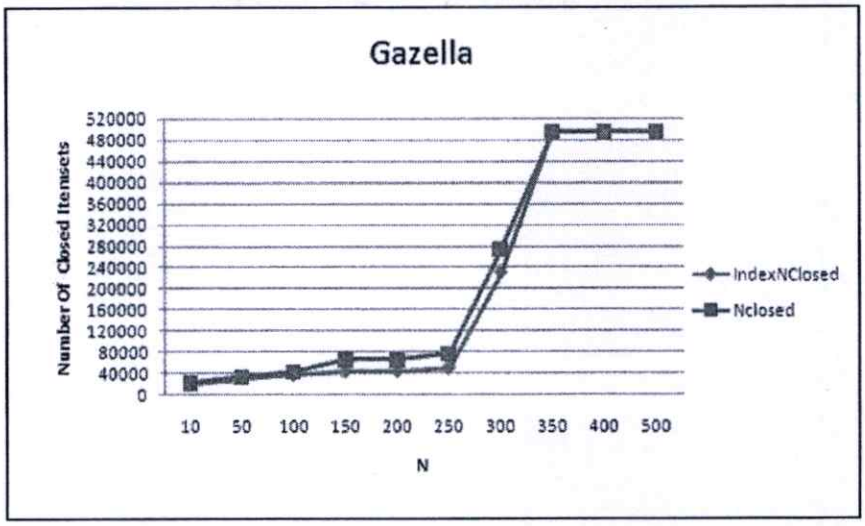
N	10	50	100	150	200	250	300	350	400	500
ชุดข้อมูล T10I4D8K										
IndexNClosed	90000	112000	138246	152708	168122	183334	196199	210469	220441	238315
NCLOSED	100208	121067	156087	171204	180331	195822	209510	233504	233504	251108
ชุดข้อมูล Gazelle										
IndexNClosed	20600	30400	36716	42227	44115	49140	230377	494592	494592	494592
NCLOSED	21624	33392	40305	65361	65035	75156	273639	494592	494592	494592

ภาพที่ 4.3 แสดงจำนวนกลุ่มข้อมูลแบบปิดที่ถูกสร้างขึ้นโดยขั้นตอนวิธีการ IndexNClosed และ NCLOSED เมื่อทดลองกับชุดข้อมูลแบบกระจาย โดยกำหนด k_{max} = 4 และ N มีค่าระหว่าง 10 ถึง 500 เห็นได้ว่า ขั้นตอนวิธีการ IndexNClosed สามารถลดพื้นที่ในการสร้างกลุ่มข้อมูลแบบปิดได้ดีกว่าขั้นตอนวิธีการ NCLOSED จากภาพที่ 4.3 (a) และ 4.3 (b) จะเห็นว่า กลุ่มข้อมูลแบบปิดถูกสร้างขึ้นจากชุดข้อมูล T10I4D8K และ Gazelle โดยขั้นตอนวิธีการ IndexNClosed มีจำนวนน้อยกว่า ขั้นตอนวิธีการ NCLOSED สำหรับทุกค่า N



(a) ชุดข้อมูล T10I4D8K

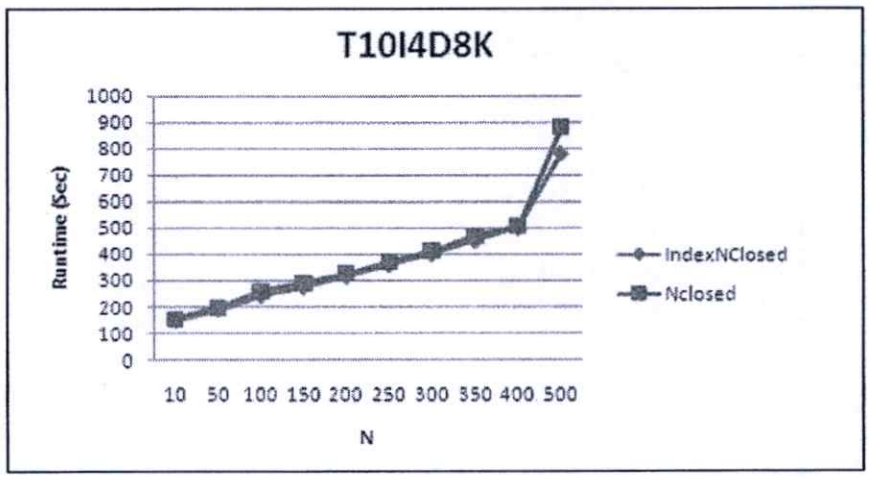
ภาพที่ 4.3 แสดงจำนวนกลุ่มข้อมูลแบบปิดที่ถูกสร้างขึ้น ข้อมูลแบบกระจาย



(b) ชุดข้อมูลGazelle

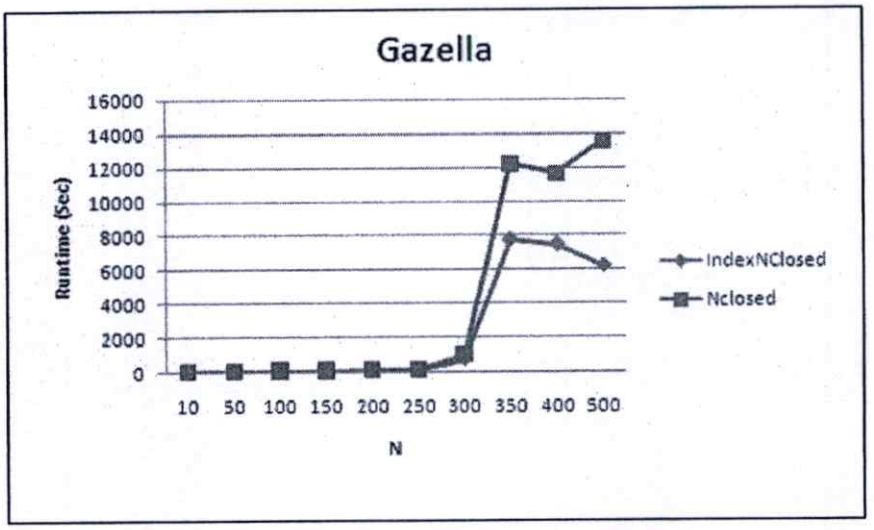
ภาพที่ 4.3 แสดงจำนวนกลุ่มข้อมูลแบบปิดที่ถูกสร้าง ข้อมูลแบบกระจาย

ภาพที่ 4.4 แสดงเวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิด N ลำดับแรกโดยขั้นตอนวิธีการ IndexNClosed และ NCLOSED เมื่อทดลองกับชุดข้อมูลแบบกระจาย โดยกำหนด k_{max} = 4 และ N มีค่าระหว่าง 10 ถึง 500 เห็นได้ว่า ประสิทธิภาพด้านเวลาที่ใช้ในการสืบค้นของทั้งสองขั้นตอนวิธีการ ไม่แตกต่างกัน เมื่อ N มีค่าน้อย แต่ขั้นตอนวิธีการ IndexNClosed มีประสิทธิภาพดีกว่าขั้นตอนวิธีการ NCLOSED เมื่อ N มีค่ามาก จากภาพที่ 4.4 (a) และ 4.4 (b) จะเห็นว่า เวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก โดยขั้นตอนวิธีการ IndexNClosed น้อยกว่าเวลาที่ใช้ในการสืบค้นโดยขั้นตอนวิธีการ NCLOSED เมื่อ N มีค่าตั้งแต่ 300 ขึ้นไป



(a) ชุดข้อมูล T10I4D8K

ภาพที่ 4.4 เวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก ข้อมูลแบบกระจาย



(b) ชุดข้อมูลGazelle

ภาพที่ 4.4 เวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก ข้อมูลแบบกระจาย

4.2.3 เปรียบเทียบจำนวนเจเนเรเตอร์เริ่มต้นก่อนการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก

ตารางที่ 4.3 แสดงจำนวนเจเนเรเตอร์เริ่มต้นก่อนการสืบค้นข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก

Datasets	#Items	NCLOSED	IndexNClosed
ชุดข้อมูลหนาแน่น(Dense Dataset)			
Chess	75	75	75
Mushroom	119	119	95
Connect	129	129	129
ชุดข้อมูลแบบกระจาย (Sparse Dataset)			
T10I4D8K	862	862	746
Gazelle	333	333	266

จากตารางที่ 4.3 เป็นการแสดงจำนวนเจเนเรเตอร์เริ่มต้นก่อนทำการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก เห็นได้ว่าจำนวนเจเนเรเตอร์ที่เก็บลงคิวก่อนการสืบค้นกลุ่มข้อมูลแบบปิดของขั้นตอนวิธีการ IndexNClosed เก็บลงในคิวน้อยกว่าขั้นตอนวิธีการ NCLOSED ทั้งนี้เนื่องจากขั้นตอนวิธีการ IndexNClosed ทำการสร้างอินเด็กซ์อาร์เรย์ก่อนเก็บลงคิว ดังนั้นสร้างอินเด็กซ์อาร์เรย์จึงสามารถลดเจเนเรเตอร์เริ่มต้นได้ดีกับข้อมูลแบบกระจาย

จากผลการทดลองทั้งในกรณีชุดข้อมูลแบบหนาแน่นและชุดข้อมูลแบบกระจาย เมื่อพิจารณาประสิทธิภาพในการลดพื้นที่การค้นหากลุ่มข้อมูลแบบปิด จะเห็นว่า ในกรณีชุดข้อมูลแบบหนาแน่น ขั้นตอนวิธีการ IndexNClosed และขั้นตอนวิธีการ NCLOSED มีประสิทธิภาพไม่แตกต่างกัน อย่างไรก็ดี ขั้นตอนวิธีการ IndexNClosed สามารถลดพื้นที่การค้นหาได้ดีกว่าขั้นตอนวิธีการ NCLOSED ในชุดข้อมูลแบบหนาแน่นน้อย สำหรับทุกค่า N ดังภาพที่ 4.1 (c) ส่วนในกรณีชุดข้อมูลแบบกระจาย ขั้นตอนวิธีการ IndexNClosed มีประสิทธิภาพดีกว่าขั้นตอนวิธีการ NCLOSED สำหรับทุกค่า N ดังภาพที่ 4.3

เมื่อพิจารณาประสิทธิภาพด้านเวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิด พบว่า ในกรณีชุดข้อมูลแบบหนาแน่น ทั้งสองขั้นตอนวิธีมีประสิทธิภาพใกล้เคียงกัน แต่เวลาที่ใช้ในชุดข้อมูลแบบกระจาย จะเห็นว่า เมื่อ N มีค่าน้อย ขั้นตอนวิธีการ IndexNClosed จะใช้เวลาใกล้เคียงกันกับขั้นตอนวิธีการ NCLOSED และ เมื่อ N มีค่าตั้งแต่ 300 ขึ้นไป ขั้นตอนวิธี IndexNClosed ใช้เวลาน้อยกว่าขั้นตอนวิธี NCLOSED

ดังนั้น ขั้นตอนวิธีการ IndexNClosed จึงเหมาะสมกับชุดข้อมูลแบบกระจายมากกว่าชุดข้อมูลแบบหนาแน่น ทั้งนี้เนื่องจาก ชุดข้อมูลแบบกระจายนั้น ชิ้นข้อมูลที่ปรากฏในลำดับรายการเปลี่ยนแปลง มีจำนวนน้อย และชิ้นข้อมูลเหล่านั้นมีความสัมพันธ์ต่อกันไม่มาก แต่สำหรับชุดข้อมูลแบบหนาแน่น ลำดับรายการเปลี่ยนแปลงมีความยาวมาก และชิ้นข้อมูลมีความสัมพันธ์ต่อกันค่อนข้างสูง

บทที่ 5
สรุปและข้อเสนอแนะ

5.1 สรุป

ขั้นตอนวิธีการ IndexNClosed เพื่อสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก เป็นขั้นตอนวิธีที่ได้นำอินเด็กซ์อาร์เรย์มาประยุกต์ใช้ในการหาชิ้นข้อมูลที่ปรากฏอยู่ด้วยกันเสมอ โดยชิ้นข้อมูลที่เกิดขึ้นพร้อมกันบ่อยครั้งและต่างก็ใช้ค่าสนับสนุนเดียวกัน ชิ้นข้อมูลเหล่านั้นจะสามารถรวมเข้าด้วยกันได้ ซึ่งเป็นการจัดเตรียมเจเนเรเตอร์เริ่มต้นก่อนนำเข้าสู่กระบวนการสืบค้นเพื่อลดเนื้อที่ในการสืบค้นข้อมูล ในส่วนของการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก ขั้นตอนวิธีการ IndexNClosed ที่นำเสนอนี้จะสร้างกลุ่มข้อมูลแบบปิดโดยไม่เก็บกลุ่มข้อมูลแบบปิดคู่แข่ง และมีการตรวจการซ้ำของกลุ่มข้อมูลก่อนการคำนวณโคลสเซอร์ เพื่อหลีกเลี่ยงการคำนวณที่ซ้ำซ้อนของกลุ่มข้อมูลปิดเดียวกัน นอกจากนี้ ยังได้ใช้เทคนิคการค้นหาคำตอบดีที่สุดเพื่อให้ได้กลุ่มข้อมูลแบบปิดที่เรียงลำดับค่าสนับสนุนจากมากไปน้อย ซึ่งเป็นการลดขั้นตอนการหาค่าสนับสนุนสุดท้าย

จากผลการทดลอง เพื่อศึกษาประสิทธิภาพของขั้นตอนวิธีการ IndexNClosed โดยเปรียบเทียบกับขั้นตอนวิธีการ NCLOSED โดยเกณฑ์ที่ใช้ในการเปรียบเทียบ คือ จำนวนกลุ่มข้อมูลแบบปิดที่ถูกสร้างขึ้น และ เวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก โดยทดสอบขั้นตอนวิธีทั้งสองกับชุดข้อมูลหนาแน่นและชุดข้อมูลกระจาย สรุปผลการทดลองได้ดังนี้

5.1.1. ในกรณีชุดข้อมูลแบบหนาแน่น ขั้นตอนวิธีการ IndexNClosed และ ขั้นตอนวิธีการ NCLOSED มีประสิทธิภาพไม่แตกต่างกัน อย่างไรก็ดี ขั้นตอนวิธีการ IndexNClosed สามารถลดพื้นที่การค้นหาได้ดีกว่าขั้นตอนวิธีการ NCLOSED ในชุดข้อมูลแบบกระจาย สำหรับทุกค่า N

5.1.2. เมื่อพิจารณาประสิทธิภาพด้านเวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิด พบว่า ในกรณีชุดข้อมูลแบบหนาแน่น ทั้งสองขั้นตอนวิธีมีประสิทธิภาพใกล้เคียงกัน แต่เวลาที่ใช้ในชุดข้อมูลแบบกระจาย จะเห็นว่า เมื่อ N มีค่าน้อย ขั้นตอนวิธีการ IndexNClosed จะใช้เวลาใกล้เคียงกันกับขั้นตอนวิธีการ NCLOSED และ เมื่อ N มีค่าตั้งแต่ 300 ขึ้นไป ขั้นตอนวิธีการ IndexNClosed ใช้เวลาน้อยกว่าขั้นตอนวิธีการ NCLOSED

สรุปได้ว่า ดังนั้น ขั้นตอนวิธีการ IndexNClosed จึงเหมาะสมกับชุดข้อมูลแบบกระจายมากกว่าชุดข้อมูลแบบหนาแน่น ทั้งนี้เนื่องจาก ชุดข้อมูลแบบกระจายนั้น ชิ้นข้อมูลที่ปรากฏในลำดับรายการเปลี่ยนแปลง มีจำนวนน้อย และชิ้นข้อมูลเหล่านั้นมีความสัมพันธ์ต่อกันไม่มาก แต่

สำหรับชุดข้อมูลแบบหนาแน่น ลำดับรายการเปลี่ยนแปลงมีความยาวมากและขนาดของความยาวของลำดับรายการเปลี่ยนแปลงจะเท่ากันทั้งชุดข้อมูล และชิ้นข้อมูลมีความสัมพันธ์ต่อกันค่อนข้างสูง

5.2 ข้อเสนอแนะ

ขั้นตอนวิธี IndexNClosed สำหรับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก จะสร้างกลุ่มข้อมูลแบบปิดด้วยการหาโคลสเซอร์ ซึ่งกลุ่มข้อมูลแบบปิดจะถูกขยายให้ใหญ่ขึ้นด้วยกลุ่มข้อมูล ซึ่งกลุ่มข้อมูลบางตัวอาจไม่เป็นคำตอบ หรือ กลุ่มข้อมูลบางตัวไม่มีความจำเป็นต้องใส่เข้าไปในคิว ทั้งนี้เพราะกลุ่มข้อมูลดังกล่าวอาจไม่ถูกสืบค้นเลย แนวทางหนึ่งในการพัฒนาขั้นตอนวิธีการ IndexNClosed ต่อไปในอนาคต คือ ควรปรับปรุงขั้นตอนวิธีให้สามารถตัดกลุ่มข้อมูลที่ไม่จำเป็นออกไป ก่อนที่จะใส่เข้าไปในคิว

จากผลการทดลองเพื่อศึกษาประสิทธิภาพของขั้นตอนวิธีการ IndexNClosed พบว่าสามารถลดพื้นที่การค้นหาข้อมูลได้ดีกับชุดข้อมูลแบบกระจาย ดังนั้น แนวทางหนึ่งในการพัฒนาขั้นตอนวิธีการ IndexNClosed ต่อไป คือ การพัฒนาขั้นตอนวิธีนี้ให้สามารถลดพื้นที่การค้นหาข้อมูลได้ดีกับชุดข้อมูลแบบหนาแน่น อย่างไรก็ดี เทคนิคการค้นหาคำตอบดีที่สุดจะใช้เวลามากขึ้นเมื่อข้อมูลมีขนาดใหญ่ ดังนั้น แนวทางหนึ่งที่น่าจะเป็นไปได้ คือ การพัฒนาขั้นตอนวิธีควบคู่กับโครงสร้างข้อมูลอื่นๆ ที่อาจทำให้ขั้นตอนวิธีนี้มีประสิทธิภาพมากขึ้น

บรรณานุกรม

[1] A.W. –C. Fu, R. W. – W. Kwong, and J. Tang, "Mining N-most interesting Itemsets", In Proc. of 12th Symposium on Methodologies for Intelligent System (ISMIS), pp. 59-67, London, UK, 2000.

[2] Y-L. Cheng and A. Fu, "An FP-tree approach for mining N-most interesting itemsets", In Proc. of the SPIE Conference on Data Mining, pp. 460-471, 2002. Proc. of the SPIE Conference on Data Mining , pp. 460-471, 2002

[3] S. Ngan, T. Lam, R.C. Wong, and A. W. Fu, "Mining N-most interesting itemsets without support threshold by COFI-tree", *Journal of Business Intelligence and Data Mining*, Vol. 1, No. 1. 2005.

[4] M. U. Arshad, M. N. Ayyaz, "Mining N-most Interesting Itemsets Using Support-Ordered Tries", pp. 592-599, Dubai, 2006.

[5] Songram, P. Boonjing, V. "N-Most Interesting Closed Itemset Mining", Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on, Vol.1, pp. 619-624, Busan, 2008

[6] Wei Song,Bingru Yang and Zhangyan Xu " Index-CloseMiner: an improved algorithm for Mining frequent closed itemset", Intelligent Data Analysis, Volume 12 , Issue 4 pp. 321-338 , 2008

[7] C. Lucchese, S. Orlando and R. Perego, "DCI_Closed: a fast and memory efficient algorithm to mine frequent closed itemsets", In Proceedings of the ICDM 2004 Workshop on Frequent Itemset Mining Implementations (FIMI'04), 2004.

[8] J. Pei, J. Han, and R. Mao. "Closet: An efficient algorithm for mining frequent closed itemsets." In SIGMOD Int'l Workshop on Data Mining and Knowledge Discovery, May 2000.

[9] J. Wang, J. Han, and J. Pei, "Closet+: Searching for the Best Strategies for Mining Frequent Closed Itemset" In KDD'03 Washington,DC,August 2003.

[11] M. Zaki and C. Hsiao. "Charm: An Efficient Algorithm for Closed Itemset Mining". In *Proceedings of SIAM'02*, Arlington,Apr. 2002.

[12] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, "Discovering Frequent Closed Itemsets for Association Rules," Proc. Of the 7th ICDT Conference, January 1999.

[13] J. Han, J. Wang, Y. Lu and P. Tzvetkov, "Mining Top-k Frequent Closed Patterns without Minimum Support," In Proc.of IEEE ICDM Conference on Data Mining, 2002.

ภาคผนวก

ผลงานที่ได้รับการตีพิมพ์

ขั้นตอนวิธี IndexNClosed สำหรับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก

An IndexNClosed Algorithm for Mining N-most Interesting Closed Itemsets

พิพิธพร โพนตุแสง[1] และ วีระ บุญจริง[1,2]

[1]ห้องปฏิบัติการวิศวกรรมระบบซอฟต์แวร์ สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ถ.ฉลองกรุง เขตลาดกระบัง กรุงเทพ 10520

[2]ศูนย์ความเป็นเลิศแห่งชาติด้านคณิตศาสตร์, สบว., กรุงเทพฯ 10400

E-mail : [1]s9067551@kmitl.ac.th, [2]kbveera@kmitl.ac.th

Abstract **N-most interesting closed itemsets mining was proposed to avoid a generation of redundant itemsets and a specification of an appropriate minimum support threshold. A too big threshold could give no answer whereas a too small one probably yields a large number of redundant itemsets. In addition, the determination of the optimal threshold is hard for users having no knowledge of mining queries and task-specific data. This paper adopts index array for mining N-most interesting closed itemsets and improve an efficient algorithm, called IndexNClosed. The index array is presented, which is used to discover those items that always appear together. Due to index array, items coinciding and sharing the same support are merged together and preserved as initial generators. Then generators are used in the first *N*-most Interesting Closed Itemsets mining process. The algorithm uses a Best-First Search strategy to mine closed itemsets in descending order of their supports. This leads to an efficient pruning of unnecessary itemsets. In addition, the experiments are conducted from large database to compare the performance of algorithm IndexNClosed with NCLOSED. The experimental results indicate that the proposed algorithm IndexNClosed outperforms. It uses search space less than the other.**

Keywords : closed itemset, N-most interesting itemset, index array

บทคัดย่อ การสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ *N* ลำดับแรกถูกเสนอขึ้นเพื่อเลี่ยงการสร้างข้อมูลที่ซ้ำซ้อนและปัญหาการกำหนดค่าสนับสนุนขั้นต่ำที่เหมาะสม เนื่องจากถ้าผู้ใช้กำหนดค่าสนับสนุนสูงเกินไป อาจพบกลุ่มข้อมูลจำนวนน้อยมาก แต่ถ้ากำหนดค่าสนับสนุนต่ำเกินไป ก็อาจพบกลุ่มข้อมูลมากเกินจำเป็น งานวิจัยนี้ได้นำอินเด็กซ์อาร์เรย์ (Index Array) มาใช้ในการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ *N* ลำดับแรก เรียกขั้นตอนวิธีการดังกล่าวว่า IndexNClosed อินเด็กซ์อาร์เรย์ถูกใช้เพื่อหาชิ้นข้อมูล (Item) ที่ปรากฏอยู่ด้วยกันเสมอ โดยชิ้นข้อมูลที่เกิดขึ้นพร้อมกันและใช้ค่าสนับสนุนเดียวกัน จะถูกรวมเข้าด้วยกันและเก็บเป็นกลุ่มข้อมูลเริ่มต้น และนำเข้ากระบวนการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ *N* ลำดับแรกด้วยเทคนิคการค้นหาคำตอบดีที่สุด วิธีดังกล่าวสร้างกลุ่มข้อมูลตามลำดับค่าสนับสนุนจากมากไปน้อย ซึ่งลดจำนวนชิ้นข้อมูลที่ไม่จำเป็น ผู้วิจัยได้ทดลองกับฐานข้อมูลขนาดใหญ่เพื่อเปรียบเทียบประสิทธิภาพของขั้นตอนวิธีการ IndexNClosed และ NCLOSED จากผลการทดลอง พบว่า IndexNClosed ที่เสนอนั้นลดพื้นที่การค้นหาได้มากกว่า NCLOSED

คำสำคัญ : กลุ่มข้อมูลแบบปิด, กลุ่มข้อมูลที่น่าสนใจ N ลำดับ, อินเด็กซ์อาร์เรย์

1. บทนำ

ในการสืบค้นกลุ่มข้อมูลที่เกิดบ่อย มักพบ ปัญหาเกี่ยวกับการกำหนดค่าสนับสนุนขั้นต่ำ ทั้งนี้เพราะในกรณีที่ผู้ใช้ไม่มีความรู้ในการกำหนดค่าสนับสนุนขั้นต่ำ หากกำหนดค่าสนับสนุนขั้นต่ำที่มีค่าสูงเกินไป อาจไม่พบกลุ่มข้อมูลใดๆ เลย หรือพบกลุ่มข้อมูลน้อยมาก หากกำหนดค่าสนับสนุนขั้นต่ำที่มีค่าต่ำเกินไป อาจพบกลุ่มข้อมูลจำนวนมากเกินความจำเป็น

เพื่อแก้ปัญหาดังกล่าวข้างต้นได้มีการเสนอแนวคิดในการสืบค้นกลุ่มข้อมูลที่เกิดบ่อยที่มีความยาว k สูงสุด โดยผู้ใช้ไม่จำเป็นต้องกำหนดค่าสนับสนุนขั้นต่ำ เพียงแต่ระบุจำนวนผลลัพธ์ที่ต้องการ N ลำดับ งานวิจัยที่ได้นำเสนอ เช่น [1] ได้เสนอ 2 ขั้นตอนวิธี ได้แก่ Itemset-Loop และ Itemset-iLoop เพื่อสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับ [2] ได้เสนอ 3 ขั้นตอนวิธีการ ได้แก่ LOOPBACK, BOLB และ BOMO เพื่อสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับโดยใช้โครงสร้าง FP-tree ซึ่งขั้นตอนวิธีทั้งสามมีประสิทธิภาพด้านเวลาสูงกว่าขั้นตอนวิธีการ Itemset-Loop เทคนิคของ COFI-tree ได้มีการนำมาประยุกต์ใช้เพื่อสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับใน [3] โดยไม่ต้องกำหนดค่าสนับสนุนขั้นต่ำ ขั้นตอนวิธีดังกล่าวมีประสิทธิภาพด้านเวลาเมื่อ k มีค่าน้อย [4] ได้เสนอ 2 ขั้นตอนวิธี ได้แก่ NFOLD-growth และ LOOPBACK- NFOLD-growth สำหรับการสืบค้นกลุ่มข้อมูลที่น่าสนใจ N ลำดับ โดยใช้โครงสร้างข้อมูล SOTrieIT (Support-Ordered Trie Itemset) เพื่อค้นหากลุ่มข้อมูลที่เกิดบ่อยที่มีความยาว k =1 และกลุ่มข้อมูลที่เกิดบ่อยที่มีความยาว k =2ได้อย่างรวดเร็ว ซึ่งขั้นตอนวิธีดังกล่าวมีประสิทธิภาพในด้านเวลาดีกว่าขั้นตอนวิธี Itemset-Loop และ BOMO ขั้นตอนวิธี NCLOSED [5] เป็นขั้นตอนวิธีสำหรับสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับ โดยใช้เทคนิคการค้นหาคำตอบดีที่สุด (Best-First Search Strategy) เพื่อลดจำนวนกลุ่มข้อมูลที่ซ้ำซ้อน อย่างไรก็ตาม ขั้นตอนวิธีดังกล่าวไม่ได้จัดเตรียมกลุ่มข้อมูลเริ่มต้นสำหรับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับ ทำให้ต้องใช้เนื้อที่มากในการค้นหา ซึ่งเนื้อที่สามารถลดได้หากมีการจัดเตรียมข้อมูลก่อนการทำการสืบค้นข้อมูล

เพื่อลดเนื้อที่ที่ใช้ในการค้นหาดังกล่าว งานวิจัยนี้จึงเสนอขั้นตอนวิธี IndexNClosed เพื่อสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับ โดยนำอินเด็กซ์อาร์เรย์ (Index Array) มาประยุกต์ใช้ เพื่อหาชิ้นข้อมูลที่ปรากฏอยู่ด้วยกันเสมอ โดยชิ้นข้อมูลที่เกิดขึ้นพร้อมกันบ่อยครั้งและต่างก็ใช้ค่าสนับสนุนเดียวกัน ชิ้นข้อมูลเหล่านั้นจะสามารถรวมเข้าด้วยกันได้ ก่อนจะถูกนำเข้าสู่กระบวนการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับ

เนื้อหาในบทความแบ่งออกเป็นส่วนต่างๆ ดังนี้ ส่วนที่ 2 กล่าวถึงบทนิยามพื้นฐาน ส่วนที่ 3 เป็นขั้นตอนวิธี IndexNClosed ส่วนที่ 4 การวิเคราะห์ผลการทดลอง และในส่วนสุดท้ายเป็นบทสรุปของงานวิจัย

2. นิยามพื้นฐาน

กำหนดให้ $D = \{t_1, t_2, \ldots, t_m\}$ เป็นเซตของลำดับรายการเปลี่ยนแปลง (Transaction id) ทั้งหมดในฐานข้อมูล และ $I = \{i_1, i_2, \ldots, i_n\}$ เป็นเซตของชิ้นข้อมูลทั้งหมดที่ปรากฏอยู่ในฐานข้อมูล เซต $T = \{t_1, t_2, \ldots, t_h\}$, T เป็นสับเซตที่ไม่ใช่เซตว่างของ D โดยที่ $h \le m$. เซต $X = \{i_1, i_2, \ldots, i_k\}$ เป็นสับเซตที่ไม่ใช่เซตว่างของ I โดยที่ $h \le n$ นิยามฟังก์ชัน f และ g ได้ดังนี้

$$f(T) = \{i \in I \mid \forall t \in T,\ i \in t\}$$

$$g(X) = \{t \in D \mid \forall i \in X, i \in t\}$$

ฟังก์ชัน f ให้กลุ่มข้อมูลที่ใหญ่ที่สุดที่อยู่ไว้ใน T และ ฟังก์ชัน g ให้เซตของลำดับรายการเปลี่ยนแปลงซึ่งบรรจุกลุ่มข้อมูล X

นิยามที่ 1 กำหนดให้ กลุ่มข้อมูล X เป็นกลุ่มข้อมูลแบบปิด (closed itemset) เรียก $\varsigma(X)$ ว่า โคลสเซอร์ (closure) ของ X และเรียก ς ว่า Galois Operator หรือ Closure Operator ถ้า $\varsigma(X) = f(g(X)) = fog(X) = X$

ตัวอย่างที่ 1 จากตัวอย่างฐานข้อมูลดังแสดงในตารางที่ 1 พิจารณา *BCF เป็นกลุ่มข้อมูลแบบปิด* ได้ดังนี้

$g(BCF) = g(B) \cap g(C) \cap g(F) = 1345 \cap 1345 \cap 1345 = 1345$

$f(1345) = f(1) \cap f(3) \cap f(4) \cap f(5) = A\ B\ C\ D\ E\ F \cap B\ C\ E\ F \cap B\ C\ D\ E\ F \cap B\ C\ F = BCF$

จากนิยามที่ 1 $\varsigma(BCF) = f(g(BCF)) = fog(BCF) = BCF$

ดังนั้น กล่าวได้ว่า BCF เป็นกลุ่มข้อมูลแบบปิด

ตารางที่ 1 ตัวอย่างฐานข้อมูล

Tid	Items
1	A B C D E F
2	A D E
3	B C E F
4	B C D E F
5	B C F

นิยามที่ 2 คลาสสมมูล (Equivalence Class) คือ เซตกลุ่มข้อมูลโคลสเซอร์เดียวกัน

นิยามที่ 3 เจเนเรเตอร์ (Generator) คือ กลุ่มข้อมูลที่มีความยาวน้อยที่สุดของคลาสสมมูล

ตัวอย่างที่ 2 กลุ่มข้อมูล D, BD, ED และ BED อยู่ในคลาสสมมูลเดียวกันเพราะมีโคลสเซอร์เดียวกัน และ กลุ่มข้อมูลจำนวนน้อยสุด ก็คือ D ฉะนั้น เจเนเรเตอร์ของคลาสนี้ คือ D

นิยามที่ 4 ขนาดหรือความยาวของกลุ่มข้อมูล X คือ จำนวนชิ้นข้อมูลใน X เขียนแทนได้ด้วย $|X|$

นิยามที่ 5 ค่าสนับสนุนของกลุ่มข้อมูล X คือ จำนวนลำดับรายการเปลี่ยนแปลงที่บรรจุ X เขียนแทนได้ด้วย $supp(X)$ หรือ $|g(X)|$

นิยามที่ 6 กลุ่มข้อมูล X คือ กลุ่มข้อมูลที่เกิดบ่อย ถ้าค่าสนับสนุนของ X มีค่าไม่น้อยกว่าค่าสนับสนุนขั้นต่ำ

นิยามที่ 7 กลุ่มข้อมูลแบบปิด k (k-closed Itemset) คือ กลุ่มข้อมูลแบบปิดที่มีความยาว k

นิยามที่ 8 กลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับแรก คือ การรวมกันของกลุ่มข้อมูบแบบปิด k สำหรับ $1 \leq k \leq k_{max}$ โดยที่ k_{max} คือ ความยาวสูงสุดของกลุ่มข้อมูลที่ต้องการ

3. ขั้นตอนวิธี IndexNClosed

ขั้นตอนวิธี IndexNClosed ถูกพัฒนาขึ้นเพื่อลดพื้นที่ในการค้นหา สำหรับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับ เนื่องจากขั้นตอนวิธี NCLOSED [5] ไม่ได้จัดเตรียมกลุ่มข้อมูลเริ่มต้นสำหรับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับ ซึ่งอาจจำเป็นต้องใช้พื้นที่ในการค้นหา ดังนั้น เพื่อแก้ปัญหาดังกล่าว เราจึงเสนอขั้นตอนวิธี IndexNClosed สำหรับการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับ โดยนำอินเด็กซ์อาร์เรย์จากขั้นตอนวิธี Index-CloseMiner [6] มาประยุกต์ใช้ โดยขั้นตอนวิธี IndexNClosed ที่เสนอนี้ประกอบด้วย 2 ขั้นตอนหลัก คือ 1) สร้าง Index Array และ 2) จัดเก็บ Index-Array ใน คิว (Queue) เพื่อใช้เป็นกลุ่มข้อมูลเริ่มต้นในการสืบค้นกลุ่มข้อมูลแบบปิดที่น่าสนใจ N ลำดับ

3.1 การสร้าง Index Array และ IndexNClosed

นิยามที่ 9 อินเด็กซ์อาร์เรย์ คือ อาร์เรย์ขนาด $m1$ เมื่อ $m1$ คือ จำนวนของกลุ่มข้อมูลที่เกิดบ่อยที่มีความยาว k=1 แต่ละสมาชิกของอาร์เรย์ปรากฏในรูปของคู่อันดับ (item, subsume) เมื่อ item คือชิ้นข้อมูล และ $subsume(item) = \{ j \in I \mid j \neq item \wedge g(item) \subseteq g(j) \}$ สำหรับแต่ละสมาชิกของ index array เราเรียก $subsume(item)$ ว่า *subsume index*

นิยามที่ 10 $subsume(item)$ เป็น กลุ่มข้อมูล ถ้า ลำดับรายการเปลี่ยนแปลงของชิ้นข้อมูล (Tid) เป็นสับเซตของลำดับรายการเปลี่ยนแปลงของ j เมื่อ $j \in subsume(item)$

ตัวอย่างที่ 3 จากตารางที่ 1 หา *subsume index* ได้ดังนี้ *subsume index* ของ A คือ DE , *subsume index* ของ B คือ CF , *subsume index* ของ C คือ BF , *subsume index* ของ D คือ E , *subsume index* ของ E คือ $\varnothing$ และ *subsume index* ของ F คือ BC เห็นได้ว่า *subsume index* ของ B C และ F มีชิ้นข้อมูลที่เกิดขึ้นเหมือนกันและมีค่าสนับสนุนเดียวกัน จึงสามารถรวมเข้าด้วยกันได้

รหัสเทียมของขั้นตอน IndexNClosed แสดงดังภาพที่ 1

Algorithm: IndexNCLOSED (D, N, k_{max})
Input: A database D, the upper bound of the length k_{max} and the desired number of k-itemsets N.
Output: N-most interesting closed itemsets NCI.
Method:

1. scan database D and find set of transaction ids in each item
2. Call **IndexClosed** subroutine find index array to construct initial generator
3. sort distinct items in descending order of their supports
4. insert the distinct items in queue q
5. $S[1] = S[2] = \ldots = S[k_{max}] = 0$
6. $C[1] = C[2] = \ldots = C[k_{max}] = 0$
7. N-mine(q)

ภาพที่ 1 รหัสเทียมขั้นตอน IndexNClosed

ฐานข้อมูล *D* ถูกอ่านครั้งแรกในทันที และเรียกซับรูทีนอินเด็กซ์โคลส (IndexClosed Subroutine) เพื่อหาอินเด็กซ์อาร์เรย์แต่ละชิ้นข้อมูล (บรรทัดที่ 30) จากนั้นชิ้นข้อมูลถูกเรียงเป็นอันดับที่แน่นอนด้วยโครงสร้างบิทแมป (Bitmap) (บรรทัดที่ 32) นำชิ้นข้อมูลที่ถูกเรียงอันดับรายการเปลี่ยนแปลง แล้วนำมาคำนวณหาอินเด็กซ์อาร์เรย์โดยพิจารณาจากลำดับรายการเปลี่ยนแปลงที่เกิดของชิ้นข้อมูล แล้วนำไปพิจารณาในแต่ละชิ้นทั้งหมดว่ามีลำดับรายการเปลี่ยนแปลงใดที่เกิดเหมือนกันกับชิ้นข้อมูลที่เป็น อินเด็กซ์ก็จะเป็นซับซัมอินเด็กซ์ของชิ้นข้อมูลนั้นและทำการเก็บไว้เป็นกลุ่มข้อมูลเริ่มต้นในคิว(Queue) (บรรทัด33-36) หลังจากผ่านขั้นตอนในขั้นตอนวิธี 1 (ภาพที่2) แล้วให้ส่งค่ากลุ่มข้อมูลเริ่มต้นที่เก็บไว้ในคิวไปยังขั้นตอนวิธี 2 (ภาพที่ 3) นำข้อมูลเริ่มต้นจากคิวที่สร้างเก็บไว้ในขั้นตอนที่ 1 มาทำการสืบค้นกลุ่มข้อมูลแบบปิด N ลำดับแรก ตรวจสอบคิวว่าเป็นคิวว่างหรือไม่ ถ้าว่างให้หยุดสืบค้น (บรรทัดที่ 8) แต่ถ้าไม่ว่าง ให้ดึงกลุ่มข้อมูลออกมาเช็คว่ามีการสร้างกลุ่มข้อมูลนี้ซ้ำ (Duplicate) หรือไม่ โดยเรียกไปที่ซับรูทีน Generator_check(*X*) (ดังภาพที่ 4) เมื่อเช็คแล้วไม่ใช่กลุ่มข้อมูลที่ซ้ำ จะนำกลุ่มข้อมูลไปสร้างโคลสเซอร์โดยเรียกไปที่ซับรูทีน Closure_calculate(*X*) (ดังภาพที่ 5) ทำขั้นตอนจนได้กลุ่มข้อมูลแบบปิด N ลำดับแรกตามที่เรากำหนดไว้ใน N ที่เราต้องการ

ขั้นตอนวิธี 1

```
Subroutine: IndexClosed
Method:
30.  for each element index[j] of index array do
31.      index[j].item = a_j
32.  Represent the database D with bitmap
33.  for each element index[j] of index array do
34.      index[j].subsume=∅
35.      Candidate = ∩_{t∈g(index[j].item)} t;
36.  Store the item corresponding to t_k^th position in
candidate to index[j].subsume (excluding index[j].subsume)
,if the value of the t_k^th bit in candidate is set
37.      for each item i in index[j].subsume do
38.          if supp(index[j].item)==supp(i) then
39.              Delete index[k] with index[k].item== i ;
40.  End for
```

ภาพที่ 2 IndexClosed Subroutine

ขั้นตอนวิธี 2

```
Subroutine: N-mine (NCI, q)
Input: set of N-most interesting closed itemsets NCI,queue q
Output: set of N-most interesting closed itemsets NCI.
Method:

7.  while(1)
8.      if queue is empty then break
9.      X = de-queue()
10.     if Generator_check(X)
11.         Y = Closure_calculate(X)
12.         if supp(Y) ≥ S[Y.len] then
13.             NCI = NCI ∪ Y
14.         C[Y.len] =C[Y.len]+1
15.         if C[Y.len] ==N then
16.             S[Y.len] = supp(Y)
17.         if Y.len < k_max then
18.             P = {Y∪i_1, Y∪i_2,..., Y∪i_n} ; i ∈ pre-items(Y)
    //Extend Y
19.             while(j<n)
20.                 en-queue(Y∪i_j)
```

ภาพที่ 3 N-mine subroutine

```
Subroutine: Generator_check(X)

Input: An itemset X

Output: If X is a non-duplicate generator, 1 is returned.
Otherwise, 0 is returned.

Method:

21. while(supp(X)=supp(l_j)) ; l_j ∈ post-items(X)
22.   if(g(X) ⊆ g(l_j))
23.     return 0
24. return 1
```

ภาพที่ 4 Generator checking subroutine

```
Subroutine: Closure_calculate(X)
Input: A itemset X
Output: A Closure of X, Y
Method:
25. Y = X
26. while(supp(X) = supp(j_j)) ; j_j ∈ pre-items(X)
27.   if(g(X) ⊆ g(j_j))
28.     Y = Y ∪ j_j
29. return Y
```

ภาพที่ 5 Closure calculation subroutine

4. วิเคราะห์ผลการทดลอง

ในการศึกษาประสิทธิภาพขั้นตอนวิธี IndexNClosed ได้ทดลองโดยเปรียบเทียบกับขั้นตอนวิธี NCLOSED และทดสอบกับชุดข้อมูลที่มีชื่นข้อมูลต่างๆ กัน โดยกำหนด k_{max} มีค่าคงที่เท่ากับ 4 และกำหนดค่า N = 10, 20, 50, 100, 1000, 5000, 10000, 50000, 100000 และ 1000000

4.1 ลักษณะของฐานข้อมูลที่นำมาทดลอง

ผู้วิจัยได้ทำการทดลองกับชุดข้อมูลหนาแน่น (Dense datasets) 3 ชุด ได้แก่ Chess, Mushroom และ Connect โดยชุดข้อมูล Chess และ Connect เป็นแบบหนาแน่นมาก ขณะที่ชุดข้อมูล Mushroom เป็นแบบหนาแน่นน้อย ลักษณะของชุดข้อมูลดังแสดงในตารางที่ 2 และฐานข้อมูลที่ใช้ในการทดลองสามารถดาวน์โหลดได้จาก http://fimi.cs.helsinki.fi/

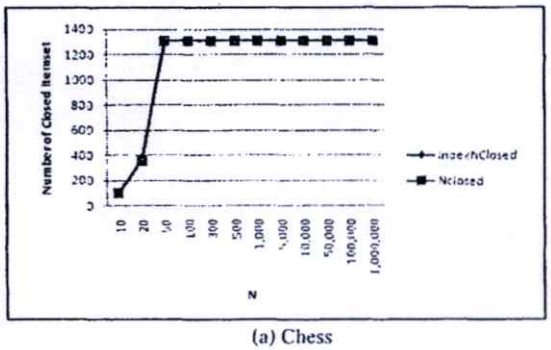
ตารางที่ 2 ลักษณะของชุดข้อมูลที่ใช้ในการทดลอง

Datasets	#Items	#Record	Avg.Length
Chess	75	3,196	37
Mushroom	119	8,124	23
Connect	129	65,557	43

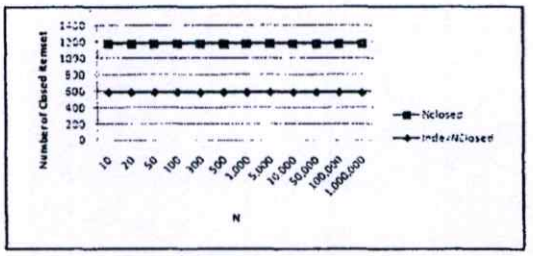
ขั้นตอนวิธี IndexNClosed และขั้นตอนวิธี NCLOSED ได้ทดลองบนเครื่องคอมพิวเตอร์ส่วนบุคคล ซึ่งมีหน่วยการประมวลผลเป็น Core 2 Duo 1.6 GHz หน่วยความจำหลัก 2 GB ฮาร์ดดิสก์ขนาด 120 GB บนระบบปฏิบัติการ Windows XP SP3 ขั้นตอนวิธีการ IndexNClosed เขียนด้วยภาษา C++ โดยใช้ g++ Compiler

4.2 ผลการทดลอง

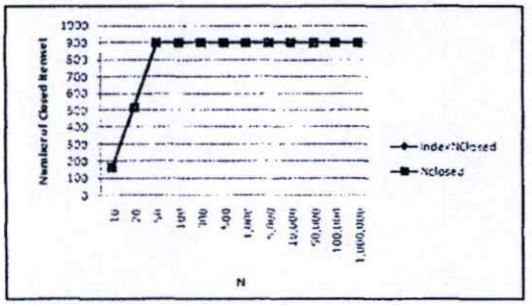
ในการศึกษาประสิทธิภาพขั้นตอนวิธี IndexNClosed จะพิจารณาจากจำนวนกลุ่มข้อมูลแบบปิดที่ถูกสร้างขึ้น และ เวลาที่ใช้ผลการทดลอง ดังแสดงในภาพที่ 7 และ ภาพที่ 8



(a) Chess



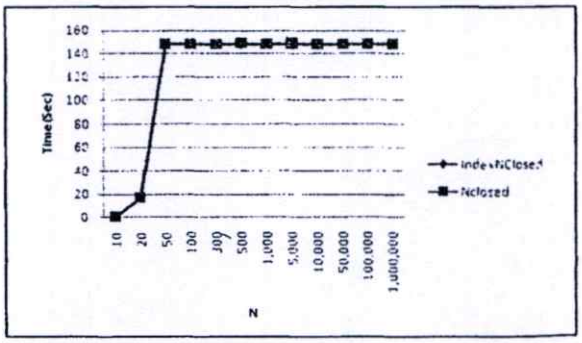
(b) Mushroom



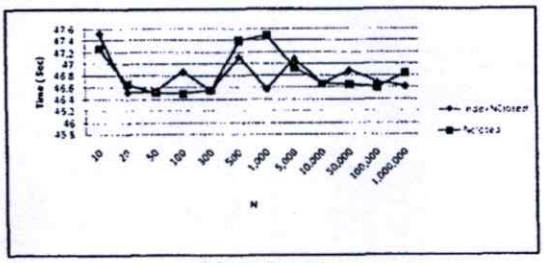
(c) Connect

ภาพที่ 7 เปรียบเทียบจำนวนกลุ่มข้อมูลแบบปิดที่สร้างขึ้น

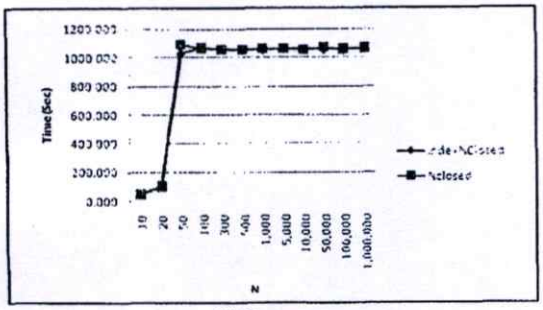
ภาพที่ 7 แสดงการเปรียบเทียบจำนวนกลุ่มข้อมูลแบบปิดที่ถูกสร้างขึ้นโดยขั้นตอนวิธี IndexNClosed และ NCLOSED กับชุดข้อมูล a) Chess b) Mushroom และ c) Connect ตามลำดับ พบว่าจำนวนกลุ่มข้อมูลแบบปิดที่ถูกสร้างขึ้นโดยทั้งสองขั้นตอนวิธีไม่แตกต่างกัน สำหรับชุดข้อมูล Chess และ Connect แต่ขั้นตอนวิธี IndexNClosed ที่เสนอนั้นมีประสิทธิภาพเหนือกว่า NCLOSED ในชุดข้อมูล Mushroom ซึ่งเป็นข้อมูลที่หนาแน่นน้อย



(a) Chess



(b) Mushroom



(c) Connect

ภาพที่ 8 เปรียบเทียบเวลาที่ใช้ในการสืบค้น

ภาพที่ 8 แสดงการเปรียบเทียบเวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิด N ลำดับแรก โดยขั้นตอนวิธี IndexNClosed และ NCLOSED กับชุดข้อมูล a) Chess b) Mushroom และ c) Connect ตามลำดับ จากผลการทดลอง พบว่า เมื่อพิจารณาโดยภาพรวม เวลาที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิด N ลำดับแรก โดยขั้นตอนวิธี IndexNClosed และ ขั้นตอนวิธี NCLOSED ใกล้เคียงกัน

5. สรุป

งานวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาประสิทธิภาพของขั้นตอนวิธี IndexNClosed ที่ใช้ในการสืบค้นกลุ่มข้อมูลแบบปิด N ลำดับ โดยนำอินเด็กซ์อาร์เรย์มาประยุกต์ใช้เพื่อหาชิ้นข้อมูลที่ปรากฏอยู่ด้วยกัน โดยชิ้นข้อมูลที่เกิดขึ้นพร้อมกันและใช้ค่าสนับสนุนเดียวกันจะถูกรวมเข้าด้วยกันและเก็บเป็นกลุ่มข้อมูลเริ่มต้น จึงสามารถลดจำนวนกลุ่มข้อมูลเริ่มต้นที่ใช้ในการสืบค้นได้ นอกจากนี้เราได้ศึกษาประสิทธิภาพของขั้นตอนวิธี IndexNClosed โดยเปรียบเทียบกับ NCLOSED จากผลการทดลองแสดงให้เห็นว่าขั้นตอนวิธี IndexNClosed ใช้เวลาในการสืบค้นกลุ่มข้อมูลแบบปิด N ลำดับ ใกล้เคียงกันกับขั้นตอนวิธี NCLOSED นอกจากนี้ ขั้นตอนวิธี IndexNClosed ที่เสนอนั้นสามารถลดพื้นที่ในการสืบค้นได้ เมื่อข้อมูลมีความหนาแน่นน้อย ดังนั้น ขั้นตอนวิธี IndexNClosed นี้จึงน่าจะลดเนื้อที่ได้มากเมื่อใช้กับ Sparse Datasets ซึ่งเป็นงานวิจัยที่จะทำในอนาคตต่อไป

เอกสารอ้างอิง

[1] A.W. –C. Fu, R. W. – W. Kwong, and J. Tang, "Mining N-most interesting Itemsets", In Proc. of 12th Symposium on Methodologies for Intelligent System(ISMIS), pp. 59-67, London, UK, 2000.

[2] Y-L. Cheng and A. Fu, "An FP-tree approach for mining N-most interesting itemsets", In Proc. of the SPIE Conference on Data Mining, pp. 460-471, 2002. Proc. of the SPIE Conference on Data Mining , pp.460-471 , 2002

[3] S. Ngan, T. Lam, R.C. Wong, and A. W. Fu, "Mining N-most interesting itemsets without support threshold by COFI-tree", *Journal of Business Intelligence and Data Mining*, Vol. 1, No. 1. 2005.

[4] M. U. Arshad, M. N. Ayyaz, "Mining N-most Interesting Itemsets Using Support-Ordered Tries",pp. 592-599, Dubai, 2006.

[5] Songram, P. Boonjing, V. "N-Most Interesting Closed Itemset Mining". Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on, Vol.1, page(s): 619-624, , Busan,2008

[6] Wei Song,Bingru Yang and Zhangyan Xu " Index-CloseMiner: an improved algorithm for Mining frequent closed itemset", Intelligent Data Analysis. Volume 12 , Issue 4 Pages 321-338 ,2008

ประวัติผู้เขียน

ชื่อ – สกุล นางสาวพิพิธพร โพนตุแสง

วัน เดือน ปีเกิด 11 มิถุนายน 2521

ที่อยู่ 86/164 รอยัลทาวเวอร์ 3 ซอยอินทามระ 25 แขวงสามเสนใน
เขตพญาไท จังหวัดกรุงเทพมหานคร 10400

ประวัติการศึกษา

2543 จบการศึกษาปริญญาวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์
จากภาควิชาวิทยาการคอมพิวเตอร์ คณะวิศวกรรมศาสตร์และ
วิทยาการคอมพิวเตอร์ มหาวิทยาลัยมหาสารคาม

2549- ปัจจุบัน กำลังศึกษาวิทยาศาสตรมหาบัณฑิต
สาขาวิทยาการคอมพิวเตอร์
ภาควิชาคณิตศาสตร์และวิทยาศาสตร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประวัติการทำงาน

ปัจจุบัน ตำแหน่งโปรแกรมเมอร์
หน่วยงานเทคโนโลยีสารสนเทศ ฝ่ายพัฒนาระบบสารสนเทศ
ธนาคารออมสิน สำนักงานใหญ่