การทำดัชนีที่แม่นยำสำหรับการค้นข้อมูลอนุกรมเวลาตามความคล้าย ภายใต้ไทม์วอร์ปปิงด้วยการเข้าถึงข้อมูลแบบลำดับโดยใช้ดัชนี

นายพงศกร เรืองรองหิรัญญา

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2551

EXACT INDEXING FOR SIMILARITY SEARCH ON TIME SERIES DATA UNDER TIME WARPING USING INDEXED SEQUENTIAL ACCESS

Mr. Pongsakorn Ruengronghirunya

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Computer Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2008

หัวข้อวิทยานิพนธ์	การทำดัชนีที่แม่นยำสำหรับการค้นข้อมูลอนุกรมเวลาตามความคล้ายภายใต้ไทม์วอร์ปปิงด้วยการเข้าถึงข้อมูลแบบลำดับโดยใช้ดัชนี
โดย	นายพงศกร เรืองรองหิรัญญา
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.โชติรัตน์ รัตนามหัทธนะ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์
(ผู้ช่วยศาสตราจารย์ ดร.โชติรัตน์ รัตนามหัทธนะ)

..... กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.กฤษณะ ไวยมัย)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ)

พงศกร เรืองรองหิรัญญา : การทำดัชนีที่แม่นยำสำหรับการค้นข้อมูลอนุกรมเวลาตามความคล้ายภายใต้ไทม์วอร์ปปิงด้วยการเข้าถึงข้อมูลแบบลำดับโดยใช้ดัชนี. (EXACT INDEXING FOR SIMILARITY SEARCH ON TIME SERIES DATA UNDER TIME WARPING USING INDEXED SEQUENTIAL ACCESS) อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.โชติรัตน์ รัตนามหัทธนะ, 103 หน้า.

การค้นคืนข้อมูลอนุกรมเวลาตามความคล้ายเป็นสิ่งที่สำคัญสำหรับงานประยุกต์มากมาย เช่น การค้นคืนข้อมูลมัลติมีเดียและการจำแนกข้อมูลเหล่านั้น สำหรับงานประยุกต์ดังกล่าวนั้นไดนามิกไทม์วอร์ปปิงจัดเป็นมาตรวัดที่ใช้สำหรับการค้นคืนข้อมูลอนุกรมเวลาที่มีความแม่นยำมากที่สุดวิธีหนึ่ง อย่างไรก็ตามการใช้ไดนามิกไทม์วอร์ปปิงสำหรับการค้นคืนข้อมูลนั้นต้องใช้เวลาในการคำนวณสูง ซึ่งปัญหาดังกล่าวเป็นประเด็นที่มีนักวิจัยมากมายสนใจที่จะพัฒนาให้มีความรวดเร็วมากยิ่งขึ้น จนเมื่อไม่นานมานี้ได้มีงานวิจัยที่ได้เสนอวิธีการแก้ปัญหาดังกล่าวด้วยฟังก์ชันขอบเขตล่างซึ่งสามารถลดทอนการคำนวณไดนามิกไทม์วอร์ปปิงและสามารถเพิ่มความเร็วในการค้นคืนข้อมูลได้อย่างมีประสิทธิภาพ แต่สำหรับในงานด้านฐานข้อมูล ประเด็นสำคัญนั้นไม่ได้อยู่เพียงแค่การลดเวลาในการคำนวณสำหรับการค้นคืนข้อมูลเท่านั้น แต่อยู่ที่จะทำอย่างไรเพื่อที่จะลดการเข้าถึงข้อมูลหรืออินพุต / เอาต์พุตให้ได้มากที่สุด ซึ่งวิธีที่ใช้กันทั่วไปก็คือวิธีการทำดัชนี แต่จนถึงปัจจุบันยังไม่มีวิธีการทำดัชนีข้อมูลอนุกรมเวลาที่มีประสิทธิภาพเนื่องจากต้องเผชิญกับปัญหาสำคัญอยู่สองประการ ประการแรกเกิดจากการที่ข้อมูลอนุกรมเวลานั้นมีจำนวนมิติที่สูงมาก และในประการที่สองเนื่องจากการคำนวณไดนามิกไทม์วอร์ปปิงนั้นจุดข้อมูลหนึ่งสามารถมีความสัมพันธ์กับจุดข้อมูลที่อยู่คนละมิติได้ ด้วยเหตุนี้วิธีการทำดัชนีข้อมูลอนุกรมเวลาที่มีอยู่ในปัจจุบันยังคงไม่สามารถลดทอนการเข้าถึงข้อมูลให้ได้มากเพียงพอกับเวลาที่ต้องเสียเพิ่มเติมจากการเข้าถึงข้อมูลแบบสุ่ม กล่าวคือการกราดตรวจข้อมูลตามลำดับนั้นสามารถค้นคืนข้อมูลได้เร็วกว่าการทำดัชนีทุกวิธีที่มีอยู่ในปัจจุบัน ดังนั้นงานวิจัยนี้จึงนำเสนอวิธีการทำดัชนีข้อมูลอนุกรมเวลาที่มีประสิทธิภาพด้วยการนำวิธีการจับกลุ่มข้อมูลมาประยุกต์ใช้ โดยใช้แนวคิดในการจัดลำดับการเข้าถึงข้อมูลในแต่ละกลุ่มข้อมูลโดยเรียงตามค่าระยะทางขอบเขตล่างที่ได้นำเสนอ ซึ่งเป็นผลทำให้การค้นคืนข้อมูลสามารถเข้าถึงข้อมูลที่มีความคล้ายได้ในช่วงต้นของกระบวนการค้น นอกจากนี้ยังสามารถลดทอนการเข้ากลุ่มข้อมูลหลายกลุ่มได้ด้วยค่าระยะทางขอบเขตล่างดังกล่าว ในการทดลองนั้นวิธีการทำดัชนีที่ได้นำเสนอสามารถค้นคืนข้อมูลได้เร็วกว่าวิธีกราดตรวจ โดยลดการเข้าถึงข้อมูลได้หลายสิบเท่า

ภาควิชา .....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต .....

สาขาวิชา.....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่ออาจารย์ที่ปรึกษา .....

ปีการศึกษา .....2551.....

5170390721 : MAJOR COMPUTER ENGINEERING

KEY WORD : TIME SERIES RETRIEVAL / DYNAMIC TIME WARPING / INDEXING

PONGSAKORN RUENGRONGHIRUNYA : EXACT INDEXING FOR SIMILARITY SEARCH ON TIME SERIES DATA UNDER TIME WARPING USING INDEXED SEQUENTIAL ACCESS. THESIS ADVISOR : ASST. PROF. CHOTIRAT RATANAMAHATANA PH.D., 103 pp.

As time series has become one of the most prevalent types of data in this digital age, similarity search occurs to be crucial for these applications, including multimedia data retrieval and classification. Dynamic time warping (DTW) is one of the most accurate distance measure exploited in the search. However, it is known to suffer from such a high computational cost, raising great amount of interest in trying to speed it up. Recently, this obstacle has been alleviated using efficient existing lower bounding functions, e.g., FTW and LB_Keogh which can greatly speed up the distance calculation for the similarity search. However, in database field, the crux of the matter is how to minimize the data access or I/O cost. The most typical approach is through indexing. Unlike other types of data, time series indexing is almost impractical due to their explosively high number of dimensions. Moreover, unlike typical Euclidean distance measure, in DTW distance measure, a data point can be in relation to data points from other dimensions. As a result, time series indexing is beyond challenging since sequential scan unexpectedly outperforms all existing indexing methods. This research proposes an efficient time series indexing structure. With clustering technique on data preprocessing, the order of the data access on each data cluster is sorted by the proposed lower bounding distance which indicates the similarity between query and data cluster. Therefore, the similar candidates can be accessed early in the search process. Moreover, many data clusters can be pruned by the proposed lower bounding distance. In the experiments, the proposed indexing method outperforms the sequential scan by over an order of magnitude in terms of data access reduction.

Department ....Computer Engineering. Student's Signature .....

Field of Study..Computer Engineering. Advisor's Signature .....

Academic Year .....2008.....

กิตติกรรมประกาศ

กว่าจะได้มาเป็นวิทยานิพนธ์เล่มสมบูรณ์ฉบับนี้ ทางผู้จัดทำจำต้องฝ่าฟันอุปสรรคนานัปการ ทั้งนี้ทางผู้จัดทำก็มิอาจผ่านพ้นอุปสรรคเหล่านั้นมาได้อย่างลุล่วงหากขาดซึ่งผู้ที่คอยอุปถัมป์สนับสนุนและคอยช่วยเหลือมาโดยตลอดช่วงเวลาที่ผ่านมา ดังนั้นผู้จัดทำจึงขอกล่าวขอบคุณทุกแรงสนับสนุนที่ทำให้ผู้จัดทำได้ประสบความสำเร็จลุล่วงไปด้วยดี

ก่อนอื่นก็ขอกราบขอบพระคุณอาจารย์ที่ปรึกษาวิทยานิพนธ์ฉบับนี้ ผู้ช่วยศาสตราจารย์ ดร.โชติรัตน์ รัตนามหัทธนะ ที่คอยให้คำปรึกษา แนะนำ และคอยผลักดันให้เกิดผลงานที่ดี รวมถึงยังเป็นแรงบันดาลใจให้ผู้จัดทำสามารถฝ่าฟันอุปสรรคต่าง ๆ ไปได้ นอกจากนี้ยังเป็นผู้ที่คอยตักเตือนและแก้ไขในส่วนบกพร่องที่เกิดจากตัวผู้จัดทำเอง ซึ่งคำแนะนำเหล่านี้นอกจากจะเกิดประโยชน์ทำให้วิทยานิพนธ์เล่มนี้เสร็จสมบูรณ์ลุล่วงได้ด้วยดีแล้ว ยังเป็นประโยชน์แก่ตัวผู้จัดทำในด้านการพัฒนากระบวนการการคิดและทักษะต่าง ๆ ที่จำเป็นสำหรับการทำวิจัย สิ่งเหล่านี้ก่อให้เกิดแนวคิดและประสบการณ์ใหม่ ๆ สำหรับตัวผู้จัดทำ ซึ่งจะเป็นประโยชน์อย่างยิ่งในภายภาคหน้า อีกทั้งยังคอยสนับสนุนในด้านทุนการทำวิจัยซึ่งเป็นสิ่งที่สำคัญที่สุดสิ่งหนึ่งที่ขาดไปไม่ได้

ขอกราบขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ทุกท่าน ประกอบด้วย ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ และรองศาสตราจารย์ ดร.กฤษณะ ไวยมัย ผู้ที่ให้แนวคิดและข้อเสนอแนะที่เป็นประโยชน์อย่างยิ่งกับวิทยานิพนธ์ฉบับนี้ โดยเป็นผู้จุดประกายแนวคิดและช่องทางในการพัฒนางานวิจัยนี้ให้ดียิ่งขึ้น รวมถึงเพิ่มพูนความสมบูรณ์ในตัวงานวิจัยสำหรับวิทยานิพนธ์ฉบับนี้ให้มากยิ่งขึ้นไปอีก

ขอขอบคุณพี่ ๆ และเพื่อน ๆ ร่วมงานซึ่งทำงานอยู่ในห้องปฏิบัติการด้วยกันมาโดยตลอด สำหรับคำแนะนำและแนวคิดต่าง ๆ ที่ก่อให้เกิดประโยชน์กับวิทยานิพนธ์ฉบับนี้ รวมถึงยังเป็นผู้แบ่งปันประสบการณ์ต่าง ๆ ที่มีค่าสำหรับการทำวิจัยให้สำเร็จลุล่วงไปได้ด้วยดี นอกจากนี้ยังร่วมด้วยช่วยกันแก้ปัญหาเล็ก ๆ น้อย ๆ ที่เกิดขึ้นทั้งหลายทั้งปวงตลอดมา

และที่สำคัญที่สุด ก็ขอกราบขอบพระคุณคุณพ่อคุณแม่ที่คอยดูแลเอาใจใส่ และเป็นกำลังใจให้ผู้จัดทำมาโดยตลอด

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญภาพ ..... ญ
สารบัญตาราง ..... ฒ
บทที่ 1 บทนำ 1
1.1 ที่มาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ของการวิจัย 3
1.3 ขอบเขตของการวิจัย 3
1.4 ประโยชน์ที่ได้รับ 4
1.5 วิธีดำเนินการวิจัย 4
1.6 ผลงานตีพิมพ์จากงานวิจัย 4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 6
2.1 ทฤษฎีที่เกี่ยวข้อง 6
2.1.1 มาตรวัดระยะทางแบบยุคลิด 6
2.1.2 มาตรวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง 6
2.1.3 การจับกลุ่ม 9
2.1.4 การวัดความสมเหตุสมผลของการจับกลุ่ม 11
2.1.5 การเข้าถึงข้อมูลแบบลำดับโดยใช้ดัชนี 16
2.2 งานวิจัยที่เกี่ยวข้อง 16
2.2.1 ฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิง 16
2.2.2 การปรับขนาดเอกรูป 21
2.2.3 การจัดทำดัชนีสำหรับการค้นคืนข้อมูลอนุกรมเวลา 22
บทที่ 3 การค้นคืนข้อมูลอนุกรมเวลาด้วยการเข้าถึงข้อมูลแบบลำดับโดยใช้ดัชนี 25
3.1 การสกัดลักษณะสำคัญของข้อมูล 25
3.2 การแปลงข้อมูลอนุกรมเวลาให้เป็นบรรทัดฐาน 27
3.3 การจัดเตรียมข้อมูลสำหรับการจัดทำดัชนีการเข้าถึงข้อมูลอนุกรมเวลา 27
3.3.1 ขั้นการจับกลุ่มข้อมูล 28
3.3.2 การกำหนดขอบเขตของกลุ่มข้อมูลสำหรับการทำดัชนีการค้นคืนข้อมูล 29

3.4 การค้นคืนข้อมูลอนุกรมเวลาด้วยการเข้าถึงข้อมูลแบบลำดับโดยใช้ดัชนี.....32
3.4.1 ฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูล 33
3.4.2 การจัดลำดับการค้นข้อมูล 37
3.4.3 การค้นตามลำดับภายในแต่ละกลุ่มข้อมูล 38
3.5 วิธีการจับกลุ่มที่ใช้ในการจัดเตรียมข้อมูลสำหรับการทำดัชนีการค้นคืนข้อมูล.....39
3.5.1 การจับกลุ่มแบบเคมีนภายใต้การลดทอนการคำนวณ 40
3.5.2 การจับกลุ่มแบบแทรก 44
3.6 การกำหนดค่าพารามิเตอร์สำหรับการจับกลุ่ม 46
3.7 การรองรับการเปลี่ยนแปลงของชุดข้อมูล 48
3.7.1 การจับกลุ่มใหม่ภายหลังการเพิ่มข้อมูลเข้าไปในชุดข้อมูล 48
3.7.2 การจับกลุ่มใหม่ภายหลังการลบข้อมูลออกจากชุดข้อมูล 50
บทที่ 4 การทดลองและวิเคราะห์ผล.....51
4.1 รูปแบบของข้อมูลทั้งหมดที่ใช้ในการทดลอง.....51
4.1.1 ข้อมูลจริงที่ใช้กันทั่วไปในงานวิจัยด้านการทำเหมืองข้อมูลอนุกรมเวลา.....51
4.1.2 ข้อมูลที่ได้จากการสังเคราะห์ขึ้น 53
4.2 การทดสอบประสิทธิภาพการลดทอนการคำนวณในการจับกลุ่มข้อมูลแบบเคมีน.....56
4.3 การทดสอบเพื่อเปรียบเทียบประสิทธิภาพของการลดทอนข้อมูลด้วยการทำดัชนีข้อมูลที่ใช้ฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่ได้นำเสนอทั้งสองวิธี 59
4.4 การทดสอบประสิทธิภาพในการกำหนดค่าพารามิเตอร์ในการจับกลุ่มข้อมูลสำหรับการทำดัชนีการค้นคืนข้อมูลด้วยวิธีการวัดความสมเหตุสมผลของการจับกลุ่ม.....62
4.5 การทดสอบประสิทธิภาพการลดทอนข้อมูลจากการค้นคืนข้อมูลด้วยดัชนีบนชุดข้อมูลในรูปแบบต่าง ๆ 65
4.6 การทดสอบประสิทธิภาพการลดทอนข้อมูลจากการค้นคืนข้อมูลด้วยดัชนีสำหรับการกำหนดขนาดของเงื่อนไขบังคับโดยรวมที่แตกต่างกัน.....66
4.7 การทดสอบประสิทธิภาพการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอ.....67
4.8 การทดสอบเพื่อเปรียบเทียบประสิทธิภาพของผลการจับกลุ่มระหว่างการจับกลุ่มแบบเคมีนกับการจับกลุ่มแบบแทรก 71
4.9 การทดสอบประสิทธิภาพการค้นคืนข้อมูลบนชุดข้อมูลขนาดใหญ่ 73

4.9.1 การทดสอบเพื่อกำหนดค่าพารามิเตอร์สำหรับการจับกลุ่มแบบแทรก 73
4.9.2 การทดสอบเพื่อเปรียบเทียบประสิทธิภาพในการค้นคืนข้อมูลระหว่าง
วิธีการค้นคืนด้วยดัชนีที่ได้นำเสนอกับวิธีที่ใช้กันอยู่ในปัจจุบัน.....74
บทที่ 5 สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ 79
5.1 สรุปผลการวิจัย.....79
5.2 ข้อเสนอแนะ 80
รายการอ้างอิง 81
ภาคผนวก 84
ภาคผนวก ก.....85
ภาคผนวก ข 94
ประวัติผู้เขียนวิทยานิพนธ์ 103

สารบัญภาพ

หน้า

รูปที่ 2.1 ตัวอย่างการเปรียบเทียบวิธีการคำนวณระยะทางของข้อมูลอนุกรมเวลา....7
รูปที่ 2.2 ตัวอย่างกรณีของการจับคู่เปรียบเทียบที่ไม่เหมาะสม....8
รูปที่ 2.3 ตัวอย่างเงื่อนไขบังคับโดยรวมสำหรับไดนามิกไทม์วอร์ปปิง....9
รูปที่ 2.4 ตัวอย่างเดนโดรแกรมจากการจับกลุ่มแบบลำดับขั้น....10
รูปที่ 2.5 ตัวอย่างการคำนวณฟังก์ชันขอบเขตล่างของระยะทางแบบไดนามิกไทม์วอร์ปปิง LB_Keogh....17
รูปที่ 2.6 ตัวอย่างการแบ่งข้อมูลอนุกรมเวลาออกเป็นช่วงย่อย ๆ สำหรับคำนวณฟังก์ชันขอบเขตล่าง....19
รูปที่ 2.7 ตัวอย่างการประมาณฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิงด้วยวิธี FTW....20
รูปที่ 2.8 ตัวอย่างการทำดัชนีด้วยโครงสร้างแบบต้นไม้สำหรับข้อมูลอนุกรมเวลาที่ถูกลดขนาดลงด้วยวิธีพีเอเอ....23
รูปที่ 3.1 ตัวอย่างการสกัดลักษณะสำคัญจากข้อมูลภาพถ่ายลายมือ....26
รูปที่ 3.2 ตัวอย่างการสกัดลักษณะสำคัญจากภาพถ่ายใบไม้....26
รูปที่ 3.3 ภาพรวมของขั้นตอนการเตรียมการก่อนการค้น....28
รูปที่ 3.4 ภาพรวมของขั้นตอนการระบุตำแหน่งการค้น....28
รูปที่ 3.5 ตัวอย่างของกลุ่มข้อมูลที่ผ่านการจับกลุ่มแบบเคมีน....30
รูปที่ 3.6 กล่องขอบเขตที่ครอบคลุมกลุ่มข้อมูลตัวอย่างในรูปที่ 3.5....31
รูปที่ 3.7 ขอบเขตของกลุ่มข้อมูลภายใต้การกำหนดเงื่อนไขบังคับโดยรวมที่สร้างจากกล่องขอบเขตในรูปที่ 3.6....32
รูปที่ 3.8 ตัวอย่างการคำนวณฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่มีการตรึงขนาดของเงื่อนไขบังคับโดยรวม....34
รูปที่ 3.9 ตัวอย่างการแปลงข้อมูลสอบถามด้วยวิธี LB_Keogh....35
รูปที่ 3.10 ตัวอย่างการคำนวณฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่สามารถกำหนดขนาดของเงื่อนไขบังคับโดยรวมได้ในแต่ละข้อมูลสอบถาม....36
รูปที่ 3.11 รหัสเทียมสำหรับการค้นคืนข้อมูลแบบลำดับโดยใช้ดัชนี....37
รูปที่ 3.12 รหัสเทียมสำหรับการค้นคืนข้อมูลจากในกลุ่มข้อมูลตามลำดับ....39
รูปที่ 3.13 รหัสเทียมสำหรับวิธีการจับกลุ่มข้อมูลอนุกรมเวลาแบบเคมีน....41
รูปที่ 3.14 รหัสเทียมสำหรับฟังก์ชันการลดการคำนวณระยะทางแบบยุคลิด....43

รูปที่ 3.15 รหัสเทียมสำหรับฟังก์ชันการจับกลุ่มแบบแทรก.....45
รูปที่ 3.16 ตัวอย่างแนวโน้มของเวลาที่ใช้ในการค้นคืนจากแต่ละกลุ่มข้อมูลที่มีการปรับเปลี่ยนจำนวนกลุ่มในการจับกลุ่ม.....48
รูปที่ 3.17 รหัสเทียมสำหรับฟังก์ชันการเพิ่มข้อมูลเข้าไปในชุดข้อมูล.....49
รูปที่ 4.1 ตัวอย่างข้อมูลที่ได้จากการสังเคราะห์ขึ้นแบบ RWI.....54
รูปที่ 4.2 ตัวอย่างข้อมูลที่ได้จากการสังเคราะห์ขึ้นแบบ RWII.....54
รูปที่ 4.3 ตัวอย่างข้อมูลซีบีเอฟทั้ง 3 คลาส.....55
รูปที่ 4.4 ผลการทดลองการเปรียบเทียบจำนวนครั้งในการวัดระยะทางระหว่างแต่ละจุดข้อมูลในการวัดระยะทางแบบยุคลิดบนการจับกลุ่มแบบเคมีนระหว่างวิธีที่ได้นำเสนอกับวิธีดั้งเดิม โดยทดสอบบนการจับกลุ่มชุดข้อมูล Mixed ด้วยการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน.....57
รูปที่ 4.5 ผลการทดลองการเปรียบเทียบจำนวนครั้งในการวัดระยะทางระหว่างแต่ละจุดข้อมูลในการวัดระยะทางแบบยุคลิดบนการจับกลุ่มแบบเคมีนระหว่างวิธีที่ได้นำเสนอกับวิธีดั้งเดิม โดยทดสอบบนการจับกลุ่มชุดข้อมูล RWI ทั้งหมด 10,000 อนุกรม แต่ละอนุกรมมีความยาว 128 จุดข้อมูล ด้วยการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน.....57
รูปที่ 4.6 ผลการทดลองการวัดจำนวนรอบการวนซ้ำบนการจับกลุ่มแบบเคมีนจากการเพิ่มข้อมูล RWI ทั้งหมด 20 อนุกรมเข้าไปในชุดข้อมูล RWI ทั้งหมด 100,000 อนุกรมที่ผ่านการจับกลุ่มมาก่อนแล้ว โดยแต่ละอนุกรมมีความยาว 128 จุดข้อมูล ด้วยการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน.....58
รูปที่ 4.7 ผลการทดลองการเปรียบเทียบจำนวนครั้งในการวัดระยะทางระหว่างแต่ละจุดข้อมูลในการวัดระยะทางแบบยุคลิดบนการจับกลุ่มแบบเคมีนระหว่างวิธีที่ได้นำเสนอกับวิธีดั้งเดิม โดยทดสอบบนการเพิ่มข้อมูล RWI ทั้งหมด 20 อนุกรมเข้าไปในชุดข้อมูล RWI ทั้งหมด 100,000 อนุกรมที่ผ่านการจับกลุ่มมาก่อนแล้ว โดยแต่ละอนุกรมมีความยาว 128 จุดข้อมูล ด้วยการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน.....59
รูปที่ 4.8 ผลการทดลองการเปรียบเทียบกำลังการลดทอนระหว่างการค้นคืนข้อมูลด้วยดัชนีที่ใช้ฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่มีการตรึงขนาดของเงื่อนไขบังคับโดยรวมกับฟังก์ชันที่สามารถปรับขนาดของเงื่อนไขบังคับโดยรวมได้.....60

รูปที่ 4.9 การเปรียบเทียบการคำนวณฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางแบบไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลระหว่างแบบที่มีการตรึงขนาดของเงื่อนไขบังคับโดยรวมกับแบบที่สามารถกำหนดเงื่อนไขบังคับโดยรวมได้ 61
รูปที่ 4.10 ค่าดัชนีดันที่วัดได้จากการจับกลุ่มแบบเคมีนบนชุดข้อมูล Wafer โดยทำการปรับค่าจำนวนกลุ่มที่แตกต่างกัน.....62
รูปที่ 4.11 ค่าดัชนีเดวีส์บูลดินที่วัดได้จากการจับกลุ่มแบบเคมีนบนชุดข้อมูล Wafer โดยทำการปรับค่าจำนวนกลุ่มที่แตกต่างกัน.....63
รูปที่ 4.12 ค่าดัชนีความสมเหตุสมผลของส่วนเบี่ยงเบนมาตรฐานที่วัดได้จากการจับกลุ่มแบบเคมีนบนชุดข้อมูล Wafer โดยทำการปรับค่าจำนวนกลุ่มที่แตกต่างกัน.....63
รูปที่ 4.13 ค่าดัชนีความสมเหตุสมผลของเอสดีบีดับเบิลยูที่วัดได้จากการจับกลุ่มแบบเคมีนบนชุดข้อมูล Wafer โดยทำการปรับค่าจำนวนกลุ่มที่แตกต่างกัน.....64
รูปที่ 4.14 ผลการทดสอบการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอบนชุดข้อมูล Wafer โดยทำการปรับค่าจำนวนกลุ่มที่แตกต่างกัน 64
รูปที่ 4.15 ผลการทดลองการเปรียบเทียบกำลังการลดทอนข้อมูลจากการเข้าถึงข้อมูลด้วยดัชนีที่ได้นำเสนอ โดยทำการเปรียบระหว่างการเข้าถึงชุดข้อมูล RWI RWII และ CBF ข้อมูลทั้งหมดมีความยาวเท่ากันเท่ากับ 128 จุดข้อมูล แต่ละชุดข้อมูลประกอบไปด้วยทั้งหมด 100,000 อนุกรม โดยการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน 66
รูปที่ 4.16 ผลการทดลองการเปรียบเทียบผลกระทบของขนาดของเงื่อนไขบังคับโดยรวมต่อประสิทธิภาพในการลดทอนข้อมูลจากการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอ โดยทดสอบบนการค้นคืนข้อมูล RWII ทั้งหมด 100 อนุกรมจากในชุดข้อมูล RWII ทั้งหมด 100,000 อนุกรมที่ผ่านการจับกลุ่มมาก่อนแล้ว โดยแต่ละอนุกรมมีความยาว 2,048 จุดข้อมูล ด้วยการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน.....67
รูปที่ 4.17 ผลการทดลองการเลือกจำนวนกลุ่มในการจับกลุ่มของชุดข้อมูล RWII ด้วยวิธีการใช้ชุดตรวจสอบความสมเหตุสมผล โดยใช้บนชุดตรวจสอบความสมเหตุสมผลที่ประกอบด้วย 20 อนุกรม บนชุดข้อมูลที่ประกอบด้วยข้อมูลจำนวนทั้งหมด 100,000 อนุกรมแต่ละอนุกรมมีความยาว 128 จุดข้อมูล ซึ่งมีการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน 68

หน้า

รูปที่ 4.18 ผลการทดลองการเลือกจำนวนกลุ่มในการจับกลุ่มของชุดข้อมูล RW II ด้วยวิธีการใช้ชุดตรวจสอบความสมเหตุสมผล โดยใช้บนชุดตรวจสอบความสมเหตุสมผลที่ประกอบด้วย 20 อนุกรม บนชุดข้อมูลที่ประกอบด้วยข้อมูลจำนวนทั้งหมด 100,000 อนุกรม แต่ละอนุกรมมีความยาว 1,024 จุดข้อมูล ซึ่งมีการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน 69
รูปที่ 4.19 ผลการทดลองการเลือกจำนวนกลุ่มในการจับกลุ่มของชุดข้อมูล RW II ด้วยวิธีการใช้ชุดตรวจสอบความสมเหตุสมผล โดยใช้บนชุดตรวจสอบความสมเหตุสมผลที่ประกอบด้วย 20 อนุกรม บนชุดข้อมูลที่ประกอบด้วยข้อมูลจำนวนทั้งหมด 100,000 อนุกรมแต่ละอนุกรมมีความยาว 2,048 จุดข้อมูล ซึ่งมีการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน 70
รูปที่ 4.20 ผลการทดลองเพื่อเปรียบเทียบเวลาในการค้นคืนข้อมูลบนชุดข้อมูล RW II ระหว่างวิธีการค้นทั้ง 3 วิธี ได้แก่ วิธีการค้นตามลำดับโดยใช้ฟังก์ชันขอบเขตล่างของค่าระยะทางไดนามิกไทม์วอร์ปปิงแบบ FTW และแบบ LB_Keogh กับวิธีที่ได้นำเสนอ โดยใช้ชุดข้อมูลสอบถามทั้งหมด 100 อนุกรม เพื่อทำการค้นบนชุดข้อมูลที่ประกอบด้วยข้อมูลจำนวนทั้งหมด 100,000 อนุกรมโดยแต่ละชุดข้อมูลมีความยาวของข้อมูล 128 1,024 และ 2,048 จุดข้อมูล 70
รูปที่ 4.21 ผลการทดลองเปรียบเทียบประสิทธิภาพของการจับกลุ่มระหว่างการจับกลุ่มแบบเคมีนกับการจับกลุ่มแบบแทรก โดยทำการทดสอบการค้นคืนข้อมูลสอบถามทั้งหมด 100 อนุกรม บนชุดข้อมูล RW II ขนาด 262,144 อนุกรม แต่ละอนุกรมมีความยาว 2,048 จุดข้อมูล ชุดข้อมูลดังกล่าวผ่านการจับกลุ่มแบบเคมีนและแบบแทรก ซึ่งมีการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน แล้วทำการวัดประสิทธิภาพในรูปแบบของกำลังการลดทอนจากการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอบนผลลัพธ์ของการจับกลุ่มแต่ละรูปแบบ.....72
รูปที่ 4.22 เวลาที่ใช้ในการจับกลุ่มแบบแทรกบนชุดข้อมูล RW II ขนาด 262,144 อนุกรม แต่ละอนุกรมมีความยาว 2,048 จุดข้อมูล ด้วยค่าพารามิเตอร์ในการจับกลุ่มแบบแทรก PageSize ที่แตกต่างกัน.....72
รูปที่ 4.23 จำนวนกลุ่มข้อมูลที่ได้จากการจับกลุ่มแบบแทรกด้วยการกำหนดค่าพารามิเตอร์ในการจับกลุ่มที่แตกต่างกันซึ่งก็คือขนาดที่ใหญ่ที่สุดสำหรับแต่ละกลุ่มข้อมูลบนชุดข้อมูล RW II ขนาด 524,288 อนุกรม แต่ละอนุกรมมีความยาว 2,048 จุดข้อมูล.....74

รูปที่ 4.24 ผลการทดลองการจับเวลาที่ใช้ในการค้นคืนข้อมูลจากชุดข้อมูล RWII ขนาด 524,288 อนุกรม แต่ละอนุกรมมีความยาว 2,048 จุด โดยทำการปรับค่าค่าพารามิเตอร์ในการจับกลุ่มที่แตกต่างกันซึ่งก็คือขนาดที่ใหญ่ที่สุดสำหรับแต่ละกลุ่มข้อมูล 74

รูปที่ 4.25 ผลการทดลองการเปรียบเทียบความเร็วในการค้นคืนข้อมูลระหว่างวิธี FTW LB_Keogh และวิธีที่ได้นำเสนอ โดยทำการทดสอบบนชุดข้อมูล RWII ที่มีขนาดแตกต่างกัน แต่ละข้อมูลจากทุกชุดข้อมูลมีความยาวเท่ากันเท่ากับ 2,048 จุด 75

รูปที่ 4.26 ผลการทดลองการเปรียบเทียบความเร็วในการค้นคืนข้อมูลระหว่างวิธี FTW LB_Keogh และวิธีที่ได้นำเสนอ โดยทำการทดสอบบนชุดข้อมูล RWII ที่มีขนาดเท่ากันเท่ากับ 524,288 อนุกรม ข้อมูลจากแต่ละชุดข้อมูลมีความยาวที่แตกต่างกัน 76

รูปที่ 4.27 ผลการทดลองการเปรียบเทียบจำนวนหน้าข้อมูลที่ทำการเข้าถึงในการค้นคืนข้อมูลระหว่างวิธี FTW LB_Keogh และวิธีที่ได้นำเสนอ โดยทำการทดสอบบนชุดข้อมูล RWII ที่มีขนาดแตกต่างกัน แต่ละข้อมูลจากทุกชุดข้อมูลมีความยาวเท่ากันเท่ากับ 2,048 จุด และทำการเฉลี่ยเวลาที่ใช้ในการค้นคืนข้อมูลสอบถามทั้งหมด 100 อนุกรม 77

รูปที่ 4.28 ผลการทดลองเพื่อเปรียบเทียบเวลาที่ใช้ในการเตรียมข้อมูลระหว่างวิธี FTW กับวิธีการทำดัชนีที่ได้นำเสนอ 78

สารบัญตาราง

หน้า

ตารางที่ 2.1 คำอธิบายสัญกรณ์ทั้งหมดที่ใช้ในฟังก์ชันการคำนวณค่าดัชนีความสมเหตุสมผลของการจับกลุ่ม 12
ตารางที่ 4.1 คุณลักษณะของชุดข้อมูลจริงที่ใช้ในการทดลอง 52

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ในปัจจุบัน ระบบสารสนเทศต่าง ๆ ต้องจัดการกับข้อมูลที่มีขนาดเพิ่มมากขึ้นเรื่อย ๆ ซึ่งในบางส่วนของระบบเหล่านี้จะต้องจัดเก็บข้อมูลทั้งในรูปแบบของลำดับของข้อมูลเชิงกาลเวลา (Temporal Data) หรือข้อมูลอนุกรมเวลา (Time Series Data) ยกตัวอย่างเช่น ในระบบสารสนเทศที่ต้องทำการวิเคราะห์ข้อมูลที่อยู่ในรูปแบบของข้อมูลอนุกรมเวลาในบางระบบจำเป็นต้องมีการค้นคืนข้อมูลตามความคล้ายคลึง เช่น ระบบการค้นหาเพลงโดยการร้องทำนอง [1, 2] เนื่องด้วยในระบบเหล่านี้จำต้องมีการติดต่อกับผู้ใช้งานโดยตรง ดังนั้นประสิทธิภาพเชิงเวลาสำหรับการค้นคืนข้อมูลจึงเป็นสิ่งจำเป็นเพื่อให้ระบบสามารถตอบสนองต่อผู้ใช้ได้อย่างทันท่วงที นอกจากประเด็นในด้านการเพิ่มความเร็วในการค้นข้อมูลซึ่งเป็นสิ่งที่นักวิจัยทั้งหลายให้ความสนใจแล้ว นักวิจัยยังต้องเล็งเห็นถึงความสำคัญของความแม่นยำในการค้นคืนข้อมูลเป็นหลัก ดังนั้นการเพิ่มความเร็วในการค้นข้อมูลจึงไม่ควรมีผลกระทบซึ่งทำให้ความแม่นยำในการค้นคืนข้อมูลลดลง เนื่องจากผู้ใช้ระบบมักมุ่งเน้นความสำคัญไปที่ความถูกต้องในการทำงานของระบบมากกว่าความเร็ว

สำหรับข้อมูลอนุกรมเวลา การสืบค้นข้อมูลตามความคล้ายคลึงโดยส่วนมากมักจะใช้การวัดระยะทางเป็นตัวเปรียบเทียบความคล้ายคลึงกันของข้อมูล เนื่องจากเป็นวิธีที่แม่นยำและมีประสิทธิภาพสูง ตัวอย่างวิธีวัดระยะทางที่เป็นที่นิยมใช้กันอย่างแพร่หลายในการเปรียบเทียบข้อมูลอนุกรมเวลา ได้แก่ วิธีวัดระยะทางแบบยุคลิด (Euclidean Distance Metric) เนื่องจากวิธีวัดระยะทางแบบยุคลิดเป็นวิธีวัดระยะทางที่สามารถคำนวณได้อย่างรวดเร็วและไม่ซับซ้อน อย่างไรก็ตามเป็นที่น่าเสียดายที่วิธีวัดระยะทางแบบยุคลิดไม่สามารถรองรับความแปรผันของข้อมูลเชิงเวลาได้อย่างสมบูรณ์ เนื่องจากค่าที่ได้จากการใช้วิธีวัดระยะทางดังกล่าวไม่เหมาะสมและอาจส่งผลให้การค้นข้อมูลทำการคืนผลที่ผิดพลาด นอกเหนือจากนั้นวิธีวัดระยะทางแบบยุคลิดยังมีข้อจำกัดที่ไม่สามารถใช้เปรียบเทียบกับข้อมูลที่มีความยาวแตกต่างกันได้ อีกทางเลือกหนึ่งสำหรับวิธีวัดระยะทางคือวิธีไดนามิกไทม์วอร์ปปิง (Dynamic Time Warping) [3] ซึ่งเป็นวิธีที่มีความยืดหยุ่นและแม่นยำในการเปรียบเทียบความคล้ายคลึงกันของข้อมูลอนุกรมเวลามากกว่าวิธีแบบยุคลิด เนื่องจากวิธีวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงอนุญาตให้มีการจับคู่ของจุดข้อมูลบนข้อมูลอนุกรมเวลาซึ่งเกิดในลำดับของเวลาที่แตกต่างกันเพื่อคำนวณหาค่าระยะทาง อย่างไรก็ตามวิธีไดนามิกไทม์วอร์ปปิงมีข้อด้อยที่เป็นปัญหาหลักนั่นก็คือการคำนวณระยะทางนั้นต้องใช้เวลาในการคำนวณสูง หรือถ้าเปรียบเทียบในเชิงของขีดจำกัดเชิงสัญกรณ์ (Asymptotic Limit) ในด้านเวลาจะอยู่ในระดับของฟังก์ชันพหุนามกำลัง

สอง $O(n^2)$ เมื่อ n คือความยาวของข้อมูลอนุกรมเวลา ด้วยเหตุนี้จึงเป็นส่วนกระตุ้นให้มีงานวิจัยจำนวนมากมุ่งพัฒนาในด้านการเพิ่มความเร็วในการค้นข้อมูลอนุกรมเวลาตามความคล้ายโดยใช้วิธีวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงเป็นตัวกำหนดความคล้าย

วิธีดั้งเดิมวิธีหนึ่งในการเพิ่มความเร็วในการค้นข้อมูลอนุกรมเวลาตามความคล้ายโดยใช้วิธีวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงเป็นตัวกำหนดความคล้ายคือการจัดทำดัชนีสำหรับข้อมูลอนุกรมเวลา (Time Series Indexing) ซึ่งวิธีดังกล่าวสามารถลดทอนข้อมูลบางส่วนออกก่อนการคำนวณหาระยะทางจริงด้วยวิธีไดนามิกไทม์วอร์ปปิง โดยใช้เวลาในการคำนวณเพียงเล็กน้อยเมื่อเทียบกับเวลาที่ใช้ในการคำนวณไดนามิกไทม์วอร์ปปิง ปัจจุบันมีหลายงานวิจัย [1, 4-7] ที่ได้นำเสนอวิธีการจัดทำดัชนีสำหรับข้อมูลอนุกรมเวลาในวิธีที่แตกต่างกัน ยกตัวอย่างเช่น เมื่อไม่นานมานี้มีงานวิจัยหนึ่งได้จัดทำดัชนีสำหรับข้อมูลอนุกรมเวลาโดยการคำนวณหาค่าขอบเขตล่างซึ่งเป็นค่าประมาณระยะทางของข้อมูลด้วยวิธีการประมาณค่าข้อมูลอนุกรมเวลาด้วยการลดความยาวของข้อมูลลง [1, 8] ซึ่งจะสามารถลดเวลาในการคำนวณระยะทางได้เนื่องจากวิธีดังกล่าวสามารถลดทอนข้อมูลบางส่วนได้จากการแทนที่การคำนวณระยะทางแบบไดนามิกไทม์วอร์ปปิงด้วยการคำนวณค่าขอบเขตล่างซึ่งใช้เวลาในการคำนวณน้อยกว่า ทำให้การค้นหาข้อมูลอนุกรมเวลาโดยใช้การคำนวณระยะทางแบบไดนามิกไทม์วอร์ปปิงสามารถนำไปใช้ได้จริงในทางปฏิบัติ แต่อย่างไรก็ตามการคำนวณค่าขอบเขตล่างจากการลดความยาวของข้อมูลอนุกรมเวลา เป็นเพียงแค่การประมาณค่าระยะทางเท่านั้น ดังนั้นค่าขอบเขตล่างที่ได้ยังคงแตกต่างจากค่าระยะทางจริงอยู่ค่อนข้างมาก ซึ่งส่งผลให้ประสิทธิภาพในการตัดทอนข้อมูลลดลง

อีกตัวอย่างหนึ่งในการจัดทำดัชนีสำหรับข้อมูลอนุกรมเวลาได้แก่ การจัดทำโครงสร้างดัชนีที่สามารถระบุตำแหน่งของข้อมูลสำหรับทำการค้น ซึ่งจะเป็นตัวบ่งชี้ได้ว่าข้อมูลที่ต้องการค้นนั้นอยู่ในส่วนใดของฐานข้อมูล รูปแบบโครงสร้างที่พบเห็นได้ในงานวิจัยด้านการค้นข้อมูลประเภทข้อมูลอนุกรมเวลา [1, 9, 10] นั้นมักจะอยู่ในรูปแบบของต้นไม้ เช่น โครงสร้างดัชนีในรูปแบบของ R-Tree [11] และ R*-Tree [12] เป็นต้น ซึ่งการค้นคืนข้อมูลผ่านโครงสร้างต้นไม้นั้นจะส่งผลต่อเวลาที่ต้องเสียไปในการอ่านข้อมูล เนื่องจากต้องทำการแบ่งข้อมูลออกเป็นส่วนย่อย ๆ ตามจำนวนโหนดใบของต้นไม้ ซึ่งจะทำให้ต้องเปิดอ่านแฟ้มข้อมูลหลายครั้ง นอกจากนี้ยังต้องทำการจัดเก็บโครงสร้างดัชนีทั้งหมดไว้ในหน่วยความจำหลัก จึงทำให้วิธีนี้ไม่รองรับกับการขยายตัวของขนาดของฐานข้อมูลอย่างมหาศาลได้ ดังนั้นจึงมีงานวิจัยใหม่ ๆ ซึ่งพยายามที่จะลดเวลาที่ต้องเสียไปจากการอ่านข้อมูลด้วยแนวคิดในการค้นข้อมูลโดยลำดับ (Sequential Search) งานวิจัยหนึ่งได้นำเสนอแนวคิดในการจัดทำดัชนีแบบใหม่เรียกว่า FTW (**F**ast Search Method for Dynamic **T**ime **W**arping) [7] ซึ่งได้ใช้แนวคิดในการค้นข้อมูลในรูปแบบใหม่โดยมีพื้นฐานอยู่บนการค้นโดยลำดับ ซึ่งจะทำการเก็บข้อมูลในฐานข้อมูลทั้งหมดไว้ในแฟ้มข้อมูลเดียวกัน แต่อย่างไรก็ตามแนวคิดนี้ยังมีข้อด้อยหลัก ๆ อยู่คือความจริงที่ว่าการค้น

ข้อมูลโดยลำดับต้องทำการเข้าถึงทุกข้อมูลในฐานข้อมูลเพื่อทำการค้นโดยเข้าถึงข้อมูลตามลำดับ (Sequential Access) ซึ่งเห็นได้ชัดว่าในกรณีที่ข้อมูลที่ระบบกำลังค้นหาอยู่นั้นถูกเก็บอยู่ในส่วนท้าย ๆ ของฐานข้อมูล จะทำให้การคำนวณค่าระยะทางขอบบนไม่สามารถตัดทอนข้อมูลได้อย่างมีประสิทธิภาพมากนัก ดังนั้นการคำนวณหาระยะทางจะล่าช้ามากกว่าจะค้นถึงข้อมูลที่ต้องการซึ่งอยู่ช่วงท้าย ๆ ของชุดข้อมูล

สำหรับงานวิจัยนี้ มีวัตถุประสงค์เพื่อนำเสนอวิธีการค้นข้อมูลประเภทข้อมูลอนุกรมเวลาตามความคล้ายโดยใช้วิธีวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงเป็นตัวบ่งชี้ความคล้ายกันของข้อมูลอนุกรมเวลา โดยสิ่งที่งานวิจัยนี้ได้พัฒนาขึ้นมาคือการจัดทำโครงสร้างดัชนีแบบใหม่ ซึ่งจะช่วยแก้ปัญหาที่เกิดขึ้นจากการค้นข้อมูลโดยลำดับ โครงสร้างดัชนีที่ได้นำเสนอนี้สามารถระบุตำแหน่งของข้อมูลที่จะทำการค้น โดยข้อดีของโครงสร้างดัชนีสำหรับระบุตำแหน่งเข้าถึงข้อมูลที่ได้นำเสนอคือ การเข้าถึงข้อมูลจะมีลำดับการเข้าถึงที่แน่นอนทั้งหมด เนื่องด้วยโครงสร้างดัชนีที่ได้นำเสนอได้ใช้วิธีการเข้าถึงลำดับดัชนี (Indexed Sequential Access) ซึ่งมีส่วนช่วยทำให้การทำงานในส่วนของอินพุต/เอาท์พุต (Input/output) สามารถทำการอ่านข้อมูลจากในฮาร์ดดิสก์มาเก็บไว้ในบัฟเฟอร์ล่วงหน้าเตรียมไว้สำหรับการค้นข้อมูลต่อไป โดยไม่ต้องรอให้หน่วยประมวลผลกลางทำการค้นข้อมูลตัวก่อนหน้าจนเสร็จ ยิ่งไปกว่านั้น โครงสร้างดัชนีที่ได้นำเสนอยังสามารถนำไปดัดแปลงเพื่อใช้ในการทำดัชนีเพื่อระบุตำแหน่งการค้นสำหรับวิธีการค้นข้อมูลในลักษณะของการค้นข้อมูลโดยลำดับได้

1.2 วัตถุประสงค์ของการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อนำเสนอวิธีการค้นหาข้อมูลอนุกรมเวลาตามความคล้ายคลึงที่ใช้วิธีวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง ซึ่งสามารถค้นคืนข้อมูลได้ภายในระยะเวลาอันสั้นสำหรับฐานข้อมูลที่มีขนาดใหญ่

1.3 ขอบเขตของการวิจัย

1. พัฒนาวิธีการทำดัชนีสำหรับค้นหาข้อมูลอนุกรมเวลาตามความคล้ายคลึงที่ใช้วิธีวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงที่สามารถทำงานได้อย่างถูกต้องและรวดเร็ว
2. พัฒนาวิธีการค้นหาข้อมูลโดยใช้ลักษณะการเปรียบเทียบข้อมูลแบบทั้งความยาวข้อมูล (Whole Sequence Matching)
3. พัฒนาวิธีการค้นหาข้อมูลอนุกรมเวลาจากข้อมูลที่มีความยาวเท่ากันทุกตัว
4. ทดสอบความแม่นยำและความเร็วของวิธีที่นำเสนอ ในการค้นหาข้อมูลอนุกรมเวลาโดยใช้ข้อมูลทดสอบที่มีแหล่งอ้างอิง ซึ่งความแม่นยำในการค้นหาข้อมูล

ของวิธีที่นำเสนอในงานวิจัยนี้ จะต้องไม่น้อยกว่าความแม่นยำจากการค้นข้อมูลด้วยวิธีดั้งเดิมสำหรับการวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง

1.4 ประโยชน์ที่ได้รับ

ได้วิธีการสืบค้นข้อมูลอนุกรมเวลาตามความคล้ายคลึงของข้อมูลที่ใช้วิธีวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง ที่สามารถค้นข้อมูลได้อย่างรวดเร็วและแม่นยำ อีกทั้งยังสามารถรองรับปริมาณข้อมูลที่เพิ่มขึ้นได้

1.5 วิธีดำเนินการวิจัย

1. ศึกษาค้นคว้างานวิจัยที่เกี่ยวข้องกับการค้นหาข้อมูลอนุกรมเวลาตามความคล้ายคลึงโดยใช้วิธีวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง พร้อมทั้งวิเคราะห์ข้อดี ข้อเสียของงานวิจัยที่เกี่ยวข้อง
2. ออกแบบและพัฒนาวิธีสำหรับการค้นหาข้อมูลโดยมีรูปแบบการเข้าถึงข้อมูลแบบลำดับโดยใช้ดัชนี
3. ออกแบบและพัฒนาวิธีการทำดัชนีจากฟังก์ชันขอบเขตล่างที่มีประสิทธิภาพสำหรับกลุ่มข้อมูล เพื่อที่จะลดทอนจำนวนข้อมูลที่ต้องทำการค้นจากฐานข้อมูลก่อนทำการคำนวณระยะทางโดยใช้วิธีไดนามิกไทม์วอร์ปปิง
4. ทดสอบความแม่นยำของวิธีที่นำเสนอโดยเทียบกับผลการทดสอบความแม่นยำที่เทียบเท่ากับการใช้การวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงโดยตรง
5. ออกแบบและพัฒนาวิธีการปรับค่าพารามิเตอร์ให้ได้ผลที่เหมาะสม
6. ทดสอบความเร็วของวิธีที่นำเสนอกับวิธีอื่นที่ได้เสนอในประเด็นปัญหาเดียวกัน
7. วิเคราะห์และสรุปผลการทดลอง
8. สรุป เรียบเรียง และจัดทำวิทยานิพนธ์

1.6 ผลงานตีพิมพ์จากงานวิจัย

ส่วนหนึ่งของงานวิทยานิพนธ์นี้ ได้รับการตีพิมพ์เป็นบทความทางวิชาการสองเรื่อง ดังนี้

- “Efficient Similarity Search under Fast Index Structure for Time Series Data” โดย พงศกร เรืองรองหิรัญญา วิชญ์ เนียรนาทตระกูล และโชติรัตน์ รัตนามหัทธนะ (Best Paper Award) ในงานประชุมวิชาการ “12th National Computer Science and Engineering Conference (NCSEC2008)” ซึ่งจัดขึ้น

ณ จังหวัดชลบุรี ประเทศไทย ระหว่างวันที่ 20 – 21 พฤศจิกายน 2551 ดังรายละเอียดในภาคผนวก ก

- “Speeding up Similarity Search on Large Time Series Dataset Under Time Warping Distance” โดย พงศกร เรืองรองหิรัญญา วิชญ์ เนียรนาทตระกูล และโชติรัตน์ รัตนามหัทธนะ ในงานประชุมวิชาการนานาชาติ “13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD2009)” ซึ่งจัดขึ้น ณ กรุงเทพมหานคร ประเทศไทย ระหว่างวันที่ 27 – 30 เมษายน 2552 ดังรายละเอียดในภาคผนวก ข

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

สำหรับบทที่ 2 จะนำเสนอทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับการค้นคืนข้อมูลอนุกรมเวลาตามความคล้าย โดยจะกล่าวถึงวิธีการกำหนดความคล้ายกันของข้อมูลด้วยการวัดระยะทางระหว่างข้อมูลในรูปแบบต่าง ๆ และวิธีการจับกลุ่มข้อมูลจากการวัดระยะทาง อีกทั้งยังได้กล่าวถึงงานวิจัยที่เกี่ยวข้องกับการเพิ่มความเร็วในการค้นคืนข้อมูลอนุกรมเวลา

2.1 ทฤษฎีที่เกี่ยวข้อง

สำหรับในหัวข้อนี้ จะนำเสนอทฤษฎีและความรู้พื้นฐานที่เกี่ยวข้องกับงานวิจัย ซึ่งจะกล่าวถึงวิธีการวัดระยะทางของข้อมูลอนุกรมเวลา และวิธีการจับกลุ่มข้อมูลที่ใช้ในงานวิจัยนี้ รวมถึงวิธีการปรับค่าพารามิเตอร์ในการจับกลุ่มให้เหมาะสมด้วยวิธีการวัดความสมเหตุสมผลของการจับกลุ่ม โดยมีรายละเอียดดังต่อไปนี้

2.1.1 มาตรวัดระยะทางแบบยุคลิด (Euclidean Distance Metric)

มาตรวัดระยะทางแบบยุคลิดเป็นมาตรวัดระยะทางสำหรับข้อมูลอนุกรมเวลาที่สามารถคำนวณได้ง่ายที่สุด เนื่องจากมีขีดจำกัดเชิงสัญกรณ์เชิงเวลาเพียงฟังก์ชันเชิงเส้นหรือ $O(n)$ เท่านั้น การคำนวณระยะทางนั้นทำได้โดยการแยกคำนวณระยะทางระหว่างข้อมูลอนุกรมเวลาในแต่ละมิติของข้อมูล โดยกล่าวรายละเอียดการคำนวณได้ดังต่อไปนี้

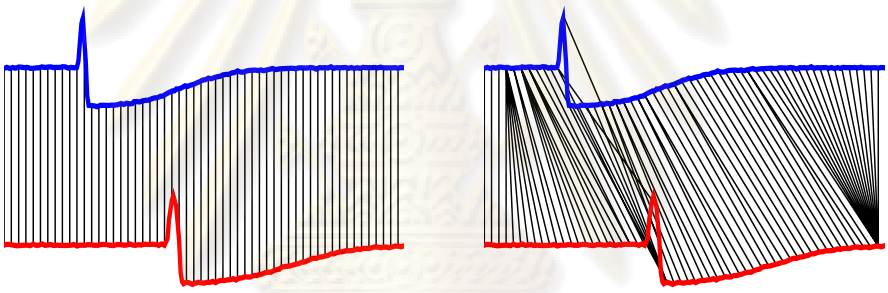
กำหนดให้มีข้อมูลอนุกรมเวลา A และ B ที่มีความยาว n โดยที่ $A = \{a_1, a_2, \ldots, a_n\}$ และ $B = \{b_1, b_2, \ldots, b_n\}$ สมการคำนวณระยะทางแบบยุคลิดดังแสดงในสมการ (2.1)

$$Euclidean(A,B) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \qquad (2.1)$$

2.1.2 มาตรวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง (Dynamic Time Warping Distance Measure)

ไดนามิกไทม์วอร์ปปิงเป็นวิธีการวัดระยะทางระหว่างข้อมูลอนุกรมเวลาซึ่งเป็นวิธีที่นิยมใช้กันในหลาย ๆ โปรแกรมประยุกต์ เนื่องจากไดนามิกไทม์วอร์ปปิงเหมาะสำหรับเปรียบเทียบข้อมูลอนุกรมเวลาที่มีความแปรผันของข้อมูลเชิงเวลา ซึ่งจะทำการเปรียบเทียบระยะทางในแต่ละคู่จุดของข้อมูลโดยเลือกคู่จุดที่เหมาะสมเพื่อนำมาเปรียบเทียบ จุดเด่นของวิธีไดนามิกไทม์วอร์ปปิงคือจะสามารถเลือกเปรียบเทียบคู่จุดข้อมูลที่ไม่ได้อยู่ในลำดับเดียวกันใน

ข้อมูลอนุกรมเวลาได้ ซึ่งจุดเด่นดังกล่าวนี้ทำให้ไดนามิกไทม์วอร์ปปิงมีความเหมาะสมกับข้อมูลอนุกรมเวลาที่มีความแปรผันเฉพาะที่เชิงเวลา (Local Variation) ดังตัวอย่างในรูปที่ 2.1 เป็นการวัดระยะทางระหว่างข้อมูลอนุกรมเวลา 2 อนุกรมที่มีความคล้ายกันมาก เพียงแต่มีความแปรผันเฉพาะที่เชิงเวลา ดังนั้นการวัดระยะทางแบบยุคลิดจึงไม่เหมาะสมที่จะนำมาเปรียบเทียบความคล้ายคลึงกันของข้อมูลอนุกรมเวลาได้ดังแสดงในรูปที่ 2.1 ก) ส่วนวิธีวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงสามารถทำการเปรียบเทียบจุดข้อมูลที่คล้ายคลึงกันแต่เกิดความแปรผันเฉพาะที่เชิงเวลาได้ดังแสดงในรูปที่ 2.1 ข) ตัวอย่างของข้อมูลที่เกิดความแปรผันเฉพาะที่เชิงเวลานั้นพบเห็นได้ทั่วไป ยกตัวอย่างเช่น ข้อมูลที่เป็นเสียงพูด ผู้พูดแต่ละคนมักจะพูดด้วยความเร็วที่ต่างกัน ดังนั้นข้อมูลอนุกรมเวลาที่เป็นข้อมูลเสียงจะมีความแปรผันในแกนของเวลาเป็นต้น ในด้านการใช้งานจริงสำหรับการค้นหาข้อมูลอนุกรมเวลาที่มีความแปรผันเฉพาะที่เชิงเวลาก็สามารถพบเห็นได้อย่างแพร่หลาย เช่น ระบบรู้จำเสียงพูด (Speech Recognition) และระบบค้นหาเพลงโดยการร้องทำนอง (Query by Humming) เป็นต้น



ก) ข)

รูปที่ 2.1 ตัวอย่างการเปรียบเทียบวิธีการคำนวณระยะทางของข้อมูลอนุกรมเวลา ก) การคำนวณระยะทางแบบยุคลิด ข) การคำนวณระยะทางแบบไดนามิกไทม์วอร์ปปิง

วิธีการคำนวณระยะทางแบบไดนามิกไทม์วอร์ปปิงนั้น จะใช้หลักการและแนวคิดของกำหนดการพลวัต (Dynamic Programming) โดยมีรายละเอียดดังต่อไปนี้

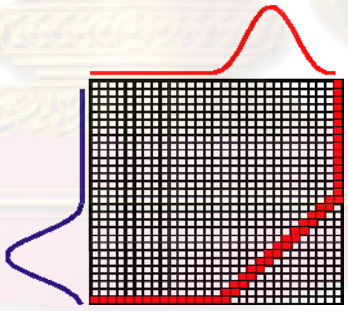
กำหนดให้มีข้อมูลอนุกรมเวลา 2 ข้อมูล ได้แก่ C และ Q ซึ่งเป็นข้อมูลที่จะนำมาคำนวณหาระยะทางแบบไดนามิกไทม์วอร์ปปิง โดยให้ C มีความยาว N ซึ่งประกอบด้วยจุดข้อมูล $c_1, c_2, c_3, \ldots, c_N$ ตามลำดับ และให้ Q มีความยาว M ซึ่งประกอบด้วยจุดข้อมูล $q_1, q_2, q_3, \ldots, q_M$ ตามลำดับ ดังนั้นจะสามารถนิยามฟังก์ชันการคำนวณระยะทางแบบไดนามิกไทม์วอร์ปปิงระหว่าง C และ Q ตามที่แสดงในสมการ (2.2) ซึ่งแสดงการคำนวณหาฟังก์ชัน $DTW(C,Q)$ ดังกล่าว

$$
\begin{aligned}
&DTW(C,Q) = \sqrt{f(N,M)} \\
&f(i,j) = d(c_i,q_j) + \min\left\{\begin{matrix} f(i-1,j-1) \\ f(i-1,j) \\ f(i,j-1) \end{matrix}\right\} \\
&f(0,0) = 0,\ f(i,0) = f(0,j) = \infty \\
&1 \le i \le N, 1 \le j \le M
\end{aligned}
\tag{2.2}
$$

จากสมการ (2.2) $d(c_i, q_j)$ แสดงฟังก์ชันการคำนวณระยะทางระหว่างจุดข้อมูล c_i กับ q_j ซึ่งการคำนวณระยะทางระหว่างจุดข้อมูลจะสามารถคำนวณได้ตามสมการ (2.3)

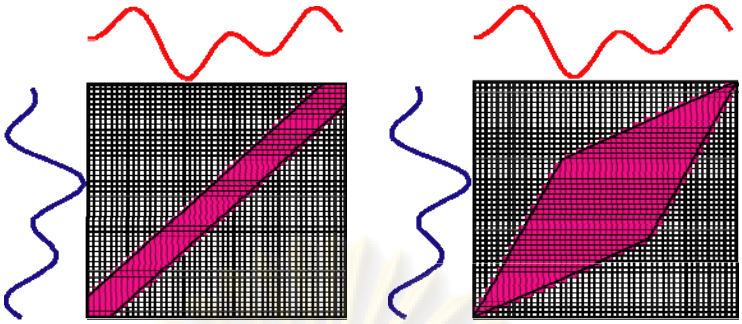
$$d(c_i,q_j) = (c_i - q_j)^2 \tag{2.3}$$

อย่างไรก็ตามในบางกรณีกำหนดการพลวัตอาจทำการจับคู่เปรียบเทียบจุดของข้อมูลอย่างไม่เหมาะสม ดังแสดงตัวอย่างในรูปที่ 2.2 โดยกำหนดการพลวัตได้ทำการจับคู่จุดที่อยู่ห่างกันมาก ซึ่งในบางสถานการณ์นั้นไม่สมควรที่จะนำมาเปรียบเทียบกัน โดยปกติแล้วข้อมูลอนุกรมเวลานั้นจะมีความแปรผันเฉพาะที่เชิงเวลาไม่มากนัก ดังนั้นจึงได้มีการกำหนดเงื่อนไขบังคับสำหรับการคำนวณกำหนดการพลวัตสำหรับไดนามิกไทม์วอร์ปปิงให้อยู่ในช่วงที่เหมาะสม



รูปที่ 2.2 ตัวอย่างกรณีของการจับคู่เปรียบเทียบที่ไม่เหมาะสม

เงื่อนไขบังคับโดยรวมสำหรับไดนามิกไทม์วอร์ปปิง (Global Constraint on Dynamic Time Warping) [13] เป็นการกำหนดขอบเขตของการคำนวณกำหนดการพลวัตสำหรับไดนามิกไทม์วอร์ปปิง โดยจะจำกัดขอบเขตในการคำนวณจุดข้อมูลมิให้ทำการเปรียบเทียบจุดข้อมูลที่เกิดขึ้นในเวลาที่แตกต่างกันมากเกินกว่าที่กำหนดไว้ในเงื่อนไขบังคับโดยรวมได้ ซึ่งการกำหนดเงื่อนไขบังคับโดยรวมนั้นมีด้วยกันหลายรูปแบบ ดังแสดงตัวอย่างในรูปที่ 2.3 ซึ่งการคำนวณกำหนดการพลวัตจะคำนวณเฉพาะในช่องที่ถูกแรเงาไว้เท่านั้น ส่วนในช่องที่ไม่ได้ถูกแรเงาจะถือว่าค่าในช่องนั้นมีค่าเป็นอนันต์



ก) ข)

รูปที่ 2.3 ตัวอย่างเงื่อนไขบังคับโดยรวมสำหรับไดนามิกไทม์วอร์ปปิง ก) เงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะ [14] ข) เงื่อนไขบังคับโดยรวมแบบอิตาคุระ [13]

การกำหนดเงื่อนไขบังคับโดยรวมสำหรับไดนามิกไทม์วอร์ปปิงนอกจากจะเป็นการเพิ่มความแม่นยำเนื่องจากจะบังคับให้เปรียบเทียบคู่จุดที่เหมาะสมเท่านั้น [3] ยังมีส่วนช่วยเพิ่มความเร็วในการทำไดนามิกไทม์วอร์ปปิงได้อีกด้วย เนื่องจากลดจำนวนครั้งในการคำนวณกำหนดการพลวัตลงจากที่ต้องคำนวณทั้งตารางเหลือเพียงแค่คำนวณในพื้นที่ที่อยู่ในขอบเขตเงื่อนไขบังคับโดยรวมเท่านั้น

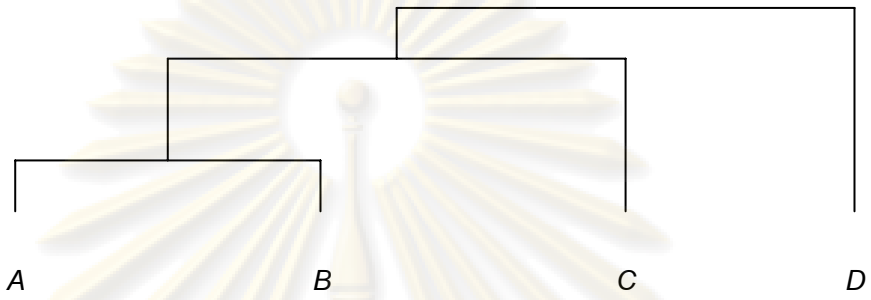
2.1.3 การจับกลุ่ม (Clustering)

การจับกลุ่มข้อมูล คือการจำแนกข้อมูลออกเป็นกลุ่มย่อย ๆ โดยที่ข้อมูลที่ถูกจำแนกอยู่ภายในกลุ่มเดียวกันจะมีคุณลักษณะที่คล้ายคลึงกัน ในด้านการทำเหมืองข้อมูลนั้นการบ่งชี้ความคล้ายกันของข้อมูลมักใช้วิธีการวัดระยะทางของข้อมูล วิธีการจับกลุ่มข้อมูลนั้นแบ่งได้เป็นสองรูปแบบ ได้แก่ การจับกลุ่มแบบลำดับขั้น (Hierarchical Clustering) และการจับกลุ่มแบบแบ่งส่วน (Partitional Clustering)

การจับกลุ่มข้อมูลแบบลำดับขั้นเป็นการจับกลุ่มข้อมูลแบบวนซ้ำ ซึ่งจะได้ผลลัพธ์ของกลุ่มข้อมูลที่มีหลายลำดับขั้นในลักษณะของต้นไม้ที่เรียกว่าเดนโดรแกรม ดังแสดงตัวอย่างในรูปที่ 2.4 ซึ่งจากรูปนั้นข้อมูล A และ B มีความคล้ายกันมากที่สุด จึงถูกจับกลุ่มไว้เป็นกลุ่มเดียวกันเป็นลำดับชั้นแรก จากนั้นข้อมูล C มีความคล้ายกับกลุ่มข้อมูลที่มีข้อมูล A และ B มากที่สุด จึงถูกจับกลุ่มรวมเข้ากับกลุ่มของข้อมูล A และ B เป็นลำดับขั้นที่ 2 และทำการจับกลุ่มข้อมูล D เข้ากับข้อมูลทั้งหมดเป็นขั้นที่ 3 ซึ่งจะเห็นได้ว่าวิธีการจับกลุ่มด้วยวิธีนี้จะทำให้ได้กลุ่มของข้อมูลที่มีความคล้ายคลึงกันของข้อมูลภายในกลุ่มเดียวกันมากที่สุดวิธีหนึ่ง อย่างไรก็ตาม วิธีนี้มีข้อเสียอยู่ที่เวลาที่ใช้ในการคำนวณเพื่อทำการจับกลุ่มนั้นสูงมาก เนื่องจากต้องทำการวัดระยะทางระหว่างทุกคู่ของข้อมูลภายในชุดข้อมูลทั้งหมด ขีดจำกัดเชิงสัญกรณ์ในด้านเวลาจึงสูงถึง $O(n^2m)$ นอกจากนี้การจัดเก็บข้อมูลเดนโดรแกรมยังมีขีดจำกัดเชิงสัญกรณ์ในด้านพื้นที่ (Space Complexity) อยู่ในระดับฟังก์ชันเชิงเส้น $O(n)$ โดยที่ n คือจำนวนข้อมูลทั้งหมด

ภายในชุดข้อมูล และขีดจำกัดเชิงสัญกรณ์ในด้านเวลาของการคำนวณค่าระยะทางระหว่างข้อมูลคู่หนึ่งเท่ากับ O(m)

การมีขีดจำกัดเชิงสัญกรณ์ที่สูงมากทั้งในด้านเวลาในการคำนวณและพื้นที่ที่ต้องใช้บนหน่วยความจำ ดังนั้นจึงเป็นไปไม่ได้ในทางปฏิบัติสำหรับการจับกลุ่มชุดข้อมูลที่มีขนาดใหญ่หรือมีจำนวนข้อมูลอยู่เป็นจำนวนมาก



รูปที่ 2.4 ตัวอย่างเดนโดรแกรมจากการจับกลุ่มแบบลำดับขั้น

การจับกลุ่มข้อมูลแบบแบ่งส่วนเป็นการจับกลุ่มข้อมูลที่ทำการแบ่งข้อมูลออกเป็นกลุ่มภายในครั้งเดียว การจับกลุ่มแบบการแบ่งส่วนนั้นมีด้วยกันหลากหลายวิธี ยกตัวอย่างเช่น การจับกลุ่มบนพื้นฐานของความหนาแน่น (Density-based Clustering) การจับกลุ่มบนพื้นฐานของกริด (Grid-based Clustering) และการจับกลุ่มแบบเคมีน (*k*-mean Clustering) เป็นต้น ซึ่งในงานวิจัยนี้ได้เลือกใช้วิธีการจับกลุ่มแบบเคมีน โดยวิธีในการจับกลุ่มแบบเคมีนจะถูกกล่าวโดยละเอียดดังต่อไปนี้

2.1.3.1 การจับกลุ่มแบบเคมีน (k-Mean Clustering)

การจับกลุ่มแบบเคมีนเป็นวิธีการจับกลุ่มที่เป็นที่นิยมใช้กันอย่างแพร่หลายสำหรับการจับกลุ่มชุดข้อมูลที่มีขนาดใหญ่ เนื่องจากเวลาที่ใช้ในการคำนวณโดยเฉลี่ยในการทำการจับกลุ่มจะน้อยกว่าเมื่อเทียบกับวิธีการจับกลุ่มวิธีอื่น อีกทั้งยังมีพารามิเตอร์ที่ต้องกำหนดเพียงค่าเดียว คือจำนวนกลุ่มที่ต้องการทำการจับกลุ่ม ดังนั้นทำให้การเรียนรู้เพื่อปรับค่าพารามิเตอร์ให้เหมาะสมเป็นไปได้ง่าย

การจับกลุ่มแบบเคมีนเริ่มจากการกำหนดว่าต้องการจับกลุ่มข้อมูลออกเป็นจำนวนกี่กลุ่ม สมมติว่าต้องการจับกลุ่มข้อมูลเป็น k กลุ่ม จากนั้นให้ทำการสุ่มข้อมูลขึ้นมา k ตัวจากในชุดข้อมูลเพื่อสร้างเป็นตัวแทนในแต่ละกลุ่มทั้งหมด k กลุ่ม จากนั้นจะทำการคำนวณแบบวนซ้ำตามขั้นตอนต่อไปนี้

1. ทำการจำแนกกลุ่มข้อมูลทุกตัวในชุดข้อมูล โดยทำการวัดระยะทางกับตัวแทนของทุกกลุ่ม และทำการจำแนกข้อมูลไปยังกลุ่มที่ได้ค่าระยะทางน้อยที่สุด
2. ทำการปรับค่าตัวแทนของกลุ่มแต่ละกลุ่มใหม่ โดยเปลี่ยนเป็นค่าเฉลี่ยของข้อมูลทุกตัวที่ถูกจำแนกอยู่ในกลุ่มนั้น ๆ
3. ทำการวนซ้ำไปยังข้อที่ 1 จนกว่าจะไม่มีข้อมูลใดเลยที่ถูกจำแนกให้เปลี่ยนกลุ่มไปจากเดิมตามในข้อที่ 2

ข้อจำกัดของการจับกลุ่มแบบเคมีนคือการกำหนดตัวแทนของแต่ละกลุ่มจากการหาค่าเฉลี่ยจากข้อที่ 2 นั้นเหมาะสำหรับข้อมูลที่อยู่บนปริภูมิยุคลิด (Euclidean Space) เท่านั้น ดังนั้นมาตรวัดระยะทางที่ใช้จึงควรเป็นมาตรวัดระยะทางแบบยุคลิด สำหรับการใช้มาตรวัดระยะทางที่ไม่มีคุณสมบัติของมาตราเมตริก (Metric Measure) การหาตัวแทนของกลุ่มข้อมูลไม่ควรใช้วิธีการหาค่าเฉลี่ย ในกรณีนี้ควรใช้วิธีการหาตัวแทนของกลุ่มด้วยวิธีอื่น ยกตัวอย่างเช่น การหาเมดอยด์ (Medoid) ของกลุ่ม หรือเรียกวิธีนี้ว่า การจับกลุ่มแบบเคเมดอยด์ (*k*-Medoid Clustering)

2.1.3.2 การจับกลุ่มแบบเคเมดอยด์ (k-Medoid Clustering)

การจับกลุ่มแบบเคเมดอยด์มีแนวคิดที่คล้ายกับการจับกลุ่มแบบเคมีน เพียงแต่ใช้การหาเมดอยด์เป็นตัวแทนกลุ่มแทนที่การใช้ค่าเฉลี่ย วิธีนี้ทำให้สามารถทำการจับกลุ่มข้อมูลที่ไม่ได้อยู่บนปริภูมิยุคลิดได้ การหาตัวแทนกลุ่ม ๆ หนึ่งด้วยการหาเมดอยด์ทำได้โดยการเลือกข้อมูลหนึ่งตัวจากในกลุ่ม ๆ นั้นโดยที่ข้อมูลตัวนั้นมีผลรวมของระยะทางจากข้อมูลตัวอื่นที่อยู่ภายในกลุ่มนั้นทุกตัวน้อยที่สุด อย่างไรก็ตามการจับกลุ่มข้อมูลบนปริภูมิยุคลิดนั้น การจับกลุ่มแบบเคมีนมักให้ผลการจับกลุ่มที่ดีกว่าเมื่อเทียบกับการจับกลุ่มแบบเคเมดอยด์ โดยวัดประสิทธิภาพของกลุ่มข้อมูลจากการวัดความสมเหตุสมผลของการจับกลุ่ม

2.1.4 การวัดความสมเหตุสมผลของการจับกลุ่ม (Cluster Validity Measurement)

การวัดความสมเหตุสมผลของการจับกลุ่มเป็นมาตรวัดหนึ่งที่สามารถบ่งชี้ถึงถึงคุณภาพของกลุ่มข้อมูลที่ได้จากการจับกลุ่มในแต่ละครั้ง โดยทั่วไปการวัดความสมเหตุสมผลของการจับกลุ่มมักใช้สำหรับเปรียบเทียบประสิทธิภาพระหว่างวิธีการจับกลุ่มด้วยรูปแบบต่าง ๆ รวมถึงการปรับค่าพารามิเตอร์สำหรับการจับกลุ่มในแต่ละวิธีให้เหมาะสมอีกด้วย

ความสมเหตุสมผลของการจัดกลุ่มสามารถวัดได้จากการเป็นไปตามคุณสมบัติของการจับกลุ่มที่ดี โดยคุณสมบัติของการจับกลุ่มซึ่งทำให้ได้กลุ่มข้อมูลตามในกรณีอุดมคติมีทั้งหมด 2 ข้อด้วยกัน ดังนี้

1. ความอัดแน่น (Compactness) ข้อมูลที่ถูกจับกลุ่มไว้อยู่ในกลุ่มเดียวกันควรมีความคล้ายคลึงกันมากที่สุด
2. การแยกออกจากกัน (Separation) ข้อมูลที่ถูกจำแนกไว้อยู่คนละกลุ่มควรมีความแตกต่างกันมากที่สุด

การวัดคุณภาพของการจับกลุ่มนั้นจะใช้วิธีการคำนวณค่าดัชนีความสมเหตุสมผล (Validity Index) ของการจับกลุ่มซึ่งเป็นจำนวนจริงค่าหนึ่ง เพื่อสามารถนำค่า ๆ นั้นไปเปรียบเทียบคุณภาพของการจับกลุ่มกับวิธีการจับกลุ่มในรูปแบบอื่น ฟังก์ชันในการคำนวณดัชนีความสมเหตุสมผลของการจับกลุ่มมักมีตัวประกอบหลักของฟังก์ชันอยู่ 2 ตัวประกอบ ได้แก่ การวัดค่าที่ทำให้เกิดกรณีการจับกลุ่มแบบอุดมคติที่เป็นไปตามคุณสมบัติทั้ง 2 ข้อตามที่กล่าวมาแล้วข้างต้นนั่นเอง

ในปัจจุบันได้มีงานวิจัยมากมายที่ได้นำเสนอฟังก์ชันในการคำนวณค่าดัชนีความสมเหตุสมผลของการจับกลุ่มในรูปแบบต่าง ๆ ซึ่งทุกฟังก์ชันที่ได้มีการเสนอมานั้นจะทำการวัดความสมเหตุสมผลของการจับกลุ่มข้อมูลที่อยู่บนปริภูมิยุคลิดเท่านั้น ก่อนจะกล่าวถึงรายละเอียดของการคำนวณค่าดัชนีความสมเหตุสมผลของการจับกลุ่ม ขออธิบายถึงสัญกรณ์ทั้งหมดที่ใช้ในการอธิบายฟังก์ชันในการคำนวณค่าดัชนีความสมเหตุสมผลของการจับกลุ่มดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 คำอธิบายสัญกรณ์ทั้งหมดที่ใช้ในฟังก์ชันการคำนวณค่าดัชนีความสมเหตุสมผลของการจับกลุ่ม

สัญกรณ์	คำอธิบาย
n_c	จำนวนกลุ่มข้อมูลทั้งหมดที่ได้จากการแบ่งกลุ่ม
n_d	จำนวนมิติของข้อมูล
$d(x, y)$	ระยะทางระหว่างข้อมูล x และ y
v_i	ตัวแทนหรือจุดศูนย์กลางของกลุ่มข้อมูลที่ i
c_i	กลุ่มข้อมูลที่ i
$\|\|c_i\|\|$	จำนวนข้อมูลในกลุ่มข้อมูลที่ i
x^p	ค่าในมิติที่ p ของข้อมูล x
$\overline{x}$	ตัวแทนหรือจุดศูนย์กลางของชุดข้อมูล
s_i	ค่าการกระจายตัวของข้อมูลในกลุ่มที่ i

ในงานวิจัยนี้ได้เลือกฟังก์ชันในการคำนวณค่าดัชนีความสมเหตุสมผลของการจับกลุ่มเพื่อใช้สำหรับการทดลองมาทั้งหมด 4 ฟังก์ชัน ดังต่อไปนี้

2.1.4.1 ดัชนีดัน (Dunn Index)

ดัชนีดัน [15] เป็นการคำนวณอัตราส่วนระหว่างระยะทางระหว่างตัวแทนของกลุ่มคู่ที่น้อยที่สุดกับระยะทางที่มากที่สุดระหว่างข้อมูลคู่หนึ่งที่ถูกจับกลุ่มอยู่ในกลุ่มเดียวกัน กำหนดให้ D เป็นดัชนีของดัน สามารถนิยามฟังก์ชันการคำนวณค่า D ได้ดังสมการ (2.4)

$$
\begin{aligned}
D &= \min_{i=1}^{n_c}\left\{\min_{j=1}^{n_c}\left(\frac{d(c_i,c_j)}{\max_{k=1}^{n_c}\left(diam(c_k)\right)}\right)\right\} \\
d(c_i,c_j) &= \min_{x\in c_i, y\in c_j}\left\{d(x,y)\right\} \\
diam(c_i) &= \max_{x,y\in c_i}\left\{d(x,y)\right\}
\end{aligned}
\tag{2.4}
$$

จากสมการ (2.4) จะเห็นได้ว่าถ้าค่าจากฟังก์ชัน $d(c_i, c_j)$ ยิ่งมากหมายความว่าระยะทางระหว่างกลุ่มข้อมูลนั้นมีค่ามาก ซึ่งสามารถแสดงคุณสมบัติของการจับกลุ่มที่ดีด้านความอัดแน่น รวมถึงถ้าค่าจากฟังก์ชัน $diam(c_i)$ ยิ่งน้อยก็สามารถสื่อได้ว่าข้อมูลที่อยู่ภายในกลุ่มเดียวกันมีระยะทางที่ใกล้กัน ซึ่งก็สามารถแสดงคุณสมบัติของการจับกลุ่มที่ดีด้านการแยกออกจากกันด้วย ดังนั้นจึงสามารถสรุปได้ว่ายิ่งดัชนีดันมีค่ามากเท่าไร ก็หมายความว่าได้กลุ่มข้อมูลที่มีคุณสมบัติเข้าใกล้กรณีอุดมคติมากขึ้นไปด้วย หรือก็คือเป็นการจับกลุ่มที่ดีตรงตามทั้งคุณสมบัติการแยกออกจากกันและความอัดแน่น

2.1.4.2 ดัชนีเดวีส์บูลดิน (Davies Bouldin index)

ดัชนีเดวีส์บูลดิน [16] ใช้มาตรวัดความคล้ายคลึงกันของแต่ละกลุ่มข้อมูลเป็นตัวบ่งชี้คุณภาพของกลุ่มข้อมูล กำหนดให้ R_{ij} เป็นมาตรวัดความคล้ายคลึงกันระหว่างกลุ่มข้อมูลที่ i และกลุ่มข้อมูลที่ j สามารถนิยามฟังก์ชันในการคำนวณค่า R_{ij} ได้ตามสมการ (2.5)

$$
R_{ij} = \frac{s_i + s_j}{d(v_i, v_j)} \tag{2.5}
$$

โดยที่ s_i เป็นค่าการกระจายตัวของข้อมูลในกลุ่มที่ i ซึ่งคำนวณได้จากสมการ (2.6)

$$
s_i = \frac{1}{\|c_i\|}\sum_{x\in c_i} d(x, v_i) \tag{2.6}
$$

จากสมการ (2.5) ค่า $d(v_i,v_j)$ เป็นความแตกต่างระหว่างกลุ่มข้อมูลโดยวัดจากระยะทางระหว่างตัวแทนของกลุ่มข้อมูลทั้งสอง ดังนั้นยิ่งได้ค่า $d(v_i,v_j)$ มากทำให้ยิ่งเป็นไปตามคุณสมบัติของการจัดกลุ่มที่ดีด้านการแยกออกจากกัน ส่วนสมการ (2.6) เป็นการวัดค่าการกระจายตัวของข้อมูลในกลุ่ม ๆ หนึ่ง โดยวัดจากค่าเฉลี่ยของระยะทางระหว่างข้อมูลทุกตัวภายในกลุ่มกับตัวแทนของกลุ่ม ยิ่งค่า s_i น้อยจะทำให้เป็นไปตามคุณสมบัติของการจัดกลุ่มที่ดีด้านความอัดแน่น สำหรับการจัดกลุ่มที่ดีที่เป็นไปตามคุณสมบัติทั้งสองข้อแล้วนั้น จึงควรได้ค่า R_{ij} ที่น้อยสำหรับทุกค่า i และ j

สำหรับการวัดความสมเหตุสมผลของกลุ่มข้อมูลทั้งหมด สามารถวัดได้จากค่าดัชนีเดวีส์บูลดิน DB โดยคำนวณได้จากสมการ (2.7) ค่าดัชเดวีส์บูลดินที่น้อยแสดงถึงการจับกลุ่มที่ดีตรงตามทั้งคุณสมบัติการแยกออกจากกันและความอัดแน่น

$$DB = \frac{1}{n_c}\sum_{i=1}^{n_c} R_i$$
$$R_i = \max_{j=1..n_c, i\neq j}\left(R_{ij}\right) \tag{2.7}$$

2.1.4.3 ดัชนีความสมเหตุสมผลของส่วนเบี่ยงเบนมาตรฐาน (SD Validity Index)

ดัชนีความสมเหตุสมผลของส่วนเบี่ยงเบนมาตรฐาน [17] แบ่งการวัดค่าความสมเหตุสมผลออกเป็น 2 ส่วน ได้แก่ ส่วนของการวัดความเปลี่ยนแปลงของส่วนเบี่ยงเบนมาตรฐานของข้อมูลหลังการจับกลุ่ม และส่วนของการวัดความกระจายตัวของกลุ่มข้อมูล

ในส่วนของการวัดความเปลี่ยนแปลงของส่วนเบี่ยงเบนมาตรฐานของข้อมูลหลังการจับกลุ่ม กำหนดให้ $\sigma(x)$ เป็นค่าความแปรปรวนของชุดข้อมูลก่อนการจับกลุ่มซึ่งสามารถคำนวณได้จากสมการ (2.8) และกำหนดให้ $\sigma(x)$ เป็นค่าความแปรปรวนของข้อมูลภายในกลุ่มที่ i ซึ่งสามารถคำนวณได้จากสมการ (2.9)

$$\sigma(x) = \sqrt{\sum_{p=1}^{n_d}\left(\frac{1}{n}\sum_{k=1}^{n}\left(x_k^p - \bar{x}^p\right)^2\right)^2} \tag{2.8}$$

$$\sigma(v_i) = \sqrt{\sum_{p=1}^{n_d}\left(\frac{1}{\|c_i\|}\sum_{k=1}^{\|c_i\|}\left(x_k^p - \bar{x}^p\right)^2\right)^2} \tag{2.9}$$

กำหนดให้ *scatt* แทนค่าความเปลี่ยนแปลงของส่วนเบี่ยงเบนมาตรฐานของข้อมูลหลังการจับกลุ่มซึ่งคำนวณได้จากอัตราส่วนระหว่างค่าเฉลี่ยของความแปรปรวนในแต่ละกลุ่มข้อมูลเทียบกับความแปรปรวนของชุดข้อมูล สามารถนิยามการคำนวณค่า *scatt* ได้จาก

สมการ (2.10) ซึ่งค่า *scatt* ที่น้อยนั้นสามารถบ่งบอกถึงคุณสมบัติด้านความอัดแน่นกันของการจับกลุ่ม

$$scatt = \frac{\frac{1}{n_c}\sum_{i=1}^{n_c}\sigma(v_i)}{\sigma(x)} \tag{2.10}$$

ในส่วนของการวัดความกระจายตัวของแต่ละกลุ่มข้อมูล กำหนดให้ *dis* แทนค่าความกระจายตัวของกลุ่มข้อมูล ซึ่งสามารถคำนวณได้จากสมการ (2.11) จากสมการนั้นแสดงถึงการคำนวณโดยการกลับส่วนค่าของระยะทางระหว่างตัวแทนของกลุ่มข้อมูล ดังนั้นค่า *dis* ที่น้อยจะแสดงถึงการจับกลุ่มที่ดีตามคุณสมบัติด้านการแยกออกจากกัน

$$dis = \frac{\max_{i,j=1}^{n_c}\left(d(v_i,v_j)\right)}{\min_{i,j=1}^{n_c}\left(d(v_i,v_j)\right)}\sum_{i=1}^{n_c}\left(\sum_{j=1}^{n_c}d(v_i,v_j)\right)^{-1} \tag{2.11}$$

สำหรับการคำนวณค่าดัชนีดัชนีความสมเหตุสมผลของส่วนเบี่ยงเบนมาตรฐาน *SD* สามารถคำนวณได้ตามสมการ (2.12)

$$SD = \alpha \cdot scatt + dis \tag{2.12}$$

โดยที่ค่า α คือค่าความกระจายตัวของแต่ละกลุ่มข้อมูล *dis* ที่ได้จากการจับกลุ่มด้วยวิธีที่ทำให้ได้จำนวนกลุ่มของข้อมูลที่มากที่สุดจากการทดลองเพื่อวัดความสมเหตุสมผลของการจับกลุ่มหลาย ๆ วิธี ซึ่งค่าดัชนีความสมเหตุสมผลของส่วนเบี่ยงเบนมาตรฐานที่น้อยบ่งบอกถึงการจับกลุ่มที่ตรงตามทั้งคุณสมบัติการแยกออกจากกันและความอัดแน่น

2.1.4.4 ดัชนีความสมเหตุสมผลของเอสดีบีดับเบิลยู (S_Dbw Validity Index)

ดัชนีความสมเหตุสมผลของเอสดีบีดับเบิลยู [18] นั้นนำเสนอคล้ายกับดัชนีความสมเหตุสมผลของส่วนเบี่ยงเบนมาตรฐานโดยยังคงใช้ค่า *scatt* เป็นตัววัดความอัดแน่นของการจับกลุ่ม ส่วนมาตรวัดการแยกออกจากกันนั้นคำนวณได้จากค่า *Dens_bw* ซึ่งมีนิยามตามสมการ (2.13) ซึ่งเป็นการเปรียบเทียบความถี่ของข้อมูลที่อยู่ใกล้กับตัวแทนกลุ่มกับความถี่ของข้อมูลที่อยู่ระหว่างตัวแทนกลุ่ม ซึ่งสามารถบ่งชี้ถึงคุณสมบัติความอัดแน่นของกลุ่มข้อมูล

$$Dens\_bw = \frac{1}{n_c(n_c-1)}\sum_{i=1}^{n_c}\left(\sum_{j=1,i\neq j}^{n_c}\frac{density(u_{ij})}{\max\{density(v_i),density(v_j)\}}\right) \tag{2.13}$$

โดยที่ u_{ij} คือค่ากลางระหว่างตัวแทนข้อมูลกลุ่มที่ i กับตัวแทนข้อมูลกลุ่มที่ j และฟังก์ชัน $density(x)$ เป็นฟังก์ชันในการนับจำนวนข้อมูลทั้งหมดจากในชุดข้อมูลที่มีระยะทางจาก x เป็นระยะทางไม่เกินค่าส่วนเบี่ยงเบนมาตรฐาน $stdev$ ซึ่งนิยามได้จากสมการ (2.14)

$$stdev = \frac{1}{n_c}\sqrt{\sum_{i=1}^{n_c}\|\sigma(v_i)\|} \tag{2.14}$$

กำหนดให้ S_Dbw แทนค่าดัชนีความสมเหตุสมผลของเอสดีบีดับเบิลยู ซึ่งสามารถนิยามได้จากสมการ (2.15)

$$S\_Dbw = scatt + Dens\_bw \tag{2.15}$$

2.1.5 การเข้าถึงข้อมูลแบบลำดับโดยใช้ดัชนี (Indexed Sequential Access)

การเข้าถึงข้อมูลแบบลำดับโดยใช้ดัชนี เป็นการเข้าถึงข้อมูลด้วยดัชนีที่เน้นการเข้าถึงข้อมูลในรูปแบบการเข้าถึงตามลำดับ (Sequential Access) ซึ่งวิธีนี้มีวัตถุประสงค์เพื่อลดเวลาที่หน่วยประมวลผลกลางต้องรอข้อมูลจากการอ่านแฟ้มข้อมูลในฮาร์ดดิสก์ โดยปกติแล้วการเข้าถึงข้อมูลด้วยดัชนีในรูปแบบทั่วไป เช่น ดัชนีในรูปแบบของต้นไม้ เป็นต้น มักจะก่อให้เกิดการเข้าถึงข้อมูลโดยสุ่ม (Random Access) ซึ่งก่อให้เกิดความล่าช้าในการเข้าถึงข้อมูลอย่างมากเมื่อเทียบกับการเข้าถึงตามลำดับ ดังนั้นการเข้าถึงลำดับดัชนีจึงเป็นการระบุตำแหน่งในการเข้าถึงข้อมูลเป็นบริเวณกว้าง และทำการเข้าถึงตามลำดับไปยังบริเวณดังกล่าว

2.2 งานวิจัยที่เกี่ยวข้อง

ในหัวข้อนี้จะกล่าวถึงงานวิจัยต่าง ๆ ที่ทำการเพิ่มความเร็วในการค้นคืนข้อมูลอนุกรมเวลาตามความคล้ายโดยใช้มาตรวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงเป็นตัวกำหนดความคล้ายกันของข้อมูล วิธีการเพิ่มความเร็วนั้นแบ่งออกเป็น 2 หัวข้อใหญ่ ๆ ได้แก่ การลดทอนข้อมูลด้วยการประมาณค่าระยะทางไดนามิกไทม์วอร์ปปิงจากฟังก์ชันขอบเขตล่าง และการจัดทำดัชนีสำหรับการค้นหาข้อมูล

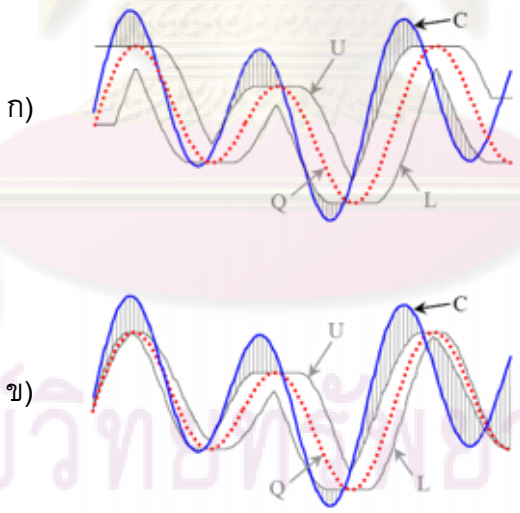
2.2.1 ฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิง (Lower Bounding Function for DTW)

เนื่องจากการคำนวณไดนามิกไทม์วอร์ปปิงนั้นต้องใช้เวลาในการคำนวณสูงโดยมีขีดจำกัดเชิงสัญกรณ์ในด้านเวลาเป็นฟังก์ชันพหุนามกำลังสองหรือ $O(n^2)$ เมื่อ n คือความยาวของข้อมูลอนุกรมเวลา ดังนั้นนับตั้งแต่ปี 1998 ได้มีงานวิจัย [5] ที่เริ่มคิดที่จะเพิ่มความเร็วในการคำนวณไดนามิกไทม์วอร์ปปิงโดยคิดค้นฟังก์ชันขอบเขตล่างซึ่งเป็นฟังก์ชันที่สามารถ

ประมาณค่าระยะทางจากการคำนวณไดนามิกไทม์วอร์ปปิงโดยที่ค่าประมาณที่ได้จะมีค่าไม่เกินค่าระยะทางจริง วิธีดังกล่าวสามารถตัดทอนข้อมูลบางส่วนได้จากการแทนที่ด้วยการคำนวณค่าขอบเขตล่างซึ่งใช้เวลาในการคำนวณน้อยกว่ามาก ฟังก์ชันดังกล่าวแบ่งได้เป็น 2 แนวคิดใหญ่ ๆ ได้แก่ การประมาณด้วยการวัดระยะทางแบบยุคลิดและการประมาณด้วยการวัดระยะทางโดยใช้กำหนดการพลวัต

2.2.1.1 การประมาณฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิงด้วยการวัดระยะทางแบบยุคลิด

นับตั้งแต่มีการคิดค้นฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิง ได้เริ่มมีงานวิจัย [5] ที่ใช้การวัดระยะทางเป็นแบบยุคลิดในการประมาณฟังก์ชันขอบเขตล่าง เนื่องจากสามารถประมาณค่าขอบเขตล่างของไดนามิกไทม์วอร์ปปิงด้วยขีดจำกัดเชิงสัญกรณ์เพียงฟังก์ชันเชิงเส้น $O(n)$ เมื่อ n คือความยาวของข้อมูลอนุกรมเวลา จากนั้นจึงได้มีงานวิจัยอีกเป็นจำนวนมากที่ได้พัฒนาฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิงด้วยการวัดระยะทางแบบยุคลิด [1, 4, 6] เพื่อให้ได้ค่าประมาณระยะทางที่ได้จากฟังก์ชันขอบเขตล่างที่เข้าใกล้ค่าระยะทางจริงที่ได้จากการคำนวณไดนามิกไทม์วอร์ปปิง จนงานวิจัยล่าสุดของ Keogh ได้เสนอฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิงด้วยการวัดระยะทางแบบยุคลิด เรียกว่า LB_Keogh ดังแสดงตัวอย่างในรูปที่ 2.5



รูปที่ 2.5 ตัวอย่างการคำนวณฟังก์ชันขอบเขตล่างของระยะทางแบบไดนามิกไทม์วอร์ปปิง LB_Keogh ก) ฟังก์ชันขอบเขตล่างภายใต้การกำหนดเขื่อนไขบังคับโดยรวมแบบซาโก-ชิบะ ข) ฟังก์ชันขอบเขตล่างภายใต้การกำหนดเขื่อนไขบังคับโดยรวมอิตาคุระ (ที่มา : Keogh และ Ratanamahatana [8])

ในส่วนบนของรูปที่ 2.5 เป็นการกำหนดเงื่อนไขบังคับโดยรวมในรูปแบบของซาโก-ชิบะ [14] และด้านล่างจะเป็นรูปแบบของอิตาคุระ [13] โดยการกำหนดขอบเขตช่วงบนของแต่ละจุดบนข้อมูลอนุกรมเวลา U และขอบเขตช่วงล่างของแต่ละจุดบนข้อมูลอนุกรมเวลา L จากค่าในการคำนวณกำหนดการพลวัตภายใต้เงื่อนไขบังคับโดยรวมสำหรับข้อมูลสอบถาม (Query) ส่วนฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิงจะคำนวณได้จากการหาระยะทางแบบยุคลิดระหว่างข้อมูลตัวเลือก (Candidate) C กับช่วงระหว่างขอบเขตล่างถึงขอบเขตบนที่สร้างจากข้อมูลสอบถาม ดังแสดงเป็นส่วนพื้นที่แรเงา

เพื่อความง่ายต่อการเข้าใจ จึงกำหนดให้ใช้การกำหนดเงื่อนไขบังคับรวมแบบซาโก-ชิบะ [14] โดยมีความกว้างของเงื่อนไขบังคับรวมเป็น r ซึ่งหมายความว่าเงื่อนไขบังคับรวมจะกำหนดให้สามารถทำการเปรียบเทียบจุดข้อมูลที่แตกต่างกันในมิติของเวลาไม่เกิน r หน่วยของเวลา และกำหนดให้มีข้อมูลสอบถาม Q ที่มีความยาว N ซึ่งประกอบด้วยจุดข้อมูล $q_1, q_2, q_3, \ldots, q_N$ ตามลำดับ สามารถนิยามขอบเขตบน $U = \{u_1, u_2, \ldots, u_N\}$ และขอบเขตล่าง $L = \{l_1, l_2, \ldots, l_N\}$ ของแต่ละจุดบนข้อมูลอนุกรมเวลาได้จากสมการ (2.16)

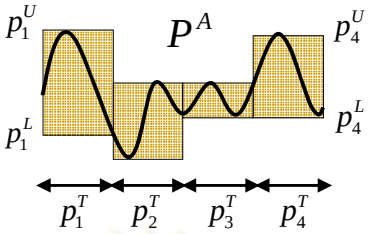
$$u_i = \max_{j=\max(0,i-r)}^{\min(i+r,N)} (q_j)$$
$$l_i = \min_{j=\max(0,i-r)}^{\min(i+r,N)} (q_j) \qquad (2.16)$$

กำหนดให้มีข้อมูลตัวเลือก C ที่มีความยาว N ซึ่งประกอบด้วยจุดข้อมูล $c_1, c_2, c_3, \ldots, c_N$ ตามลำดับ และกำหนดให้ฟังก์ชัน $LB\_Keogh(U, L, C)$ เป็นฟังก์ชันขอบเขตล่างของระยะทางแบบไดนามิกไทม์วอร์ปปิง LB_Keogh โดยมีนิยามตามสมการ

$$LB\_Keogh(U,L,C) = \sum_{i=1}^{N} \begin{cases} (c_i - u_i)^2 & \text{if } c_i > u_i \\ (l_i - c_i)^2 & \text{if } c_i < l_i \\ 0 & \text{otherwise} \end{cases} \qquad (2.17)$$

2.2.1.2 การประมาณฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิงด้วยการวัดระยะทางโดยใช้กำหนดการพลวัต

งานวิจัยของ Sakurai และคณะ [7] ได้เสนอฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิงแบบใหม่โดยยังคงคำนวณด้วยวิธีกำหนดการพลวัต เรียกว่า วิธี FTW โดยทำการลดจำนวนมิติของข้อมูลอนุกรมเวลาลง โดยทำการแบ่งข้อมูลอนุกรมเวลาออกเป็นช่วงย่อย ๆ แต่ละช่วงของข้อมูลจะเก็บค่าสูงสุด ค่าต่ำสุด และความกว้างของช่วงนั้น ดังแสดงตัวอย่างในรูปที่ 2.6 ซึ่งจะแบ่งข้อมูลออกเป็น 3 ช่วง แต่ละช่วงจะเก็บค่าสูงสุด p^U ค่าต่ำสุด p^L และความยาวของช่วงนั้น p^T

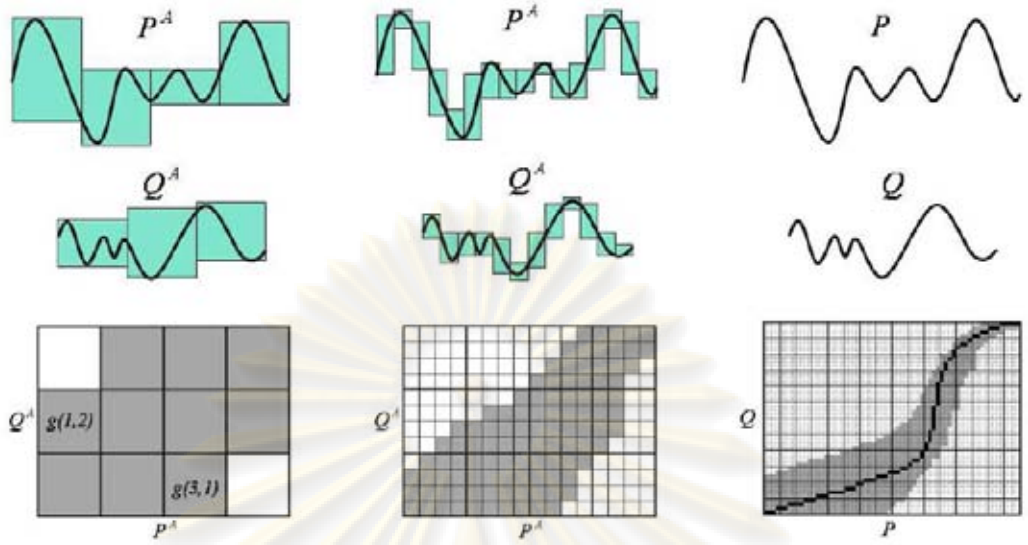


รูปที่ 2.6 ตัวอย่างการแบ่งข้อมูลอนุกรมเวลาออกเป็นช่วงย่อย ๆ สำหรับคำนวณฟังก์ชันขอบเขตล่าง (ที่มา : Sakurai และคณะ [7])

จากนี้จะขอนิยามการแบ่งข้อมูลอนุกรมเวลาออกเป็นช่วงย่อย ๆ โดยสมมติว่าต้องการแบ่งข้อมูล Q ที่มีความยาว N ซึ่งประกอบด้วยจุดข้อมูล $q_1, q_2, q_3, \ldots, q_N$ ตามลำดับ เป็น Q^A ซึ่งประกอบด้วย M ช่วงข้อมูล $Q_1^A, Q_2^A, Q_3^A, ..., Q_M^A$ แต่ละช่วงของข้อมูลใด ๆ Q_i^A ประกอบด้วยค่าสูงสุดของช่วง Q_i^U ค่าต่ำสุดของช่วง Q_i^L และความยาวของช่วง Q^T จะสามารถคำนวณค่าต่าง ๆ ดังกล่าวได้ตามสมการ

$$
\begin{aligned}
Q_i^U &= \max_{j \in \left(\left\lceil \frac{N}{M} * (i-1) \right\rceil, \left\lceil \frac{N}{M} * i \right\rceil\right]} \left(Q_j\right) \\
Q_i^L &= \min_{j \in \left(\left\lceil \frac{N}{M} * (i-1) \right\rceil, \left\lceil \frac{N}{M} * i \right\rceil\right]} \left(Q_j\right) \\
Q_i^T &= \left\lceil \frac{N}{M} * i - \left\lceil \frac{N}{M} * (i-1) \right\rceil \right\rceil
\end{aligned}
\qquad (2.18)
$$

ในส่วนของการคำนวณฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิงด้วยวิธีนี้จะเริ่มจากการแบ่งข้อมูลออกเป็นช่วงจำนวนน้อย ๆ ก่อน หรือคือเป็นการประมาณข้อมูลอนุกรมเวลาอย่างหยาบ แล้วทำการคำนวณระยะทางแบบไดนามิกไทม์วอร์ปปิงโดยให้แต่ละช่วงของข้อมูลเปรียบเสมือนจุดหนึ่งของข้อมูล ถ้าหากว่าค่าระยะทางที่ได้จากการประมาณยังคงไม่เกินค่าที่ระยะทางที่ต้องการค้น ก็จะทำการเพิ่มความละเอียดสำหรับการประมาณโดยการลดจำนวนช่วงที่ทำการแบ่งข้อมูลลงจนสุดท้ายจะทำการคำนวณระยะทางแบบไดนามิกไทม์วอร์ปปิงจริง ดังแสดงตัวอย่างในรูปที่ 2.7 ทางด้านซ้ายมือจะเป็นการประมาณฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิงอย่างหยาบ โดยทำการแบ่งช่วงของข้อมูล Q และ P ให้เป็นลำดับของช่วงข้อมูล Q^A และ P^A ซึ่งประกอบไปด้วย 3 และ 4 ช่วงตามลำดับ ซึ่งสามารถคำนวณค่าประมาณขอบเขตล่างของระยะทางไดนามิกไทม์วอร์ปปิงได้อย่างรวดเร็ว จากนั้นถ้าระยะทางขอบเขตล่างนั้นมีค่าเกินกว่าค่าระยะทางจริงตามที่ต้องการ ก็จะทำการเพิ่มจำนวนช่วงของข้อมูลให้มากขึ้นเป็น 12 และ 16 ช่วงตามลำดับดังแสดงตัวอย่างในส่วนกลางของรูปที่ 2.7 แต่ถ้าระยะทางที่ได้ยังคงเกินกว่าค่าระยะทางจริงตามที่ต้องการอยู่ และไม่สามารถเพิ่มจำนวนช่วงของข้อมูลได้อีก ถึงจะทำการเข้าถึงข้อมูลจริงเพื่อทำการคำนวณระยะทางแบบไดนามิกไทม์วอร์ปปิงกับตัวข้อมูลจริงดังแสดงในส่วนขวาของรูปที่ 2.7



รูปที่ 2.7 ตัวอย่างการประมาณฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิงด้วยวิธี FTW (ที่มา : Sakurai และคณะ [7])

สำหรับระยะทางระหว่างช่วงของข้อมูลสามารถคำนวณได้ดังต่อไปนี้ สมมติว่าต้องการจะหาระยะทางระหว่างช่วงข้อมูล p และ q โดยที่ช่วง p มีค่าสูงสุดและค่าต่ำสุดของช่วงคือ p^U และ p^L ตามลำดับและมีความยาวของช่วงคือ p^T ส่วนช่วง q มีค่าสูงสุดและค่าต่ำสุดของช่วงคือ q^U และ q^L ตามลำดับและมีความยาวของช่วงคือ q^T จะสามารถคำนวณระยะห่างระหว่างช่วง p และ q ได้ดังสมการ (2.19)

$$d(p,q) = x \times \min(p^T, q^T)$$
$$x = \begin{cases} (p^L - q^U)^2 & \text{if } p^L > q^u \\ (q^L - p^U)^2 & \text{if } q^L > p^u \\ 0 & \text{otherwise} \end{cases} \qquad (2.19)$$

ในขั้นการเตรียมข้อมูลสำหรับการค้นคืนข้อมูลด้วยฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิงด้วยวิธี FTW นั้น สามารถทำการแบ่งข้อมูลออกเป็นช่วงย่อย ๆ หลายระดับความละเอียดของช่วงข้อมูลไว้ก่อนล่วงหน้าก่อนทำการค้นคืนข้อมูล สำหรับจำนวนช่วงในแต่ละระดับความละเอียดนั้น ในงานวิจัยที่ได้นำเสนอวิธี FTW ได้กำหนดไว้ว่า จำนวนช่วงที่มีความละเอียดที่สุดจะมีค่าเท่ากับครึ่งหนึ่งของความยาวของข้อมูล จากนั้นจะทำการลดความละเอียดลงระดับละ 1 ใน 4 ของจำนวนช่วงเดิมไปเรื่อย ๆ จนกว่าจำนวนช่วงจะมีค่าไม่เกิน 16 ยกตัวอย่างเช่น ถ้าข้อมูลมีความยาว 2,048 จุดข้อมูล จำนวนช่วงที่มีการแบ่งด้วยความละเอียดสูงสุดคือ 1,024 ช่วงข้อมูล จากนั้นระดับความละเอียดที่ลดลงจะมีจำนวนช่วงลดลงเหลือ 256 64 และ 16 ตามลำดับ โดยในขั้นตอนการค้นคืนข้อมูล จะทำการเข้าถึงช่วงย่อย ๆ หลายระดับของข้อมูลทีละตัว โดยเริ่มจากการประมาณฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิง

ด้วยวิธี FTW กับระดับความละเอียดที่มีจำนวนช่วงข้อมูลน้อยที่สุดซึ่งเท่ากับ 16 ก่อน และถ้าค่าที่ได้จากฟังก์ชันดังกล่าวมีค่าไม่เกินกว่าค่าระยะทางที่น้อยที่สุดที่ต้องการค้นคืน ก็จะทำการเพิ่มความละเอียดเป็นจำนวนช่วงเท่ากับ 64 แต่ถ้าค่าที่ได้นั้นเกินกว่าค่าระยะทางที่น้อยที่สุดแล้วก็จะทำการละทิ้งข้อมูลนี้โดยไม่ต้องเข้าถึงตัวข้อมูลจริงเลย จากนั้นจะดำเนินการดังนี้ไปเรื่อย ๆ จนกว่าจะถึงระดับความละเอียดสูงที่สุดซึ่งเท่ากับ 1,024 ช่วงข้อมูล ถึงจะทำการเข้าถึงตัวข้อมูลจริงเพื่อทำการคำนวณระยะทางแบบไดนามิกไทม์วอร์ปปิงจริง

ด้วยวิธีดังกล่าวจะสามารถลดทอนการคำนวณไดนามิกไทม์วอร์ปปิงกับข้อมูลแต่ละตัวด้วยการแทนที่ด้วยการคำนวณระยะทางไดนามิกไทม์วอร์ปปิงกับช่วงย่อยของข้อมูลซึ่งสามารถคำนวณได้เร็วกว่ามาก นอกจากนี้ยังสามารถลดทอนการเข้าถึงข้อมูลจริงได้ด้วยการเข้าถึงช่วงข้อมูลย่อย ๆ แทน แต่ถ้าสังเกตดี ๆ แล้ว งานวิจัยนี้มีจุดอ่อนอยู่ที่เวลาที่ต้องใช้ในการเข้าถึงข้อมูลทั้งหมด โดยสังเกตได้ว่าจริง ๆ แล้ว ผลรวมของขนาดพื้นที่ที่ใช้จัดเก็บช่วงย่อยของข้อมูลในทุกระดับนั้นมีขนาดใหญ่กว่าขนาดข้อมูลจริงเสียอีก ด้วยเหตุนี้การค้นคืนข้อมูลด้วยวิธี FTW นั้นทำให้ขนาดของข้อมูลที่ต้องทำการเข้าถึงทั้งหมดนั้นมากขึ้นกว่าปกติถึงเกือบสามเท่าตัว นอกจากนี้การเข้าถึงตัวข้อมูลจริงแต่ละครั้งนั้นยังเป็นการเข้าถึงข้อมูลแบบสุ่ม (Random Access) เนื่องจากจะทำการเข้าถึงข้อมูลจริงเพียงบางส่วนเท่านั้น การเข้าถึงข้อมูลแบบสุ่มนั้นต้องใช้เวลามากกว่าการเข้าถึงตามลำดับ (Sequential Access) ถึงสิบเท่าตัว [19, 20] ดังนั้นวิธีนี้จึงมีจุดด้อยอยู่ที่เวลาที่ต้องเสียไปในส่วนของอินพุต/เอาท์พุต

2.2.2 การปรับขนาดเอกรูป (Uniform Scaling)

การปรับขนาดเอกรูปเป็นการปรับขนาดความยาวของข้อมูลอนุกรมเวลาให้ได้ตามความต้องการโดยที่รูปร่างของข้อมูลอนุกรมเวลายังคงสภาพที่คล้ายเดิม โดยสามารถนิยามการปรับขนาดเอกรูปได้ดังต่อไปนี้

กำหนดให้มีข้อมูลอนุกรมเวลา Q ที่มีความยาว N ซึ่งประกอบด้วยจุดข้อมูล $q_1, q_2, q_3, \ldots, q_N$ ตามลำดับ การปรับขนาดเอกรูปของ Q ให้มีความยาว M ซึ่งประกอบด้วยจุดข้อมูล $q'_1, q'_2, q'_3, \ldots, q'_M$ ตามลำดับทำได้ตามสมการ (2.20)

$$q'_i = q_{\lfloor i*M/N \rfloor} \tag{2.20}$$

การปรับขนาดเอกรูปนั้นสามารถนำไปแก้ปัญหาที่เกิดขึ้นกับการประมาณฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิงด้วยการวัดระยะทางแบบยุคลิด เนื่องจากข้อจำกัดของการวัดระยะทางแบบยุคลิดคือข้อมูลทั้งสองข้อมูลที่จะนำมาวัดระยะทางกันจำเป็นต้องมีความยาวที่เท่ากัน แต่สำหรับการวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงนั้นสามารถใช้กับข้อมูลที่มีความยาวแตกต่างกันได้ อย่างไรก็ตามคุณสมบัติของการวัดระยะทางระหว่างข้อมูลที่มีความยาวแตกต่างกันนั้นไม่เป็นประเด็นสาระสำคัญ เนื่องจากได้มีงานวิจัยหนึ่ง

[3] ได้พิสูจน์แล้วว่าการแก้ปัญหาดังกล่าวสามารถทำได้ง่ายดายเพียงทำการปรับความยาวของข้อมูลอนุกรมเวลาให้เท่ากันด้วยการปรับขนาดเอกรูป โดยที่การปรับขนาดเอกรูปนี้จะส่งผลต่อผลลัพธ์ที่ได้จากการวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงอย่างไม่มีนัยสำคัญ

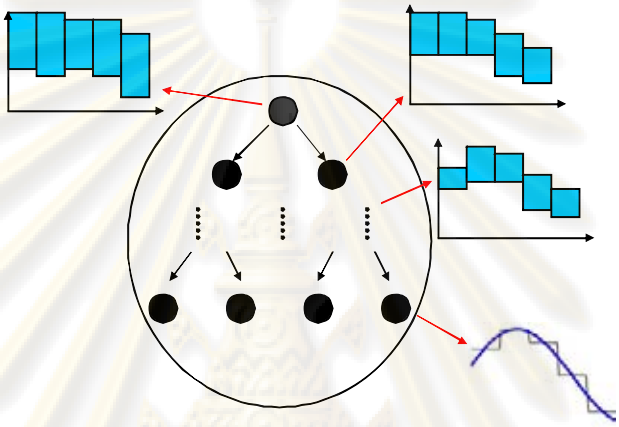
2.2.3 การจัดทำดัชนีสำหรับการค้นคืนข้อมูลอนุกรมเวลา

การทำดัชนีสำหรับการค้นคืนข้อมูลนั้นเป็นการทำการประมวลผลก่อน (Preprocessing) กับชุดข้อมูลหรือฐานข้อมูลเพื่อให้สามารถลดเวลาที่ใช้ในการค้น โดยการค้นหาข้อมูลนั้นสามารถเลือกค้นข้อมูลเฉพาะในส่วนที่ดัชนีได้บ่งชี้ไปเท่านั้น จึงสามารถลดจำนวนข้อมูลที่ต้องทำการเข้าถึงได้สูงมาก ในด้านการค้นคืนข้อมูลจากฐานข้อมูล (Database Retrieval) การค้นหาข้อมูลในฐานข้อมูลนั้นเพียงค้นหาข้อมูลที่มีบางลักษณะประจำ (Attribute) เหมือนกันกับข้อมูลสอบถาม การทำดัชนีนั้นอาจทำได้อย่างง่ายดายเพียงทำการเพิ่มลักษณะประจำ (Attribute) ซึ่งเป็นกุญแจหลัก (Primary Key) สำหรับตารางของชุดข้อมูลนั้น และทำการจัดเรียงข้อมูลตามค่าของกุญแจหลัก สำหรับการค้นคืนข้อมูลนั้นอาจใช้วิธีค้นหาตามกุญแจด้วยการค้นแบบทวิภาค (Binary Search) ซึ่งมีขีดจำกัดเชิงสัญกรณ์ในด้านเวลาอยู่ในระดับ $O(\log n)$ เท่านั้น

สำหรับการค้นคืนข้อมูลตามความคล้ายนั้นจัดว่าเป็นวิธีการทำเหมืองข้อมูลวิธีหนึ่งซึ่งมีความซับซ้อนสูงกว่าการค้นคืนข้อมูลจากฐานข้อมูล โดยเฉพาะอย่างยิ่งกับข้อมูลที่มีจำนวนมิติสูงเช่นข้อมูลอนุกรมเวลานั้นไม่สามารถนำข้อมูลมาเรียงกันเป็นลำดับได้โดยตรง การทำดัชนีสำหรับการค้นหาข้อมูลนั้นจึงไม่สามารถจำกัดขอบเขตของข้อมูลสำหรับการค้นให้อยู่ภายในบริเวณใดบริเวณหนึ่งได้ เนื่องจากเราไม่สามารถระบุได้ว่าข้อมูลใดมีความคล้ายกันกับข้อมูลสอบถามมากที่สุดจนกว่าจะทำการเปรียบเทียบกับข้อมูลครบทั้งชุดข้อมูล อย่างไรก็ตามการทำดัชนีสามารถช่วยลดจำนวนการเข้าถึงข้อมูลได้ด้วยการแทนที่การเปรียบเทียบข้อมูลด้วยการประมาณค่าด้วยระยะทางขอบเขตล่าง

สำหรับข้อมูลอนุกรมเวลา การค้นคืนข้อมูลตามความคล้ายคลึงเป็นหนึ่งในงานสำคัญสำหรับการทำเหมืองข้อมูล วิธีดั้งเดิมวิธีหนึ่งในการค้นหาข้อมูลได้แก่วิธีการค้นตามลำดับ (Sequential Search) โดยทำการคำนวณระยะทางระหว่างข้อมูลสอบถามกับข้อมูลจากฐานข้อมูลทีละตัวและทำการกราดตรวจไปตามลำดับการจัดเก็บของฐานข้อมูลจนครบทั้งฐานข้อมูล จวบจนปี 1993 ได้มีงานวิจัยหนึ่ง [9] ที่ได้เสนอวิธีการค้นหาข้อมูลประเภทข้อมูลอนุกรมเวลาโดยการค้นจากดัชนี (Indexed Search) โดยโครงสร้างดัชนีที่ใช้นั้นอยู่ในรูปแบบของ R*-Tree [12] แต่อย่างไรก็ตามวิธีดังกล่าวประสบปัญหาคำสาปเชิงมิติ (Curse of Dimensionality) ซึ่งเกิดขึ้นกับการทำดัชนีบนปริภูมิที่มีจำนวนมิติสูง เนื่องจากขนาดของปริภูมิจะขยายตัวในระดับเลขชี้กำลัง (Exponential) สำหรับจำนวนมิติที่เพิ่มขึ้น ดังนั้นจึงได้มีงานวิจัยอีกมากมาย ที่ได้นำเสนอและพัฒนาวิธีแก้ปัญหาคำสาปเชิงมิติจากการค้นหาข้อมูลด้วย

โครงสร้างดัชนีในรูปแบบต่าง ๆ โดยทำการลดจำนวนมิติของข้อมูลลงด้วยวิธีที่แตกต่างกัน เช่น วิธีพีเอเอ (PAA: **P**iecewise **A**ggregate **A**pproximation) [1, 8] และวิธีเอพีซีเอ (APCA: **A**daptive **P**iecewise **C**onstant **A**pproximation) [21] เป็นต้น โดยรูปที่ 2.8 แสดงตัวอย่างโครงสร้างดัชนีในรูปแบบต้นไม้สำหรับข้อมูลอนุกรมเวลาที่ถูกลดจำนวนมิติลงด้วยวิธีพีเอเอ โดยในแต่ละโหนดใบ (Leaf Node) จะเก็บข้อมูลอนุกรมเวลาข้อมูลหนึ่งซึ่งถูกลดจำนวนมิติลงด้วยวิธีพีเอเอ ส่วนในแต่ละโหนดภายใน (Internal Node) จะเก็บกล่องขอบเขต (Bounding Box) ซึ่งเป็นช่วงของข้อมูลทั้งหมดที่เก็บอยู่ในปมใบที่เป็นลูกหลานของโหนดนั้น ๆ



รูปที่ 2.8 ตัวอย่างการทำดัชนีด้วยโครงสร้างแบบต้นไม้สำหรับข้อมูลอนุกรมเวลาที่ถูกลดขนาดลงด้วยวิธีพีเอเอ

อย่างไรก็ตามปัญหาที่เกิดจากการทำดัชนีสำหรับค้นคืนข้อมูลอนุกรมเวลาไม่ได้มีเพียงแค่ปัญหาคำสาปเชิงมิติเท่านั้น แต่ยังมีปัญหาที่ยังเป็นอุปสรรคอยู่จนถึงปัจจุบันอยู่สองประเด็น ได้แก่ ปัญหาด้านการเข้าถึงข้อมูลแบบสุ่ม (Random Access) และปัญหาด้านพื้นที่ในการจัดเก็บโครงสร้างดัชนี

ในปัญหาด้านการเข้าถึงข้อมูล (I/O) การเข้าถึงข้อมูลแบบสุ่มนั้นใช้เวลานานกว่าการเข้าถึงข้อมูลแบบลำดับมาก โดยอาจมากกว่าถึงสิบเท่าตัวได้ การเข้าถึงข้อมูลจริงซึ่งเก็บอยู่ในโหนดใบของโครงสร้างดัชนีแต่ละโหนดนั้นมีลำดับการเข้าถึงที่ไม่แน่นอน ดังนั้นจึงเกิดการเข้าถึงข้อมูลแบบสุ่มในทุก ๆ โหนดใบ ซึ่งเวลาที่ใช้ในการค้นคืนข้อมูลด้วยโครงสร้างดัชนีนั้นมักเสียไปกับเวลาในการเข้าถึงข้อมูลเสียเป็นส่วนใหญ่

ในปัญหาด้านพื้นที่ในการจัดเก็บโครงสร้างดัชนี โครงสร้างดัชนีทั้งหมดต้องถูกอ่านมาเก็บไว้ในหน่วยความจำรองก่อนทั้งหมด เนื่องจากลำดับในการเข้าถึงแต่ละโหนดของโครงสร้างดัชนีนั้นไม่แน่นอน การที่จะค่อย ๆ อ่านโครงสร้างดัชนีตามลำดับการเข้าถึงนั้นจะกลายเป็นการเข้าถึงข้อมูลแบบสุ่มในทุก ๆ โหนด ซึ่งกลายเป็นการลดทอนประสิทธิภาพการค้น

คืนข้อมูล ดังนั้นการขยายตัวของโครงสร้างดัชนีในระดับเชิงเส้นเมื่อเทียบกับการขยายตัวของขนาดของชุดข้อมูลอาจทำให้เกิดปัญหาพื้นที่จัดเก็บบนหน่วยความจำรองไม่เพียงพอได้

บทที่ 3

การค้นคืนข้อมูลอนุกรมเวลาด้วยการเข้าถึงข้อมูลแบบลำดับโดยใช้ดัชนี

แนวคิดที่ได้นำเสนอสำหรับงานวิจัยนี้เป็นการนำเสนอวิธีการทำดัชนีที่แม่นยำสำหรับค้นข้อมูลอนุกรมเวลาที่รวดเร็วยิ่งขึ้นสำหรับการวัดระยะทางด้วยไดนามิกไทม์วอร์ปปิงที่มีการกำหนดเงื่อนไขบังคับโดยรวม และข้อมูลทุกตัวต้องมีความยาวเท่ากัน งานวิจัยนี้เน้นไปที่การทำดัชนีที่สามารถระบุดัชนีข้อมูลได้อย่างแม่นยำโดยไม่ก่อให้เกิดความล่าช้าที่เกิดจากการเข้าถึงข้อมูลมากนัก เนื่องจากงานวิจัยนี้ได้คิดค้นโครงสร้างดัชนีรูปแบบใหม่ที่ใช้วิธีการเข้าถึงข้อมูลแบบลำดับโดยใช้ดัชนี ซึ่งจะเป็นโครงสร้างดัชนีที่เน้นด้านการลดเวลาที่ต้องเสียไปจากการเข้าถึงข้อมูลด้วยดัชนี นอกจากนี้โครงสร้างดัชนีในรูปแบบใหม่ไม่ต้องทำการเก็บโครงสร้างไว้ในหน่วยความจำหลักทั้งหมด ซึ่งจะมีส่วนช่วยในการลดการใช้พื้นที่หน่วยความจำหลักได้อย่างมากเมื่อเทียบกับวิธีการทำดัชนีวิธีอื่น

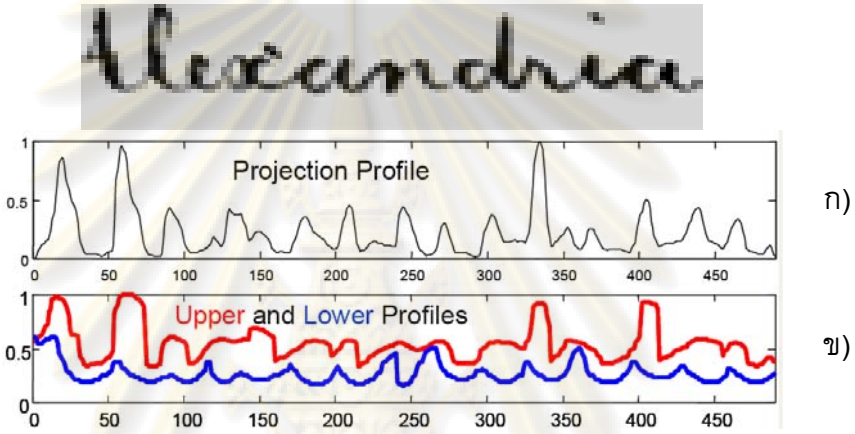
สำหรับในบทที่ 3 จะนำเสนอกรอบงานในการค้นคืนข้อมูลอนุกรมเวลาเป็นลำดับขั้นตอน เริ่มตั้งแต่ที่มาของข้อมูลอนุกรมเวลาซึ่งได้จากการสกัดลักษณะสำคัญของข้อมูล (Feature Extraction) การแปลงข้อมูลอนุกรมเวลาให้เป็นบรรทัดฐาน (Normalization) การทำดัชนีที่แม่นยำสำหรับการค้นคืนข้อมูลอนุกรมเวลา การค้นคืนข้อมูลโดยใช้ดัชนีดังกล่าว การจับกลุ่มข้อมูลอนุกรมเวลา (Clustering) ที่ใช้ในการทำดัชนีข้อมูล และการปรับค่าพารามิเตอร์สำหรับการจับกลุ่มเพื่อให้ได้ดัชนีการค้นคืนข้อมูลที่มีประสิทธิภาพ

3.1 การสกัดลักษณะสำคัญของข้อมูล

ในปัจจุบัน งานประยุกต์ต่าง ๆ ที่ทำการค้นคืนข้อมูลนั้นมักมีให้พบเห็นได้ทั่วไป และเนื่องจากเทคโนโลยีด้านการจัดเก็บข้อมูลนั้นได้มีการพัฒนาไปอย่างต่อเนื่อง การจัดเก็บข้อมูลที่มีความซับซ้อนสูงจึงไม่เป็นอุปสรรคอีกต่อไป เช่น การจัดเก็บข้อมูลในรูปแบบของข้อมูลภาพ ข้อมูลเสียง อีกทั้งยังรวมถึงข้อมูลในรูปแบบของสื่อประสมต่าง ๆ ทั้งนี้เพื่อให้งานประยุกต์เหล่านี้สามารถเพิ่มความสะดวกสบายให้กับผู้ใช้ได้สูงสุด ซึ่งการค้นคืนข้อมูลในรูปแบบดังกล่าวนี้มีความยุ่งยากและซับซ้อนมากเมื่อเทียบกับการค้นคืนข้อมูลบนฐานข้อมูลทั่วไป

การเปรียบเทียบข้อมูลที่มีความซับซ้อนสูงนั้นในบางกรณีการทำการเปรียบเทียบข้อมูลกันโดยตรงตามรูปแบบการจัดเก็บนั้นอาจให้ผลการเปรียบเทียบที่ไม่ดี เช่น การเปรียบเทียบความคล้ายคลึงกันของรูปภาพ โดยปกติข้อมูลรูปภาพจะถูกจัดเก็บไว้ในรูปแบบตาราง 2 มิติของค่าสีจากในแต่ละจุดภาพ การเปรียบเทียบความคล้ายคลึงกันโดยตรงโดยการเปรียบเทียบความคล้ายกันของค่าสีในแต่ละจุดภาพนั้นเป็นการไม่สมเหตุสมผลในการบ่งชี้ถึงความคล้ายกันของรูปภาพ โดยทั่วไปในงานประยุกต์ด้านการประมวลผลรูปภาพมักใช้

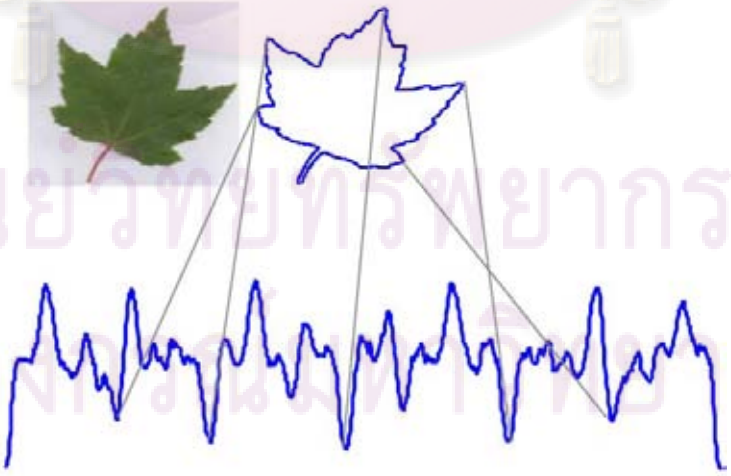
วิธีการสกัดลักษณะสำคัญ ซึ่งทำการสกัดเฉพาะคุณลักษณะที่บ่งบอกถึงเอกลักษณ์ของข้อมูลเหล่านั้นได้ โดยที่ในหลาย ๆ งานประยุกต์จะสกัดลักษณะสำคัญออกมาในรูปแบบของข้อมูลอนุกรมเวลา ดังแสดงตัวอย่างในรูปที่ 3.1 และรูปที่ 3.2 โดยที่รูปที่ 3.1 เป็นการสกัดลักษณะสำคัญจากข้อมูลภาพถ่ายลายมือให้อยู่ในรูปแบบของข้อมูลอนุกรมเวลา ซึ่งสามารถสกัดออกได้ 2 รูปแบบ ได้แก่ การสกัดลักษณะสำคัญจากโพรไฟล์ของภาพฉาย (Projection Profile) ดังแสดงในรูปที่ 3.1 ก) และการสกัดลักษณะสำคัญจากขอบบนและล่างของภาพถ่ายลายมือดังแสดงในรูปที่ 3.1 ข) ส่วนในรูปที่ 3.2 แสดงตัวอย่างการสกัดลักษณะสำคัญจากภาพถ่ายใบไม้ให้อยู่ในรูปแบบของข้อมูลอนุกรมเวลา



ก)

ข)

รูปที่ 3.1 ตัวอย่างการสกัดลักษณะสำคัญจากข้อมูลภาพถ่ายลายมือ ก) การสกัดลักษณะสำคัญจากภาพถ่ายลายมือด้วยโพรไฟล์ของภาพฉาย ข) การสกัดลักษณะสำคัญจากขอบบนและล่างของภาพถ่ายลายมือ (ที่มา : Ratanamahatana และ Keogh [22])



รูปที่ 3.2 ตัวอย่างการสกัดลักษณะสำคัญจากภาพถ่ายใบไม้ (ที่มา : Ratanamahatana และ Keogh [22])

3.2 การแปลงข้อมูลอนุกรมเวลาให้เป็นบรรทัดฐาน

การเปรียบเทียบความคล้ายกันระหว่างข้อมูลอนุกรมเวลาด้วยการวัดระยะทางไดนามิกไทม์วอร์ปปิงมีจุดแข็งเมื่อเทียบกับระยะทางแบบยุคลิดตรงที่สามารถตรวจจับความคล้ายกันเชิงรูปร่างของข้อมูลอนุกรมเวลาได้ จึงเป็นวิธีการวัดระยะทางที่ให้ผลความแม่นยำในการค้นคืนข้อมูลมากกว่า อย่างไรก็ตามการตรวจจับความคล้ายกันเชิงรูปร่างของข้อมูลอนุกรมเวลายังมีอุปสรรคเมื่อข้อมูลนั้นมีรูปร่างคล้ายกัน เพียงแต่อยู่ในมาตราส่วน (Scale) ที่แตกต่างกัน หรือรวมถึงมีความแตกต่างกันเพียงขนาดของแอมพลิจูดเท่านั้น ซึ่งอุปสรรคเหล่านี้อาจก่อให้เกิดปัญหาการตรวจจับความคล้ายกันเชิงรูปร่างด้วยไดนามิกไทม์วอร์ปปิง ปัญหาเหล่านี้แก้ได้โดยการแปลงข้อมูลอนุกรมเวลาให้เป็นบรรทัดฐาน ซึ่งการแปลงข้อมูลอนุกรมเวลาให้เป็นบรรทัดฐานนั้นเป็นการปรับมาตราส่วนและแอมพลิจูดของข้อมูลอนุกรมเวลาให้อยู่ในระดับเดียวกัน ในงานวิจัยทั่วไปมักใช้วิธีการแปลงข้อมูลอนุกรมเวลาให้เป็นบรรทัดฐานด้วยวิธีการใช้คะแนน Z (Z-score Normalization) โดยสามารถนิยามได้ดังนี้

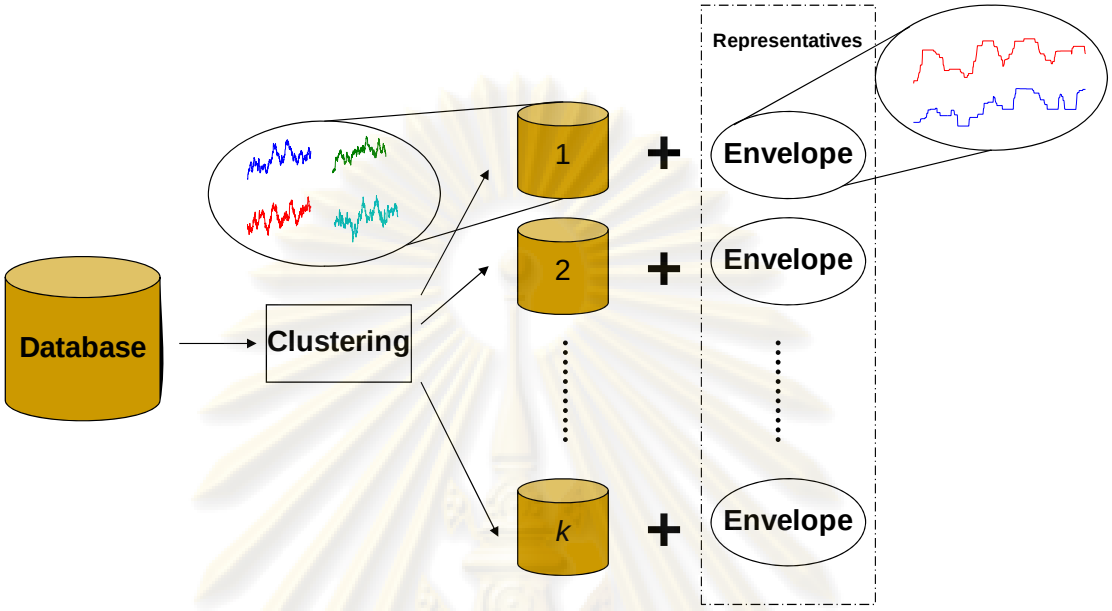
กำหนดให้ข้อมูลอนุกรมเวลา Q มีความยาว N ซึ่งประกอบด้วยจุดข้อมูล q_1, q_2, q_3, … , q_N วิธีการแปลงข้อมูลอนุกรมเวลาให้เป็นบรรทัดฐานจะทำการแทนที่ทุกจุดข้อมูลบนข้อมูลอนุกรมเวลาด้วยค่าคะแนน Z ของแต่ละจุดข้อมูล โดยกำหนดให้ qz แทนข้อมูลที่ได้จากการแปลงข้อมูล Q ให้เป็นบรรทัดฐาน โดยประกอบด้วยจุดข้อมูล qz_1, qz_2, qz_3, … , qz_N สามารถนิยามการคำนวณ qz ได้จากสมการ (3.1)

$$qz_i = \frac{q_i - \overline{q}}{SD}$$
$$\overline{q} = \frac{\sum_{i=1}^{n} q_i}{n} \qquad (3.1)$$
$$SD = \sum_{i=1}^{n} \sqrt{\frac{(q_i - \overline{q})^2}{n}}$$

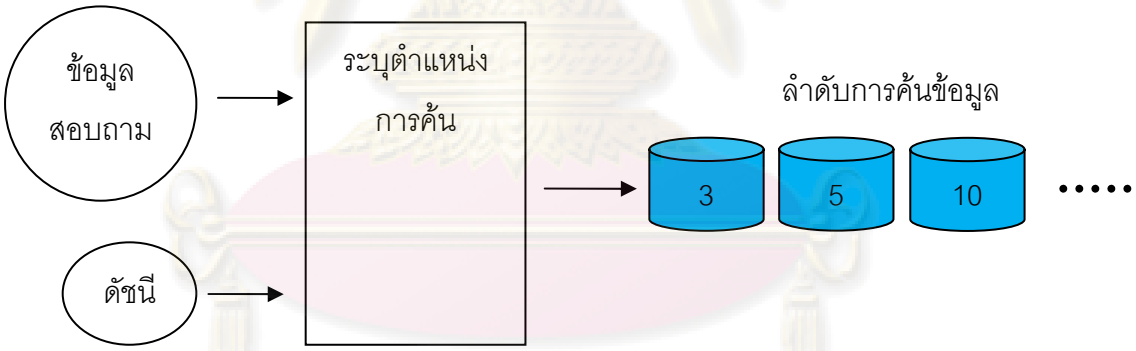
3.3 การจัดเตรียมข้อมูลสำหรับการจัดทำดัชนีการเข้าถึงข้อมูลอนุกรมเวลา

ในภาพรวมของแนวคิดในงานวิจัยนี้ ได้เสนอการเตรียมการก่อนการค้นหาข้อมูลเพื่อสามารถระบุตำแหน่งในการค้นคืนข้อมูลได้ โดยทำการแบ่งกลุ่มข้อมูลในฐานข้อมูลออกเป็นกลุ่มย่อย ๆ และจะทำการกำหนดดัชนีขึ้นมาสำหรับแต่ละกลุ่มข้อมูลเพื่อเป็นตัวแทนสำหรับกลุ่มข้อมูลนั้น ดังแสดงในรูปที่ 3.3 แต่ละกลุ่มของข้อมูลจะถูกนำเสนอด้วยขอบเขตของข้อมูลทั้งหมดที่อยู่ภายในกลุ่มนั้น ๆ ส่วนขั้นตอนการค้นข้อมูลจะทำการนำข้อมูลสอบถามมาคำนวณฟังก์ชันขอบเขตล่างของระยะทางแบบไดนามิกไทม์วอร์ปปิงของทั้งกลุ่มข้อมูลเพื่อนำไปเรียงลำดับกลุ่มข้อมูลสำหรับการค้น นอกจากนี้ค่าขอบเขตล่างที่ได้ยังสามารถนำไปใช้ในการตัด

ทอนกลุ่มข้อมูลบางส่วนออกก่อนการค้นด้วย เพื่อเป็นการลดจำนวนข้อมูลที่จะต้องทำการค้นลง โดยที่ไม่จำเป็นต้องทำการเข้าถึงข้อมูลเหล่านั้นเลย ดังแสดงในรูปที่ 3.4



รูปที่ 3.3 ภาพรวมของขั้นตอนการเตรียมการก่อนการค้น



รูปที่ 3.4 ภาพรวมของขั้นตอนการระบุตำแหน่งการค้น

จากนี้จะกล่าวถึงรายละเอียดขั้นตอนการเตรียมข้อมูลซึ่งประกอบด้วยการจับกลุ่มข้อมูล และการกำหนดตัวแทนของแต่ละกลุ่มข้อมูล

3.3.1 ขั้นการจับกลุ่มข้อมูล

ในด้านการทำเหมืองข้อมูลโดยทั่วไป การจับกลุ่มข้อมูลมักนำไปใช้สำหรับงานด้านการจำแนกข้อมูล รวมถึงสามารถทำการค้นพบองค์ความรู้จากชุดข้อมูลว่าทั้งชุดข้อมูลมีคุณลักษณะเด่น ๆ ของข้อมูลเป็นรูปแบบใดบ้าง ดังนั้นคุณสมบัติของการจับกลุ่มที่ดีจึงมีอยู่ 2

ข้อใหญ่ ๆ ดังที่ได้กล่าวไว้ในบทที่ 2 คือ ความอัดแน่นของข้อมูลภายในกลุ่มเดียวกัน และการแยกออกจากกันของข้อมูลหลังการจัดกลุ่ม แต่สำหรับในงานวิจัยนี้ได้นำแนวคิดของการจับกลุ่มข้อมูลมาใช้ในงานด้านการทำดัชนีสำหรับการค้นคืนข้อมูล ดังนั้นการจับกลุ่มที่ใช้ในงานวิจัยนี้จึงมีวัตถุประสงค์ที่แตกต่างจากการจับกลุ่มที่ใช้ในงานด้านการทำเหมืองข้อมูลทั่วไป โดยในหัวข้อนี้จะนำเสนอวิธีการเตรียมข้อมูลด้วยการจับกลุ่มที่เหมาะสมกับงานวิจัยนี้ทั้งหมด 2 วิธี ได้แก่ วิธีการจับกลุ่มแบบเคมีนภายใต้การลดทอนการคำนวณ และวิธีการจับกลุ่มแบบแทรก (Insertion Clustering) ซึ่งเป็นวิธีการจับกลุ่มใหม่ที่ผู้วิจัยได้คิดค้นขึ้นเพื่อให้เหมาะสมตามวัตถุประสงค์ของงานวิจัยนี้ ซึ่งจะกล่าวโดยละเอียดอีกครั้งภายหลังในหัวข้อที่ 3.5 เนื่องจากต้องกล่าวถึงการนำกลุ่มข้อมูลที่ได้จากการจับกลุ่มไปใช้ในการจัดทำดัชนีก่อนเพื่อให้เข้าใจถึงวัตถุประสงค์ของการจับกลุ่มในงานวิจัยนี้ ในหัวข้อถัดไปจะกล่าวถึงการนำกลุ่มข้อมูลไปใช้งานในด้านการทำดัชนี ซึ่งสิ่งที่นำไปใช้จากกลุ่มข้อมูลก็คือขอบเขตของกลุ่มข้อมูล

3.3.2 การกำหนดขอบเขตของกลุ่มข้อมูลสำหรับการทำดัชนีการค้นคืนข้อมูล

ในด้านการค้นคืนข้อมูลอนุกรมเวลาตามความคล้ายโดยใช้การวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงเป็นตัวบ่งชี้ความคล้ายกันของข้อมูล การคำนวณระยะทางไดนามิกไทม์วอร์ปปิงในแต่ละครั้งนั้นสามารถแทนที่ได้ด้วยการคำนวณค่าขอบเขตล่างของไดนามิกไทม์วอร์ปปิงด้วยวิธีต่าง ๆ เช่น LB_Keogh [8] และ FTW [7] เป็นต้น ซึ่งใช้เวลาในการคำนวณน้อยกว่ามาก อย่างไรก็ตามประสิทธิภาพในการแทนที่การคำนวณไดนามิกไทม์วอร์ปปิงด้วยค่าขอบเขตล่างนั้นขึ้นอยู่กับค่าระยะทางที่มากที่สุดที่ระบบการค้นจะทำการเลือกข้อมูลนั้น ๆ เป็นคำตอบของการค้นคืน หรือที่เรียกว่า ระยะทาง *best-so-far* ดังนั้นถ้าค่าขอบเขตล่างขอบระยะทางมีค่าเกินกว่าค่า *best-so-far* แล้ว จะสามารถยุติการคำนวณระยะทางกับข้อมูลนั้นได้ จึงทำให้สามารถลดทอนการคำนวณไดนามิกไทม์วอร์ปปิงได้ ด้วยเหตุนี้ค่า *best-so-far* ที่น้อยทำให้โอกาสที่จะสามารถลดทอนการคำนวณไดนามิกไทม์วอร์ปปิงด้วยค่าขอบเขตล่างนั้นมีสูงมากขึ้นไปด้วย

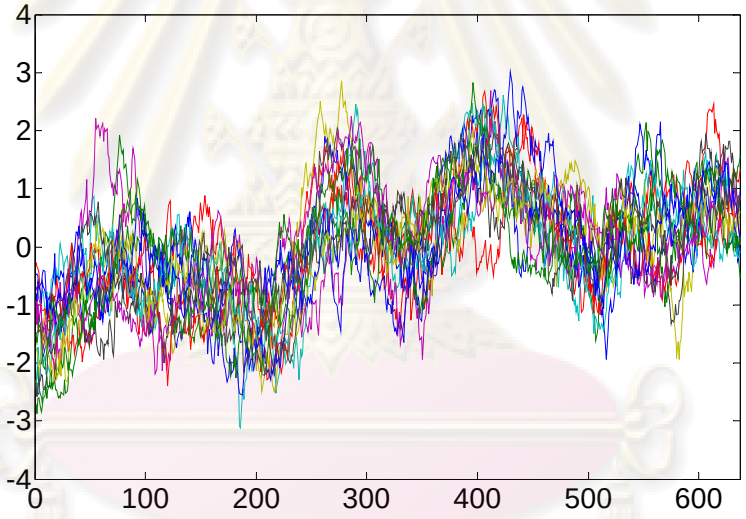
การที่จะเพิ่มประสิทธิภาพสูงสุดให้กับการประมาณค่าขอบเขตล่างนั้น จำต้องลดค่า *best-so-far* ให้เหลือน้อยโดยเร็วที่สุด ซึ่งทำได้โดยทำการค้นหาข้อมูลที่มีระยะทางห่างจากข้อมูลสอบถามน้อยที่สุดให้พบตั้งแต่การค้นหาข้อมูลช่วงแรกเริ่ม งานวิจัยนี้ได้เสนอการค้นคืนข้อมูลจากชุดข้อมูลที่ถูกแบ่งออกเป็นกลุ่ม ๆ โดยได้นำเสนอวิธีการทำดัชนีเพื่อเพิ่มความเร็วในการค้นหาข้อมูลด้วยวิธีการจัดลำดับกลุ่มข้อมูลที่จะทำการค้นหา เพื่อให้สามารถเลือกค้นกลุ่มข้อมูลที่มีแนวโน้มว่าจะค้นพบข้อมูลที่มีความคล้ายคลึงกับข้อมูลสอบถามก่อน

ในการคาดเดาว่ากลุ่มข้อมูลใดจากในชุดข้อมูลนั้นน่าจะมีข้อมูลที่คล้ายคลึงกับข้อมูลสอบถาม งานวิจัยนี้ได้เสนอวิธีคาดเดาจากค่าฟังก์ชันขอบเขตล่างของค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลซึ่งจะกล่าวโดยละเอียดในหัวข้อที่ 3.4.1 โดยฟังก์ชัน

ดังกล่าวมีองค์ประกอบในการคำนวณอยู่ 2 องค์ประกอบ ได้แก่ ข้อมูลสอบถาม และขอบเขตของกลุ่มข้อมูล ในหัวข้อนี้จะกล่าวถึงวิธีการกำหนดขอบเขตของกลุ่มข้อมูล โดยแบ่งออกเป็น 2 วิธี ได้แก่ การกำหนดขอบเขตของกลุ่มข้อมูลโดยยังไม่มีการกำหนดขนาดของเงื่อนไขบังคับโดยรวมสำหรับไดนามิกไทม์วอร์ปปิง และการกำหนดขอบเขตของกลุ่มข้อมูลภายใต้การกำหนดขนาดของเงื่อนไขบังคับโดยรวมสำหรับไดนามิกไทม์วอร์ปปิงที่แน่นอนสำหรับการค้นคืนข้อมูลทุกรูปแบบ

3.3.2.1 การกำหนดขอบเขตของกลุ่มข้อมูลโดยยังไม่มีการกำหนดขนาดของเงื่อนไขบังคับโดยรวมสำหรับไดนามิกไทม์วอร์ปปิง

การกำหนดขอบเขตของกลุ่มข้อมูลในรูปแบบนี้ใช้สำหรับชุดข้อมูลที่ยังไม่มีการกำหนดขนาดของเงื่อนไขบังคับโดยรวมตั้งแต่ช่วงของการเตรียมข้อมูล แต่จะทำการกำหนดขนาดของเงื่อนไขบังคับโดยรวมสำหรับในการทำการค้นคืนข้อมูลแต่ละครั้ง

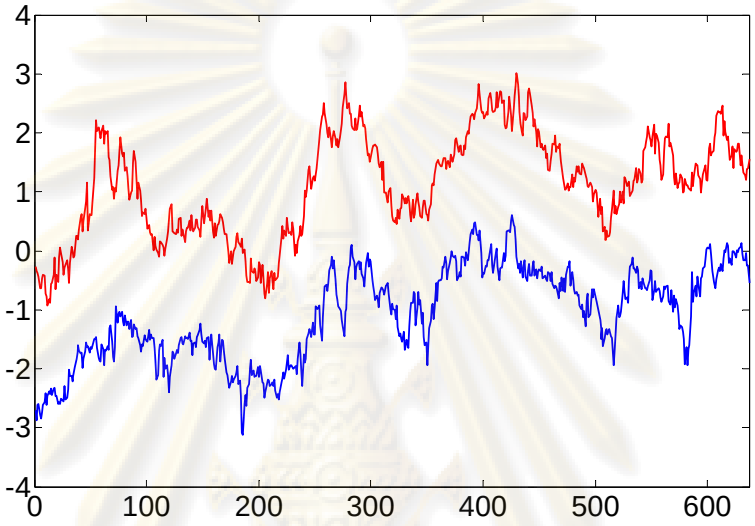


รูปที่ 3.5 ตัวอย่างของกลุ่มข้อมูลที่ผ่านการจับกลุ่มแบบเคมีน

ขอบเขตของกลุ่มข้อมูลแต่ละกลุ่มจะใช้แนวคิดของกล่องขอบเขต (Bounding Box) ซึ่งนำเสนอด้วยค่าที่มากที่สุดและน้อยที่สุดจากในแต่ละมิติของข้อมูลทุกตัวในกลุ่มข้อมูล กำหนดให้มีกลุ่มข้อมูล g ประกอบไปด้วยสมาชิกในกลุ่มทั้งหมด c ตัว ได้แก่ ข้อมูล p_1, p_2, p_3, … , p_c ซึ่งสมาชิกทุกตัวในกลุ่ม g จะเป็นข้อมูลอนุกรมเวลาที่มีความยาวเท่ากับ n โดยที่ทุกข้อมูล p_i ใด ๆ จะประกอบด้วยจุดข้อมูล p_{i1}, p_{i2}, p_{i3}, … , p_{in} จะสามารถนำเสนอขอบเขตของกลุ่มข้อมูลด้วยค่ากล่องขอบเขตช่วงบน B_u ซึ่งประกอบไปด้วยจุดข้อมูล B_{u1}, B_{u2}, B_{u3}, … , B_{un} และค่าขอบเขตช่วงล่าง B_l ซึ่งประกอบด้วยจุดข้อมูล B_{l1}, B_{l2}, B_{l3} จนถึง B_{ln} ได้จากสมการ (3.2) โดยในรูปที่ 3.5 แสดงตัวอย่างกลุ่มของข้อมูลอนุกรมเวลา g ที่ได้จากการแบ่งกลุ่มด้วยวิธี

เคมีน และในรูปที่ 3.6 แสดงกล่องขอบเขตที่สร้างขึ้นจากกลุ่มข้อมูล g ซึ่งตัวกล่องขอบเขตของแต่ละกลุ่มข้อมูลทำหน้าที่เป็นดัชนีของกลุ่มข้อมูลนั้น ๆ

$$B_{ui} = \max_{j=1}^{c} p_{ji}$$
$$B_{li} = \min_{j=1}^{c} p_{ji} \qquad (3.2)$$



รูปที่ 3.6 กล่องขอบเขตที่ครอบคลุมกลุ่มข้อมูลตัวอย่างในรูปที่ 3.5

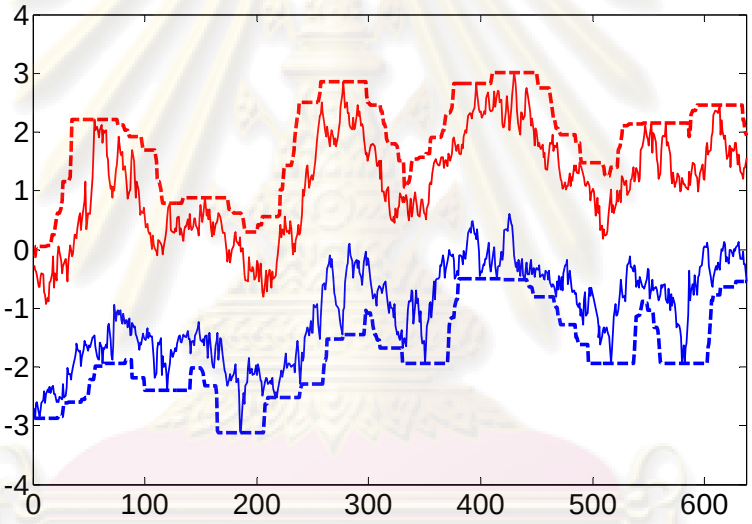
3.3.2.2 การกำหนดขอบเขตของกลุ่มข้อมูลภายใต้การกำหนดขนาดของเงื่อนไขบังคับโดยรวมสำหรับไดนามิกไทม์วอร์ปปิงที่แน่นอนสำหรับการค้นคืนข้อมูลทุกรูปแบบ

การกำหนดขอบเขตของกลุ่มข้อมูลในรูปแบบนี้ใช้สำหรับการค้นคืนที่มีการกำหนดขนาดของเงื่อนไขบังคับโดยรวมที่ตายตัวสำหรับชุดข้อมูลนี้โดยไม่มีการเปลี่ยนแปลงภายหลัง ภายใต้การกำหนดเงื่อนบังคับโดยรวมสำหรับไดนามิกไทม์วอร์ปปิง ประกอบกับการนำแนวคิดในการประมาณค่าขอบเขตล่างของไดนามิกไทม์วอร์ปปิงในรูปแบบ LB_Keogh มาประยุกต์ จะสามารถกำหนดขอบเขตช่วงบนและขอบเขตช่วงล่างของกลุ่มข้อมูลสำหรับการคำนวณฟังก์ชันขอบเขตล่างของค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูล โดยจะกล่าวถึงรายละเอียดวิธีการกำหนดขอบเขตช่วงบนและขอบเขตช่วงล่างของกลุ่มข้อมูลได้ดังต่อไปนี้

เริ่มต้นจากการสร้างกล่องขอบเขตของกลุ่มข้อมูลได้เป็น B_u และ B_l ตามการคำนวณในหัวข้อที่ 3.3.2.1 และกำหนดให้ใช้รูปแบบเงื่อนบังคับโดยรวมเป็นแบบซาโก-ชิบะโดย

มีค่าความกว้างของเงื่อนไขบังคับโดยรวมเท่ากับ r ซึ่งหมายความว่าการคำนวณไดนามิกไทม์วอร์ปปิงจะไม่ทำการเปรียบเทียบคู่จุดข้อมูลใด ๆ ที่อยู่ห่างกันเกินกว่า r ในมิติของเวลา จะสามารถนำเสนอขอบเขตของกลุ่มข้อมูลได้ด้วยค่าขอบเขตช่วงบน U ซึ่งประกอบไปด้วยจุดข้อมูล $u_1, u_2, u_3, \ldots, u_n$ และค่าขอบเขตช่วงล่าง L ซึ่งประกอบไปด้วยจุดข้อมูล $l_1, l_2, l_3, \ldots, l_n$ ได้จากสมการ (3.3) ในรูปที่ 3.7 แสดงถึงขอบเขตของกลุ่มข้อมูลที่ถูกสร้างจากกล่องขอบเขตที่ได้ดังกล่าว โดยนำเสนอเส้นขอบเขตส่วนบนและล่างในรูปของเส้นประ และเส้นกล่องขอบเขตในรูปของเส้นทึบ ซึ่งตัวเส้นประทั้งสองเส้นนี้ทำหน้าที่เป็นดัชนีของกลุ่มข้อมูล g

$$u_i = \max_{j=\max(1,i-r)}^{\min(i+r,c)} B_{uj}$$
$$l_i = \min_{j=\max(1,i-r)}^{\min(i+r,c)} B_{lj} \qquad (3.3)$$



รูปที่ 3.7 ขอบเขตของกลุ่มข้อมูลภายใต้การกำหนดเงื่อนไขบังคับโดยรวมที่สร้างจากกล่องขอบเขตในรูปที่ 3.6

3.4 การค้นคืนข้อมูลอนุกรมเวลาด้วยการเข้าถึงข้อมูลแบบลำดับโดยใช้ดัชนี

ในหัวข้อนี้ได้นำเสนอวิธีการค้นคืนข้อมูลอนุกรมเวลาจากชุดข้อมูลที่ถูกแบ่งออกเป็นกลุ่มย่อย ๆ โดยใช้แนวทางในการค้นหาด้วยดัชนีที่จัดทำขึ้นด้วยวิธีที่ได้กล่าวไว้ในหัวข้อที่ 3.3 มาใช้ในการจัดลำดับการค้นหาข้อมูลว่าควรทำการค้นข้อมูลจากกลุ่มข้อมูลใดก่อนหลัง โดยเลือกค้นกลุ่มข้อมูลที่มีโอกาสพบข้อมูลที่คล้ายกับข้อมูลสอบถามมากกว่าก่อน ในการคาดเดาโอกาสที่จะพบข้อมูลที่คล้ายกับข้อมูลสอบถามนั้น งานวิจัยนี้ได้เสนอฟังก์ชันขอบเขตล่างของค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลซึ่งสามารถบ่งบอกถึงโอกาสที่

น่าจะพบข้อมูลที่คล้ายกัน อีกทั้งยังสามารถตัดทอนกลุ่มข้อมูลที่ค่าจากฟังก์ชันดังกล่าวบ่งบอกว่าไม่มีโอกาสที่จะพบข้อมูลที่คล้ายกับข้อมูลสอบถามเลย ต่อจากนี้จะกล่าวถึงรายละเอียดของวิธีการคำนวณฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูล และวิธีการค้นคืนข้อมูลโดยใช้ค่าฟังก์ชันดังกล่าวเป็นดัชนีการค้นข้อมูล

3.4.1 ฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูล

ฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลเป็นฟังก์ชันในการประมาณค่าขอบเขตล่างของค่าระยะทางแบบไดนามิกไทม์วอร์ปปิงที่น้อยที่สุดระหว่างข้อมูลสอบถามกับข้อมูลที่เป็นสมาชิกทุกตัวภายในกลุ่ม ค่าที่ได้จากฟังก์ชันขอบเขตล่างดังกล่าวนี้สามารถนำมาช่วยพิจารณาว่าควรทำการค้นหาข้อมูลภายในกลุ่มข้อมูลดังกล่าวหรือไม่ หากค่าที่ได้จากฟังก์ชันนี้ไม่น้อยกว่าค่า *best-so-far* จากกระบวนการค้นหาข้อมูลแล้ว จะสามารถสรุปได้ว่าค่าระยะทางไดนามิกไทม์วอร์ปปิงระหว่างข้อมูลที่เป็นสมาชิกภายในกลุ่มนี้กับข้อมูลสอบถามต้องไม่น้อยกว่าค่า *best-so-far* เช่นกัน ดังนั้นจึงสามารถละทิ้งการค้นหาข้อมูลทุกตัวที่เป็นสมาชิกภายในกลุ่มดังกล่าวได้โดยไม่จำเป็นต้องทำการเข้าถึงข้อมูลดังกล่าวเลยแม้แต่ตัวเดียว

วิธีการคำนวณฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลนั้นแบ่งออกเป็น 2 รูปแบบ ขึ้นอยู่กับวิธีการกำหนดขอบเขตของกลุ่มข้อมูลในขั้นตอนของการเตรียมข้อมูลว่ามีการกำหนดขนาดของเงื่อนไขบังคับโดยรวมที่ตายตัวไว้ก่อนหรือไม่ ดังนั้นวิธีการคำนวณฟังก์ชันดังกล่าวจึงแบ่งออกเป็นฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่มีการตรึงขนาดของเงื่อนไขบังคับโดยรวม และฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่สามารถปรับขนาดของเงื่อนไขบังคับโดยรวมได้

3.4.1.1 ฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่มีการตรึงขนาดของเงื่อนไขบังคับโดยรวม

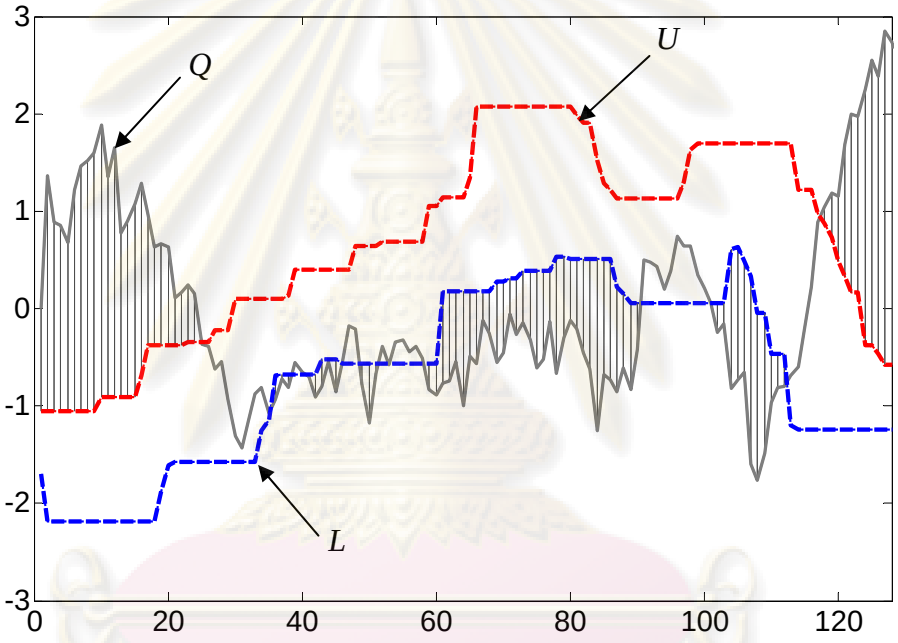
การคำนวณฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่มีการตรึงขนาดของเงื่อนไขบังคับโดยรวมนั้นใช้สำหรับชุดข้อมูลที่มีการกำหนดขนาดของเงื่อนไขบังคับไว้แล้วในขั้นตอนการเตรียมข้อมูลซึ่งใช้วิธีการเตรียมข้อมูลตามหัวข้อที่ 3.3.2.2 โดยวิธีการคำนวณค่าฟังก์ชันดังกล่าวจะกล่าวโดยละเอียดดังต่อไปนี้

กำหนดให้ข้อมูลอนุกรมสอบถาม Q มีความยาว N โดยประกอบด้วยจุดข้อมูล $q_1, q_2, q_3, \ldots, q_N$ และกำหนดให้มีกลุ่มข้อมูลหนึ่งโดยมีการกำหนดขอบเขตของกลุ่มข้อมูล B โดยมีการกำหนดขนาดของเงื่อนไขบังคับโดยรวมไว้แล้ว ซึ่งประกอบด้วยขอบเขตช่วงบน U

และขอบเขตช่วงล่าง L โดยขอบเขตช่วงบน U ประกอบด้วยจุดข้อมูล $u_1, u_2, u_3, \ldots, u_N$ และขอบเขตช่วงล่าง L ประกอบด้วยจุดข้อมูล $l_1, l_2, l_3, \ldots, l_N$ จะสามารถนิยามฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูล $LBG(Q, B)$ ได้ตามสมการ (3.4)

$$LBG(Q,B) = \sum_{i=1}^{N} d(q_i, B)$$

$$d(q_i, B) = \begin{cases} (q_i - u_i)^2 & \text{if } q_i > u_i \\ (l_i - q_i)^2 & \text{if } q_i < l_i \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$



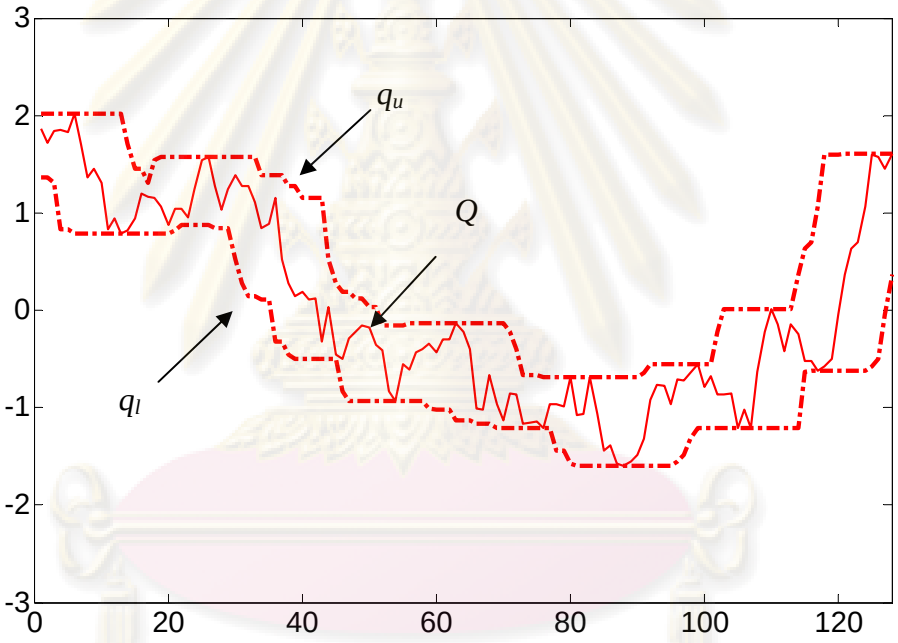
รูปที่ 3.8 ตัวอย่างการคำนวณฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่มีการตรึงขนาดของเงื่อนไขบังคับโดยรวม โดยทำการคำนวณค่าฟังก์ชันดังกล่าวระหว่างข้อมูลสอบถาม Q กับขอบเขตบน U และขอบเขตล่าง L ของกลุ่มข้อมูล ซึ่งแสดงเป็นพื้นที่ในส่วนที่แรเงาไว้

รูปที่ 3.8 แสดงตัวอย่างในการคำนวณฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่มีการตรึงขนาดของเงื่อนไขบังคับโดยรวม โดยให้เส้นประแทนขอบเขตของกลุ่มข้อมูลที่มีการกำหนดขนาดของเงื่อนไขบังคับรวมที่แน่นอนในขั้นการเตรียมข้อมูลตามในหัวข้อที่ 3.3.2.2 โดยเส้นประเส้นบนและล่างแทนขอบเขตช่วงบน U และขอบเขตช่วงล่าง L ตามลำดับ และให้เส้นทึบแทนข้อมูลสอบถาม Q จะสามารถคำนวณค่า

ฟังก์ชันขอบเขตล่างของกลุ่มข้อมูลได้เป็นขนาดของพื้นที่ที่แรเงาซึ่งเป็นส่วนของพื้นที่ที่ข้อมูลสอบถามอยู่นอกขอบเขตของกลุ่มข้อมูล

3.4.1.2 ฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่สามารถปรับขนาดของเงื่อนไขบังคับโดยรวมได้

การคำนวณฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่ยังไม่มีการกำหนดขนาดของเงื่อนไขบังคับโดยรวมในขั้นตอนการเตรียมข้อมูลนั้นใช้สำหรับการค้นคืนข้อมูลอนุกรมเวลาที่มีการกำหนดขนาดของเงื่อนไขบังคับโดยรวมในขั้นตอนของการค้นข้อมูล ซึ่งวิธีนี้จะต้องทำการแปลงข้อมูลสอบถามเสียก่อน โดยวิธีการคำนวณค่าฟังก์ชันดังกล่าวจะกล่าวโดยละเอียดดังต่อไปนี้



รูปที่ 3.9 ตัวอย่างการแปลงข้อมูลสอบถาม Q ด้วยวิธี LB_Keogh ให้เป็นขอบเขตช่วงบน q_u และช่วงล่างของข้อมูล q_l

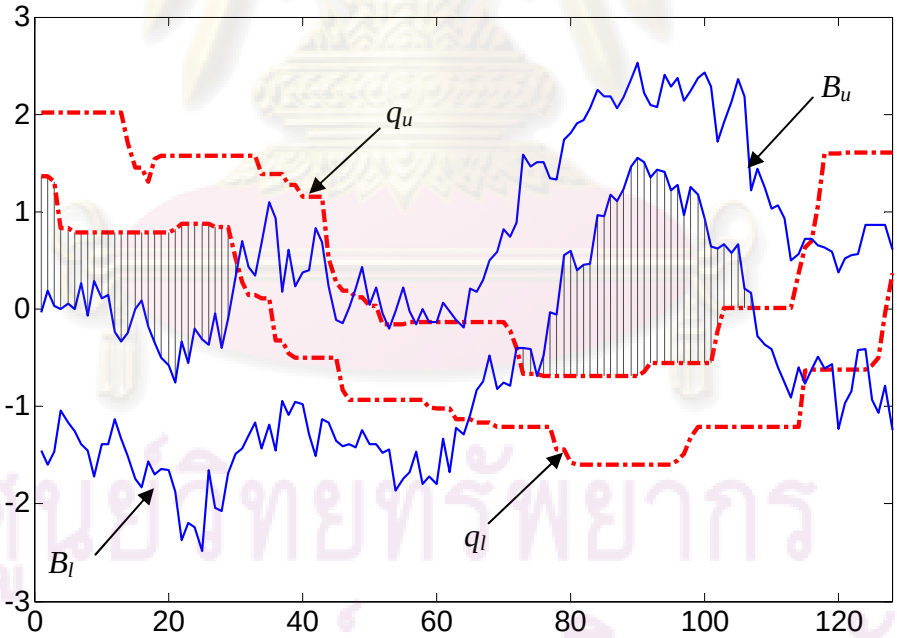
กำหนดให้ข้อมูลอนุกรมสอบถาม Q มีความยาว N โดยประกอบด้วยจุดข้อมูล $q_1, q_2, q_3, \ldots, q_N$ และมีการกำหนดขนาดของเงื่อนไขบังคับโดยรวมสำหรับการค้นหาข้อมูล Q มีความกว้างเท่ากับ r การคำนวณฟังก์ชันขอบเขตล่างนั้นจะต้องทำการแปลงข้อมูลเป็นรูปแบบของขอบเขตของข้อมูลภายใต้เงื่อนไขบังคับโดยรวมด้วยวิธี LB_Keogh โดยจะได้ผลของการแปลงข้อมูลเป็นขอบเขตช่วงบน $q_u = \{q_{u1}, q_{u2}, \ldots, q_{uN}\}$ และขอบเขตช่วงล่าง $q_l = \{q_{l1}, q_{l2}, \ldots, q_{lN}\}$ ของแต่ละจุดบนข้อมูลอนุกรมเวลาได้จากสมการ (3.5) รูปที่ 3.9 แสดงตัวอย่าง

ขอบเขตของข้อมูลสอบถามที่ได้จากการแปลงด้วยวิธี LB_Keogh โดยเส้นประเส้นบนแทนขอบเขตช่วงบน q_u ส่วนเส้นประเส้นล่างแทนขอบเขตช่วงล่าง q_l และเส้นทึบแทนตัวข้อมูลสอบถาม Q

$$q_{ui} = \max_{j=\max(0,i-r)}^{\min(i+r,N)} (q_j)$$
$$q_{li} = \min_{j=\max(0,i-r)}^{\min(i+r,N)} (q_j) \quad (3.5)$$

ในการคำนวณฟังก์ชันขอบเขตล่างดังกล่าวจะทำการหาระยะทางแบบยุคลิดระหว่างขอบเขตของข้อมูลสอบถาม q_u และ q_l กับกล่องขอบเขตของกลุ่มข้อมูล B_u และ B_l จากที่ได้จากการคำนวณในขั้นตอนการเตรียมข้อมูลในหัวข้อที่ 3.3.2.1 โดยกำหนดให้ฟังก์ชัน $LBG(q_u, q_l, B_u, B_l)$ เป็นฟังก์ชันขอบเขตล่างดังกล่าว ซึ่งคำนวณได้จากสมการ (3.6)

$$LBG(q_u, q_l, B_u, B_l) = \sum_{i=1}^{N} \begin{cases} (q_{ui} - B_{li})^2 & \text{if } q_{ui} < B_{li} \\ (q_{li} - B_{ui})^2 & \text{if } q_{li} > B_{ui} \\ 0 & \text{otherwise} \end{cases} \quad (3.6)$$



รูปที่ 3.10 ตัวอย่างการคำนวณฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่สามารถกำหนดขนาดของเงื่อนไขบังคับโดยรวมได้ในแต่ละข้อมูลสอบถาม ซึ่งทำการคำนวณกับขอบเขตบน q_u และขอบเขตล่าง q_l ของข้อมูลสอบถาม รวมถึงขอบเขตบน B_u และขอบเขตล่าง B_l ของกลุ่มข้อมูล

รูปที่ 3.10 แสดงตัวอย่างในการคำนวณฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่ใช้การกำหนดขนาดของเงื่อนไขบังคับโดยรวมบนข้อมูลสอบถาม โดยให้เส้นประแทนขอบเขตของข้อมูลสอบถาม เส้นประเส้นบนและล่างแทนขอบเขตช่วงบน q_u และขอบเขตช่วงล่าง q_l ตามลำดับ และให้เส้นทึบแทนกล่องขอบเขตของกลุ่มข้อมูล โดยที่เส้นทึบเส้นบนและล่างแทนกล่องขอบเขตช่วงบน B_u และกล่องขอบเขตช่วงล่าง B_l ตามลำดับ ดังนั้นจะสามารถคำนวณค่าฟังก์ชันขอบเขตล่างของกลุ่มข้อมูลได้เป็นขนาดของพื้นที่แรเงาซึ่งเป็นส่วนของพื้นที่ที่ขอบเขตของข้อมูลสอบถามอยู่นอกกล่องขอบเขตของกลุ่มข้อมูล

3.4.2 การจัดลำดับการค้นข้อมูล

Algorithm 1 : *IndexedSequentialSearch*(*Q*,*IndexList*,*GroupList*)

```
1:  If no global constraint is assigned in IndexList then
2:    Generate LB_Keogh's envelope from Q
3:  EndIf
4:
5:  Queue SearchList
6:  %Each node in SearchList consists of (pointer,lowerBound)
7:
8:  For i = 1 to IndexList.length do
9:    dist := LBG(Q,IndexList[i])
10:   Add (i,dist) to SearchList
11: EndFor
12:
13: Sort SearchList by dist
14:
15: Foreach node in SearchList do
16:   If node.lowerBound > best-so-far then
17:      Return best-match
18:   EndIf
19:
20:   SequentialSearch(Q,GroupList[node.pointer])
21:   update best-so-far
22: EndFor
```

รูปที่ 3.11 รหัสเทียมสำหรับการค้นคืนข้อมูลแบบลำดับโดยใช้ดัชนี

ในส่วนการค้นคืนข้อมูลตามความคล้ายจากชุดข้อมูลที่ผ่านการจับกลุ่มแบบเคมีนดังที่ได้อธิบายในหัวข้อที่ 3.3.1 งานวิจัยนี้ได้เสนอวิธีการค้นข้อมูลในรูปแบบของการเข้าถึงข้อมูลแบบลำดับโดยใช้ดัชนี โดยการจัดลำดับการเข้าถึงกลุ่มข้อมูลจากฟังก์ชันขอบเขต

ล่างที่ได้นำเสนอในหัวข้อที่ 3.4.1 ลำดับการเข้าถึงข้อมูลนั้นจะเรียงจากกลุ่มข้อมูลที่ได้ค่าฟังก์ชันขอบเขตล่างที่น้อยไปยังกลุ่มข้อมูลที่ได้ค่ามาก และทำการค้นตามลำดับ (Sequential Search) ในแต่ละกลุ่มข้อมูล นอกจากนี้ค่าที่ได้จากฟังก์ชันขอบเขตล่างของกลุ่มข้อมูลยังเป็นเกณฑ์ในการยุติการค้นข้อมูลและทำการคืนคำตอบจากการค้นข้อมูล รูปที่ 3.11 แสดงรหัสเทียมของฟังก์ชัน *IndexedSequentialSearch*(*Q*,*IndexList*,*GroupList*) ซึ่งอธิบายการทำงานของฟังก์ชันการค้นคืนข้อมูลแบบลำดับโดยใช้ดัชนี โดยกำหนดให้มีพารามิเตอร์ตัวแรก *Q* แทนข้อมูลสอบถามสำหรับการค้น พารามิเตอร์ตัวที่สอง *IndexList* แทนรายการของดัชนีตัวแทนกลุ่มข้อมูลทั้งหมด ซึ่งก็คือขอบเขตของแต่ละกลุ่มข้อมูลที่ได้จากการเตรียมข้อมูลในหัวข้อที่ 3.3.2 และให้พารามิเตอร์ตัวสุดท้าย *GroupList* แทนรายการของกลุ่มข้อมูลทั้งหมดที่ได้จากการจับกลุ่มแบบเคมีนกับชุดข้อมูลดังที่ได้กล่าวไว้ในหัวข้อที่ 3.3.1

จากรหัสเทียมในรูปที่ 3.11 บรรทัดที่ 1 ถึง 3 แสดงการแปลงข้อมูลสอบถามให้อยู่ในรูปแบบของ LB_Keogh สำหรับกรณีที่ไม่มีการกำหนดขนาดของเงื่อนไขบังคับโดยรวมตั้งแต่ในขั้นตอนการเตรียมข้อมูล ดังที่ได้กล่าวรายละเอียดไปแล้วในหัวข้อที่ 3.3.2.1 บรรทัดที่ 5 ถึง 6 เป็นการประกาศรายการ *SearchList* สำหรับจัดเก็บลำดับการค้นข้อมูล โดยแต่ละโหนดนั้นจะประกอบด้วย *pointer* และ *lowerBound* ตามลำดับ โดยที่ *pointer* แทนตัวชี้ไปยังกลุ่มข้อมูล และ *lowerBound* เป็นค่าฟังก์ชันขอบเขตล่างของกลุ่มข้อมูลนั้น จากนั้นในบรรทัดที่ 7 ถึง 10 จะทำการคำนวณค่าฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของแต่ละกลุ่มข้อมูลจนครบทุกกลุ่มข้อมูล แล้วในบรรทัดที่ 12 จะทำการเรียงลำดับกลุ่มข้อมูล *SearchList* จากค่า *lowerBound* โดยเรียงจากน้อยไปมาก จากนั้นในบรรทัดที่ 14 ถึง 17 จะทำการค้นหาข้อมูลในแต่ละกลุ่มข้อมูลโดยค้นตามลำดับที่ได้จัดเรียงไว้ใน *SearchList* ซึ่งการค้นในแต่ละกลุ่มข้อมูลจะใช้วิธีการค้นตามลำดับภายในกลุ่มข้อมูลดังแสดงในบรรทัดที่ 19 โดยจะกล่าวโดยละเอียดให้หัวข้อต่อไป หลังจากทำการค้นกลุ่มข้อมูลหนึ่ง ๆ เสร็จ ในบรรทัดที่ 20 จะทำการปรับค่า *best-so-far* สำหรับการค้นให้เป็นปัจจุบัน แต่ก่อนที่จะทำการค้นในแต่ละกลุ่มข้อมูล บรรทัดที่ 15 ถึง 17 จะทำการตรวจสอบว่าค่าที่ได้จากฟังก์ชันขอบเขตล่างที่น้อยที่สุดใน *SearchList* นั้นมีค่าเกินกว่าค่า *best-so-far* หรือไม่ ถ้าเกินกว่านั้นจะยุติการค้นแล้วทำการคืนผลของการค้นคืนข้อมูลเป็นข้อมูลที่มีระยะทางน้อยที่สุดจากที่ได้ทำการค้นมาทั้งหมดเป็นคำตอบ เนื่องจากกลุ่มที่ยังคงหลงเหลืออยู่ใน *SearchList* นั้นจะต้องมีค่าขอบเขตล่างที่มากกว่า *best-so-far* เสมอ ดังนั้นจึงสามารถตัดทอนกลุ่มข้อมูลเหล่านี้โดยไม่ต้องทำการเข้าถึงข้อมูลเลยแม้แต่น้อย

3.4.3 การค้นตามลำดับภายในแต่ละกลุ่มข้อมูล

ในการทำการค้นหาข้อมูลจากในแต่ละกลุ่มข้อมูลนั้น งานวิจัยนี้เสนอให้ทำการค้นตามลำดับโดยใช้ฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิงในการลดการคำนวณไดนามิกไทม์วอร์ปปิงลง สาเหตุที่เสนอให้ทำการค้นตามลำดับแทนการค้นด้วยดัชนีอีกรอบ

เนื่องจากหากการจับกลุ่มข้อมูลนั้นมีประสิทธิภาพสูง จะทำให้ได้กลุ่มข้อมูลที่สมาชิกในกลุ่มเดียวกันมีความคล้ายคลึงกันสูง ดังนั้นการทำดัชนีจึงไม่อาจระบุเฉพาะเจาะจงไปยังข้อมูลส่วนใดส่วนหนึ่งได้ จึงเสนอให้ทำการค้นตามลำดับจะเป็นการดีที่สุด

```
Algorithm 2 : SequentialSearch(Q,DataGroup)
 1:   Foreach C in DataGroup do
 2:     If lowerBound(Q,C) < best-so-far then
 3:        dist := DTW(Q,C)
 4:        If dist < best-so-far then
 5:           best-match := C
 6:           best-so-far := dist
 7:        EndIf
 8:     EndIf
11:   EndFor
```

รูปที่ 3.12 รหัสเทียมสำหรับการค้นคืนข้อมูลจากในกลุ่มข้อมูลตามลำดับ

การค้นตามลำดับในแต่ละกลุ่มข้อมูลสามารถเลือกใช้ฟังก์ชันขอบเขตล่างวิธีใดก็ได้ ไม่ว่าจะเป็นการประมาณระยะทางขอบเขตล่างด้วยระยะทางแบบยุคลิดของ Keogh และคณะ หรือว่าเป็นการประมาณระยะทางขอบเขตล่างด้วยการคำนวณไดนามิกไทม์วอร์ปปิงโดยทำการลดความละเอียดของข้อมูลลงของ Sakurai และคณะ ดังที่ได้กล่าวรายละเอียดไว้ในบทที่ 2 กำหนดให้มีฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิงใด ๆ *lowerBound*(*Q*,*C*) ซึ่งเป็นค่าขอบเขตล่างของระยะทางไดนามิกไทม์วอร์ปปิงระหว่าง *Q* กับ *C* จะสามารถนิยามฟังก์ชัน *SequentialSearch*(*Q*,*DataGroup*) ดังแสดงในรูปที่ 3.12 ซึ่งเป็นฟังก์ชันการค้นตามลำดับภายในกลุ่มข้อมูล โดย *Q* แทนข้อมูลสอบถามและ *DataGroup* แทนกลุ่มข้อมูลเริ่มจากการเข้าถึงข้อมูลใด ๆ *C* ภายในกลุ่มข้อมูล *DataGroup* ตามลำดับ บรรทัดที่ 2 จะทำการคำนวณค่าขอบเขตล่างระหว่าง *Q* และ *C* ถ้าค่าระยะทางขอบเขตล่างมีค่าไม่เกิน *best-so-far* จึงทำการคำนวณค่าระยะทางไดนามิกไทม์วอร์ปปิง *DTW*(*Q*,*C*) ในบรรทัดที่ 3 และถ้าค่าระยะทางไดนามิกไทม์วอร์ปปิงจริงมีค่าน้อยกว่าค่า *best-so-far* จึงทำการบันทึกข้อมูล *C* เป็นคำตอบที่ดีที่สุดในขณะนี้

3.5 วิธีการจับกลุ่มที่ใช้ในการจัดเตรียมข้อมูลสำหรับการทำดัชนีการค้นคืนข้อมูล

การเลือกใช้วิธีการจับกลุ่มข้อมูลนั้นควรเลือกวิธีที่ให้ผลลัพธ์ของการจับกลุ่มตรงตามวัตถุประสงค์ที่ต้องการจากการจับกลุ่มข้อมูล โดยในการทำดัชนีสำหรับการค้นคืนข้อมูลที่ได้นำเสนอนั้น สิ่งที่การค้นคืนข้อมูลด้วยดัชนีนำไปใช้จากการจับกลุ่มนั้นมีเพียงกล่องขอบเขต (Bounding Box) ของแต่ละกลุ่มข้อมูล ดังนั้นการจับกลุ่มที่มีประสิทธิภาพสำหรับงานวิจัยนี้คือ

การจับกลุ่มที่ทำให้กล่องขอบเขตของทุกกลุ่มข้อมูลมีความกระชับกับตัวข้อมูลภายในกลุ่มมากที่สุด กล่องขอบเขตของกลุ่มข้อมูลควรบอกคุณลักษณะของข้อมูลภายในกลุ่มได้อย่างชัดเจน และสามารถนำไปแยกความแตกต่างระหว่างข้อมูลภายในกลุ่มเดียวกันกับข้อมูลที่อยู่ต่างกลุ่มข้อมูลกันได้เป็นอย่างดี งานวิจัยนี้จึงได้นำเสนอวิธีการจับกลุ่มที่จะใช้ในการจัดเตรียมข้อมูลสำหรับการทำดัชนีการค้นคืนข้อมูลทั้งหมด 2 วิธี ได้แก่ วิธีการจับกลุ่มแบบเคมีนซึ่งเป็นวิธีที่พบเห็นได้ทั่วไปในงานด้านการจับกลุ่มข้อมูล ซึ่งน่าจะเป็นวิธีที่ให้ผลลัพธ์ของการจับกลุ่มที่ตรงตามวัตถุประสงค์ของงานวิจัยนี้ เนื่องจากผลลัพธ์ของการจับกลุ่มแบบเคมีนที่ได้นั้น ข้อมูลภายในกลุ่มข้อมูลหนึ่ง ๆ จะมีระยะทางจากจุดศูนย์กลางของข้อมูลหรือค่าเฉลี่ยของข้อมูลน้อยที่สุด ด้วยเหตุนี้จึงทำให้ขนาดของกล่องขอบเขตของแต่ละกลุ่มข้อมูลมีขนาดเล็กที่สุดด้วย นอกเหนือจากการนำวิธีการจับกลุ่มแบบเคมีนไปใช้แล้ว งานวิจัยนี้ยังเสนอวิธีการลดทอนการคำนวณในการจับกลุ่มแบบเคมีนที่สามารถเพิ่มความเร็วในการจับกลุ่มได้อย่างดีเยี่ยม และยังทำให้ระบบการค้นคืนข้อมูลด้วยวิธีที่นำเสนอสามารถรองรับกับการเปลี่ยนแปลงของข้อมูลได้เป็นอย่างดี ไม่ว่าจะเป็นการเพิ่มข้อมูลหรือการลบข้อมูลออกจากชุดข้อมูล นอกจากนี้งานวิจัยนี้ยังได้นำเสนอวิธีการจับกลุ่มวิธีใหม่ได้คิดค้นขึ้นมาใหม่เพื่อเน้นในด้านความเร็วในการคำนวณการจับกลุ่ม ทำให้สามารถรองรับกับการจัดเตรียมข้อมูลบนชุดข้อมูลขนาดใหญ่มาก โดยยังคงได้ผลลัพธ์ที่ตรงตามวัตถุประสงค์ของงานวิจัยนี้ นั่นก็คือวิธีการจับกลุ่มแบบกล่องขอบเขต (Bounding Box Clustering)

3.5.1 การจับกลุ่มแบบเคมีนภายใต้การลดทอนการคำนวณ

ในขั้นการเตรียมข้อมูล จะทำการจับกลุ่มข้อมูลทั้งหมดในชุดข้อมูลด้วยวิธีการจับกลุ่มแบบเคมีนโดยใช้วิธีวัดระยะทางของข้อมูลแบบยุคลิดในการจับกลุ่ม แต่เนื่องจากในงานวิจัยนี้มุ่งเน้นในด้านการจัดการกับข้อมูลอนุกรมเวลาซึ่งเป็นข้อมูลที่มีจำนวนมิติสูง ดังนั้นการวัดระยะทางแบบยุคลิดแต่ละครั้งนั้นมีขีดจำกัดเชิงสัญกรณ์ในด้านเวลาอยู่ในระดับฟังก์ชันเชิงเส้นของจำนวนมิติของข้อมูล เมื่อคำนึงไปถึงความซับซ้อนทั้งหมดสำหรับการจับกลุ่มแล้วอาจมีขีดจำกัดเชิงสัญกรณ์สูงถึงระดับฟังก์ชันพหุนามกำลังสองของจำนวนครั้งที่ต้องทำการวัดระยะทางของข้อมูล ถึงแม้ว่าในการจัดเตรียมข้อมูลนั้นสามารถจัดเตรียมการประมวลผลไว้ก่อนการทำการค้นคืนข้อมูลได้ แต่เพื่อเป็นการเพิ่มประสิทธิภาพของงานวิจัยนี้ จึงได้เสนอวิธีการเพิ่มความเร็วในการจับกลุ่มข้อมูลอนุกรมเวลาด้วยแนวคิดที่จะนำเสนอเป็นรหัสเทียมดังแสดงในรูปที่ 3.13 ในรหัสเทียมดังกล่าวได้นำเสนอฟังก์ชัน *clustering*(*Dataset*,*MeanList*) ซึ่งเป็นฟังก์ชันในการจับกลุ่มแบบเคมีนที่รวดเร็วยิ่งขึ้นจากการลดจำนวนครั้งในการคำนวณระยะทางยุคลิด และการลดเวลาในการคำนวนระยะทางแต่ละครั้ง โดยกำหนดค่าพารามิเตอร์คือ *Dataset* แทนแหล่งเก็บชุดข้อมูลทั้งหมด และ *MeanList* แทนค่าเฉลี่ยเริ่มต้นของกลุ่มข้อมูลโดยได้จากการสุ่มเลือกข้อมูลจากใน *Dataset* ซึ่งจำนวนข้อมูลที่ได้จากการสุ่มเลือกมาจะบ่งบอกถึงจำนวนกลุ่มที่จะเป็นผลลัพธ์จากการจับกลุ่ม

เริ่มต้นจากการกำหนดค่าเฉลี่ยของแต่ละกลุ่มข้อมูลไว้ใน *MeanList* ซึ่งได้มาจากการสุ่มข้อมูลขึ้นมา *k* ตัว โดย *k* คือจำนวนกลุ่มที่จะทำการจับกลุ่ม จากนั้นจะทำการจำแนกข้อมูลทั้งหมดจากใน *Dataset* ไปยังกลุ่มข้อมูลทั้ง *k* กลุ่ม วิธีการจำแนกข้อมูลแต่ละตัวนั้นแสดงในบรรทัดที่ 5 ถึง 15 โดยจะทำการวัดระยะทางแบบยุคลิดระหว่างข้อมูลหนึ่ง ๆ จากใน *Dataset* กับค่าเฉลี่ยของทุกกลุ่มข้อมูลจากใน *MeanList* และจะทำการจำแนกข้อมูลนั้นไปยังกลุ่มข้อมูลที่มีระยะทางจากค่าเฉลี่ยของกลุ่มน้อยที่สุด แต่เพื่อเป็นการลดจำนวนครั้งในการคำนวณระยะทางแบบยุคลิด ในบรรทัดที่ 7 ได้กำหนดว่าจะทำการคำนวณระยะทางกับค่าเฉลี่ยของกลุ่มเฉพาะในกรณีที่ในกลุ่มข้อมูลนั้น (*group*[*i*]) เกิดการเปลี่ยนแปลงของสมาชิกในกลุ่มจากการจำแนกข้อมูลในรอบก่อนหน้า หรือในกรณีที่กลุ่มข้อมูลที่ข้อมูลนั้นถูกจำแนกไว้ในการจำแนกรอบก่อนหน้า (*group*[*p*]) เกิดการเปลี่ยนแปลงสมาชิกภายในกลุ่ม ซึ่งสามารถลดทอนจำนวนครั้งในการคำนวณระยะทางกับค่าเฉลี่ยของกลุ่มข้อมูลตามที่กล่าวไว้ในทฤษฎีบทที่ 1

***Algorithm* 3 : *Clustering*(*Dataset*,*MeanList*)**

```
 1:  While at least 1 sequence from MeanList has been changed do
 2:    Foreach C in Dataset do
 3:      best-so-far := previous best-so-far of C
 4:      nearest_group := null
 5:      For i = 1 to k do
 6:        p := the group that C recently contained in
 7:        If group[i] or group[p] has been changed then
 8:          D := EarlyAbandon(C,MeanList[i],best-so-far)
 9:          If D < best-so-far then
10:            best-so-far := D
11:            nearest_group := i
12:          EndIf
13:        EndIf
14:      EndFor
15:      Move C to group[nearest_group]
16:    EndFor
17:    For i = 1 to k where the i^th group has been changed do
18:      Recalculate the mean of the i^th group into MeanList[i]
19:    EndFor
20:  EndWhile
21:  Return group
```

รูปที่ 3.13 รหัสเทียมสำหรับวิธีการจับกลุ่มข้อมูลอนุกรมเวลาแบบเคมีน

ทฤษฎีบทที่ 1 *การจำแนกข้อมูลหนึ่งในแต่ละรอบการทำงานสำหรับการจับกลุ่มแบบเคมีน ในกรณีที่กลุ่มที่ข้อมูลนั้นถูกจำแนกไว้ในรอบก่อนหน้าไม่มีการเปลี่ยนแปลงของสมาชิกภายในกลุ่มนั้น เป็นเหตุให้ทำให้ข้อมูลนั้นไม่จำเป็นต้องพิจารณาการจำแนกข้อมูลไปยังกลุ่มข้อมูลอื่นที่ไม่เกิดการเปลี่ยนแปลงของข้อมูลภายในกลุ่มเช่นกัน*

พิสูจน์

กำหนดให้การดำเนินไปของการจับกลุ่มอยู่ที่การจำแนกข้อมูล x ไปยังกลุ่มใดกลุ่มหนึ่งใน k กลุ่ม โดยที่ x ได้ถูกจำแนกไปยังกลุ่มข้อมูลที่ p ในการจำแนกข้อมูลรอบก่อนหน้า และกำหนดให้ $D[i]$ เป็นระยะทางระหว่างข้อมูล x กับค่าเฉลี่ยของกลุ่มข้อมูลที่ i ในรอบการจำแนกก่อนหน้า และเนื่องจากจากนิยามในการจำแนกกลุ่มของการจับกลุ่มแบบเคมีนซึ่งจะทำการจำแนกข้อมูลแต่ละตัวไปยังกลุ่มข้อมูลที่มีค่าเฉลี่ยของกลุ่มข้อมูลอยู่ใกล้กับข้อมูลนั้นมากที่สุด อีกนัยหนึ่งคือระยะทางระหว่างข้อมูลนั้นกับค่าเฉลี่ยของกลุ่มข้อมูลที่จะทำการจำแนกข้อมูลนั้นไปยังกลุ่มดังกล่าวจะต้องน้อยที่สุดเมื่อเทียบกับระยะทางกับค่าเฉลี่ยของกลุ่มอื่น จะได้ว่า

$$\forall i \neq p \rightarrow D[p] \leq D[i] \tag{3.7}$$

กำหนดให้ $D'[i]$ เป็นระยะทางระหว่างข้อมูล x กับค่าเฉลี่ยของกลุ่มข้อมูลที่ i ในรอบการจำแนกปัจจุบัน และจากที่ได้กำหนดไว้ในทฤษฎีบทนี้ว่าทุกกลุ่มข้อมูล q ใด ๆ และกลุ่มข้อมูลที่ข้อมูล x ถูกจำแนกไว้ก่อนหน้า หมายความว่ากลุ่มข้อมูลที่ p ไม่เกิดการเปลี่ยนแปลงสมาชิกภายในกลุ่ม ดังนั้นค่าเฉลี่ยของกลุ่มข้อมูลที่ p และ q ก็จะไม่เกิดการเปลี่ยนแปลงเช่นกัน จึงได้ว่า

$$\begin{aligned} D'[p] &= D[p] \\ \forall q \rightarrow D'[q] &= D[q] \end{aligned} \tag{3.8}$$

และจากสมการ (3.7) จะได้ว่า

$$\forall q \rightarrow D'[p] \leq D'[q] \tag{3.9}$$

ดังนั้นข้อมูล x จึงสามารถแทนที่การจำแนกกลุ่มไปยังกลุ่มที่ q ใด ๆ ด้วยการจำแนกไปยังกลุ่มที่ p เนื่องจากค่าระยะทางระหว่าง x กับค่าเฉลี่ยของกลุ่มที่ p นั้นไม่มากกว่าค่าระยะทางจากค่าเฉลี่ยของทุกกลุ่ม q จึงสามารถสรุปได้ว่า การจำแนกกลุ่มข้อมูล x นั้นไม่จำเป็นต้องพิจารณาที่จะจำแนกไปยังกลุ่ม q ใด ๆ ซึ่งยังคงเป็นไปตามนิยามการจับกลุ่มแบบเคมีนตามสมการ (3.7)

ในการจำแนกข้อมูลแต่ละตัวสำหรับในแต่ละรอบของการจับกลุ่มแบบเคมีน การวัดระยะทางกับค่าเฉลี่ยของทุกกลุ่มข้อมูลเพียงเพื่อหากลุ่มข้อมูลที่ได้ระยะทางน้อยที่สุดเพื่อทำการจำแนกข้อมูลนั้นไปยังกลุ่มที่มีระยะทางน้อยที่สุด ดังนั้นในงานวิจัยนี้จึงได้นำเสนอฟังก์ชัน *EarlyAbandon*(*C, Q, best-so-far*) ซึ่งเป็นฟังก์ชันที่สามารถลดเวลาในการคำนวณระยะทางกำลังสองแบบยุคลิดระหว่างข้อมูล C และ Q โดยกำหนดให้ Q มีความยาว N ประกอบด้วยจุดข้อมูล $q_1, q_2, q_3, \ldots, q_N$ และ C มีความยาว N ประกอบด้วยจุดข้อมูล $c_1, c_2, c_3, \ldots, c_N$ โดยค่าระยะทางที่จะนำไปใช้ควรมีค่าไม่เกิน *best-so-far* การคำนวณฟังก์ชัน *EarlyAbandon*(*C, Q, best-so-far*) สามารถนิยามได้จากรหัสเทียมในรูปที่ 3.14 โดยในบรรทัดที่ 7 ถึง 9 จะทำการหยุดการคำนวณค่าระยะทางถ้าผลรวมของค่าระยะทางกำลังสองของแต่ละคู่จุดข้อมูลในแต่ละมิติมีค่าเกินกว่าค่าระยะทางที่น้อยที่สุดตามที่ฟังก์ชันต้องการ *best-so-far* และคืนค่าระยะทางที่เป็นอนันต์เป็นผลของการวัดระยะทาง เนื่องจากถึงแม้จะทำการคำนวณต่อจนจบ ผลที่ได้จะต้องมีค่าไม่น้อยกว่าค่า *best-so-far* อย่างแน่นอน และกลุ่มนั้นก็จะถูกละทิ้งไป เนื่องจากการวัดระยะทางแบบยุคลิดที่ใช้ในการจับกลุ่มข้อมูลแบบเคมีนนั้นใช้สำหรับเพียงแค่หากลุ่มที่ใกล้กับข้อมูลนั้นมากที่สุด โดยสรุปแล้วจึงสามารถใช้ค่า *best-so-far* มาช่วยลดการคำนวณลงได้

Algorithm 4 : *EarlyAbandon*(*C, Q, best-so-far*)

```
1:   dist := 0
2:
3:   For i = 1 to N
4:     dist := dist + (c_i − q_i)^2
5:     If dist ≥ best-so-far then
6:       Return INFINITY
7:     EndIf
8:   EndFor
9:   Return dist
```

รูปที่ 3.14 รหัสเทียมสำหรับฟังก์ชันการลดการคำนวณระยะทางแบบยุคลิด

ในทุกครั้งที่ทำการจำแนกข้อมูลใด ๆ แล้วทำให้ข้อมูลนั้นเกิดการเปลี่ยนกลุ่มไปจากเดิม บรรทัดที่ 16 และ 18 ของรหัสเทียมในรูปที่ 3.13 จะทำการบันทึกว่าทั้งกลุ่มที่ข้อมูลนั้นเคยถูกจำแนกไว้ก่อนหน้าและกลุ่มที่ถูกจำแนกใหม่ได้เกิดการเปลี่ยนแปลงของสมาชิกภายในกลุ่มขึ้น หลังจากทำการจำแนกข้อมูลทุกตัวไปยังกลุ่ม k กลุ่มจนหมดแล้ว ในบรรทัดที่ 19 ถึง 21 จะทำการคำนวณค่าเฉลี่ยของข้อมูลภายในกลุ่มใหม่สำหรับทุกกลุ่มข้อมูลที่เกิดการเปลี่ยนแปลงสมาชิกภายในกลุ่ม และทำการวนซ้ำกลับไปทำงานที่บรรทัดที่ 3 ใหม่ คือการจำแนกข้อมูลทุกตัวใหม่อีกรอบ จนกว่าจะไม่มีกลุ่มข้อมูลในที่เกิดการเปลี่ยนแปลงสมาชิกภายในกลุ่มเลย เท่านี้ก็เป็นการเสร็จสิ้นการจับกลุ่มข้อมูล

แม้ว่าการลดทอนการคำนวณในการจับกลุ่มแบบเคมีนนั้นจะสามารถลดเวลาในการคำนวณได้ในระดับหนึ่ง แต่เนื่องจากการจับกลุ่มแบบเคมีนบนกลุ่มข้อมูลขนาดใหญ่นั้นอาจต้องใช้เวลาในการคำนวณสูงมากจนไม่สามารถประมาณเวลาที่ต้องใช้ได้ ซึ่งอาจใช้เวลามากจนไม่เหมาะที่จะนำไปใช้งานจริงบนการค้นคืนข้อมูลจากชุดข้อมูลขนาดใหญ่มาก ดังนั้นงานวิจัยนี้จึงได้เสนอวิธีการจับกลุ่มวิธีใหม่ที่เหมาะกับงานวิจัยนี้โดยเฉพาะ อีกทั้งยังสามารถคำนวณได้อย่างรวดเร็วกว่าการจับกลุ่มแบบเคมีนมาก ขอเรียกการจับกลุ่มวิธีใหม่นี้ว่าการจับกลุ่มแบบแทรก (Insertion Clustering)

3.5.2 การจับกลุ่มแบบแทรก (Insertion Clustering)

การจับกลุ่มแบบแทรกเป็นการจับกลุ่มวิธีใหม่สำหรับขั้นการเตรียมข้อมูลก่อนการทำดัชนีสำหรับการค้นคืนที่ได้เสนอในงานวิจัยนี้โดยเฉพาะ แนวคิดของการจับกลุ่มด้วยวิธีนี้จะเน้นไปที่การทำให้ขนาดของกล่องขอบเขตของแต่ละกลุ่มข้อมูลมีขนาดเล็กที่สุดด้วยการรวมเอาข้อมูลหลาย ๆ ตัวเข้าด้วยกันโดยที่ข้อมูลแต่ละตัวที่ถูกรวมเข้าในกลุ่มเดียวกันจะต้องทำให้ขนาดของกล่องขอบเขตของกลุ่มข้อมูลขยายใหญ่ขึ้นน้อยที่สุด ด้วยเหตุนี้งานวิจัยนี้จึงได้เสนอมาตรวัดในรูปแบบใหม่ที่สามารถบ่งบอกถึงการขยายตัวของกลุ่มข้อมูลจากการรวมข้อมูลเพิ่มเข้าไปแต่ละตัว โดยจะกล่าวโดยละเอียดดังต่อไปนี้

กำหนดให้มีกลุ่มของข้อมูลอนุกรมเวลา C ที่ข้อมูลแต่ละตัวมีความยาว N มีกล่องขอบเขตที่ประกอบด้วยขอบเขตบน B_{u1} ซึ่งประกอบไปด้วยจุดข้อมูล B_{u1}, B_{u2}, $B_{u3}, \ldots,$ B_{un} และค่าขอบเขตช่วงล่าง B_l ซึ่งประกอบด้วยจุดข้อมูล B_{l1}, B_{l2}, $B_{l3}, \ldots, B_{ln}$ โดยจะทำการแทรกข้อมูล Q เข้าไปในกลุ่มข้อมูล C โดยกำหนดให้ข้อมูลอนุกรมเวลา Q ซึ่งประกอบด้วยจุดข้อมูล q_1, q_2, q_3, … , q_N ดังนั้นจึงขอกำหนดมาตรวัดที่ใช้บ่งบอกถึงการขยายตัวของกล่องขอบเขต C จากการเพิ่มข้อมูล Q เข้าไปเป็นค่าที่ได้จากฟังก์ชัน $BoundExtension(Q, C)$ โดยฟังก์ชันดังกล่าวมีนิยามการคำนวณดังแสดงในสมการ (3.10)

$$BoundExtension(Q,C) = \sum_{i=0}^{N} \begin{cases} (q_i - B_{li})^2 \text{ if } q_i > B_{ui} \\ (B_{ui} - q_i)^2 \text{ if } q_i < B_{li} \\ 0 \quad \text{otherwise} \end{cases} \tag{3.10}$$

จากนี้จะกล่าวถึงรายละเอียดวิธีการจับกลุ่มแบบแทรก โดยก่อนจะทำการจับกลุ่มด้วยวิธีนี้จะต้องทำการกำหนดค่าพารามิเตอร์อยู่ค่าหนึ่ง นั่นก็คือค่า `PageSize` ซึ่งเป็นค่าที่บ่งบอกว่าแต่ละกลุ่มข้อมูลจะมีจำนวนสมาชิกในกลุ่มไม่เกินค่า `PageSize` ซึ่งการกำหนดค่า `PageSize` ก็จะให้ผลลัพธ์คล้ายกับการกำหนดจำนวนกลุ่มในการจับกลุ่มแบบเคมีนกล่าวคือเมื่อกำหนดค่า `PageSize` ให้มากขึ้น จะทำให้จำนวนกลุ่มจากการจับกลุ่มที่น้อยลง และในทางกลับกันถ้ากำหนดค่า `PageSize` ให้น้อยลงก็ทำให้จำนวนกลุ่มที่ได้เพิ่มขึ้นเช่นกัน

แนวคิดของการจับกลุ่มแบบแทรกจะนำเสนอในรูปของรหัสเทียมดังแสดงในรูปที่ 3.15 โดยเป็นการอธิบายฟังก์ชัน *InsertionClustering*(*Dataset*,*PageSize*) ซึ่งเป็นฟังก์ชันการจับกลุ่มแบบแทรก

Algorithm 5 : *InsertionClustering*(*Dataset*,*PageSize*)

```
 1:   ArrayList groupList
 2:
 3:   Foreach data in Dataset do
 4:     If groupList is empty then add data to groupList[0]
 5:     Else
 6:        mindist := INFINITY
 7:        Foreach group in groupList do
 8:           dist := BoundExtension(data,group)
 9:           If dist < mindist then
10:              mindist := dist
11:              bestGroup := group
12:           EndIf
13:        EndFor
14:        Add data to bestGroup
15:        If bestGroup.size > pageSize then
16:           Split bestGroup into 2 groups
17:        EndIf
18:     EndIf
19:   EndFor
```

รูปที่ 3.15 รหัสเทียมสำหรับฟังก์ชันการจับกลุ่มแบบแทรก

จากรหัสเทียมในรูปที่ 3.15 เป็นการแทรกข้อมูลจากในชุดข้อมูล *Dataset* ทีละข้อมูลเข้าไปในกลุ่มใดกลุ่มหนึ่งใน *groupList* ในบรรทัดที่ 4 เป็นการกำหนดเงื่อนไขเริ่มต้นคือทำการสร้างกลุ่มข้อมูลแรกจากข้อมูลแรกสุดจากใน *Dataset* จากนั้นในบรรทัดที่ 6 ถึง 14 จะทำการเลือกกลุ่มข้อมูลที่จะทำการเพิ่มข้อมูลเข้าไป โดยเลือกจากกลุ่มข้อมูลที่ได้ค่าจากฟังก์ชัน *BoundExtension(data,group)* จากสมการ (3.10) ที่น้อยที่สุด ซึ่งก็คือกลุ่มข้อมูลที่เมื่อเพิ่มข้อมูลตัวใหม่เข้าไปแล้วทำให้ขนาดของกล่องขอบเขตขยายเพิ่มขึ้นน้อยที่สุด หลังจากทำการเพิ่มข้อมูลเข้าไปในกลุ่มข้อมูลดังกล่าวแล้ว ในบรรทัดที่ 15 ถึง 17 จะตรวจสอบว่ากลุ่มข้อมูลนั้นมีสมาชิกในกลุ่มเกินกว่าค่า *PageSize* หรือไม่ ถ้าจำนวนสมาชิกมีค่าเกินให้ทำการแบ่งกลุ่มข้อมูลดังกล่าวออกเป็น 2 กลุ่มด้วยวิธีการจับกลุ่มแบบเคมีน กระบวนการนี้จะวนซ้ำจนกว่าจะทำการเพิ่มข้อมูลเข้าไปจนครบทั้งชุดข้อมูล และจะได้ผลลัพธ์ของกลุ่มข้อมูลทั้งหมดจากการจับกลุ่มแบบแทรกอยู่ใน *groupList*

3.6 การกำหนดค่าพารามิเตอร์สำหรับการจับกลุ่ม

ในขั้นการเตรียมข้อมูลด้วยการจับกลุ่มจากที่ได้กล่าวไว้ในหัวข้อที่ 3.3.1 นั้นมีพารามิเตอร์สำหรับการจับกลุ่มอยู่ค่าหนึ่งนั่นคือจำนวนกลุ่มที่จะทำการจับกลุ่ม ซึ่งจำนวนกลุ่มที่ได้จากการจับกลุ่มนั้นมีผลต่อประสิทธิภาพในการค้นคืนข้อมูลที่ได้นำเสนอในงานวิจัยนี้เป็นอย่างมาก โดยการจับกลุ่มข้อมูลออกเป็นจำนวนกลุ่มที่น้อยเกินไป จะทำให้กล่องขอบเขตของกลุ่มข้อมูลมีขนาดใหญ่เกินไป ซึ่งส่งผลให้ค่าที่ได้จากฟังก์ชันขอบเขตล่างสำหรับระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลมีค่าห่างจากค่าระยะทางไดนามิกไทม์วอร์ปปิงจริงกับข้อมูลภายในกลุ่ม ทำให้ประสิทธิภาพในการตัดทอนข้อมูลลดลง แต่สำหรับการจับกลุ่มด้วยจำนวนกลุ่มที่มากเกินไป จะส่งผลให้ต้องเสียเวลาในการคำนวณฟังก์ชันขอบเขตล่างมากขึ้น เนื่องจากต้องทำการคำนวณระยะทางขอบเขตล่างกับทุก ๆ กลุ่มข้อมูล นอกจากนี้ยังส่งผลให้เกิดความล่าช้าจากการเข้าถึงข้อมูลอีกด้วย เนื่องจากการเข้าถึงกลุ่มข้อมูลแต่ละครั้งจะต้องทำการเข้าถึงข้อมูลโดยสุ่ม (Random Access) ซึ่งก่อให้เกิดความล่าช้าในส่วนของอินพุต / เอาต์พุตอย่างมาก ซึ่งปัญหานี้ยังพบได้ในงานวิจัยที่เสนอวิธีการค้นคืนข้อมูลด้วยดัชนีที่มีโครงสร้างในรูปแบบของต้นไม้

สำหรับการกำหนดค่าพารามิเตอร์สำหรับการจับกลุ่มโดยทั่วไปแล้วมักใช้วิธีการวัดความสมเหตุสมผลของการจับกลุ่ม แต่เนื่องจากการจับกลุ่มที่ใช้กันทั่วไปนั้นมีวัตถุประสงค์ที่แตกต่างจากการจับกลุ่มที่ใช้ในงานวิจัยนี้ ดังนั้นจึงขอนิยามคุณสมบัติของการจับกลุ่มที่ดีสำหรับงานวิจัยนี้ไว้ 2 ข้อดังต่อไปนี้

- กล่องขอบเขตของแต่ละกลุ่มข้อมูลที่ได้จากการจับกลุ่มควรมีขนาดเล็กที่สุด เนื่องจากกล่องขอบเขตที่มีความกระชับกับข้อมูลจะส่งผลให้การประมาณระยะทางขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลมีค่าเข้าใกล้ค่าระยะทางไดนามิกไทม์วอร์ปปิงจริงมากยิ่งขึ้น
- จำนวนกลุ่มข้อมูลควรมีค่าน้อยที่สุด เพื่อลดจำนวนครั้งในการคำนวณฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูล และลดจำนวนครั้งในการเข้าถึงข้อมูลโดยสุ่มจากในแต่ละกลุ่มข้อมูล

สังเกตได้ว่าคุณสมบัติทั้งสองข้อนี้กลับเป็นภาวะถ่วงดุล (Tradeoff) ซึ่งกันและกัน กล่าวคือถ้าต้องการจับกลุ่มให้ได้กล่องขอบเขตของแต่ละกลุ่มข้อมูลที่เล็กลง สามารถทำได้ง่ายเพียงเพิ่มจำนวนกลุ่มข้อมูลในการจับกลุ่มซึ่งก็ไปขัดแย้งกับอีกคุณสมบัติหนึ่ง การปรับพารามิเตอร์สำหรับการจับกลุ่มจึงเป็นการปรับสมดุลระหว่างสองคุณสมบัติดังกล่าวให้เหมาะสมที่สุด ดังนั้นวิธีการวัดความสมเหตุสมผลของการจับกลุ่มนั้นไม่สามารถกำหนดจำนวนกลุ่มที่เหมาะสมได้ เนื่องจากคุณสมบัติของการจับกลุ่มที่ดีนั้นแตกต่างกัน งานวิจัยนี้จึงเสนออีกวิธีหนึ่ง

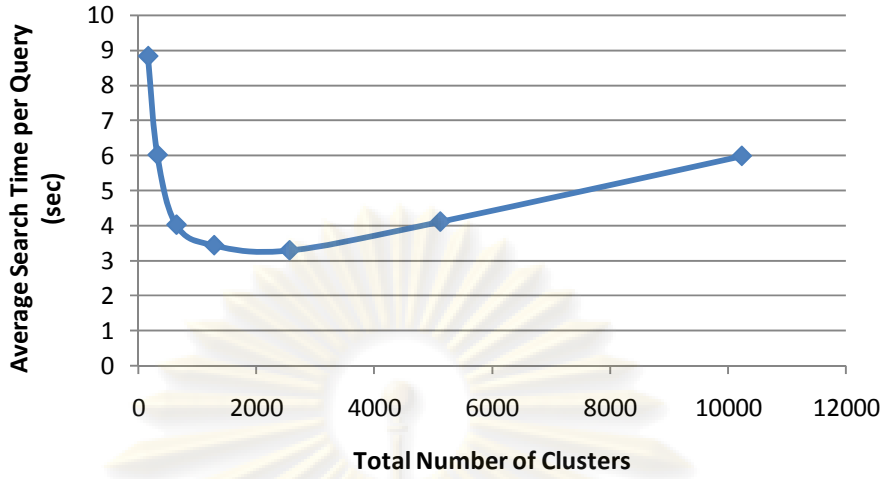
ที่สามารถกำหนดจำนวนกลุ่มที่สามารถปรับเปลี่ยนได้กับทุกรูปแบบของงานได้ นั่นคือวิธีการกำหนดชุดตรวจสอบความสมเหตุสมผล (Validation Set)

วิธีการกำหนดชุดตรวจสอบความสมเหตุสมผลเป็นการทดสอบว่าสามารถนำกลุ่มข้อมูลไปใช้งานตามต้องการได้อย่างมีประสิทธิภาพมากน้อยเพียงใด ในงานวิจัยนี้ได้ใช้วิธีการทดสอบการค้นคืนข้อมูลสำหรับการจับกลุ่มข้อมูลหลาย ๆ รูปแบบ โดยมีขั้นตอนการทำงานดังต่อไปนี้

1. ทำการสุ่มแยกข้อมูลส่วนหนึ่งออกจากชุดข้อมูลมาไว้ในชุดตรวจสอบความสมเหตุสมผล
2. นำข้อมูลที่เหลืออยู่ในชุดข้อมูลมาจับกลุ่มโดยทำการปรับค่าจำนวนกลุ่มตั้งแต่จำนวนกลุ่มน้อย ๆ จนถึงจำนวนกลุ่มมาก ๆ ซึ่งจะได้ชุดของกลุ่มข้อมูลหลาย ๆ ชุด ไว้สำหรับทดสอบว่าชุดของกลุ่มข้อมูลที่มีจำนวนกลุ่มเท่าไรถึงจะดีที่สุด
3. ทำการค้นคืนข้อมูลจากในแต่ละชุดของกลุ่มข้อมูล โดยใช้ชุดตรวจสอบความสมเหตุสมผลเป็นข้อมูลสอบถาม แล้วทำการจดบันทึกเวลาที่ใช้ในการค้นคืนข้อมูลจากในแต่ละชุดของกลุ่มข้อมูล แล้วเลือกชุดของกลุ่มข้อมูลที่ใช้เวลาในการค้นน้อยที่สุดเป็นชุดข้อมูลผลลัพธ์
4. ทำการเพิ่มข้อมูลจากในชุดตรวจสอบความสมเหตุสมผลกลับเข้าไปในชุดข้อมูลผลลัพธ์

เวลาในการค้นคืนข้อมูลจากในแต่ละกลุ่มข้อมูลมักจะมีแนวโน้มคล้ายกับกราฟที่แสดงในรูปที่ 3.16 ซึ่งจุดที่ต่ำที่สุดของกราฟนั้นแสดงว่าไม่ว่าจะทำการเพิ่มหรือลดจำนวนกลุ่มในการจับกลุ่มก็ล้วนแต่มีแนวโน้มที่จะลดทอนประสิทธิภาพในการค้นคืนข้อมูล ดังนั้นจึงได้ว่ากลุ่มข้อมูลที่ให้ผลการค้นคืนดีที่สุดเป็นการจับกลุ่มที่มีประสิทธิภาพ

รูปที่ 3.16 แสดงผลการทดสอบเพื่อหารูปแบบการแบ่งกลุ่มที่เหมาะสมที่สุด ซึ่งมีแนวโน้มที่จะให้ผลจากการค้นคืนข้อมูลได้เร็วที่สุด โดยการนำชุดตรวจสอบความสมเหตุสมผลมาเป็นข้อมูลสอบถามสำหรับค้นคืนข้อมูลจากชุดข้อมูลที่ผ่านการจับกลุ่มด้วยจำนวนกลุ่มข้อมูลที่แตกต่างกันตามที่แสดงในแกนนอน และแกนตั้งแสดงเวลาที่ใช้ในการค้นคืนข้อมูลโดยเฉลี่ยต่อหนึ่งข้อมูลสอบถาม ซึ่งจากกราฟแสดงให้เห็นว่าการจับกลุ่มโดยแบ่งข้อมูลประมาณ 2,000 กลุ่มนั้นให้ผลการค้นที่เร็วที่สุด ดังนั้นเราจึงเลือกที่จะนำกลุ่มข้อมูลดังกล่าวไปใช้งานจริง โดยต้องทำการเพิ่มข้อมูลทั้งหมดจากในชุดตรวจสอบความสมเหตุสมผลกลับเข้าไปในกลุ่มดังกล่าวก่อนเพื่อให้ได้ชุดข้อมูลที่มีข้อมูลครบถ้วนดังเดิม



รูปที่ 3.16 ตัวอย่างแนวโน้มของเวลาที่ใช้ในการค้นคืนจากแต่ละกลุ่มข้อมูลที่มีการปรับเปลี่ยนจำนวนกลุ่มในการจับกลุ่ม

3.7 การรองรับการเปลี่ยนแปลงของชุดข้อมูล

ในหัวข้อนี้จะกล่าวถึงวิธีการจัดการกับชุดข้อมูลในกรณีที่ชุดข้อมูลเกิดการเปลี่ยนแปลง โดยได้นำเสนอวิธีการปรับเปลี่ยนชุดข้อมูลจากการเพิ่มข้อมูลใหม่เข้าไปในชุดข้อมูล และการลบข้อมูลบางตัวออกจากชุดข้อมูล โดยการเปลี่ยนแปลงของข้อมูลอาจก่อให้เกิดการเปลี่ยนแปลงผลของการจับกลุ่มข้อมูล ในงานวิจัยนี้จึงได้นำเสนอการจับกลุ่มใหม่ที่รวดเร็วสำหรับการเพิ่มหรือลบข้อมูลเป็นจำนวนน้อยเมื่อเทียบกับขนาดของชุดข้อมูล โดยใช้วิธีการลดทอนการคำนวณระยะทางสำหรับการจับกลุ่มแบบเคมีนซึ่งได้นำเสนอไปในหัวข้อที่ 3.3.1

3.7.1 การจับกลุ่มใหม่ภายหลังการเพิ่มข้อมูลเข้าไปในชุดข้อมูล

สำหรับการจัดการกับการเพิ่มข้อมูลเข้าไปในชุดข้อมูล หัวข้อนี้จะนำเสนอวิธีการจับกลุ่มใหม่ในรูปแบบของรหัสเทียมของฟังก์ชันการเพิ่มข้อมูลเข้าไปในชุดข้อมูล *Insert*(*InsertList*,*GroupList*,*MeanList*) ดังแสดงในรูปที่ 3.17 โดยกำหนดให้มีพารามิเตอร์ดังต่อไปนี้

- *InsertList* แทนรายการของข้อมูลที่จะทำการเพิ่มเข้าไปในชุดข้อมูล
- *GroupList* แทนรายการของกลุ่มข้อมูล
- *MeanList* แทนรายการของค่าเฉลี่ยของแต่ละกลุ่มข้อมูล

การเพิ่มข้อมูลแต่ละตัวเข้าไปในชุดข้อมูลจะต้องทำการจำแนกข้อมูลเหล่านั้นเข้าไปอยู่ในกลุ่ม ๆ หนึ่งโดยใช้การวัดระยะทางแบบยุคลิดระหว่างข้อมูลนั้นกับค่าเฉลี่ยของกลุ่ม

ข้อมูลเป็นเกณฑ์ในการจำแนก โดยจะจำแนกข้อมูลนั้นไปยังกลุ่มข้อมูลที่ได้ระยะทางดังกล่าวน้อยที่สุด ซึ่งได้แสดงการคำนวณดังกล่าวในรหัสเทียมดังรูปที่ 3.17 บรรทัดที่ 1 ถึง 14 การคำนวณระยะทางแบบยุคลิดสามารถใช้ฟังก์ชัน *EarlyAbandon*(*C,Q,best-so-far*) ตามที่ได้แสดงไว้ในรหัสเทียมรูปที่ 3.14

Algorithm 6 : *Insert*(*InsertList*,*GroupList*,*MeanList*)

```
 1:  Foreach C in InsertList do
 2:    Add C to Dataset
 3:    mindist := INFINITY
 4:    nearestGroup := null
 5:    For i = 1 to MeanList.length do
 6:       dist := EarlyAbandon(C,MeanList[i],mindist)
 7:       If dist < mindist then
 8:          mindist := dist
 9:          nearestGroup := i
10:       EndIf
11:    EndFor
12:    Add C to GroupList[nearestGroup]
13:    Set flag that MeanList[nearestGroup] has been changed
14:  EndFor
15:
16:  For i = 1 to MeanList.length do
17:    If MeanList[i] has been changed then
18:       Recalculate MeanList[i]
19:    EndIf
20:  EndFor
21:
22:  Clustering(Dataset,MeanList)
```

รูปที่ 3.17 รหัสเทียมสำหรับฟังก์ชันการเพิ่มข้อมูลเข้าไปในชุดข้อมูล

หลังจากทำการจำแนกข้อมูลที่จะเพิ่มเข้าไปในชุดข้อมูลจนครบแล้ว ในบรรทัดที่ 16 ถึง 20 ให้ทำการคำนวณค่าเฉลี่ยของทุกกลุ่มข้อมูลที่มีข้อมูลถูกจำแนกเพิ่มเข้าไปและทำการบันทึกไว้ว่ากลุ่มเหล่านี้เกิดการเปลี่ยนแปลงของสมาชิกภายในกลุ่ม จากนั้นจะทำการเรียกฟังก์ชัน *Clustering*(*Dataset,MeanList*) ซึ่งเป็นฟังก์ชันการจับกลุ่มแบบเคมีนที่ได้นำเสนอไว้ในรูปที่ 3.13 เพื่อดำเนินการจับกลุ่มแบบเคมีนต่อโดยยังคงใช้การบันทึกการเปลี่ยนแปลงของสมาชิกภายในกลุ่มที่เกิดขึ้นในฟังก์ชันการเพิ่มข้อมูล ดังนั้นถ้ามีการเพิ่มข้อมูลเพียงจำนวนน้อย จะทำให้การคำนวณการจับกลุ่มใหม่สามารถทำได้อย่างรวดเร็ว เนื่องจากมี

กลุ่มข้อมูลที่เกิดการเปลี่ยนแปลงสมาชิกภายในกลุ่มอยู่เป็นจำนวนน้อย ทำให้สามารถลดการคำนวณตามวิธีในทฤษฎีบทที่ 1 ในหน้าที่ 42 ได้อย่างมีประสิทธิภาพ

3.7.2 การจับกลุ่มใหม่ภายหลังการลบข้อมูลออกจากชุดข้อมูล

สำหรับการลบข้อมูลนั้น สามารถทำการจับกลุ่มใหม่ด้วยวิธีที่คล้ายกับการเพิ่มข้อมูล เพียงทำการคำนวณค่าเฉลี่ยของกลุ่มข้อมูลทุกกลุ่มที่มีสมาชิกภายในกลุ่มถูกลบออกไป และทำการเรียกฟังก์ชัน `Clustering(Dataset,MeanList)` เพื่อดำเนินการจับกลุ่มแบบเคมีนต่อโดยยังคงใช้การบันทึกการเปลี่ยนแปลงของสมาชิกภายในกลุ่มที่เกิดจากการลบข้อมูลออกเท่านั้น

จากที่ได้กล่าวมาทั้งหมดจากในบทนี้เป็นการนำเสนอวิธีการเพิ่มความเร็วในการค้นคืนข้อมูลอนุกรมเวลาด้วยการเข้าถึงข้อมูลที่ผ่านการจับกลุ่มมาแล้วโดยใช้ดัชนี ซึ่งวิธีที่ได้นำเสนอนี้จะสามารถเพิ่มความเร็วได้มากน้อยเพียงใด และเร็วกว่าวิธีอื่นที่มีอยู่ในปัจจุบันหรือไม่ อีกทั้งยังเหมาะสมกับข้อมูลในรูปแบบใด หรือมีจุดเด่นและจุดด้อยอยู่ที่ใดบ้าง งานวิจัยนี้จึงได้ทำการทดลองในหลาย ๆ รูปแบบ เพื่อแสดงให้เห็นถึงศักยภาพที่เหนือว่าวิธีที่ดีที่สุดที่มีอยู่ในปัจจุบัน โดยจะนำเสนอผลการทดลองและการวิเคราะห์ผลในบทถัดไป

บทที่ 4

การทดลองและวิเคราะห์ผล

เพื่อเป็นการแสดงให้เห็นถึงศักยภาพของแนวคิดทั้งหมดที่ได้นำเสนอในงานวิจัยนี้ ในบทนี้จึงได้นำเสนอการทดลองเพื่อทดสอบประสิทธิภาพในด้านต่าง ๆ ของแต่ละขั้นตอนจากในกระบวนการการทำการค้นคืนข้อมูลอนุกรมเวลาที่ได้นำเสนอในงานวิจัยนี้ ซึ่งประกอบด้วย การทดสอบประสิทธิภาพของการลดการคำนวณในการจัดกลุ่มข้อมูล การทดสอบประสิทธิภาพของการกำหนดพารามิเตอร์ของการจับกลุ่มข้อมูลด้วยวิธีการกำหนดชุดตรวจสอบความสมเหตุสมผล การทดสอบประสิทธิภาพของฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูล และการทดสอบประสิทธิภาพของงานวิจัยนี้เมื่อเทียบกับงานวิจัยอื่น ๆ และก่อนจะกล่าวถึงการทดลองในงานวิจัยนี้ ในบทนี้จะเริ่มต้นกล่าวถึงรูปแบบของข้อมูลทั้งหมดที่ใช้ในการทดลอง

4.1 รูปแบบของข้อมูลทั้งหมดที่ใช้ในการทดลอง

ในงานวิจัยนี้ได้แบ่งชนิดของข้อมูลที่ใช้ในการทดลองออกเป็น 2 ประเภท ได้แก่ ข้อมูลจริงที่ใช้กันทั่วไปในงานวิจัยด้านการทำเหมืองข้อมูลอนุกรมเวลา และข้อมูลที่ได้จากการสังเคราะห์ขึ้นเอง โดยสังเคราะห์ตามวิธีที่ใช้กันทั่วไป

4.1.1 ข้อมูลจริงที่ใช้กันทั่วไปในงานวิจัยด้านการทำเหมืองข้อมูลอนุกรมเวลา

ในงานวิจัยนี้ได้นำชุดข้อมูลอนุกรมเวลาที่ได้รับการเปิดเผยสำหรับทุกงานวิจัยที่สนใจในด้านข้อมูลอนุกรมเวลา ซึ่งรายละเอียดของชุดข้อมูลทั้งหมดสามารถหาได้ในเว็บไซต์ Archive ของ University of California, Riverside [23] โดยในตารางที่ 4.1 แสดงคุณลักษณะของชุดข้อมูลทั้งหมดที่งานวิจัยนี้ได้นำมาเป็นข้อมูลในการทดลอง

แต่อย่างไรก็ตามชุดข้อมูลอนุกรมเวลาจริงที่พบเห็นได้ทั่วไปนั้นมักเป็นชุดข้อมูลที่มีขนาดเล็กเนื่องจากข้อมูลขนาดใหญ่มักพบเห็นอยู่ในรูปแบบของข้อมูลเชิงธุรกิจหรือเป็นข้อมูลที่มีมูลค่าซึ่งไม่สามารถนำมาเปิดเผยต่อสาธารณชนได้ ดังนั้นจึงไม่มีผู้ใดสามารถเปิดเผยข้อมูลที่มีขนาดใหญ่มากผ่านอินเทอร์เน็ตได้ โดยเฉพาะอย่างยิ่งขอบเขตของงานวิจัยนี้เน้นไปในด้านการค้นคืนข้อมูลอนุกรมเวลาจากชุดข้อมูลที่มีขนาดใหญ่ งานวิจัยนี้จึงได้รวมชุดข้อมูลทั้งหมดที่แสดงในตารางที่ 4.1 ให้เป็นชุดข้อมูลเดียวกัน โดยทำการปรับขนาดเอกรูปกับทุกข้อมูลให้มีความยาวเท่ากับ 637 ซึ่งเท่ากับความยาวของชุดข้อมูลที่ยาวที่สุดนั่นก็คือ Lightning-2 ดังนั้นจึงได้ชุดข้อมูลใหม่โดยขอเรียกชุดข้อมูลดังกล่าวว่า Mixed ซึ่งเป็นชุดข้อมูลที่

มีจำนวนคลาสทั้งหมด 178 คลาส มีจำนวนข้อมูลในชุดข้อมูลทั้งหมด 18602 อนุกรม ซึ่งได้จากการรวมข้อมูลสอบถามทั้งหมดจากในทุกชุดข้อมูลที่แสดงในตารางที่ 4.1

ตารางที่ 4.1 คุณลักษณะของชุดข้อมูลจริงที่ใช้ในการทดลอง

ชุดข้อมูล	*จำนวนคลาส*	*จำนวนข้อมูลในชุดข้อมูล*	*จำนวนข้อมูลสอบถาม*	*ความยาว (มิติ)*
Synthetic Control	6	300	300	60
Gun-Point	2	50	150	150
CBF	3	30	900	128
Face (all)	14	560	1690	131
OSU Leaf	6	200	242	427
Swedish Leaf	15	500	625	128
50Words	50	450	455	270
Trace	4	100	100	275
Two Patterns	4	1000	4000	128
Wafer	2	1000	6174	152
Face (four)	4	24	88	350
Lightning-2	2	60	61	637
Lightning-7	7	70	73	319
ECG	2	100	100	96
Adiac	37	390	391	176
Yoga	2	300	3000	426
Fish	7	175	175	463
Beef	5	30	30	470
Coffee	2	28	28	286
OliveOil	4	30	30	570
รวม (Mixed)	178	18612	5397	637

สำหรับข้อมูลทดสอบบนชุดข้อมูล Mixed นั้นมีจำนวนข้อมูลสอบถามทั้งหมด 5397 อนุกรม ซึ่งได้จากการรวมข้อมูลจากชุดข้อมูลทั้งหมด ซึ่งสังเกตได้ว่าในชุดข้อมูล Mixed ได้ทำการสลับกันระหว่างชุดข้อมูลและข้อมูลสอบถาม เหตุที่ต้องทำการสลับกันเนื่องจากต้องการได้ชุดข้อมูลที่มีขนาดใหญ่ จึงได้เลือกเอาข้อมูลสอบถามมาเป็นชุดข้อมูลแทน ดังนั้นจึงได้เลือกใช้ข้อมูลที่ได้จากการสังเคราะห์ขึ้นเองด้วย เพื่อให้สามารถกำหนดขนาดของชุดข้อมูลได้เองตามความต้องการ

4.1.2 ข้อมูลที่ได้จากการสังเคราะห์ขึ้น

วิธีการสังเคราะห์ข้อมูลที่นำมาใช้ในงานวิจัยนี้แบ่งออกเป็น 2 รูปแบบ ได้แก่ การสังเคราะห์ข้อมูลเดินสุ่ม (Randomwalk) และการสังเคราะห์ข้อมูลซีบีเอฟ (CBF)

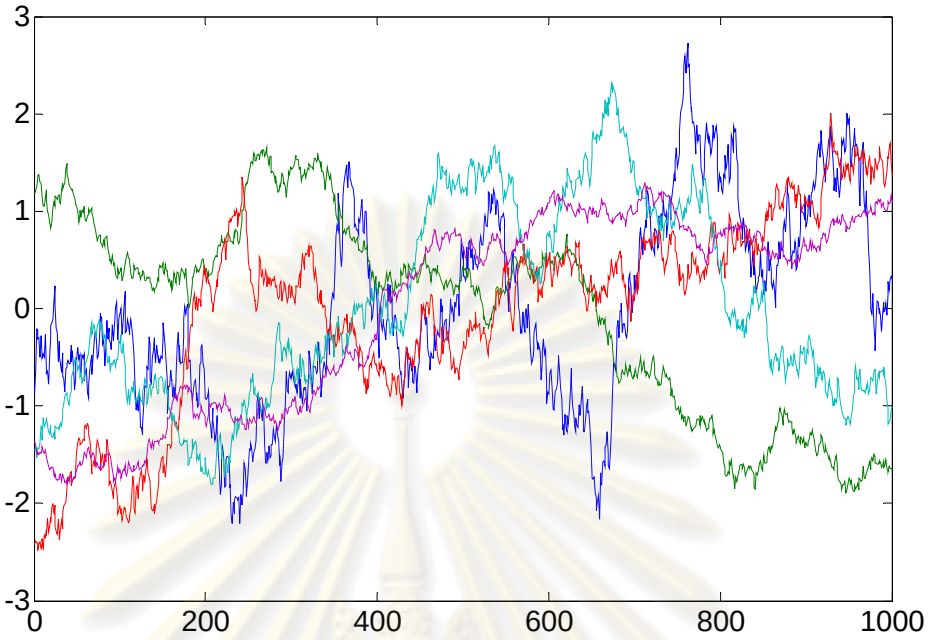
4.1.2.1 การสังเคราะห์ข้อมูลเดินสุ่ม

รูปแบบการสังเคราะห์ข้อมูลแบบเดินสุ่มนั้นเป็นที่นิยมแพร่หลายในงานวิจัยทั่วไปด้านการทำดัชนีบนข้อมูลอนุกรมเวลา [7, 8, 24, 25] ซึ่งพฤติกรรมของข้อมูลที่ได้จากการสังเคราะห์นั้นจะมีรูปแบบที่ไม่แน่นอน จึงทำให้ข้อมูลทั้งหมดไม่สามารถกำหนดคลาสของข้อมูลได้ โดยจะกล่าวถึงรายละเอียดวิธีการสังเคราะห์ข้อมูลเดินสุ่มดังต่อไปนี้

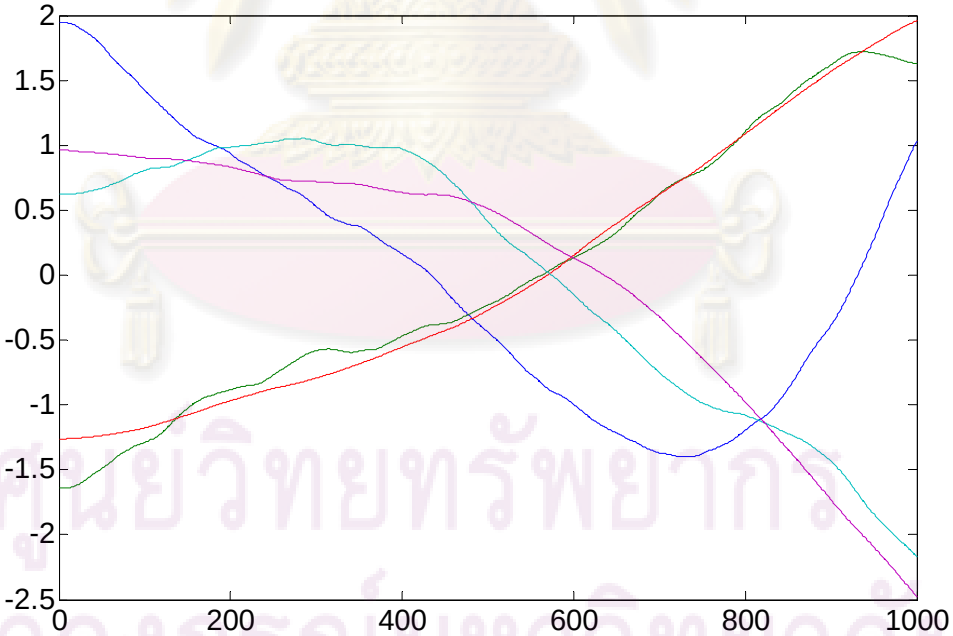
กำหนดให้ข้อมูลอนุกรมเวลา Q มีความยาว N โดยประกอบด้วยจุดข้อมูล q_1, q_2, q_3 จนถึง q_N จะสามารถสังเคราะห์ Q ด้วยวิธีเดินสุ่มได้ตามสมการ (4.1)

$$\begin{aligned} q_1 &= \mu(0,10) \\ q_i &= \eta(q_{i-1},1) \end{aligned} \tag{4.1}$$

โดยที่ฟังก์ชัน $\mu(0,10)$ คือค่าที่ได้จากการแจกแจงเอกรูป (Uniform Distribution) ที่มีค่าในช่วง 0 ถึง 10 และฟังก์ชัน $\eta(q_{i-1},1)$ เป็นค่าที่ได้จากการแจกแจงปกติ (Normal Distribution) ที่มีค่าเฉลี่ยเท่ากับ q_{i-1} และมีส่วนเบี่ยงเบนมาตรฐานเท่ากับ 1 โดยจะเรียกชุดข้อมูลที่สร้างจากสมการ (4.1) ว่า RWI (RandomWalkI) รูปที่ 4.1 แสดงตัวอย่างของข้อมูล RWI ทั้งหมด 5 ข้อมูล จะสังเกตได้ว่ารูปร่างของข้อมูลแต่ละข้อมูลมีลักษณะของสัญญาณรบกวนสูง (Noisy) ซึ่งข้อมูลในลักษณะดังกล่าวมักเป็นอุปสรรคสำหรับการประมาณค่าขอบเขตล่างของระยะทางแบบไดนามิกไทม์วอร์ปปิง ไม่ว่าจะเป็นวิธีการประมาณค่าขอบเขตล่างด้วยวิธี LB_Keogh การที่ข้อมูลมีลักษณะของสัญญาณรบกวนสูงจะทำให้การสร้างขอบเขตของข้อมูลจะได้ขอบเขตที่ไม่กระชับกับตัวข้อมูล หรือว่าจะเป็นการประมาณค่าขอบเขตล่างด้วยการลดมิติของข้อมูลลง ซึ่งการลดมิติของข้อมูลในลักษณะดังกล่าวมักทำให้สูญเสียลักษณะสำคัญ (Feature) ของตัวข้อมูลไป ดังนั้นการทำเหมืองข้อมูลโดยทั่วไปมักทำการปรับเรียบข้อมูล (Smooth) ก่อน



รูปที่ 4.1 ตัวอย่างข้อมูลที่ได้จากการสังเคราะห์ขึ้นแบบ RW I



รูปที่ 4.2 ตัวอย่างข้อมูลที่ได้จากการสังเคราะห์ขึ้นแบบ RW II

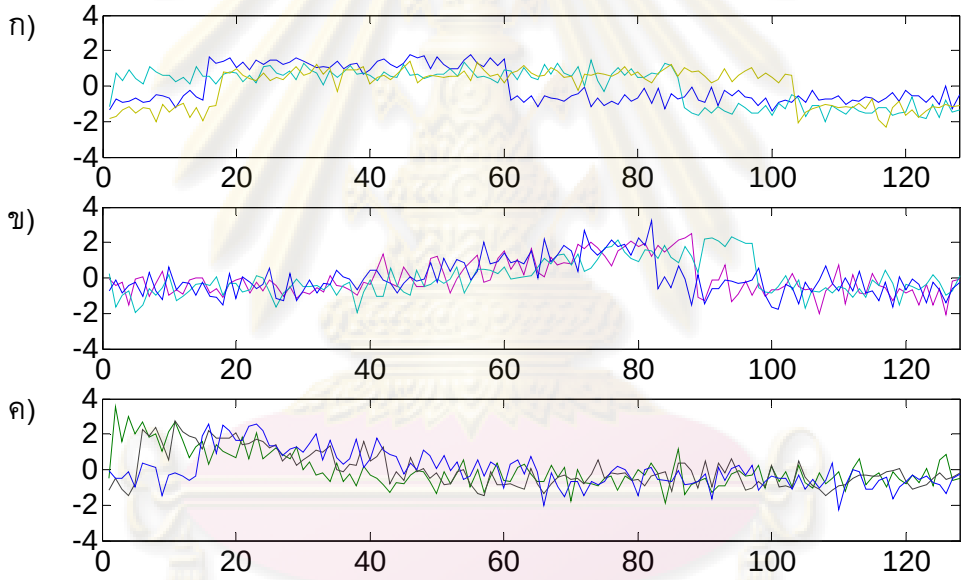
ในหลาย ๆ งานวิจัย [8, 26] จึงได้เลือกที่จะสังเคราะห์ข้อมูลให้มีลักษณะที่เรียบมากยิ่งขึ้น โดยแทนที่การสังเคราะห์จากสมการ (4.1) ด้วยสมการ (4.2) โดยจะเรียกชุดข้อมูลที่

สร้างจากสมการ (4.2) ว่า RW II (RandomWalk II) รูปที่ 4.2 แสดงตัวอย่างของข้อมูล RW II ทั้งหมด 5 อนุกรม

$$q_1 = \mu(0,10)$$
$$q_2 = \eta(q_1,1) \qquad (4.2)$$
$$q_i = \eta(2q_{i-1} - q_{i-2},1)$$

4.1.2.2 การสังเคราะห์ข้อมูลซีบีเอฟ (CBF: Cylinder Bell Funnel)

ข้อมูลซีบีเอฟ [27] เป็นข้อมูลสังเคราะห์ที่สามารถแบ่งออกได้เป็น 3 คลาส ได้แก่ คลาสกระบอก (Cylinder) คลาสระฆัง (Bell) และคลาสกรวย (Funnel) โดยข้อมูลซีบีเอฟทุกตัวจะมีความยาวเท่ากับ 128 จุดข้อมูล รายละเอียดในการสังเคราะห์ข้อมูลซีบีเอฟนั้นมีดังต่อไปนี้



รูปที่ 4.3 ตัวอย่างข้อมูลซีบีเอฟทั้ง 3 คลาส ก) คลาสกระบอก ข) คลาสระฆัง ค) คลาสกรวย

ในการสังเคราะห์ข้อมูลซีบีเอฟแต่ละตัวจะต้องทำการสุ่มค่าตัวแปร α และ β โดยที่ α นั้นมีค่าอยู่ในช่วง 16 ถึง 32 และค่า $\beta - \alpha$ จะอยู่ในช่วง 32 ถึง 96 จากนั้นกำหนดให้ข้อมูลคลาสกระบอก $C = \{C_1, C_2, C_3, \ldots, C_{128}\}$ ข้อมูลคลาสระฆัง $B = \{B_1, B_2, B_3, \ldots, B_{128}\}$ และข้อมูลคลาสกรวย $F = \{F_1, F_2, F3, \ldots, F_{128}\}$ จะสามารถสร้างข้อมูลทั้ง 3 คลาสได้จากสมการ (4.3) รูปที่ 4.3 แสดงตัวอย่างข้อมูลซีบีเอฟทั้ง 3 คลาส คลาสละ 3 ตัว สังเกตได้ว่าข้อมูลซีบีเอฟมีความเป็นสัญญาณรบกวนสูงมาก ซึ่งมักเป็นอุปสรรคต่อการประมาณค่าขอบเขตล่างของระยะทางไดนามิกไทม์วอร์ปปิง

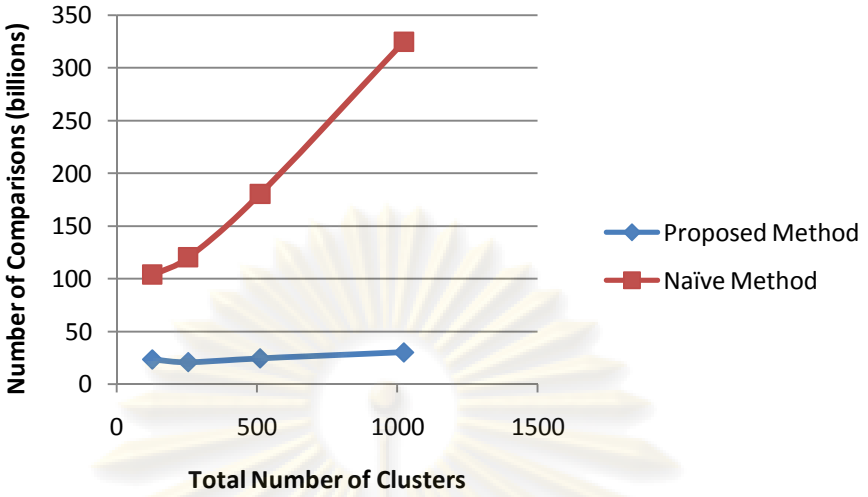
$$
\begin{aligned}
C_i &= \eta(6,1) * \chi_{[\alpha,\beta]}(i) + \eta(0,1) \\
B_i &= \eta(6,1) * \chi_{[\alpha,\beta]}(i) * (i-\alpha)/(\beta-\alpha) + \eta(0,1) \\
F_i &= \eta(6,1) * \chi_{[\alpha,\beta]}(i) * (\beta-i)/(\beta-\alpha) + \eta(0,1) \\
\chi_{[\alpha,\beta]}(i) &= \begin{cases} 1 & \text{if } \alpha \le i \le \beta \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
\tag{4.3}
$$

จากนี้จะกล่าวถึงการทดลองทั้งหมดในงานวิจัยนี้ การทดลองทั้งหมดแบ่งออกเป็นการทดสอบประสิทธิภาพการลดทอนการคำนวณในการจับกลุ่มข้อมูลแบบเคมีนที่ได้นำเสนอไว้ในหัวข้อที่ 3.3.1 การทดสอบประสิทธิภาพการลดทอนข้อมูลจากการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอในงานวิจัยนี้บนชุดข้อมูลในรูปแบบต่าง ๆ และการทดสอบประสิทธิภาพการค้นคืนข้อมูลบนชุดข้อมูลขนาดใหญ่

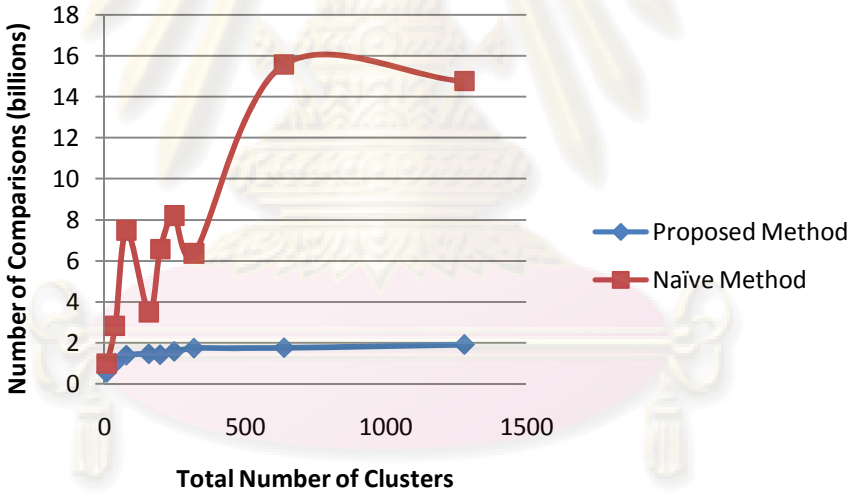
4.2 การทดสอบประสิทธิภาพการลดทอนการคำนวณในการจับกลุ่มข้อมูลแบบเคมีน

ในหัวข้อนี้จะนำเสนอผลการทดลองเพื่อวัดประสิทธิภาพในการลดทอนการคำนวณจากการจับกลุ่มแบบเคมีนโดยใช้มาตรวัดระยะทางแบบยุคลิด มาตรวัดที่ใช้ทดสอบประสิทธิภาพการลดทอนการคำนวณในการทดลองนี้ใช้วิธีการนับจำนวนครั้งในการวัดระยะทางระหว่างจุดข้อมูลหนึ่งคู่จุดจากการวัดระยะทางแบบยุคลิดทุกครั้งในการจับกลุ่ม โดยคิดจากจำนวนครั้งในการคำนวณที่ถูกลดทอนลงเป็นเปอร์เซ็นต์เมื่อเทียบกับจำนวนครั้งที่ต้องใช้คำนวณทั้งหมดโดยไม่ทำการลดทอนใด ๆ รูปที่ 4.4 แสดงผลการเปรียบเทียบการคำนวณจากการจับกลุ่มแบบเคมีนบนชุดข้อมูล Mixed ระหว่างทำการลดทอนการคำนวณและไม่ทำการลดทอนใด ๆ โดยทำการปรับค่าจำนวนกลุ่มเป็น 128 256 512 และ 1,024 กลุ่ม สังเกตได้ว่าวิธีการจับกลุ่มแบบเคมีนโดยปกติแล้ว จำนวนครั้งในการคำนวณมีแนวโน้มที่จะเพิ่มขึ้นเมื่อทำการจับกลุ่มด้วยจำนวนกลุ่มที่มากขึ้นอย่าเห็นได้ชัด แต่ด้วยการลดทอนการคำนวณด้วยวิธีที่ได้นำเสนอทำให้จำนวนครั้งในการคำนวณนั้นเกือบจะเป็นค่าคงที่สำหรับทุกจำนวนกลุ่มที่ทำการจับกลุ่ม ซึ่งสามารถลดการคำนวณลงได้มากที่สุดถึง 10 เท่า

รูปที่ 4.5 แสดงผลการเปรียบเทียบการคำนวณจากการจับกลุ่มแบบเคมีนบนชุดข้อมูล RWI จำนวน 10,000 อนุกรม โดยแต่ละอนุกรมมีความยาว 128 จุดข้อมูล จะสังเกตได้ว่าจำนวนครั้งในการคำนวณการจับกลุ่มนั้นค่อนข้างแปรปรวนและไม่แน่นอน เนื่องจากชุดข้อมูล RWI เป็นชุดข้อมูลสังเคราะห์ที่ไม่มีคลาสของข้อมูล ดังนั้นการดำเนินไปของการจับกลุ่มแบบเคมีนนั้นจึงขึ้นกับข้อมูลที่สุ่มมาเพื่อเป็นค่าเฉลี่ยเริ่มต้นของกลุ่ม การจับกลุ่มแต่ละครั้งจึงใช้เวลาที่แตกต่างกัน อย่างไรก็ตามแนวโน้มของจำนวนครั้งในการคำนวณก็ยังคงสูงขึ้นตามจำนวนกลุ่มที่เพิ่มมากขึ้น และอีกเช่นเคยด้วยวิธีที่นำเสนอก็ยังคงสามารถจับกลุ่มข้อมูลด้วยจำนวนครั้งในการคำนวณที่ค่อนข้างคงที่ซึ่งสามารถลดการคำนวณลงได้มากที่สุดถึง 8 เท่า



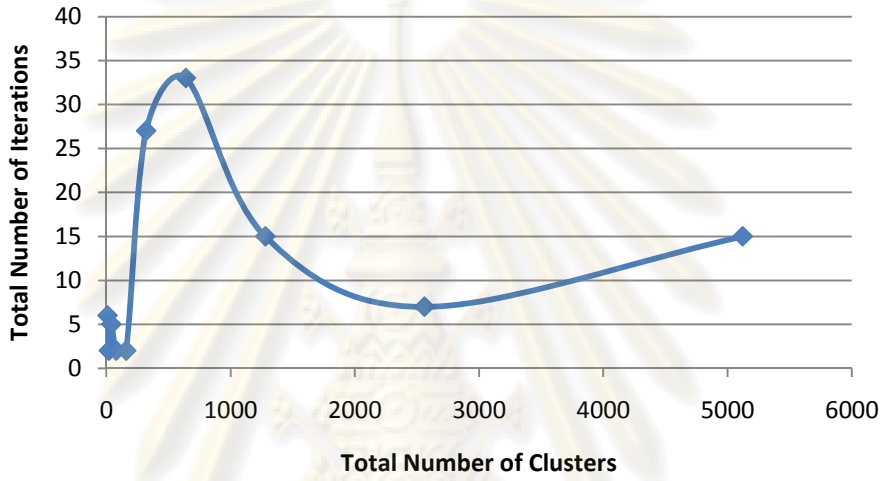
รูปที่ 4.4 ผลการทดลองการเปรียบเทียบจำนวนครั้งในการวัดระยะทางระหว่างแต่ละจุดข้อมูลในการวัดระยะทางแบบยุคลิดบนการจับกลุ่มแบบเคมีนระหว่างวิธีที่ได้นำเสนอกับวิธีดั้งเดิม โดยทดสอบบนการจับกลุ่มชุดข้อมูล Mixed ด้วยการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน



รูปที่ 4.5 ผลการทดลองการเปรียบเทียบจำนวนครั้งในการวัดระยะทางระหว่างแต่ละจุดข้อมูลในการวัดระยะทางแบบยุคลิดบนการจับกลุ่มแบบเคมีนระหว่างวิธีที่ได้นำเสนอกับวิธีดั้งเดิม โดยทดสอบบนการจับกลุ่มชุดข้อมูล RWI ทั้งหมด 10,000 อนุกรม แต่ละอนุกรมมีความยาว 128 จุดข้อมูล ด้วยการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน

จากผลการทดลองในรูปที่ 4.4 และรูปที่ 4.5 แสดงให้เห็นว่าการลดทอนการคำนวณนั้นจะมีประสิทธิภาพมากขึ้นสำหรับจำนวนกลุ่มที่มาก เนื่องจากการจับกลุ่มแบบเคมีนในแต่ละรอบการทำงานต้องทำการวัดระยะทางระหว่างข้อมูลทุกตัวกับค่าเฉลี่ยของทุกกลุ่มข้อมูล ดังนั้นสำหรับจำนวนกลุ่มมีค่ามากจะทำให้การลดทอนการคำนวณด้วยฟังก์ชัน

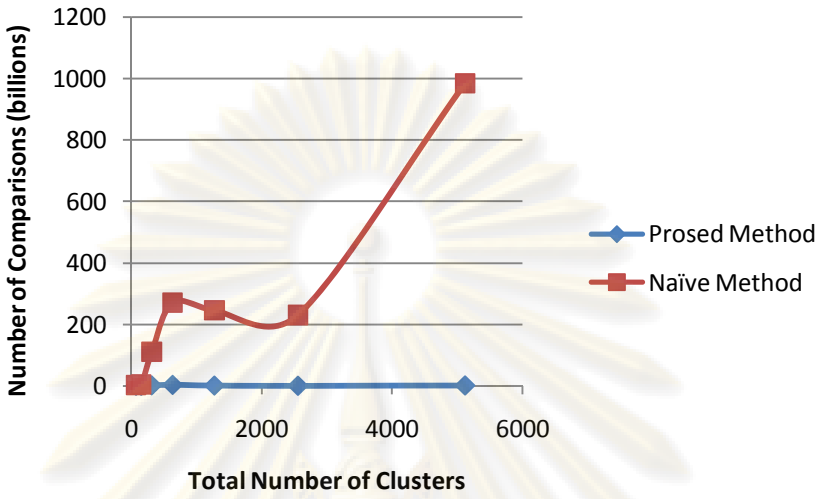
EarlyAbandon(*C*, *Q*, *best-so-far*) จากในรูปที่ 3.14 มีประสิทธิภาพสูงตามไปด้วย นอกจากนี้การที่มีจำนวนกลุ่มที่มากยังมีแนวโน้มที่จะมีกลุ่มข้อมูลที่ค่าเฉลี่ยของกลุ่มสามารถลู่เข้าสู่จุดที่เหมาะที่สุด (Local Optimal) ได้ด้วยจำนวนครั้งในการวนซ้ำที่น้อยกว่ากลุ่มข้อมูลอื่น ซึ่งส่งผลให้กลุ่มข้อมูลเหล่านั้นไม่เกิดการเปลี่ยนแปลงของสมาชิกในกลุ่มในรอบการวนซ้ำหลัง ๆ และส่งผลต่อประสิทธิภาพในการลดทอนการคำนวณตามทฤษฎีบทที่ 1 ดังนั้นจึงกล่าวได้ว่าในรอบการวนซ้ำหลัง ๆ ซึ่งกลุ่มข้อมูลหลายกลุ่มได้ลู่เข้าสู่จุดที่เหมาะที่สุดแล้วจะต้องทำการคำนวณน้อยกว่ารอบการวนซ้ำในช่วงต้น ๆ



รูปที่ 4.6 ผลการทดลองการวัดจำนวนรอบการวนซ้ำบนการจับกลุ่มแบบเคมีนจากการเพิ่มข้อมูล RWI ทั้งหมด 20 ตัวเข้าไปในชุดข้อมูล RWI ทั้งหมด 100,000 อนุกรมที่ผ่านการจับกลุ่มมาก่อนแล้ว โดยแต่ละอนุกรมมีความยาว 128 จุดข้อมูล ด้วยการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน

สำหรับการเพิ่มข้อมูลเข้าไปในชุดข้อมูลที่ผ่านการจับกลุ่มมาก่อนแล้ว การเพิ่มข้อมูลไปเป็นจำนวนน้อย ๆ จะส่งผลทำให้กลุ่มของข้อมูลเกิดการเปลี่ยนแปลงเพียงบางกลุ่มเป็นส่วนน้อยเท่านั้น ดังนั้นจึงส่งผลให้การลดทอนการคำนวณตามทฤษฎีบทที่ 1 สามารถลดทอนการคำนวณได้สูงมากดังแสดงได้จากผลการทดลองการเพิ่มข้อมูล RWI ทั้งหมด 20 ตัวเข้าไปในชุดข้อมูล RWI ทั้งหมด 100,000 อนุกรมที่ผ่านการจับกลุ่มมาก่อนแล้ว โดยแต่ละอนุกรมมีความยาว 128 จุดข้อมูล สังเกตได้ว่าถ้าจำนวนกลุ่มมีน้อยซึ่งหมายถึงจำนวนสมาชิกในแต่ละกลุ่มข้อมูลมีมาก การเพิ่มข้อมูลไปเป็นจำนวนน้อยอาจไม่ส่งผลให้กลุ่มข้อมูลเกิดการเปลี่ยนแปลงจนกว่าจำนวนกลุ่มข้อมูลจะมากถึงระดับหนึ่ง ซึ่งตรงกับจำนวน 320 กลุ่มตามที่แสดงในรูปที่ 4.6 ทำให้ต้องทำการวนซ้ำการคำนวณการจับกลุ่มเพิ่มขึ้น และส่งผลให้จำนวนครั้งในการคำนวณสูงขึ้นอย่างเห็นได้ชัดสำหรับวิธีการจับกลุ่มแบบเคมีนโดยปกติดังแสดงในรูปที่ 4.7 แต่ด้วยวิธีการลดทอนการคำนวณที่ได้นำเสนอนั้นทำให้สามารถเพิ่มข้อมูลเข้าไปได้ด้วย

เวลาที่เกือบจะคงที่สำหรับทุกค่าจำนวนกลุ่มทั้งหมดของชุดข้อมูล และน้อยกว่ามากเมื่อเทียบการจับกลุ่มโดยปกติซึ่งสามารถลดทอนการคำนวณได้มากที่สุดถึง 741 เท่าเลยทีเดียว



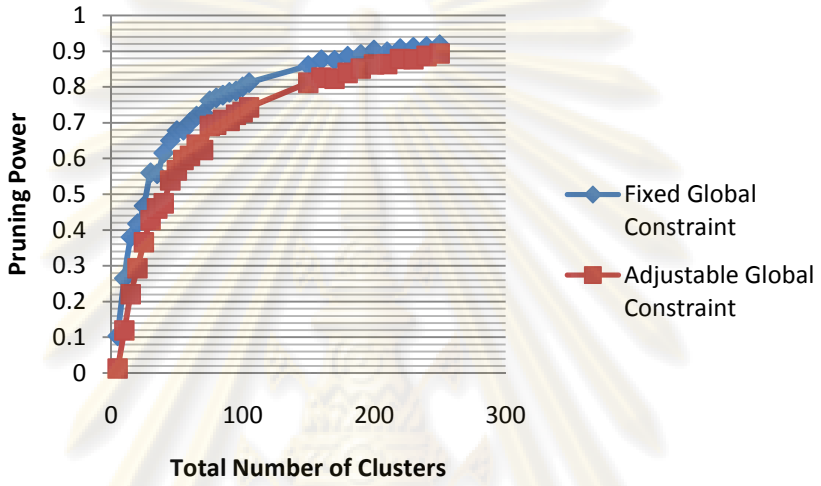
รูปที่ 4.7 ผลการทดลองการเปรียบเทียบจำนวนครั้งในการวัดระยะทางระหว่างแต่ละจุดข้อมูลในการวัดระยะทางแบบยุคลิดบนการจับกลุ่มแบบเคมีนระหว่างวิธีที่ได้นำเสนอกับวิธีดั้งเดิม โดยทดสอบบนการเพิ่มข้อมูล RWI ทั้งหมด 20 ตัวเข้าไปในชุดข้อมูล RWI ทั้งหมด 100,000 อนุกรมที่ผ่านการจับกลุ่มมาก่อนแล้ว โดยแต่ละอนุกรมมีความยาว 128 จุดข้อมูล ด้วยการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน

จากการทดลองทั้งหมดสำหรับการวัดประสิทธิภาพของการลดทอนการคำนวณสำหรับการจับกลุ่มแบบเคมีนนั้น สามารถสรุปได้ว่าวิธีการลดทอนการคำนวณที่ได้นำเสนอนั้นจะมีประสิทธิภาพสูงเมื่อทำการจับกลุ่มด้วยจำนวนกลุ่มข้อมูลที่มาก และจะมีประสิทธิภาพสูงมากเมื่อทำการจับกลุ่มที่ผ่านการจับกลุ่มมาก่อนแล้ว เพียงแต่เกิดการเปลี่ยนแปลงของข้อมูลเพียงเล็กน้อยซึ่งเกิดจากการเพิ่มหรือลดข้อมูลเข้าไปในชุดข้อมูล ดังนั้นวิธีที่ได้นำเสนอจึงทำให้ระบบการค้นคืนข้อมูลสามารถรองรับกับการเปลี่ยนแปลงของข้อมูลได้อย่างดีเยี่ยม

4.3 การทดสอบเพื่อเปรียบเทียบประสิทธิภาพของการลดทอนข้อมูลด้วยการทำดัชนีข้อมูลที่ใช้ฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่ได้นำเสนอทั้งสองวิธี

ในหัวข้อนี้จะทำการทดสอบเพื่อเปรียบเทียบประสิทธิภาพในการลดทอนข้อมูลด้วยการทำดัชนีข้อมูลที่ใช้ฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่ได้นำเสนอทั้งสองวิธีจากในหัวข้อที่ 3.4.1 ได้แก่ ฟังก์ชันที่ทำการตรึงขนาดของเงื่อนไขบังคับโดยรวมสำหรับการค้นคืนข้อมูลทุกรูปแบบบนชุดข้อมูลนั้น ๆ ซึ่งได้นำเสนอไว้ใน

หัวข้อที่ 3.4.1.1 และฟังก์ชันที่สามารถกำหนดขนาดของเงื่อนไขบังคับโดยรวมสำหรับข้อมูลสอบถามแต่ละตัวได้ ซึ่งได้นำเสนอไว้ในหัวข้อที่ 3.4.1.2 โดยวัดประสิทธิภาพของการลดทอนข้อมูลจากการค้นคืนด้วยการใช้แต่ละฟังก์ชันในรูปแบบของกำลังการลดทอน (Pruning Power) ซึ่งคำนวณได้จากอัตราส่วนระหว่างข้อมูลทั้งหมดที่ถูกลดทอนจากการค้นคืนเทียบกับจำนวนข้อมูลทั้งหมดในชุดข้อมูล การทดลองนี้วัดผลด้วยการทดสอบการค้นคืนข้อมูลด้วยชุดข้อมูล 50Words โดยผลการทดลองดังกล่าวได้แสดงไว้ในรูปที่ 4.8

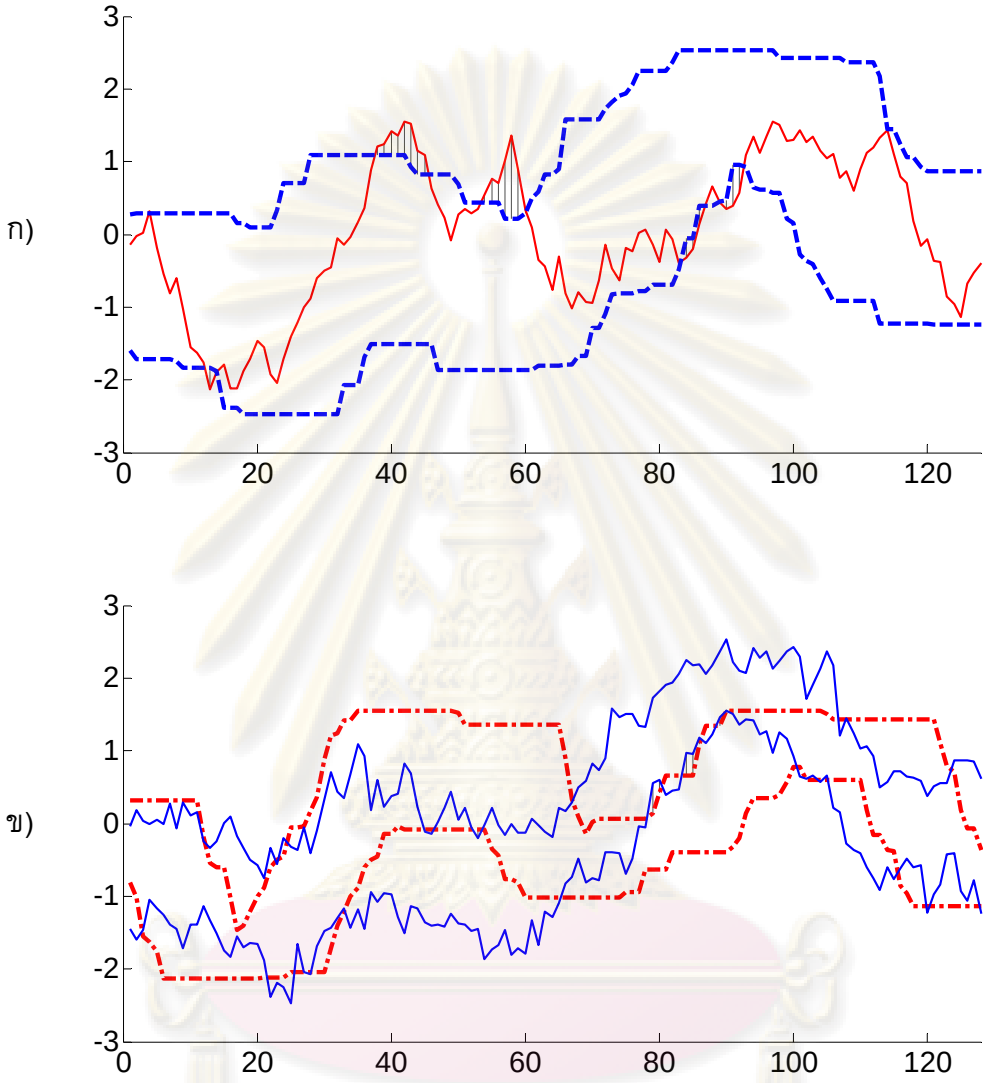


รูปที่ 4.8 ผลการทดลองการเปรียบเทียบกำลังการลดทอนระหว่างการค้นคืนข้อมูลด้วยดัชนีที่ใช้ฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่มีการตรึงขนาดของเงื่อนไขบังคับโดยรวมกับฟังก์ชันที่สามารถปรับขนาดของเงื่อนไขบังคับโดยรวมได้

จากผลการทดลองในรูปที่ 4.8 จะเห็นได้ว่าฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่มีการตรึงขนาดของเงื่อนไขบังคับโดยรวมสามารถลดทอนข้อมูลได้มากกว่าฟังก์ชันแบบที่สามารถกำหนดเงื่อนไขบังคับโดยรวมได้ เนื่องมาจากค่าขอบเขตล่าง

รูปที่ 4.9 แสดงการเปรียบเทียบรูปแบบระหว่างการคำนวณฟังก์ชันทั้งสอง สังเกตได้ว่าฟังก์ชันที่มีการตรึงขนาดของเงื่อนไขบังคับโดยรวมดังแสดงในรูปที่ 4.9 ก) จะทำการขยายขอบเขตของข้อมูลที่กลุ่มของข้อมูลโดยแสดงเป็นเส้นประ และข้อมูลสอบถามแสดงในรูปของเส้นทึบ ค่าที่ได้จากฟังก์ชันขอบเขตล่างยังคงสามารถแสดงให้เห็นในส่วนพื้นที่แรเงาซึ่งเป็นส่วนที่ข้อมูลสอบถามอยู่นอกขอบเขตของกลุ่มข้อมูล เปรียบเทียบกับการคำนวณฟังก์ชันขอบเขตล่างที่สามารถกำหนดขนาดของเงื่อนไขบังคับโดยรวมได้ดังแสดงในรูปที่ 4.9 ข) ฟังก์ชันดังกล่าวจะทำการขยายขอบเขตที่ตัวข้อมูลสอบถามแทนดังแสดงในส่วนขอบเขตเส้นประ และทำการคำนวณระยะทางกับกล่องขอบเขตของกลุ่มข้อมูลดังแสดงในส่วนของเส้น

ทึบ จะเห็นได้ว่าค่าที่ได้จากฟังก์ชันนั้นมีค่าน้อยมาก เนื่องจากการขยายขอบเขตจากส่วนของข้อมูลเดียวมักทำให้พื้นที่ขยายขึ้นมากกว่าการขยายขอบเขตจากกลุ่มข้อมูล

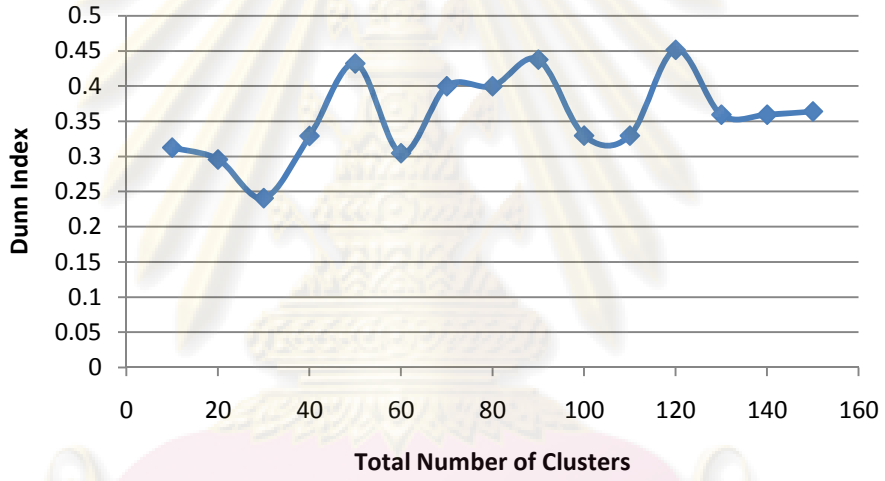


รูปที่ 4.9 การเปรียบเทียบการคำนวณฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางแบบไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลระหว่าง ก) แบบที่มีการตรึงขนาดของเงื่อนไขบังคับโดยรวมกับ ข) แบบที่สามารถกำหนดเงื่อนไขบังคับโดยรวมได้

การทดลองในหัวข้อนี้ได้แสดงให้เห็นว่าการกำหนดขนาดของเงื่อนไขบังคับโดยรวมที่แน่นอนสำหรับการค้นคืนข้อมูลจากในชุดข้อมูลนั้น ๆ จะทำให้ได้ค่าฟังก์ชันขอบเขตล่างสำหรับค่าระยะทางแบบไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่ใกล้เคียงกับระยะทางแบบไดนามิกไทม์วอร์ปปิงจริงมากกว่าการกำหนดขนาดของเงื่อนไขบังคับรวมบนข้อมูลสอบถาม

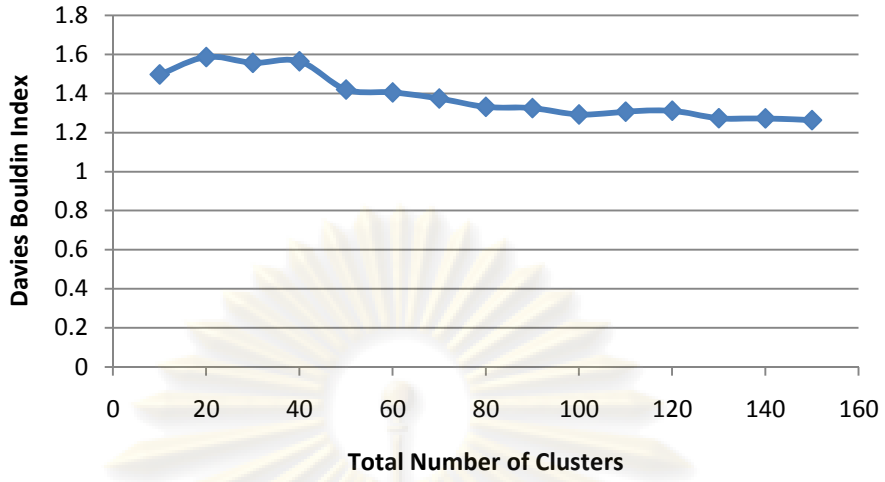
4.4 การทดสอบประสิทธิภาพในการกำหนดค่าพารามิเตอร์ในการจับกลุ่มข้อมูลสำหรับการทำดัชนีการค้นคืนข้อมูลด้วยวิธีการวัดความสมเหตุสมผลของการจับกลุ่ม (Cluster Validity Measurement)

ในหัวข้อนี้จะนำเสนอผลการทดลองการกำหนดค่าพารามิเตอร์ในการจับกลุ่มด้วยวิธีการวัดความสมเหตุสมผลของการจับกลุ่ม โดยทำการทดสอบการใช้ค่าความสมเหตุสมผลของการจับกลุ่มด้วยวิธีต่าง ๆ มาใช้กำหนดจำนวนกลุ่มในการจับกลุ่มเพื่อเตรียมข้อมูลสำหรับการทำดัชนีการค้นคืนข้อมูลด้วยวิธีที่นำเสนอ และทำการวัดผลว่าการจับกลุ่มข้อมูลที่ได้เลือกนั้นทำให้กลุ่มข้อมูลที่สามารถทำดัชนีได้อย่างมีประสิทธิภาพมากน้อยเพียงใด ด้วยการทดสอบการค้นคืนข้อมูลจริง และทำการจับเวลาเพื่อเปรียบเทียบระหว่างดัชนีที่ได้จากการจับกลุ่มที่เลือกไว้เทียบกับการจับกลุ่มด้วยค่าพารามิเตอร์อื่น ๆ

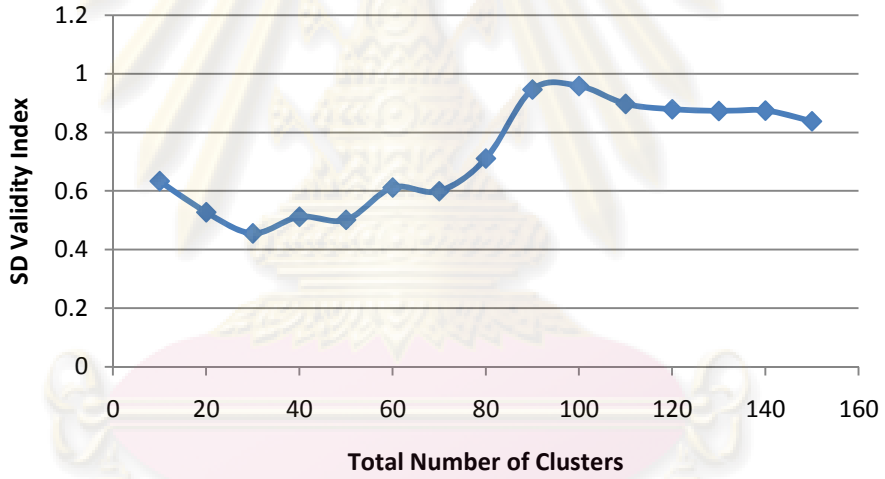


รูปที่ 4.10 ค่าดัชนีดันที่วัดได้จากการจับกลุ่มแบบเคมีนบนชุดข้อมูล Wafer โดยทำการปรับค่าจำนวนกลุ่มที่แตกต่างกัน

ชุดข้อมูลที่ใช้ในการทดสอบในหัวข้อนี้คือชุดข้อมูล Wafer เนื่องจากเป็นชุดข้อมูลจริงที่มีขนาดใหญ่ที่สุดจากในตารางที่ 4.1 โดยทำการจับกลุ่มแบบเคมีนบนชุดข้อมูลดังกล่าวด้วยการปรับค่าจำนวนกลุ่มที่แตกต่างกัน จากนั้นนำผลการจับกลุ่มด้วยค่าพารามิเตอร์ทั้งหมดมาวัดความสมเหตุสมผลด้วยดัชนีวัดความสมเหตุสมผลในรูปแบบต่าง ๆ ได้แก่ ดัชนีดันดัชนีเดวีส์บูลดิน ดัชนีความสมเหตุสมผลของส่วนเบี่ยงเบนมาตรฐาน และดัชนีความสมเหตุสมผลของเอสดีบีดับเบิลยู รูปที่ 4.10 แสดงค่าดัชนีดันที่ได้จากการจับกลุ่มทั้งหมด ซึ่งกลุ่มข้อมูลที่ดัชนีดันบ่งบอกว่าเป็นกลุ่มข้อมูลที่ดีที่สุดหรือค่าดัชนีที่ได้มีค่าน้อยที่สุดคือการจับกลุ่มด้วยจำนวนกลุ่มเท่ากับ 30 กลุ่ม ส่วนรูปที่ 4.11 แสดงค่าดัชนีเดวีส์บูลดินที่ได้จากการจับกลุ่มทั้งหมด ซึ่งกลุ่มข้อมูลที่ดัชนีเดวีส์บูลดินบ่งบอกว่าเป็นกลุ่มข้อมูลที่ดีที่สุดหรือค่าดัชนีที่ได้มีค่ามากที่สุดคือการจับกลุ่มด้วยจำนวนกลุ่มเท่ากับ 20 กลุ่ม

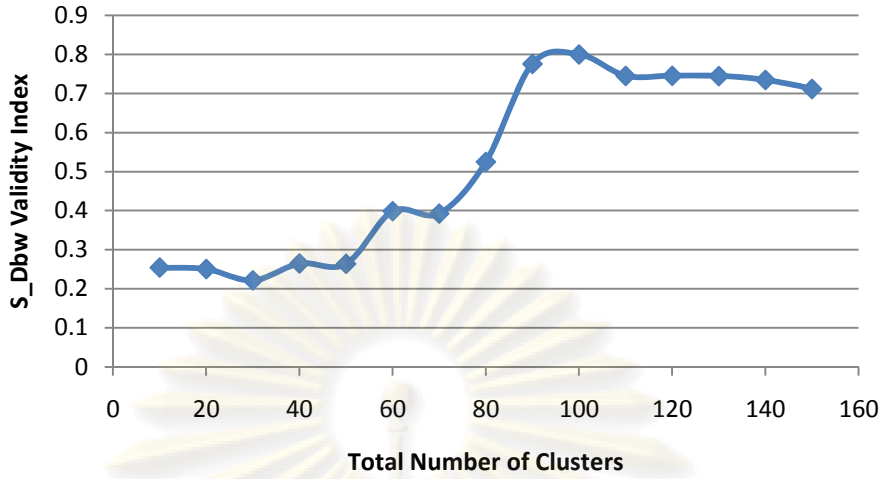


รูปที่ 4.11 ค่าดัชนีเดวีส์บูลดินที่วัดได้จากการจับกลุ่มแบบเคมีนบนชุดข้อมูล Wafer โดยทำการปรับค่าจำนวนกลุ่มที่แตกต่างกัน



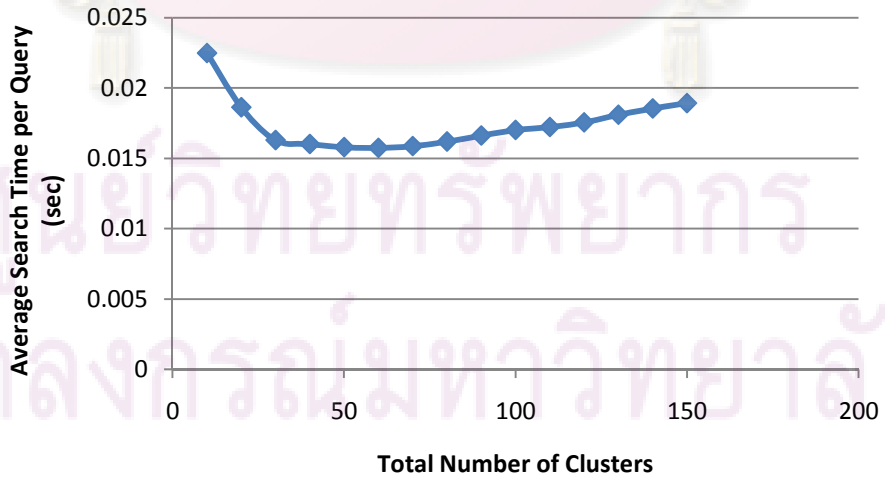
รูปที่ 4.12 ค่าดัชนีความสมเหตุสมผลของส่วนเบี่ยงเบนมาตรฐานที่วัดได้จากการจับกลุ่มแบบเคมีนบนชุดข้อมูล Wafer โดยทำการปรับค่าจำนวนกลุ่มที่แตกต่างกัน

รูปที่ 4.12 แสดงค่าดัชนีความสมเหตุสมผลของส่วนเบี่ยงเบนมาตรฐานที่ได้จากการจับกลุ่มทั้งหมด ซึ่งกลุ่มข้อมูลที่ดัชนีเดวีส์บูลดินบ่งบอกว่าเป็นกลุ่มข้อมูลที่ดีที่สุดหรือค่าดัชนีที่ได้มีค่าน้อยที่สุดคือการจับกลุ่มด้วยจำนวนกลุ่มเท่ากับ 30 กลุ่ม



รูปที่ 4.13 ค่าดัชนีความสมเหตุสมผลของเอสดีบีดับเบิลยูที่วัดได้จากการจับกลุ่มแบบเคมีนบนชุดข้อมูล Wafer โดยทำการปรับค่าจำนวนกลุ่มที่แตกต่างกัน

รูปที่ 4.13 แสดงค่าดัชนีความสมเหตุสมผลของเอสดีบีดับเบิลยูที่ได้จากการจับกลุ่มทั้งหมด ซึ่งกลุ่มข้อมูลที่ดัชนีเดวีส์บูลดินบ่งบอกว่าเป็นกลุ่มข้อมูลที่ดีที่สุดหรือค่าดัชนีที่ได้มีค่าน้อยที่สุดคือการจับกลุ่มด้วยจำนวนกลุ่มเท่ากับ 30 กลุ่ม ส่วนรูปที่ 4.14 แสดงผลการจับเวลาในการค้นคืนข้อมูล สังเกตได้ว่าการจับกลุ่มด้วยจำนวนกลุ่มที่เพิ่มขึ้นจำให้ได้ผลการค้นคืนที่รวดเร็วยิ่งขึ้นจนถึงค่าจำนวนกลุ่มเท่ากับ 30 จากนั้นการเพิ่มจำนวนกลุ่มจะมีผลต่อการค้นคืนเพียงเล็กน้อย ส่วนการจับกลุ่มด้วยจำนวนกลุ่มที่เพิ่มจาก 70 กลุ่มนั้นจะทำให้ผลการค้นคืนนั้นแย่ลง ดังนั้นจึงสามารถสรุปได้ว่าดัชนีความสมเหตุสมผลแต่ละรูปแบบนั้นสามารถกำหนดค่าพารามิเตอร์ได้ดีในระดับหนึ่ง ยกเว้นดัชนีเดวีส์บูลดินที่ให้ผลการเลือกกลุ่มข้อมูลที่ผิด



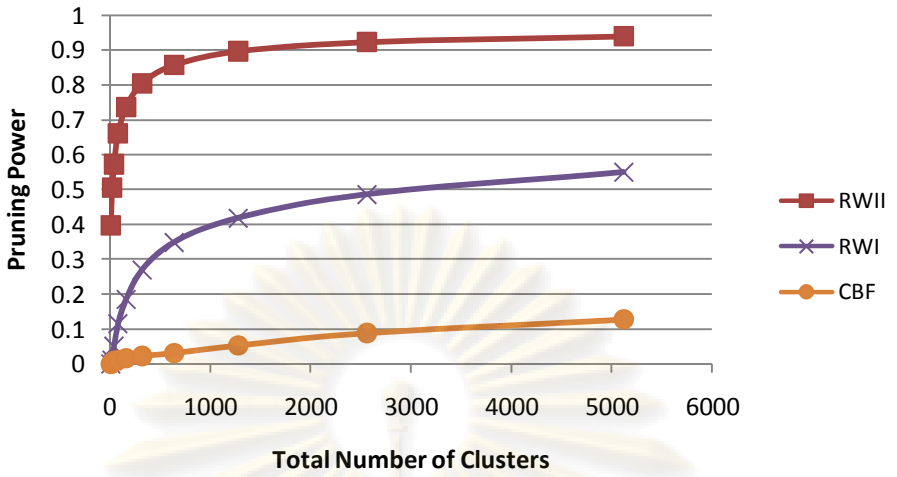
รูปที่ 4.14 ผลการทดสอบการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอบนชุดข้อมูล Wafer โดยทำการปรับค่าจำนวนกลุ่มที่แตกต่างกัน

อย่างไรก็ตามการคำนวณค่าดัชนีความสมเหตุสมผลในแต่ละรูปแบบนั้นต้องใช้เวลาในการคำนวณสูงมากโดยสังเกตได้จากรายละเอียดวิธีการคำนวณค่าดัชนีความสมเหตุสมผลจากในหัวข้อที่ 2.1.4 ดังนั้นสำหรับการค้นคืนข้อมูลบนชุดข้อมูลขนาดใหญ่นั้น การกำหนดค่าพารามิเตอร์ในการจับกลุ่มด้วยวิธีการวัดความสมเหตุสมผลของการจับกลุ่มอาจไม่เหมาะในการนำไปใช้จริง งานวิจัยนี้จึงได้เลือกวิธีการกำหนดค่าพารามิเตอร์สำหรับการจับกลุ่มด้วยการใช้ชุดข้อมูลตรวจสอบความสมเหตุสมผล (Validation Set)

4.5 การทดสอบประสิทธิภาพการลดทอนข้อมูลจากการค้นคืนข้อมูลด้วยดัชนีบนชุดข้อมูลในรูปแบบต่าง ๆ

ในหัวข้อนี้จะนำเสนอผลการทดลองเพื่อวัดประสิทธิภาพในการลดทอนข้อมูลจากการค้นคืนข้อมูลด้วยดัชนีบนชุดกับข้อมูลในรูปแบบใด และไม่เหมาะสมกับข้อมูลในรูปแบบใดข้อมูลในรูปแบบที่ต่างกันเพื่อค้นหาจุดเด่นและจุดด้อยของวิธีการค้นคืนข้อมูลที่ได้นำเสนอว่าเหมาะสม

การทดลองดังกล่าวนี้จะทำการวัดประสิทธิภาพในการลดทอนข้อมูลจากการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอ โดยทำการทดลองจากการค้นคืนข้อมูลบนชุดข้อมูล RWI RWII และ CBF ข้อมูลทั้งหมดมีความยาวเท่ากันเท่ากับ 128 จุดข้อมูล แต่ละชุดข้อมูลประกอบไปด้วยทั้งหมด 100,000 อนุกรม โดยแต่ละชุดข้อมูลจะใช้ข้อมูลสอบถามทั้งหมด 100 อนุกรมที่เป็นข้อมูลประเภทเดียวกันกับชุดข้อมูลนั้น ๆ แล้วทำการวัดประสิทธิภาพในการลดทอนในรูปแบบของกำลังการลดทอน (Pruning Power) โดยผลการทดลองดังกล่าวได้แสดงไว้ในรูปที่ 4.15 จากผลการทดลองจะเห็นได้ว่าการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอนั้นสามารถลดทอนข้อมูลจากในชุดข้อมูล RWII ได้สูงมาก แต่สำหรับชุดข้อมูล CBF นั้นมีกำลังการลดทอนที่ต่ำมากจนเป็นการไม่เหมาะสมที่จะทำการค้นคืนข้อมูลด้วยดัชนีบนชุดข้อมูล CBF เนื่องจากชุดข้อมูล CBF นั้นมีความเป็นสัญญาณรบกวนที่สูงมาก อีกทั้งค่าของข้อมูลทั้งหมดยังอยู่ในระดับมาตราส่วน (Scale) ที่ใกล้เคียงกัน ซึ่งส่งผลให้ขอบเขตของแต่ละกลุ่มข้อมูลทั้งหมดนั้นมีลักษณะที่เหมือนกันจนไม่สามารถแยกออกจากกันได้ไม่ว่าจะทำการจับกลุ่มอย่างละเอียดมากเพียงใดก็ตาม เพื่อเป็นการเปรียบเทียบให้เห็นเด่นชัดขึ้น ให้สังเกตระหว่างชุดข้อมูล RWI และ RWII ซึ่งต่างเป็นข้อมูลที่ได้จากการสุ่มทั้งคู่ เพียงแต่ชุดข้อมูล RWI จะมีความเป็นสัญญาณรบกวนอยู่ในระดับหนึ่ง ด้วยเหตุนี้จึงทำให้ประสิทธิภาพการลดทอนข้อมูลด้วยการค้นคืนข้อมูลด้วยดัชนีลดต่ำลงอย่างมาก สรุปคือการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอนั้นจะเหมาะสมกับข้อมูลที่มีความราบเรียบ (Smooth) มากกว่าข้อมูลที่มีความเป็นสัญญาณรบกวนสูง (Noisy)

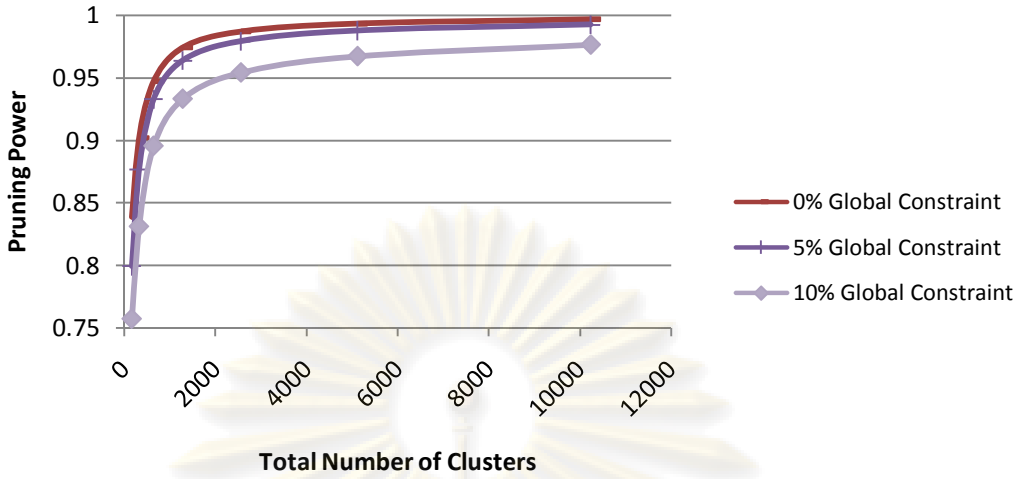


รูปที่ 4.15 ผลการทดลองการเปรียบเทียบกำลังการลดทอนข้อมูลจากการเข้าถึงข้อมูลด้วยดัชนีที่ได้นำเสนอ โดยทำการเปรียบระหว่างการเข้าถึงชุดข้อมูล RWI RWII และ CBF ข้อมูลทั้งหมดมีความยาวเท่ากันเท่ากับ 128 จุดข้อมูล แต่ละชุดข้อมูลประกอบไปด้วยทั้งหมด 100,000 อนุกรม โดยการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน

4.6 การทดสอบประสิทธิภาพการลดทอนข้อมูลจากการค้นคืนข้อมูลด้วยดัชนีสำหรับการกำหนดขนาดของเงื่อนไขบังคับโดยรวมที่แตกต่างกัน

ในหัวข้อนี้จะทำการทดลองเพื่อวัดผลกระทบของขนาดของเงื่อนไขบังคับโดยรวมที่มีต่อประสิทธิภาพในการลดทอนข้อมูลจากการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอ การทดลองนี้ได้ทำการทดสอบบนชุดข้อมูล RWII ที่มีขนาด 100,000 อนุกรม แต่ละข้อมูลมีความยาว 2,048 จุดข้อมูลที่ผ่านการจับกลุ่มด้วยจำนวนกลุ่มที่แตกต่างกัน โดยทำการค้นคืนข้อมูลด้วยข้อมูลสอบถามทั้งหมด 100 อนุกรมที่มีความยาวเท่ากันเท่ากับ 2,048 จุดข้อมูล แล้วทำการวัดกำลังการลดทอนสำหรับการค้นคืนข้อมูลที่มีการกำหนดขนาดของเงื่อนไขบังคับโดยรวมที่แตกต่างกันโดยวัดเป็นเปอร์เซ็นต์เทียบกับความยาวของข้อมูล ผลการทดลองดังกล่าวแสดงใน

รูปที่ 4.16 สังเกตได้ว่าขนาดของเงื่อนไขบังคับโดยรวมที่เพิ่มขึ้นนั้นทำให้ประสิทธิภาพในการลดทอนลดลง อย่างไรก็ตามในงานประยุกต์ที่ใช้การกำหนดเงื่อนไขบังคับโดยรวมสำหรับไดนามิกไทม์วอร์ปปิงนั้นมักใช้กันทั่วไปไม่เกิน 10% เนื่องจากการกำหนดขนาดที่มากกว่า 10% มักทำให้ความแม่นยำในการเปรียบเทียบข้อมูลลดลง [3]



รูปที่ 4.16 ผลการทดลองการเปรียบเทียบผลกระทบของขนาดของเงื่อนไขบังคับโดยรวมต่อประสิทธิภาพในการลดทอนข้อมูลจากการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอ โดยทดสอบบนการค้นคืนข้อมูล RWII ทั้งหมด 100 อนุกรมจากในชุดข้อมูล RWII ทั้งหมด 100,000 อนุกรมที่ผ่านการจับกลุ่มมาก่อนแล้ว โดยแต่ละข้อมูลมีความยาว 2,048 จุดข้อมูล ด้วยการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน

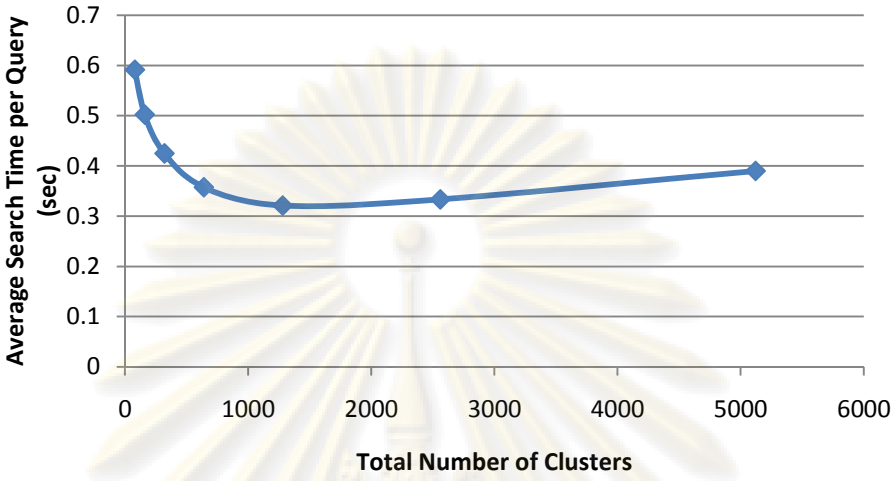
โดยสรุปแล้วการทดลองต่อจากนี้จะทำการกำหนดขนาดของเงื่อนไขบังคับโดยรวมเท่ากับ 10% ของความยาวข้อมูลเพื่อให้เกิดกรณีที่แย่ที่สุดสำหรับการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอ ถึงแม้จะเป็นกรณีที่แย่ที่สุดก็ตาม วิธีที่ได้นำเสนอก็ยังคงสามารถเพิ่มความเร็วในการค้นคืนข้อมูลได้อย่างมีนัยสำคัญ โดยจะใช้การทดลองต่อจากนี้ทั้งหมดเพื่อแสดงให้เห็นถึงศักยภาพของวิธีที่ได้นำเสนอที่เหนือกว่าวิธีการค้นอื่นที่มีอยู่ในปัจจุบัน

4.7 การทดสอบประสิทธิภาพการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอ

ในหัวข้อนี้จะทำการทดลองเพื่อวัดประสิทธิภาพในด้านความเร็วจากการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอ โดยการทดลองทั้งหมดในหัวข้อนี้จะทำการดำเนินงานบนเครื่องคอมพิวเตอร์ที่ใช้ซีพียู Intel Pentium 4 ความเร็ว 3.06 กิกะเฮิรตซ์ และใช้แรมขนาด 1 กิกะไบต์ งานทั้งหมดดำเนินงานภายใต้ระบบปฏิบัติการ Windows XP โดยใช้ชุดคำสั่งภาษาจาวาทั้งหมด

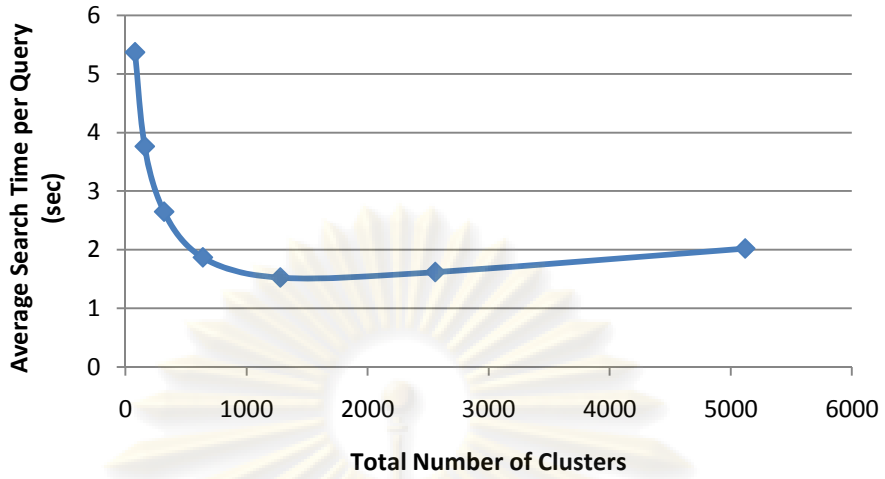
ในการทดลองเพื่อวัดความเร็วในการค้นคืนข้อมูลนั้น ได้ทำการทดลองบนชุดข้อมูล RWII ที่มีความยาวที่แตกต่างกัน ได้แก่ ชุดข้อมูลที่มีความยาว 128 1,024 และ 2,048 จุดข้อมูล แต่ละชุดข้อมูลมีขนาดเท่ากันเท่ากับ 100,000 อนุกรม และทำการวัดเวลาที่ใช้ในการค้นคืนข้อมูลโดยเฉลี่ยในแต่ละข้อมูลสอบถาม จากการทดลองทั้ง 3 ชุดข้อมูลดังกล่าวนั้น จะต้องทำการกำหนดจำนวนกลุ่มในการจับกลุ่มแต่ละชุดข้อมูลด้วยด้วยวิธีการใช้ชุดตรวจสอบความ

สมเหตุสมผล (Validation Set) โดยใช้บนชุดตรวจสอบความสมเหตุสมผลทั้งหมด 20 ข้อมูล เพื่อทำการกำหนดค่าจำนวนกลุ่มในแต่ละชุดข้อมูล



รูปที่ 4.17 ผลการทดลองการเลือกจำนวนกลุ่มในการจับกลุ่มของชุดข้อมูล RW II ด้วยวิธีการใช้ชุดตรวจสอบความสมเหตุสมผล (Validation Set) โดยใช้บนชุดตรวจสอบความสมเหตุสมผลที่ประกอบด้วย 20 ข้อมูล บนชุดข้อมูลที่ประกอบด้วยข้อมูลจำนวนทั้งหมด 100,000 อนุกรมแต่ละอนุกรมมีความยาว 128 จุดข้อมูล ซึ่งมีการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน

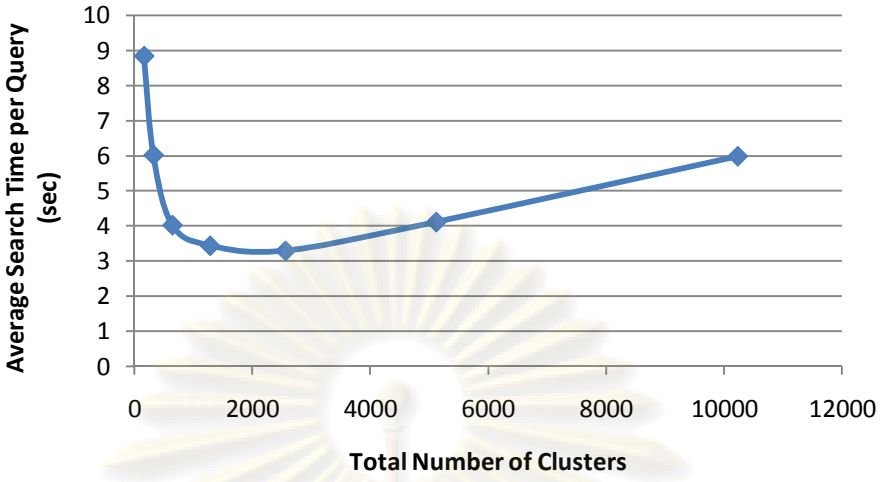
รูปที่ 4.17 แสดงผลการทดลองการวัดเวลาที่ใช้โดยเฉลี่ยสำหรับการค้นคืนข้อมูลสอบถามแต่ละข้อมูลทั้งหมด 20 อนุกรมด้วยการค้นข้อมูลด้วยดัชนีที่ได้นำเสนอ การวัดเวลาจะวัดจากการค้นคืนข้อมูลสอบถามชุดเดียวกันบนชุดข้อมูลเดียวกันขนาด 100,000 อนุกรม แต่ละอนุกรมมีความยาว 128 จุดข้อมูล โดยทำการจับกลุ่มแบบเคมีนที่มีการกำหนดจำนวนกลุ่มที่แตกต่างกัน สังเกตได้ว่าจากการค้นคืนข้อมูลจะมีประสิทธิภาพมากขึ้นบนชุดข้อมูลที่มีการจับกลุ่มที่มีจำนวนกลุ่มมาก แต่หลังจากทำการเพิ่มจำนวนกลุ่มข้อมูลให้มากเกินกว่า 1,280 กลุ่มจะทำให้ประสิทธิภาพในการค้นคืนข้อมูลด้วยดัชนีลดลง เนื่องจากเมื่อมีจำนวนกลุ่มที่มากเกินไปจะทำให้ต้องเสียเวลาในการคำนวณค่าฟังก์ชันขอบเขตล่างของกลุ่มข้อมูลแต่ละกลุ่มที่ได้นำเสนอในงานวิจัยนี้มากขึ้นไปด้วย ดังนั้นการทดลองนี้จึงสรุปได้ว่าควรเลือกการจับกลุ่มด้วยจำนวนกลุ่มเท่ากับ 1,024 กลุ่ม



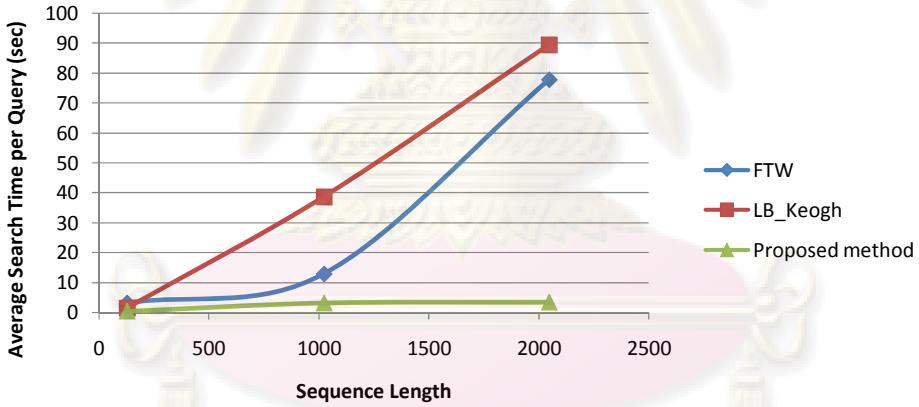
รูปที่ 4.18 ผลการทดลองการเลือกจำนวนกลุ่มในการจับกลุ่มของชุดข้อมูล RW II ด้วยวิธีการใช้ชุดตรวจสอบความสมเหตุสมผล (Validation Set) โดยใช้บนชุดตรวจสอบความสมเหตุสมผลที่ประกอบด้วย 20 อนุกรม บนชุดข้อมูลที่ประกอบด้วยข้อมูลจำนวนทั้งหมด 100,000 อนุกรมแต่ละข้อมูลมีความยาว 1,024 จุดข้อมูล ซึ่งมีการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน

รูปที่ 4.18 แสดงผลการทดลองที่คล้ายกับการทดลองจากในรูปที่ 4.17 เพียงแต่ทำการเปลี่ยนความยาวของข้อมูลทั้งหมดให้เป็น 1,024 จุดข้อมูล ซึ่งสังเกตได้ว่าผลการทดลองก็ยังคงคล้ายกับผลการทดลองบนชุดข้อมูลที่มีความยาวของแต่ละข้อมูล 128 จุดข้อมูล กล่าวคือเวลาที่ใช้ในการค้นคืนข้อมูลจะลดลงเมื่อทำการจับกลุ่มข้อมูลด้วยจำนวนกลุ่มที่เพิ่มขึ้นจนถึงค่าจำนวนกลุ่มเท่ากับ 1,024 โดยการจับกลุ่มข้อมูลด้วยจำนวนกลุ่มที่มากกว่านั้นมีแนวโน้มที่จะให้ผลของการค้นคืนข้อมูลที่แย่ลง โดยสรุปแล้วในชุดข้อมูลนี้จึงเลือกการจับกลุ่มด้วยจำนวนกลุ่มเท่ากับ 1,024 กลุ่ม ในกรณีเดียวกันสำหรับการทดลองเพื่อหาจำนวนกลุ่มข้อมูลที่เหมาะสมบนชุดข้อมูลที่แต่ละข้อมูลมีความยาวเท่ากับ 2,048 จุดข้อมูล จากผลการทดลองในรูปที่ 4.19 สามารถสรุปได้ว่าควรทำการจับกลุ่มด้วยจำนวนกลุ่มเท่ากับ 2,560 กลุ่ม

โดยสรุปแล้ว การทดลองที่ผ่านมาทั้งหมดในหัวข้อนี้เป็นการใช้ชุดตรวจสอบความสมเหตุสมผลสำหรับกำหนดจำนวนกลุ่มในการจับกลุ่มบนชุดข้อมูลทั้ง 3 ชุด ซึ่งก็คือชุดข้อมูล RW II ที่มีขนาดเท่ากันเท่ากับ 100,000 อนุกรม โดยแต่ละชุดข้อมูลจะมีความยาวของข้อมูลที่แตกต่างกัน ได้แก่ 128 1,024 และ 2,048 จุดข้อมูล ซึ่งผลสรุปที่ได้คือจำนวนกลุ่มข้อมูลสำหรับการจับกลุ่มในแต่ละชุดข้อมูลเท่ากับ 1,024 1,024 และ 2,048 กลุ่มข้อมูลตามลำดับ ซึ่งจะนำชุดข้อมูลทั้งสามนี้และค่าพารามิเตอร์ของแต่ละชุดข้อมูลไปใช้ในการทดสอบประสิทธิภาพในด้านความเร็วในการค้นคืนข้อมูลจริงโดยนำไปเปรียบเทียบกับวิธีการค้นคืนข้อมูลที่แต่ละวิธีเป็นที่ยอมรับว่าเป็นวิธีที่เร็วที่สุดในปัจจุบันวิธีหนึ่ง นั่นก็คือการค้นตามลำดับ (Sequential Search) ด้วยการใช้ฟังก์ชันขอบเขตล่างของระยะทางแบบไดนามิกไทม์วอร์ปปิง FTW [7] และ LB_Keogh [8] โดยในรูปที่ 4.20 แสดงผลการทดลองจากการทดลองดังกล่าว



รูปที่ 4.19 ผลการทดลองการเลือกจำนวนกลุ่มในการจับกลุ่มของชุดข้อมูล RW II ด้วยวิธีการใช้ชุดตรวจสอบความสมเหตุสมผล (Validation Set) โดยใช้บนชุดตรวจสอบความสมเหตุสมผลที่ประกอบด้วย 20 อนุกรม บนชุดข้อมูลที่ประกอบด้วยข้อมูลจำนวนทั้งหมด 100,000 อนุกรมแต่ละอนุกรมมีความยาว 2,048 จุดข้อมูล ซึ่งมีการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน



รูปที่ 4.20 ผลการทดลองเพื่อเปรียบเทียบเวลาในการค้นคืนข้อมูลบนชุดข้อมูล RW II ระหว่างวิธีการค้นทั้ง 3 วิธี ได้แก่ วิธีการค้นตามลำดับโดยใช้ฟังก์ชันขอบเขตล่างของค่าระยะทางไดนามิกไทม์วอร์ปปิงแบบ FTW และแบบ LB_Keogh กับวิธีที่ได้นำเสนอ โดยใช้ชุดข้อมูลสอบถามทั้งหมด 100 อนุกรม เพื่อทำการค้นบนชุดข้อมูลที่ประกอบด้วยข้อมูลจำนวนทั้งหมด 100,000 อนุกรมโดยแต่ละชุดข้อมูลมีความยาวของข้อมูล 128 1,024 และ 2,048 จุดข้อมูล

อย่างไรก็ตามเนื่องจากพื้นที่ที่ใช้ในการจัดเก็บชุดข้อมูลแต่ละชุดที่ทำการทดลองทั้งหมดมีขนาดมากที่สุดเพียงประมาณ 800 เมกะไบต์สำหรับชุดข้อมูลที่มีความยาวข้อมูล 2,048 จุดข้อมูล ซึ่งเป็นขนาดพื้นที่สามารถจัดเก็บไว้บนแรมได้ทั้งหมดสำหรับวิธีการค้นตามลำดับโดยใช้ฟังก์ชันขอบเขตล่างของค่าระยะทางไดนามิกไทม์วอร์ปปิงแบบ LB_Keogh

ดังนั้นการเข้าถึงข้อมูลในชุดข้อมูลจึงเป็นเพียงการอ่านค่าจากแรม เวลาที่เสียไปในส่วนของการอ่านข้อมูลจากการเข้าถึงข้อมูลแบบสุ่ม (Random Access) จึงไม่มีผลกระทบมากนัก เนื่องมาจากไม่ใช่การอ่านข้อมูลจากจานบันทึกแม่เหล็กเหมือนดังเช่นในการเข้าถึงข้อมูลจากฮาร์ดดิสก์ แต่สำหรับวิธี FTW นั้นต้องทำการจัดเก็บช่วงย่อยของข้อมูลของแต่ละข้อมูลเพิ่มขึ้นอีก จึงทำให้ขนาดพื้นที่จัดเก็บเพิ่มขึ้นกว่าเท่าตัว ดังนั้นการค้นคืนข้อมูลจึงต้องทำการเข้าถึงตัวข้อมูลจากฮาร์ดดิสก์ซึ่งทำให้เกิดความล่าช้าจากการเข้าถึงข้อมูลแบบสุ่ม จึงทำให้เวลาที่ต้องใช้ในการค้นคืนข้อมูลจากชุดข้อมูลดังกล่าวเพิ่มขึ้นอย่างมีนัยเมื่อเทียบกับการค้นคืนข้อมูลจากชุดข้อมูลที่มีขนาดเล็กกว่า ไม่ว่าอย่างไรวิธีการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอก็ยังคงใช้เวลาในการค้นคืนที่น้อยมากและเกือบจะคงที่สำหรับขนาดของข้อมูลที่เพิ่มขึ้น ซึ่งเร็วกว่าวิธี FTW มากที่สุดถึง 23 เท่าเลยทีเดียว และเนื่องด้วยขอบเขตของงานวิจัยนี้ได้เน้นไปที่การค้นคืนข้อมูลจากชุดข้อมูลขนาดใหญ่ ซึ่งหมายถึงชุดข้อมูลที่ขนาดใหญ่เกินกว่าที่จะสามารถจัดเก็บบนแรมทั้งหมดได้ ด้วยเหตุนี้การลดทอนการเข้าถึงข้อมูลลงด้วยการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอจะส่งผลต่อความเร็วในการค้นคืนมากยิ่งขึ้นไปกว่าเท่าที่สังเกตได้จากการทดลองในหัวข้อนี้ ดังนั้นในหัวข้อถัดไปจะทำการทดสอบประสิทธิภาพการค้นคืนข้อมูลบนชุดข้อมูลขนาดใหญ่

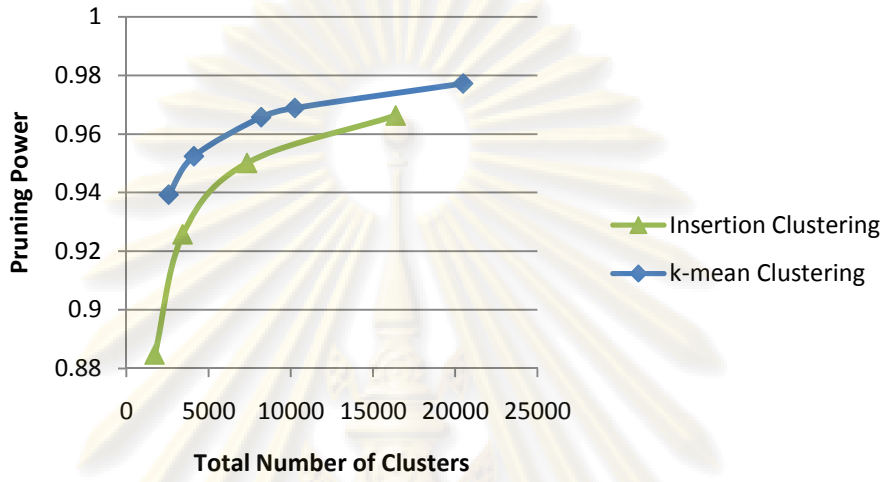
แต่สำหรับการค้นคืนข้อมูลบนชุดข้อมูลขนาดใหญ่นั้น การจับกลุ่มแบบเคมีนอาจเป็นปัญหาสำคัญเนื่องจากต้องใช้เวลาในการคำนวณสูงมาก ดังนั้นงานวิจัยนี้จึงได้ใช้การจับกลุ่มแบบแทรกแทนที่การจับกลุ่มแบบเคมีนซึ่งได้ในเสนอไว้ในหัวข้อที่ 3.5.2 ดังนั้นในหัวข้อต่อไปจะทำการทดสอบเพื่อเปรียบเทียบประสิทธิภาพของผลการจับกลุ่มระหว่างการจับกลุ่มแบบเคมีนกับการจับกลุ่มแบบแทรก โดยนำผลการจับกลุ่มของทั้งสองวิธีมาทำดัชนีสำหรับการค้นคืนข้อมูล และเปรียบเทียบเวลาที่ใช้ในการค้นคืนข้อมูลจากดัชนีดังกล่าว

4.8 การทดสอบเพื่อเปรียบเทียบประสิทธิภาพของผลการจับกลุ่มระหว่างการจับกลุ่มแบบเคมีนกับการจับกลุ่มแบบแทรก

การทดลองในหัวข้อนี้จะทำการทดสอบบนชุดข้อมูล RW II จำนวน 262,144 อนุกรม แต่ละข้อมูลมีความยาว 2,048 จุดข้อมูล สังเกตได้ว่าขนาดพื้นที่ที่ใช้ในการจัดเก็บชุดข้อมูลทั้งหมดมีขนาดถึง 2 กิกะไบต์ ซึ่งมีขนาดใหญ่เกินกว่าที่จะทำการจัดเก็บบนแรมที่มีขนาดเพียง 1 กิกะไบต์ได้ จึงถือได้ว่าเป็นชุดข้อมูลที่มีขนาดใหญ่เพียงพอสำหรับการทดสอบนี้

จากผลการทดลองในรูปที่ 4.21 จะเห็นได้ว่าการจับกลุ่มแบบเคมีนให้ผลการจับกลุ่มที่เมื่อนำไปทำดัชนีสำหรับการค้นคืนข้อมูลแล้ว สามารถลดทอนข้อมูลจากการค้นคืนได้มากกว่าการจับกลุ่มแบบแทรกเล็กน้อย แต่เนื่องจากว่าการจับกลุ่มแบบเคมีนในแต่ละครั้งนั้นต้องใช้เวลาในการคำนวณมากกว่า 3 วัน ซึ่งไม่เหมาะสมและไม่น่าจะเป็นที่ยอมรับได้ในการนำไปใช้งานจริงอย่างยิ่ง เมื่อเปรียบเทียบกับเวลาที่ใช้ในการจับกลุ่มแบบแทรกแล้ว การจับกลุ่มแบบแทรกสามารถทำได้อย่างรวดเร็วกว่ามาก โดยเวลาที่ใช้ในการจับกลุ่มแต่ละครั้งแสดงไว้ใน

รูปที่ 4.22 ซึ่งจะสังเกตได้ว่าการจับกลุ่มแบบแทรกนั้นใช้เวลามากที่สุดเพียง 7,866 วินาที แต่สำหรับรูปแบบการจับกลุ่มที่มีประสิทธิภาพที่สุดจะกำหนดค่าพารามิเตอร์สำหรับการจับกลุ่มแบบแทรกเท่ากับ 128 ซึ่งจะแสดงเป็นผลการทดลองในหัวข้อถัดไป เวลาที่ใช้ในการจับกลุ่มในรูปแบบที่เหมาะสมดังกล่าวใช้เวลาเพียง 2,851 วินาที



รูปที่ 4.21 ผลการทดลองเปรียบเทียบประสิทธิภาพของการจับกลุ่มระหว่างการจับกลุ่มแบบเคมีนกับการจับกลุ่มแบบแทรก โดยทำการทดสอบการค้นคืนข้อมูลสอบถามทั้งหมด 100 อนุกรม บนชุดข้อมูล RW II ขนาด 262,144 อนุกรม แต่ละอนุกรมมีความยาว 2,048 จุดข้อมูล ชุดข้อมูลดังกล่าวผ่านการจับกลุ่มแบบเคมีนและแบบแทรก ซึ่งมีการปรับค่าจำนวนกลุ่มในการจับกลุ่มที่แตกต่างกัน แล้วทำการวัดประสิทธิภาพในรูปแบบของกำลังการลดทอนจากการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอบนผลลัพธ์ของการจับกลุ่มแต่ละรูปแบบ

รูปที่ 4.22 เวลาที่ใช้ในการจับกลุ่มแบบแทรกบนชุดข้อมูล RW II ขนาด 262,144 อนุกรม แต่ละอนุกรมมีความยาว 2,048 จุดข้อมูล ด้วยค่าพารามิเตอร์ในการจับกลุ่มแบบแทรก PageSize ที่แตกต่างกัน

โดยสรุปแล้ว ถึงแม้การจับกลุ่มแบบแทรกจะให้ผลการลดทอนข้อมูลสำหรับการค้นคืนข้อมูลด้อยกว่าการจับกลุ่มแบบเคมีนเล็กน้อย แต่เนื่องด้วยเวลาที่ใช้ในการจับกลุ่มนั้นเร็วกว่าถึง 2 อันดับของขนาด (Order of Magnitude) ซึ่งนับว่าคุ้มค่ามากเมื่อแลกกับประสิทธิภาพที่ลดลงเพียงเล็กน้อยเท่านั้น ดังนั้นในการทดสอบการค้นคืนข้อมูลบนชุดข้อมูลขนาดใหญ่ที่จะกล่าวถึงในหัวข้อถัดไป จะเลือกใช้การจับกลุ่มแบบแทรกแทนที่การจับกลุ่มแบบเคมีน

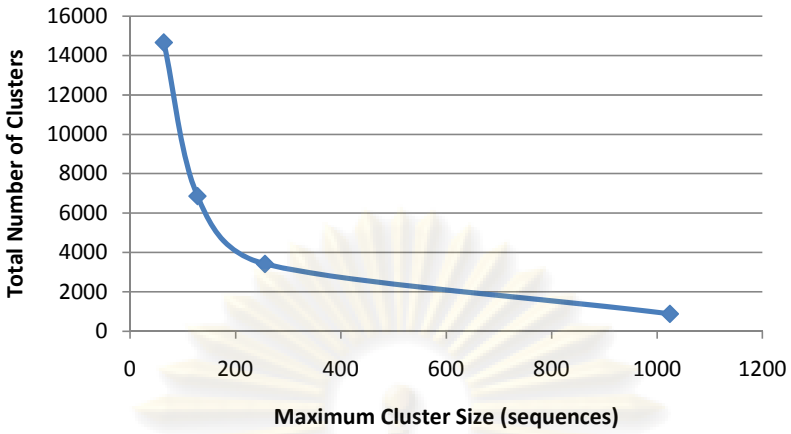
4.9 การทดสอบประสิทธิภาพการค้นคืนข้อมูลบนชุดข้อมูลขนาดใหญ่

ในการทดสอบการค้นคืนข้อมูลจากชุดข้อมูลขนาดใหญ่ จะใช้ชุดข้อมูล RWII ที่แต่ละชุดข้อมูลมีการปรับความยาวและจำนวนข้อมูลที่แตกต่างกัน แล้วใช้วิธีการจับกลุ่มแบบแทรกเพื่อทำการเตรียมข้อมูลก่อนการทำดัชนีสำหรับการค้นคืนข้อมูลด้วยวิธีที่ได้นำเสนอ โดยทำการเปรียบเทียบประสิทธิภาพกับวิธีการค้นคืนข้อมูลที่ใช้กันอยู่ในปัจจุบัน แต่ก่อนอื่นจะต้องทำการกำหนดค่าพารามิเตอร์สำหรับการจับกลุ่มซึ่งก็คือขนาดที่ใหญ่ที่สุดสำหรับแต่ละกลุ่มข้อมูลหรือค่า PageSize สำหรับการจับกลุ่มแบบแทรก ซึ่งได้กล่าวโดยละเอียดในหัวข้อที่ 3.5.2 โดยใช้ชุดข้อมูลตรวจสอบความสมเหตุสมผลในการกำหนดค่า PageSize ที่เหมาะสม

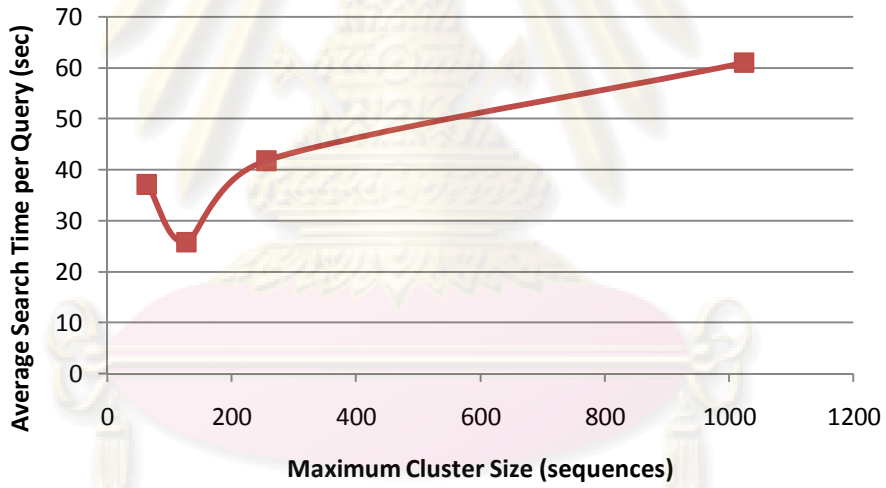
4.9.1 การทดสอบเพื่อกำหนดค่าพารามิเตอร์สำหรับการจับกลุ่มแบบแทรก

การทดลองนี้ทำการดำเนินงานบนเครื่องคอมพิวเตอร์ที่ใช้ซีพียู Intel Core 2 Duo E4600 ความเร็ว 2.40 กิกะเฮิรตซ์ และใช้แรมขนาด 2 กิกะไบต์ ภายใต้ระบบปฏิบัติการ Windows XP โดยเริ่มต้นจากการทดสอบเพื่อกำหนดค่าพารามิเตอร์สำหรับการจับกลุ่มแบบแทรกจะทำการกำหนดค่า PageSize สำหรับการจับกลุ่ม โดยทำการทดสอบบนชุดข้อมูล RWII ขนาด 524,288 อนุกรม แต่ละข้อมูลมีความยาว 2,048 จุดข้อมูล โดยทำการปรับเปลี่ยนค่า PageSize เป็น 64 128 256 และ 1,024 โดยจำนวนกลุ่มที่ได้จากการปรับค่า PageSize แต่ละค่าได้แสดงไว้ในรูปที่ 4.23

รูปที่ 4.24 แสดงผลการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอบนชุดข้อมูลที่ได้จากการจับกลุ่มด้วยค่าพารามิเตอร์ PageSize ต่าง ๆ ซึ่งสังเกตได้ว่าการจับกลุ่มแบบแทรกด้วยค่า PageSize เท่ากับ 128 นั้นให้ผลการค้นคืนข้อมูลที่เร็วที่สุด โดยสรุปแล้วการทดลองในหัวข้อนี้ได้บ่งบอกว่าการจับกลุ่มแบบแทรกด้วยค่า PageSize เท่ากับ 128 นั้นน่าจะเป็นการจับกลุ่มที่เหมาะสมที่สุด ดังนั้นการทดลองต่อจากที่จะทำการกำหนดค่า PageSize สำหรับการจับกลุ่มในทุกชุดข้อมูลเท่ากับ 128



รูปที่ 4.23 จำนวนกลุ่มข้อมูลที่ได้จากการจับกลุ่มแบบแทรกด้วยการกำหนดค่าพารามิเตอร์ในการจับกลุ่มที่แตกต่างกันซึ่งก็คือขนาดที่ใหญ่ที่สุดสำหรับแต่ละกลุ่มข้อมูลบนชุดข้อมูล RW II ขนาด 524,288 อนุกรม แต่ละอนุกรมมีความยาว 2,048 จุดข้อมูล

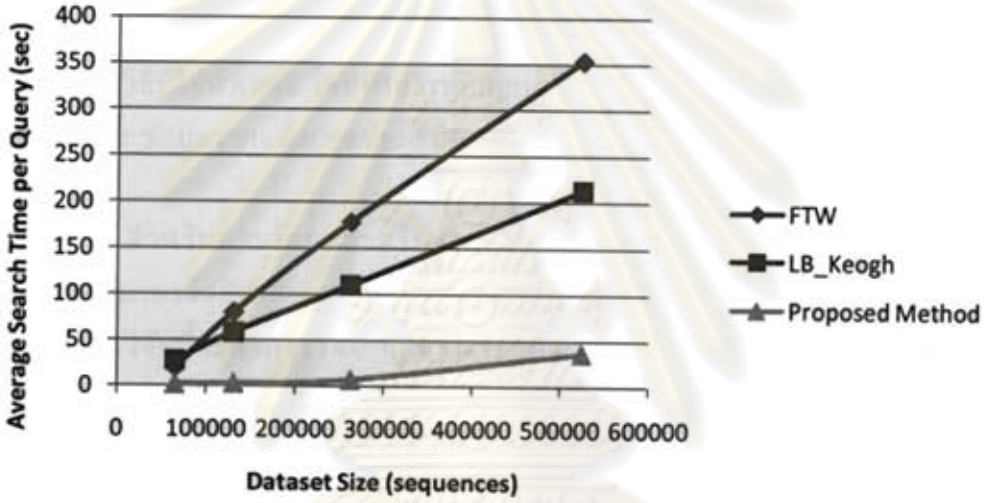


รูปที่ 4.24 ผลการทดลองการจับเวลาที่ใช้ในการค้นคืนข้อมูลจากชุดข้อมูล RW II ขนาด 524,288 อนุกรม แต่ละอนุกรมมีความยาว 2,048 จุดข้อมูล โดยทำการปรับค่าค่าพารามิเตอร์ในการจับกลุ่มที่แตกต่างกันซึ่งก็คือขนาดที่ใหญ่ที่สุดสำหรับแต่ละกลุ่มข้อมูล

4.9.2 การทดสอบเพื่อเปรียบเทียบประสิทธิภาพในการค้นคืนข้อมูลระหว่างวิธีการค้นคืนด้วยดัชนีที่ได้นำเสนอกับวิธีที่ใช้กันอยู่ในปัจจุบัน

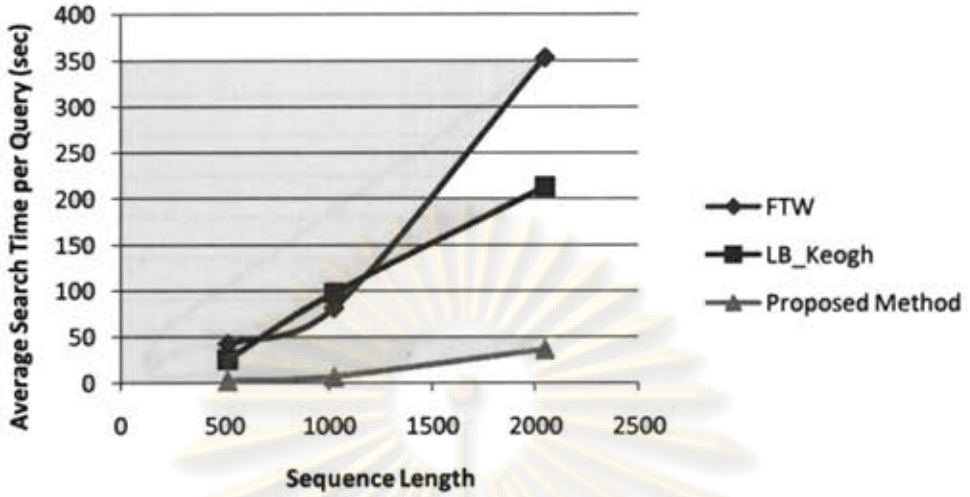
การทดลองนี้ทำการดำเนินงานบนเครื่องคอมพิวเตอร์ที่ใช้ซีพียู Intel Core 2 Duo E7300 ความเร็ว 2.66 กิกะเฮิรตซ์ และใช้แรมขนาด 2 กิกะไบต์ งานทั้งหมดดำเนินงานภายใต้ระบบปฏิบัติการ Windows XP โดยการทดลองในหัวข้อนี้จะทำการเปรียบเทียบ

ประสิทธิภาพในด้านความเร็วในการค้นคืนข้อมูลระหว่างวิธีที่ได้นำเสนอกับวิธีที่มีอยู่ในปัจจุบัน ได้แก่ วิธีการค้นคืนข้อมูลตามลำดับโดยใช้ฟังก์ชันขอบเขตล่างของระยะทางแบบไดนามิกไทม์วอร์ปปิงด้วยวิธี FTW และ LB_Keogh โดยในรูปที่ 4.25 แสดงผลการเปรียบเทียบประสิทธิภาพในด้านความเร็วการค้นคืนข้อมูลบน ชุดข้อมูล RW II ที่มีขนาดแตกต่างกัน แต่ละข้อมูลจากทุกชุดข้อมูลมีความยาวเท่ากันเท่ากับ 2,048 จุด สังเกตได้ว่าวิธีการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอสามารถค้นคืนข้อมูลได้รวดเร็วกว่าวิธีอื่นอย่างเห็นได้ชัดในทุกขนาดของชุดข้อมูล ไม่ว่าจะเป็นชุดข้อมูลขนาดเล็กที่มีจำนวนข้อมูลเพียง 65,536 อนุกรมซึ่งสามารถจัดเก็บไว้ในแรมได้อย่างง่ายดาย หรือว่าจะเป็นชุดข้อมูลขนาดใหญ่ที่สุดซึ่งมีจำนวนข้อมูลกว่า 524,288 อนุกรมซึ่งใช้ขนาดพื้นที่ในการจัดเก็บกว่า 4 กิกะไบต์



รูปที่ 4.25 ผลการทดลองการเปรียบเทียบความเร็วในการค้นคืนข้อมูลระหว่างวิธี FTW LB_Keogh และวิธีที่ได้นำเสนอ โดยทำการทดสอบบนชุดข้อมูล RW II ที่มีขนาดแตกต่างกัน แต่ละข้อมูลจากทุกชุดข้อมูลมีความยาวเท่ากันเท่ากับ 2,048 จุดข้อมูล

ส่วนในรูปที่ 4.26 เป็นผลการทดลองการค้นคืนข้อมูลที่มีการปรับความยาวของข้อมูล โดยแต่ละชุดข้อมูลมีขนาดเท่ากันเท่ากับ 524,288 อนุกรม สังเกตได้ว่าวิธีที่นำเสนอก็ยังคงสามารถค้นคืนได้เร็วกว่าวิธีอื่นอย่างเห็นได้ชัด โดยความยาวที่เพิ่มขึ้นของข้อมูลนั้นมีผลทำให้เวลาในการค้นคืนข้อมูลด้วยวิธี FTW นั้นเพิ่มขึ้นในระดับที่สูงกว่าฟังก์ชันเชิงเส้นอย่างเห็นได้ชัด เนื่องจากฟังก์ชันการประมาณค่าขอบเขตล่างของระยะทางไดนามิกไทม์วอร์ปปิงด้วยวิธี FTW นั้นเป็นการคำนวณด้วยไดนามิกไทม์วอร์ปปิงเช่นกัน ดังนั้นขีดจำกัดเชิงสัญกรณ์ในด้านเวลาของการแทนที่การคำนวณระยะทางไดนามิกไทม์วอร์ปปิงด้วยค่าขอบเขตล่างก็ยังคงอยู่ในระดับฟังก์ชันพหุนามกำลังสอง ซึ่งแตกต่างกับการประมาณค่าขอบเขตล่างด้วยระยะทางยุคลิดดังเช่นในวิธี LB_Keogh และวิธีที่ได้นำเสนอที่ทำให้เวลาที่ใช้ในการค้นคืนข้อมูลเพิ่มขึ้นในระดับเชิงเส้น

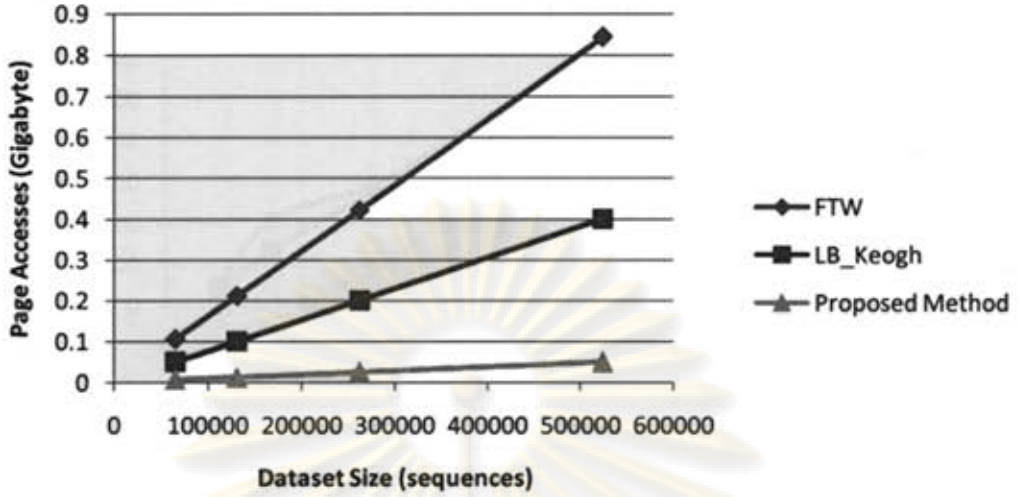


รูปที่ 4.26 ผลการทดลองการเปรียบเทียบความเร็วในการค้นคืนข้อมูลระหว่างวิธี FTW LB_Keogh และวิธีที่ได้นำเสนอ โดยทำการทดสอบบนชุดข้อมูล RWII ที่มีขนาดเท่ากันเท่ากับ 524,288 อนุกรม ข้อมูลจากแต่ละชุดข้อมูลมีความยาวที่แตกต่างกัน

สำหรับการวัดผลสำหรับงานในด้านการทำดัชนีสำหรับการค้นคืนข้อมูลนั้น โดยทั่วไปมักใช้มาตรวัดอินพุต/เอาต์พุตในรูปแบบของจำนวนหน้าข้อมูลที่ทำการเข้าถึง (Page Access) ในงานวิจัยนี้จะทำการวัดจำนวนหน้าข้อมูลที่ทำการเข้าถึงในรูปแบบของขนาดของข้อมูลที่ต้องทำการเข้าถึงทั้งหมดในรูปแบบของพื้นที่ที่ใช้จัดเก็บ โดยได้คำนึงถึงตัวประกอบที่มีผลต่อการเข้าถึงตามลำดับ (Sequential Access) [28] โดยใช้ค่าตัวประกอบเท่ากับ 10 ซึ่งหมายถึงการคิดค่าจำนวนหน้าข้อมูลที่ทำการเข้าถึงจากการเข้าถึงตามลำดับจะมีผลเพียงแค่ 1 ใน 10 ของการเข้าถึงข้อมูลแบบสุ่ม โดยค่าจำนวนหน้าข้อมูลที่ทำการเข้าถึงทั้งหมดสามารถคำนวณได้ตามสมการ (4.4)

$$PageAccess = \frac{SequentialAccess}{10} + RandomAccess \qquad (4.4)$$

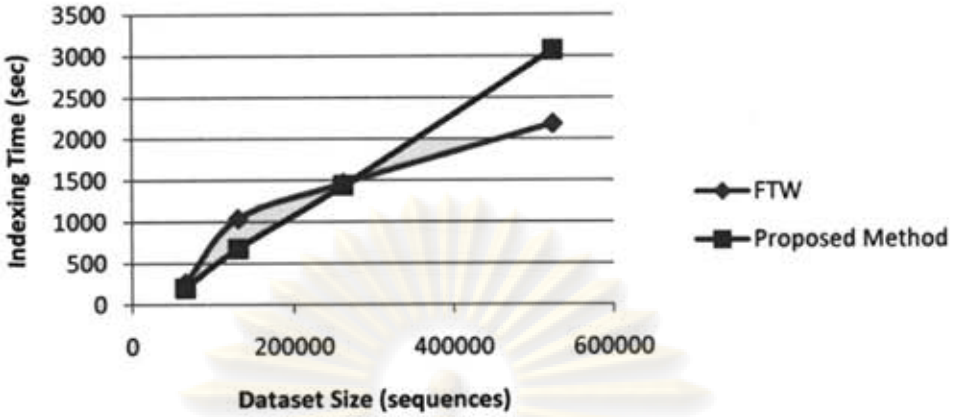
รูปที่ 4.27 แสดงการเปรียบเทียบจำนวนหน้าข้อมูลที่ทำการเข้าถึงระหว่างวิธี FTW LB_Keogh และวิธีที่ได้นำเสนอ ซึ่งอันที่จริงแล้ววิธี LB_Keogh นั้นเป็นวิธีการค้นคืนข้อมูลตามลำดับโดยไม่มีการทำดัชนีใด ๆ ทั้งสิ้น จึงถือเป็นการเข้าถึงข้อมูลตามลำดับทั้งหมด ค่าจำนวนหน้าข้อมูลที่ทำการเข้าถึงจึงมีขนาด 1 ใน 10 ของขนาดของชุดข้อมูล สำหรับวิธี FTW นั้น เนื่องจากต้องทำการกำหนดช่วงย่อยของข้อมูลเพิ่มเติม จึงทำให้ขนาดของข้อมูลที่ต้องทำการเข้าถึงเพิ่มขึ้น แต่สำหรับวิธีการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอนั้นเพียงแค่ทำการเข้าถึงขอบเขตของกลุ่มข้อมูลทั้งหมด และข้อมูลจริงเฉพาะในบางกลุ่มที่ไม่ได้ถูกลดทอนด้วยดัชนี ดังนั้นค่าจำนวนหน้าข้อมูลที่ทำการเข้าถึงจึงน้อยกว่าวิธีอื่นทั้งสองวิธีอย่างเห็นได้ชัด



รูปที่ 4.27 ผลการทดลองการเปรียบเทียบจำนวนหน้าข้อมูลที่ทำการเข้าถึงในการค้นคืนข้อมูลระหว่างวิธี FTW LB_Keogh และวิธีที่ได้นำเสนอ โดยทำการทดสอบบนชุดข้อมูล RWII ที่มีขนาดแตกต่างกัน แต่ละข้อมูลจากทุกชุดข้อมูลมีความยาวเท่ากันเท่ากับ 2,048 จุดข้อมูล และทำการเฉลี่ยเวลาที่ใช้ในการค้นคืนข้อมูลสอบถามทั้งหมด 100 อนุกรม

สังเกตได้ว่าผลการทดลองเพื่อเปรียบเทียบจำนวนหน้าข้อมูลที่ทำการเข้าถึงในรูปที่ 4.27 นั้นมีแนวโน้มที่คล้ายกับผลการทดลองเพื่อเปรียบเทียบเวลาที่ใช้ในการค้นคืนข้อมูลในรูปที่ 4.25 ซึ่งหมายความว่าการวัดประสิทธิภาพในการค้นคืนข้อมูลจากในชุดข้อมูลขนาดใหญ่สามารถวัดได้จากจำนวนหน้าข้อมูลที่ทำการเข้าถึงโดยต้องคำนึงถึงตัวประกอบที่มีผลต่อการเข้าถึงตามลำดับด้วย การใช้มาตรวัดในรูปแบบดังกล่าวนั้นดีกว่าการใช้วิธีการจับเวลาที่ใช้ในการค้นคืนข้อมูล เนื่องจากการจับเวลานั้นอาจเกิดข้อผิดพลาดขึ้นได้ โดยสรุปแล้ววิธีการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอนั้นมีประสิทธิภาพสูงกว่าวิธีอื่นถึงกว่าสิบเท่า

ถ้าจะมองในแง่มุมของเวลาที่ใช้ในการจัดเตรียมข้อมูล ถึงแม้จะสามารถคำนวณจากการประมวลผลออฟไลน์ (Offline Processing) วิธีการทำดัชนีที่ได้นำเสนอก็ยังคงใช้เวลาไม่มากนักเมื่อเทียบกับเวลาในการเตรียมข้อมูลด้วยวิธี FTW ซึ่งต้องทำการการแบ่งช่วงของข้อมูลแต่ละตัวออกเป็นช่วงย่อย ๆ โดยผลการทดลองดังกล่าวแสดงไว้ในรูปที่ 4.28 สังเกตได้ว่าเวลาที่ใช้ในการเตรียมข้อมูลในวิธีที่ได้นำเสนอหรือก็คือการจับกลุ่มข้อมูลแบบแทรกมีแนวโน้มที่จะเพิ่มขึ้นในระดับฟังก์ชันเชิงเส้นเมื่อเทียบกับขนาดของชุดข้อมูลที่เพิ่มขึ้น ดังนั้นวิธีการจับกลุ่มแบบแทรกจึงสามารถรองรับการขยายขนาดของจำนวนข้อมูลจากในชุดข้อมูลได้เป็นอย่างดี



รูปที่ 4.28 ผลการทดลองเพื่อเปรียบเทียบเวลาที่ใช้ในการเตรียมข้อมูลระหว่างวิธี FTW กับวิธีการทำดัชนีที่ได้นำเสนอ

บทที่ 5

สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

งานวิจัยนี้ได้นำเสนอการทำดัชนีสำหรับการค้นคืนข้อมูลอนุกรมเวลาตามความคล้ายจากชุดข้อมูลขนาดใหญ่ โดยใช้มาตรวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงเป็นตัวกำหนดความคล้ายกันของข้อมูล วิธีการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอสามารถค้นคืนข้อมูลจากในชุดข้อมูลขนาดใหญ่ได้ในเวลาอันสั้น และเร็วกว่าวิธีการค้นคืนข้อมูลที่มีอยู่ในปัจจุบันอย่างเห็นได้ชัดจากผลการทดลองที่ได้นำเสนอไว้ในบทที่ 4 โดยจากผลการทดลองทั้งหมดสามารถสรุปได้ดังนี้

5.1 สรุปผลการวิจัย

วิธีการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอเหมาะสำหรับการค้นคืนข้อมูลอนุกรมที่มีความราบเรียบและมีความเป็นสัญญาณรบกวนน้อย และข้อมูลแต่ละตัวจากในชุดข้อมูลควรมีความแตกต่างซึ่งกันและกัน กล่าวคือสำหรับชุดข้อมูลที่ข้อมูลแต่ละตัวมีความคล้ายคลึงกันเองเป็นส่วนมาก หรือในกรณีที่ข้อมูลมีความเป็นสัญญาณรบกวนสูง จะทำให้การแยกข้อมูลแต่ละตัวออกจากกันด้วยดัชนีนั้นเป็นไปได้ยาก ด้วยเหตุนี้การค้นคืนข้อมูลด้วยดัชนีจึงไม่สามารถระบุตำแหน่งในการค้นที่ถูกต้องเหมาะสมได้ โดยสรุปแล้วข้อมูลในลักษณะดังกล่าวจะเป็นอุปสรรคสำหรับวิธีที่ได้นำเสนอในงานวิจัยนี้

การค้นคืนข้อมูลตามความคล้ายโดยใช้ระยะทางแบบไดนามิกไทม์วอร์ปปิงเป็นตัวบ่งบอกความคล้ายกันของข้อมูลนั้น ถ้าชุดข้อมูลมีขนาดใหญ่เกินกว่าจะสามารถจัดเก็บไว้บนแรมได้ เวลาที่ต้องเสียไปในการค้นคืนข้อมูลส่วนมากมักขึ้นกับเวลาที่ใช้ในการเข้าถึงข้อมูล เนื่องจากถึงแม้ขีดจำกัดเชิงสัญกรณ์ในการคำนวณระยะทางแบบไดนามิกไทม์วอร์ปปิงจะสูงถึง $O(n^2)$ ก็ตาม แต่ด้วยการแทนที่การคำนวณไดนามิกไทม์วอร์ปปิงด้วยค่าระยะทางขอบเขตล่างจากวิธีที่มีอยู่ในปัจจุบันนั้น อาจกล่าวได้ว่าขีดจำกัดเชิงสัญกรณ์นั้นแทบจะลดเหลือเพียงแค่ฟังก์ชันเชิงเส้นเท่านั้น [3] ดังนั้นการลดจำนวนข้อมูลที่ต้องทำการเข้าถึงข้อมูลจึงส่งผลต่อประสิทธิภาพด้านความเร็วในการค้นคืนมากที่สุด และด้วยวิธีการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอ สามารถลดขนาดของข้อมูลที่ต้องทำการเข้าถึงได้มากที่สุดถึง 17 เท่า

ระบบการค้นคืนข้อมูลที่ใช้วิธีการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอนั้นสามารถรองรับกับการเปลี่ยนแปลงของข้อมูลได้เป็นอย่างดีไม่ว่าจะเป็นการเพิ่มหรือลบข้อมูลจากชุดข้อมูล ด้วยวิธีการลดทอนการคำนวณบนการจับกลุ่มแบบเคมีนที่ได้นำเสนอ ทำให้สามารถปรับเปลี่ยนรูปแบบของกลุ่มข้อมูลได้ตามการเปลี่ยนแปลงข้อมูลได้อย่างรวดเร็วโดยประสิทธิภาพของกลุ่มข้อมูลยังคงดีใกล้เคียงเดิม

วิธีการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอสามารถรองรับกับขนาดของชุดข้อมูลที่เพิ่มมากขึ้นได้เป็นอย่างดี ไม่ว่าจะเป็นการเพิ่มความยาวของข้อมูลซึ่งโดยปกติมักเป็นปัญหาใหญ่สำหรับการทำดัชนีการค้นคืนข้อมูลเนื่องจากปัญหาคำสาปเชิงมิติ (Curse of Dimensionality) หรือว่าจะเป็นการเพิ่มขึ้นของจำนวนข้อมูลซึ่งโดยปกติมักเป็นปัญหาสำหรับการจับกลุ่มข้อมูล แต่ด้วยวิธีการจับกลุ่มแบบแทรกซึ่งได้นำเสนอในงานวิจัยนี้ ทำให้สามารถจับกลุ่มข้อมูลได้อย่างรวดเร็ว เนื่องจากเวลาที่ใช้ในการจับกลุ่มแบบแทรกนั้นมีแนวโน้มที่จะเพิ่มขึ้นในระดับฟังก์ชันเชิงเส้นสำหรับการเพิ่มขึ้นของจำนวนข้อมูลในชุดข้อมูล นอกจากนี้การปรับค่าพารามิเตอร์ในการจับกลุ่มแบบแทรกสามารถลดเวลาในการคำนวณลงได้ โดยแลกกับกลุ่มข้อมูลที่มีประสิทธิภาพลดลงบ้าง

5.2 ข้อเสนอแนะ

การทำดัชนีบนข้อมูลที่ผ่านการจับกลุ่มโดยใช้มาตรวัดระยะทางแบบยุคลิดอาจดูไม่สมเหตุสมผลสำหรับวัตถุประสงค์ที่แท้จริงในการนำกลุ่มข้อมูลไปใช้ นั่นก็คือการคำนวณค่าขอบเขตล่างของระยะทางแบบไดนามิกไทม์วอร์ปปิง แต่เนื่องจากปัจจุบันยังไม่สามารถนำระยะทางแบบไดนามิกไทม์วอร์ปปิงไปใช้ในการจับกลุ่มข้อมูลได้เนื่องจากระยะทางแบบไดนามิกไทม์วอร์ปปิงไม่มีคุณสมบัติของความเป็นมาตรวัดเมตริก กล่าวคือไม่มีคุณสมบัติของอสมการสามเหลี่ยม (Triangular Inequality) นอกจากนี้ในปัจจุบันยังไม่มีวิธีหาจุดศูนย์กลางของกลุ่มข้อมูลที่ใช้มาตรวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงที่มีประสิทธิภาพ ดังนั้นถ้าหากปัญหาเหล่านี้ถูกแก้ได้อย่างมีประสิทธิภาพแล้ว อาจส่งผลให้สามารถจับกลุ่มด้วยระยะทางแบบไดนามิกไทม์วอร์ปปิงได้อย่างมีประสิทธิภาพ และส่งผลให้มีแนวโน้มที่จะได้ค่าประมาณขอบเขตล่างสำหรับค่าระยะทางไดนามิกไทม์วอร์ปปิงของกลุ่มข้อมูลที่มีค่าใกล้เคียงกับระยะทางแบบไดนามิกไทม์วอร์ปปิงจริงมากยิ่งขึ้น ซึ่งส่งผลให้สามารถลดทอนข้อมูลสำหรับการค้นคืนข้อมูลได้มากยิ่งขึ้นไปอีก

นอกจากนี้วิธีการค้นคืนด้วยดัชนีที่ได้นำเสนอยังมีข้อจำกัดอยู่ที่สามารถค้นคืนข้อมูลด้วยการเปรียบเทียบข้อมูลแบบทั้งความยาวข้อมูลเท่านั้น ในหลาย ๆ งานประยุกต์มีความต้องการที่จะค้นหาลำดับย่อยของข้อมูลที่มีความคล้ายกับข้อมูลสอบถามมากที่สุด ซึ่งวิธีการค้นคืนข้อมูลด้วยดัชนีที่ได้นำเสนอไม่สามารถแก้ปัญหาดังกล่าวได้เนื่องจากจะติดอยู่ที่ปัญหาในการจับกลุ่มลำดับย่อยของข้อมูล (Subsequence Clustering) เนื่องจากในปัจจุบันยังไม่มีวิธีการจับกลุ่มที่มีประสิทธิภาพในการจับกลุ่มลำดับย่อยของข้อมูลที่ให้ผลการจับกลุ่มที่มีความหมาย [29] ดังนั้นถ้าสามารถแก้ปัญหาการจับกลุ่มลำดับย่อยของข้อมูลได้ ก็จะทำให้วิธีการค้นคืนข้อมูลด้วยดัชนีที่ได้เสนอสามารถรองรับกับการค้นคืนข้อมูลที่เป็นลำดับย่อย (Subsequence Matching) ได้

รายการอ้างอิง

[1] Zhu, Y., and Shasha, D. (2003). Warping indexes with envelope transforms for query by humming. Proceedings of the 2003 ACM SIGMOD international conference on Management of data, pp. 181–192. San Diego, CA, USA: ACM Press.

[2] Ghias, A., Logan, J., Chamberlin, D., and Smith, B.C. (1995). Query by humming: musical information retrieval in an audio database. Proceedings of 3rd ACM international conference on Multimedia, pp. 231–236. San Francisco, CA, USA: ACM Press.

[3] Ratanamahatana, C.A., and Keogh, E. (2005). Three Myths about Dynamic Time Warping. Proceedings of SIAM International Conference on Data Mining, pp. 506–510. Newport Beach, CA, USA: SIAM.

[4] Keogh, E. (2002). Exact Indexing of Dynamic Time Warping. Proceedings of 28th International Conference on Very Large Data Bases, pp. 406–417. Hong Kong, China: VLDB Endowment.

[5] Yi, B.-K., Jagadish, H.V., and Faloutsos, C. (1998). Efficient Retrieval of Similar Time Sequences under Time Warping. Proceedings of 14th International Conference on Data Engineering, pp. 201–208. Orlando, FL, USA.

[6] Kim, S.-W., Park, S., and Chu, W.W. (2001). An Index-based Approach for Similarity Search Supporting Time Warping in Large Sequence Databases. Proceedings of 17th International Conference on Data Engineering, pp. 607–614. Heidelberg, Germany: IEEE Computer Society.

[7] Sakurai, Y., Yoshikawa, M., and Faloutsos, C. (2005). FTW: Fast Similarity Search under the Time Warping Distance. Proceedings of 24th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 326–337. Baltimore, MA, USA: ACM Press.

[8] Keogh, E., and Ratanamahatana, C.A. (2005). Exact Indexing of Dynamic Time Warping. Knowledge and Information Systems (KAIS) 7: pp. 358–386.

[9] Agrawal, R., Faloutsos, C., and Swami, A. (1993). Efficient Similarity Search In Sequence Databases. Proceedings of 4th International Conference of Foundations of Data Organization and Algorithms (FODO), pp. 69–84. Chicago, IL, USA: Springer-Verlag.

[10] Wei, L., Keogh, E., Herle, H.V., and Mafra-Neto, A. (2005). Atomic Wedgie: Efficient Query Filtering for Streaming Time Series. Proceedings of 5th IEEE International Conference on Data Mining pp. 490–497. Houston, TX, USA: IEEE Computer Society.

[11] Guttman, A. (1984). R-Trees: A Dynamic Index Structure for Spatial Searching. Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 1984), pp. 47–57. Boston, MA, USA: Morgan Kaufmann Publishers Inc.

[12] Beckmann, N., Kriegel, H.-P., Schneider, R., and Seeger, B. (1990). The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles. Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, pp. 322–331. Atlantic City, NJ, USA: ACM.

[13] Itakura, F. (1975). Minimum Prediction Residual Principle Applied to Speech Recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing 23: pp. 67–72.

[14] Sakoe, H., and Chiba, S. (1978). Dynamic Programming Algorithm Optimization for Spoken Word Recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing 26: pp. 43–49.

[15] Dunn , J.C. (1974). Well-Separated Clusters and Optimal Fuzzy Partitions. Cybernetics and Systems 4: pp. 95–104.

[16] Davies, D.L., and Bouldin, W. (1979). A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence 1: pp. 224–227.

[17] Halkidi, M., Vazirgiannis, M., and Batistakis, Y. (2000). Quality Scheme Assessment in the Clustering Process. Proceedings of 4th European Conference on Principles of Data Mining and Knowledge Discovery, pp. 265–276. Lyon, France: Springer-Verlag.

[18] Halkidi, M., and Vazirgiannis, M. (2001). Clustering Validity Assessment: Finding the Optimal Partitioning of a Data Set. Proceedings of the 2001 IEEE International Conference on Data Mining, pp. 187–194. San Jose, CA, USA: IEEE Computer Society.

[19] Hellerstein, J.M., Koutsoupias, E., and Papadimitriou, C.H. (1997). On the analysis of indexing schemes. Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, pp. 249–256. Tucson, AZ, USA: ACM.

[20] Roussopoulos, N., Kelley, S., and Vincent, F. (1995). Nearest neighbor queries. Proceedings of the 1995 ACM SIGMOD international conference on Management of data, pp. 71–79. San Jose, CA, USA: ACM.

[21] Quanzhong, L., Ines Fernando Vega, L., and Bongki, M. (2004). Skyline Index for Time Series Data. IEEE Trans. on Knowl. and Data Eng. 16: pp. 669-684.

[22] Ratanamahatana, C.A., and Keogh, E. (2007). Indexing and Mining Large Time Series Databases. Tutorial at 12th International Conference on Database Systems for Advanced Applications (DASFAA 2007), Bangkok, Thailand.

[23] Keogh, E., Xi, X., Wei, L., and Ratanamahatana, C.A. (2009). The UCR Time Series Classification/Clustering Homepage [Online]. Available from: www.cs.ucr.edu/~eamonn/time_series_data/ [1 Jan 2009]

[24] Chan, F.K.-P., Fu, A.W.-C., and Yu, C. (2003). Haar Wavelets for Efficient Similarity Search of Time-Series: With and Without Time Warping. IEEE Transactions on Knowledge and Data Engineering 15: pp. 686–705.

[25] Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S. (2001). Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases. Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, pp. 151–162. Santa Barbara, CA, USA: ACM.

[26] Ira, A., Ralph, K., Farzad, A., and Thomas, S. (2008). The TS-tree: efficient time series search and retrieval. Proceedings of 11th International Conference on Extending Database Technology: Advances in Database Technology, pp. 252–263. Nantes, France: ACM.

[27] Saito Naoki. (1994). Local feature extraction and its applications using a library of bases. Doctoral dissertation, Department of Mathematics, Yale University.

[28] Weber, R., Schek, H.-J., and Blott, S. (1998). A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. Proceedings of 24th International Conference on Very Large Data Bases, pp. 194–205. New York City, NY, USA: Morgan Kaufmann Publishers Inc.

[29] Keogh, E., Lin, J., and Truppel, W. (2003). Clustering of Time Series Subsequences is Meaningless: Implications for Previous and Future Research. Proceedings of 3rd IEEE International Conference on Data Mining, pp. 115–122. Melbourne, FL, USA: IEEE Computer Society.

ภาคผนวก

ภาคผนวก ก

บทความทางวิชาการเรื่อง “Efficient Similarity Search Under Fast Index Structure for Time Series Data” โดยพงศกร เรืองรองหิรัญญา วิชญ์ เนียรนาทตระกูล และ โชติรัตน์ รัตนามหัทธนะ ในงานประชุมวิชาการ “12th National Computer Science and Engineering Conference” ซึ่งจัดขึ้น ณ จังหวัดชลบุรี ประเทศไทย ระหว่างวันที่ 20 – 21 พฤศจิกายน 2551

Efficient Similarity Search Under Fast Index Structure for Time Series Data

Pongsakorn Ruengronghirunya Vit Niennattrakul Chotirat Ann Ratanamahatana
Department of Computer Engineering, Chulalongkorn University
254 Phyathai Road, Patumwan, Bangkok Thailand. 10330
{g51prn, g49vnn ,ann}@cp.eng.chula.ac.th

Abstract

As time series has become one of the most prevalent types of data in this digital age, necessity of similarity search for these applications is countless, including multimedia data retrieval and classification, and a Query-by-Humming system. Dynamic Time Warping (DTW) is one of the most accurate distance measure exploited in the search. However, DTW distance is known to suffer from such a high computational cost, raising great amount of interest in trying to speed it up. Recently, Fast Search Method for Dynamic Time Warping (FTW) was proposed and has become one of the most efficient similarity search method. In this paper, we will point out its drawback of sequential search exploited within, and propose a new Fast Index Structure (FIS) which could improve any sequential search performance, including the one in FTW. Our experimental results demonstrate our speedup as high as 21 times over FTW, while eliminating up to 87% of the candidates sequences for the search.

Key Words: Similarity Search, Dynamic Time Warping, Time Series Data

1. Introduction

With an explosion of data essentially in every domain around us, large portions of these data are in the forms of data sequences, temporal data, or time series, such as statistical data, multimedia data, etc. In turn, an accurate and efficient similarity search mechanism is necessary.

Particularly, in time series domains, distance-based similarity search methods have been largely demonstrated to work exceptionally well, both in terms of efficiency and effectiveness. One of the most popular distance measures, due to its simplicity, is Euclidean distance metric. Unfortunately, it is not quite robust on the data with variation within time axis, apart from the constraint that both query and candidate time series must have the same length. An alternative distance measurement is Dynamic Time Warping (DTW) distance measure [1], a more flexible and accurate distance measure for time series data that exhibit some shifting in the time axis. However, DTW distance has a major drawback with its large complexity of $O(n^2)$, where n is the length of each time series data, giving rise to significant amount of research in an attempt to speed up DTW calculation.

A classic approach to speed up DTW computation is time series indexing, with a capability to prune out large amount of candidate sequences with a relatively small computational cost. Much research [2-6] has proposed ways to index time series, making time series mining with DTW distance much more feasible in practice. For example, Zhu et al. [3] proposed a method which calculates lower bounding distance by reducing dimensionality of the data from length n to m where $m << n$, which could reduce the time complexity down by a few orders of magnitude. However, tightness of this lower bounding function is not optimal because dimensionality reduction is simply an approximated calculation of the time series distance, which could be far off the real DTW distance.

Wei et al. [7] proposed a method to reduce the amount of data being considered during the search by formulating a hierarchical index structure. However, this hierarchical structure is a tree-based structure typically contains large number of nodes, each of which is part of overall lower bounding calculations. As a result, this leads to a large computational cost.

In addition to the similarity matching problem, there are some other issues to be considered. For example, an I/O for data retrieval can pose as an overhead cost that could significantly degrade the performance of similarity search algorithms. Since the long proposed multidimensional index structure such as R-Tree [8] and R*-Tree [9] could lead to a large I/O overhead, recent work has tried to resolve this problem by exploiting sequential search, proposing a Fast Search Method for Dynamic Time Warping (FTW) [6] that stores all the data from the database within only one single file. However, its major drawback obviously is the fact that sequential search method has to retrieve all the data from the whole database. Evidently, if the most similar sequence is located at the end of the data-

base, lower bounding would not help much, giving very low pruning power , and we have to access every single data sequence until result is found at the very end.

In this work, we propose a similarity matching method using Fast Index Structure (FIS) which limits a scope of a search and retrieves data sequences with the least I/O overhead. It outperforms other search methods, especially, FTW which is to our knowledge the best existing method, in terms of processing time and data accesses. The idea of our method is to cluster all data sequences in a database into groups and to access only fractions of the database as needed. As a result, our method can prune off most of the time series data in the database that are guaranteed not to be the best match, while consuming only small amount of I/O overhead. Note that our approach is generalized to be used with any sequential search method.

The rest of the paper is organized as follows. In Section 2, we describe Background and related research work. We describe our proposed method, FIS, in Section 3, and in Section 4, we demonstrate the performance of our method in terms of time consumption and number of data accesses. Finally, Section 5 concludes our work with some suggestions for further research.

2. Background and Related Work

As similarity search is one of the most important tasks in data mining, various approaches have been proposed to improve performance of the search and retrieval. The most straightforward approach is simply a sequential search, which computes the distance between a query and a candidate sequence one by one, and consecutively scans through the whole database in order. Agrawal et al. [10] first proposed an indexed search using the R*-tree that largely reduces the number of data accesses. However, this method is inefficient on high-dimensional data, which is an underlying characteristic of time series data. Moreover, Vlachos et al. [11] propose a method that uses R-trees to store Minimum Bounding Rectangles (MBR), the approximated multidimensional time series, for the use of fast retrieval for the appropriate candidate with similar attributes. Unfortunately, this method uses longest common subsequence (LCSS) as the distance measure, which is susceptible to parameter tuning. Recently, Sakurai et al. [6] proposed FTW, the best existing search method to date; however, FTW has an obvious drawback, an issue that will be extensively discussed in Section 2.2.

Before moving on to explain our proposed work, we first provide background knowledge necessary for understanding our methodology.

2.1 Dynamic Time Warping Distance Measure

Dynamic Time Warping (DTW) distance measure [1], a shape-based similarity measure for time series data, well-known for its high accuracy in similarity matching problems. Unlike Euclidean distance, DTW measure allows warping in time axis. Consider DTW distance calculation between two time series data. Let C be a time series of length N, where $C = \{c_1, c_2, \ldots, c_i, \ldots, c_N\}$, and Q be a time series of length M, where $Q = \{q_1, q_2, \ldots, q_j, \ldots, q_M\}$. DTW distance $DTW(C,Q)$ is calculated according to Equation 1 below.

$$DTW(C,Q) = \sqrt[p]{f(N,M)}$$

$$f(i,j) = d(c_i,q_j) + \min\begin{cases} f(i-1,j-1) \\ f(i-1,j) \\ f(i,j-1) \end{cases} \quad (1)$$

$$f(0,0) = 0, f(i,0) = f(0,j) = \infty$$

$$1 \le i \le N, 1 \le j \le M$$

where $d(c_i, q_j)$ is a distance between data points c_i and q_j from time series C and Q, respectively. The distance is calculated by using the following Lp-norm.

$$d(x,y) = |x-y|^p \quad (2)$$

2.2 Fast Search Method for Dynamic Time Warping

As mentioned earlier, Sakurai et al. have recently proposed a new similarity search method called FTW (Fast Search Method for Dynamic Time Warping) [6]. In their work, they utilized three important ideas, i.e., LBS (Lower Bounding distance measure with Segmentation), EarlyStopping, and Refinement, which can effectively improve the overall search performance.

LBS is a lower bounding distance measure which calculates lower bounding distance between time series sequences in much fewer dimensions than the original sequences'. LBS divides a data sequence into segments, and then, the maximum and minimum of data points within each segment are used as a representation. This dimensionality reduction technique helps speed up the lower bounding function.

EarlyStopping approach is used to early abandon DTW distance calculation. It prunes off some elements in the distance calculation where the distances are larger than the best-so-far distance. Note that the best-so-far distance is the smallest distance between the query and the candidate sequences during the search.

Refinement technique is applied to the lower bounding function LBS for DTW distance in a multi-

resolution scheme as needed. This can greatly reduce the lower bounding computation cost.

Despite the efficiency of FTW, its worst case is impractical; the essence of the method is to come across similar matches as fast as possible since it needs the good-quality best-so-far distance to prune out the rest of the database. In the following section, we will describe our proposed method that could overcome this downside of FTW.

3. Proposed Method

In this section, we introduce nearest neighbor search method under Fast Index Structure (FIS), which guarantees no false dismissal. The architecture of our method consists of two main components, i.e., indexing phase and querying phase. Indexing phase or preprocessing method can be completed offline before the search. The architecture of the indexing phase is shown in Figure 2 and described in Section 3.1.

The querying process is shown in Figure 3. After we have obtained all envelopes generated from the indexing phase, we calculate lower bounding distances for clustered time series between generated envelopes and a query sequence, called LBC, which is further described in Section 3.2. Finally, these LBCs are used to select appropriate data groups to advance the search which is described in greater detail in Section 3.3.

3.1 Indexing Phase

In our preprocessing phase, all data sequences from the dataset are clustered into groups. In our experiments, we use *k*-means clustering [12-14] as the clustering method. Note that various clustering techniques exist [15], including *k*-medoids clustering [16, 17], CURE [18], and DBSCAN [19]; any clustering technique can be used, depending on the nature of the data at hands. Some may be suitable for density-based approach, and some may work best with noisy data or outliers. However, the choice of clustering method does not affect the performance of our work since most of the clustering approaches will give reasonable groups of data that our approach can benefit from. According to Figure 2, we first cluster the data within the database in advance (offline). As a result, we obtain k groups of data, where k is a desired number of data files. The larger the k value, the less the number of data accesses. After that, we build our index structure, FIS, by composing an envelope for each data group. Specifically, let P be a group of N time series data, where $P = \{P_1, P_2, \ldots, P_i, \ldots, P_N\}$, and let $P_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,j}, \ldots, p_{i,n}\}$. We construct an envelope Env_P of a group of time series P from Equation 3. As illustrated on the right portion of Figure 2, we index each data group accordingly. As a result, we obtain an envelope for each group of data.

$$\begin{aligned} Env_P &= \{e_1, e_2, \ldots, e_i, \ldots, e_n\} \\ e_i &= \{u_i, l_i\} \\ u_i &= \max_{1 \le j \le N} p_{i,j} \\ l_i &= \min_{1 \le j \le N} p_{i,j} \end{aligned} \tag{3}$$

where u_i and l_i are the maximum and minimum of the i^{th} data point from all data sequences, respectively.

Figure 1 illustrates an envelope of a group of clustered time series data. The upper profile illustrates the upper bound of the envelope, and the lower profile illustrates the lower bound of the envelope.

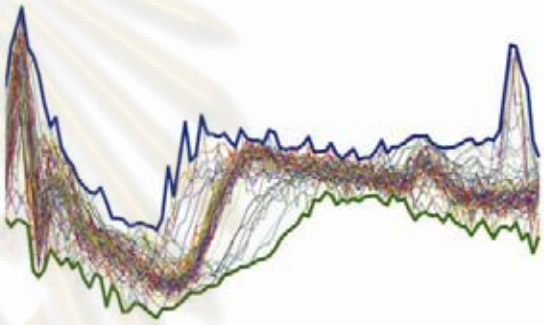


Figure 1. Illustration of the envelope of a group of time series

3.2 Lower Bounding Distance for Clustered Time Series Data (LBC)

As described in Section 3.1, an envelope for each clustered group of time series data has been constructed, assuming that there are k clustered groups. Consider our proposed LBC between a query Q and a candidate envelope Env_p, where $Q = \{q_1, q_2, \ldots, q_i, \ldots, q_N\}$ and $Env_p = \{\{u_1, l_1\}, \{u_2, l_2\}, \ldots, \{u_i, l_j\}, \ldots, \{u_M, l_M\}\}$. LBC is calculated as follows.

$$LBC(Q, Env_p) = f(N, M)$$

$$f(i,j) = d(q_i, \{u_j, l_j\}) + \min \begin{cases} f(i-1, j-1) \\ f(i-1, j) \\ f(i, j-1) \end{cases} \tag{4}$$

$$f(0,0) = 0, f(i,0) = f(0,j) = \infty$$
$$1 \le i \le N, 1 \le j \le M$$

where distance between a data point q_i from the query Q and an envelope element $\{u_j, l_j\}$ from the envelope Env_p are defined in Equation 5.

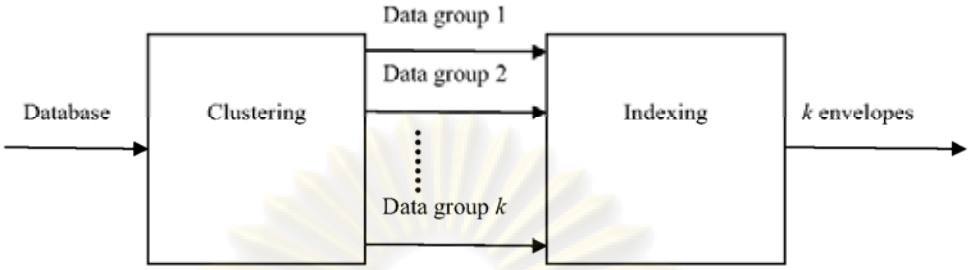


Figure 2. Illustration of our proposed indexing phase

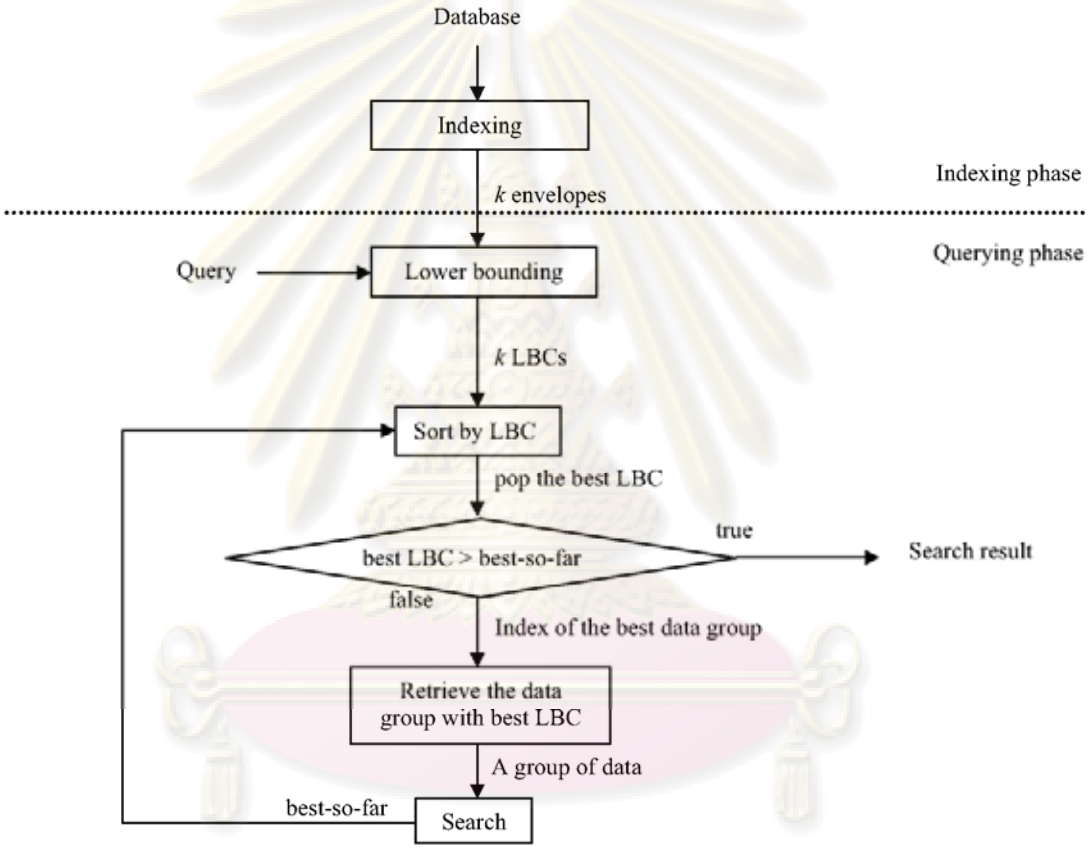


Figure 3. A flowchart of our proposed method in querying phase

$$d(q_i,\{u_j,l_j\}) = \begin{cases} (q_i - u_j)^2 & ; q_i > u_j \\ (q_i - l_j)^2 & ; q_i < l_j \\ 0 & ; \text{otherwise} \end{cases} \quad (5)$$

Simply, this lower bounding distance is the distance between the envelope and the portion of the query that does not fall within the envelope.

```
1   Algorithm FIS(Q, C[], Env[])
2   for i = 1 to k do
3     LBCdistances[i] = LBC(Q, Env[i]);
4   endfor
5   sort(LBCdistances);
6   globalBestSoFar = PositiveInfinite;
7   globalBestMatch = null;
8   for i = 1 to k do
9     if pop(LBCdistances) > globalBestSoFar then
10      return globalBestSoFar;
11    endif
12    index := popIndex(LBCdistances);
13    [bestMatch,bestSoFar] := search(Q, C[index]);
14    if globalBestSoFar > bestSoFar then
15      globalBestSoFar = bestSoFar;
16      globalBestMatch = bestMatch;
17    endif
18  endfor
19  return globalBestMatch;
```

Figure 4. Pseudo code of our proposed Fast Index Structure in query phase

3.3 Sequential Search within Each Time Series Group

After we have completed LBC calculations among a query and all the envelopes from each group, we sort these groups of time series data by LBC values. We then perform a sequential search in the first group with the smallest LBC value. This sequential search can replace any existing sequential search techniques such as LB_Keogh [20], FTW, etc. In our experiment, FTW is used in the comparison since it is the best existing method to date. After the search in the chosen group has been completed, we obtain both the best matched time series and the best-so-far distance. We then compare whether this best-so-far distance (from the current group) is smaller than the global best-so-far distance. If so, we set the global best-so-far and the global best match to this current best-so-far distance and the best matched sequence. If the LBC value of the next group is less than the global best-so-far distance, we proceed with the sequential search in the next group; otherwise, we terminate the search and return the global best match as a result of the search. In essence, we can prune out and ignore the rest of the data groups, which are known to have DTW distance larger than the global best-so-far distance since LBC values of these data will also be larger.

Figure 4 shows the pseudo code of our method in the query phase. The inputs are Q, $C[]$, and $Env[]$, where Q is a query data sequence, $C[]$ is an array of pointers to each data group, and $Env[]$ is an array of pointers to each envelope. Lines 2-4 calculate the LBC distances between query Q and each envelope, and store the result in a priority queue. Line 12 pops the pointer to the group with minimum LBC distance. Line 13 searches the query within that group. Line 14 updates the global best-so-far distance of the search. In line 9, if the best LBC distance in the priority queue is larger than the global best-so-far distance, it terminates and returns the result of the search.

4. Experiments

In this section, we demonstrate the efficiency of our proposed method, by comparing with FTW [6], the best existing search method, both in terms of computational cost and number of data accesses. Both implementations of FIS and FTW are done in Java, running on an Intel Pentium 4 ® CPU 3.06GHz platform with 1GB of memory.

4.1 Time Consumption

We compared the performance of the nearest neighbor search by using our method with the variation of the total number of data groups. Performance is evaluated both in terms of CPU time and wall clock time, comparing with the best existing method, FTW. A database is generated by a random-walk model with the total of 100,000 sequences, each with 512 data points in length. Ten additional sequences with the same length are also generated to be used as queries.

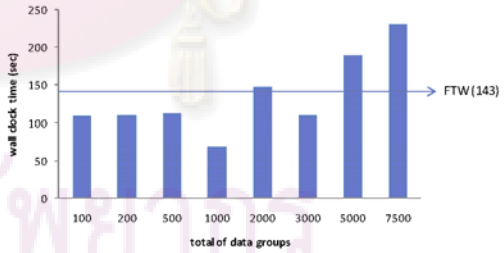


Figure 5. Wall clock time, comparing FTW with our proposed FIS with a variation in the number of groups.

As shown in Figure 5, we compared the wall clock time of FTW with our method, FIS, where the number of data groups varies from 100 to 7,500. The experimental results show that FIS gives the best performance when the total data groups are around 1,000, doubling the performance of FTW. Note that too large number of data groups would lead to an increase in wall clock time due to some I/O overhead for file accesses. If the number of data groups is too small, we would not benefit much from clustering algorithm and in selecting a good candidate for the nearest neighbor. In an extreme case of one total data group, it is identical to FTW method. Therefore, selecting the right number of data groups is essential to the overall performance of the method. In practice, we could simply determine an optimal or near-optimal number of groups through parameter tuning from training data that could be done offline. On the other hand, from Figure 6, we can see that the variation of the number of groups does not significantly affect the performance in terms of CPU time since there is no I/O time involved, i.e., FIS still outperforms FTW in all settings.

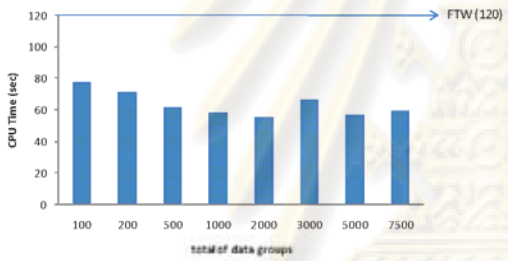


Figure 6. CPU time, comparing FTW with our proposed FIS with a variation in the number of groups

4.2 Total Number of Data Accesses

Our experiments also measure the total number of data accesses which would constitute the I/O cost. The same database and queries as in Section 4.1 are used for evaluation. The results are shown in Figure 7. We can see that the total data accesses of FTW are quite high since it needs to sequentially search through all the data, requiring 1,000,000 accesses for searching of 10 queries in the database size of 100,000 sequences.

Our experimental results show that FIS does reduce the number of data needed to be retrieved for the search. As the total groups of data in FIS increases, FIS can reduce a huge number of data accesses, in turn effectively reducing I/O cost. As shown in Figure 7, the total data access is reduced by 87%, giving a speedup of as high as 8 times when the database is clustered into 7,500 groups.

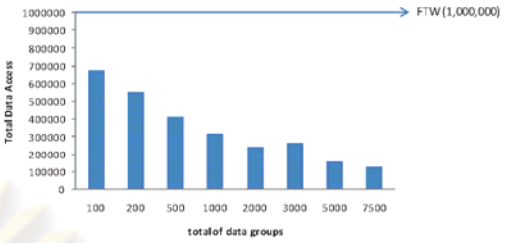


Figure 7. Total number of data accesses, comparing FTW with our proposed FIS with a variation in the number of groups

4.3 Real-World Application

To evaluate the utility of our algorithm in real-world application, we mix real-world dataset, Lightning-2 from UCR time series data archive [21], and the generated random-walk dataset together. Since time series data in Lightning-2 dataset are 637 data points in length, we also generate random-walk data with this same length. In addition, forty data sequences from Lightning-2 training dataset are randomly selected for our database, and another three sequences are used as queries for our experimental evaluation.

In Figure 8, it is apparent that FIS can significantly reduce the total number of data accesses, which also leads to the computational cost reduction for the search, as shown in Figure 9. Another obvious advantage of our proposed method is that the number of data access is unaffected as the dataset size increases, whereas that of FTW linearly increases.

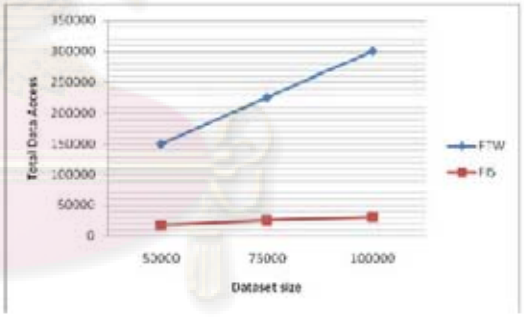


Figure 8. Total number of data accesses, comparing FTW with our proposed FIS with a variation in the dataset size

In Figure 9, we give a comparison of the overall wall clock time among the classic DTW, FTW, and our proposed FIS. Generally, FTW can speed up the normal DTW distance calculation (with no indexing) by a large margin. However, in this case, FTW can only

give 4 times speedup over normal DTW. This also illustrates a drawback of FTW when large amount of sequential search is needed. Since the 40 data sequences that are similar to the query are randomly placed within a 100,000 random walk sequences, there is quite a small chance that FTW will come across one of those forty very early. That means FTW will have large overhead for the search.

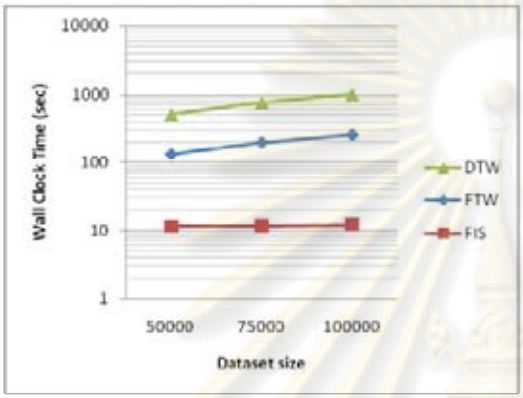


Figure 9. Wall clock time, comparing among DTW, FTW, and our proposed FIS, with a variation in the dataset size

However, this no longer is the problem for our proposed FIS, where it could achieve up to 2 orders of magnitude speedup over classic DTW and FTW. This is simply because our FIS method will try to retrieve the most similar group of data first, Furthermore, as mentioned earlier, our FIS is quite scalable, where the dataset size does not affect its computational cost. In other words, the computational cost of FIS grows in sublinear time as a function of the dataset size.

5. Conclusion

We propose a new index structure, called FIS, which can be incorporated into any search method to increase its performance. FIS offers a new lower bounding distance measure which can approximate the lower bounding distance of a similarly-clustered group of data. As a result, FIS can reduce the number of data needed to be accessed for the search by pruning off all the data that are guaranteed not to be one of the best matches, using our new lower bounding distance, LBC. Our index structure incurs only little I/O overhead because the data retrieval methodology is still based on sequential retrieval. For further work, FIS could be improved so that the lower bounding distance would support the dataset with different-length time series data.

References

[1] C. A. Ratanamahatana and E. Keogh, "Three Myths about Dynamic Time Warping," in Proceedings of SIAM International Conference on Data Mining Newport Beach, CA, USA, 2005, pp. 506–510.

[2] E. Keogh and C. A. Ratanamahatana, "Exact Indexing of Dynamic Time Warping," Knowledge and Information Systems (KAIS), vol. 7, pp. 358–386, March 2005.

[3] Y. Zhu and D. Shasha, "Warping indexes with envelope transforms for query by humming," in Proceedings of the 2003 ACM SIGMOD international conference on Management of data, San Diego, CA, USA, 2003, pp. 181–192.

[4] B.-K. Yi, H. V. Jagadish, and C. Faloutsos, "Efficient Retrieval of Similar Time Sequences under Time Warping," in Proceedings of 14th International Conference on Data Engineering, Orlando, FL, USA, 1998, pp. 201–208.

[5] S.-W. Kim, S. Park, and W. W. Chu, "An Index-based Approach for Similarity Search Supporting Time Warping in Large Sequence Databases," in Proceedings of 17th International Conference on Data Engineering, Heidelberg, Germany, 2001, pp. 607–614.

[6] Y. Sakurai, M. Yoshikawa, and C. Faloutsos, "FTW: Fast Similarity Search under the Time Warping Distance," in Proceedings of 24th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, Baltimore, MA, USA, 2005, pp. 326–337.

[7] L. Wei, E. Keogh, H. V. Herle, and A. Mafra-Neto, "Atomic Wedgie: Efficient Query Filtering for Streaming Time Series," in Proceedings of 5th IEEE International Conference on Data Mining 2005, pp. 490–497.

[8] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," in Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 1984), Boston, Massachusetts, USA, 1984, pp. 47-57.

[9] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles," in Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, Atlantic City, NJ, USA, 1990, pp. 322-331.

[10] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient Similarity Search In Sequence Databases," in Proceedings of the 4th International Confe-

rence of Foundations of Data Organization and Algorithms (FODO) Chicago, Illinois, 1993, pp. 69-84.
[11] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh, "Indexing Multi-Dimensional Time-Series with Support for Multiple Distance Measures " in Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 2003, pp. 216-225.
[12] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations," in Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281-297
[13] J. A. Hartigan, Clustering Algorithms: Wiley, 1975.
[14] J. A. Hartigan and M. A. Wong, "A K-Means Clustering Algorithm," Applied Statistics, vol. 28, pp. 100-108, 1979.
[15] T. Warren Liao, "Clustering of time series data--a survey," Pattern Recognition, vol. 38, pp. 1857-1874, 2005.
[16] L. Kaufman and P. Rousseeuw, "Finding groups in data: An introduction to cluster analysis," Applied Probability and Statistics, 1990.
[17] R. T. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining," in Proceedings of 20th International Conference on Very Large Data Bases (VLDB'94), Santiago de Chile, Chile, 1994, pp. 144–155.
[18] S. Guha, R. Rastogi, and K. Shim, "Cure: An efficient clustering algorithm for large databases," in Proceedings ACM SIGMOD International Conference on Management of Data (SIGMOD 1998), Seattle, WA, USA, 1998, pp. 73–84.
[19] M. Ester, A. Frommelt, H.-P. Kriegel, and J. Sander, "Spatial data mining: Database primitives, algorithms and efficient dbms support," Data Mining and Knowledge Discovery, vol. 4, pp. 193–216, 2000.
[20] E. Keogh, "Exact Indexing of Dynamic Time Warping," in Proceedings of 28th International Conference on Very Large Data Bases, Hong Kong, China, 2002, pp. 406–417.
[21] E. Keogh, "UCR Time Series Classification/Clustering Page."

Pongsakorn Ruengronghirunya is a master student in computer engineering at Chulalongkorn University. He also received his undergraduate study in computer engineering from Chulalongkorn University. His research interests include data mining, particularly in time series data and information retrieval.

Vit Niennattrakul is a PhD candidate in computer engineering at Chulalongkorn University, Bangkok, Thailand. He received his undergraduate study in computer engineering from Chulalongkorn University. His research interests include data mining, information retrieval, machine learning, and natural language processing. Currently, he has received a grant in part from the Thailand Research Fund given through the Royal Golden Jubilee PhD Program.

Chotirat Ann Ratanamahatana is a lecturer at Computer Engineering Department, Chulalongkorn University, Bangkok, Thailand. She received her undergraduate and graduate studies from Carnegie Mellon University and Harvard University, respectively, and received her Ph.D. from the University of California, Riverside. Her research interests include time series data mining, information retrieval, machine learning, and human-computer interaction

ภาคผนวก ข

บทความทางวิชาการเรื่อง “Speeding up Similarity Search on Large Time Series Dataset Under Time Warping Distance” โดย พงศกร เรืองรองหิรัญญา วิชญ์ เนียรนาทตระกูล และโชติรัตน์ รัตนามหัทธนะ ในงานประชุมวิชาการนานาชาติ “13th Pacific-Asia Conference on Knowledge Discovery and Data Mining” ซึ่งจัดขึ้น ณ กรุงเทพมหานคร ประเทศไทย ระหว่างวันที่ 27 – 30 เมษายน 2552 ดังรายละเอียดในภาคผนวก ข

Speeding up Similarity Search on a Large Time Series Dataset under Time Warping Distance

Pongsakorn Ruengronghirunya, Vit Niennattrakul, and Chotirat Ann Ratanamahatana

Department of Computer Engineering, Chulalongkorn University,
254 Phyathai Road, Patumwan, Bangkok Thailand 10330
{g51prn, g49vnn, ann}@cp.eng.chula.ac.th

Abstract. Time series data are a ubiquitous data type appearing in many domains such as statistics, finance, multimedia, etc. Similarity search and measurement on time series data are typically different from on other data types since time series data have the associations among adjacent dimensions. Accordingly, the classic Euclidean distance metric is not an accurate similarity measure for time series. Therefore, Dynamic Time Warping (DTW) has become a better choice for similarity measurement on time series in various applications regardless of its high computational cost. To speed up the calculation, many research works attempt to speed up DTW calculation using indexing method, which always has a tradeoff between indexing efficiency and I/O cost. In this paper, we propose a novel method to balance this tradeoff under indexed sequential access using Sequentially Indexed Structure (SIS), an approach to time series indexing with low computational cost and small overheads on I/O. Finally, we conduct experiments to demonstrate our superiority in speed performance over the best existing method.

Keywords: Similarity Search, Dynamic Time Warping, Time Series Data, Indexing, Lower Bounding Function

1 Introduction

Behind the scenes of many innovative applications, the information from these applications sometimes appears in a complex format such as multimedia data. To extract knowledge from this complex information, data are usually represented as time series for the ease of data mining tasks since time series are just a sequence of real number or, in a different aspect, a multi-dimensional variable.

Comparing with other types of multi-dimensional data, time series data have their unique property that they have shifting in time domain. In another aspect, each value in a dimension can be related to another value from different dimensions. An approach for the distance measurement in general multi-dimensional data is to directly calculate the distance from each pair of values on the same dimension, well-known as Euclidean distance metric. A more flexible distance measurement is Dynamic Time Warping (DTW) distance measure [1]. DTW distance calculation allows shifting in time domain which can possibly match related data points from

contiguous dimensions. Therefore, similarity search on time series data with time shifting using DTW distance for the similarity measurement gives more accurate result than using the Euclidean distance. However, DTW distance has a major drawback on speed performance. The time complexity of the calculation for DTW distance is $O(n^2)$, where n is a total number of the dimensions or length of time series data, which is too far to be acceptable in practice. Therefore, many research works attempt to speed up DTW distance calculation.

A popular approach to speed up DTW calculation is to further compute the lower bound of DTW distance. The lower bounding function, which can approximate the DTW distance with a relatively small computational cost, helps the similarity search method ignore some of the DTW calculations. Many research works [2-4] have proposed various lower bounding functions. For example, Keogh and Ratanamahatana [3] proposed a lower bounding function using the constraint on time warping. This function uses Euclidean distance in calculation which requires only $O(n)$ time complexity. Even though lower bounding techniques can speed up the distance calculation, the distance calculation is just a small part of the similarity search. Some other tasks still need to be considered.

Indexing time series data seems to be cumbersome since each time series consists of a large number of dimensions. This leads to a great complexity for many tree-based index structures such as R*-Tree [5]. Furthermore, the tree-based structure has a major drawback that the order of the tree traversing is not sequential. If the search scans through the tree and retrieves indexed data from a leaf node, this can lead to large overheads on I/O to retrieve a data sequence using random access on the disk. However, an indexed sequential access can solve these problems that incur in tree-based index structure.

In this work, we propose a new similarity search method on time series data using Sequentially Indexed Structure (SIS), an index structure under indexed sequential access which can directly index to the appropriate part of the dataset and retrieve the indexed data using sequential access on disk. Furthermore, SIS can also ignore most parts of the dataset which are pruned off by our new lower bounding function, LBG, for the search. Since SIS is under indexed sequential access, this leads to only small overheads on disk access. In addition, SIS is generalized to be used with any sequential search methods such as FTW (Fast Search Method for Dynamic Time Warping) [6], the fastest existing similarity search method on time series data under DTW distance. As a result, our method can significantly speed up the similarity search compared with the original FTW.

The rest of the paper is organized as follows. In Section 2, we describe background and related research work. We describe our proposed method, SIS, in Section 3, and in Section 4, we demonstrate the performance of our method in terms of pruning power and time consumption. Finally, Section 5 concludes our work with some suggestion for further research.

2. Background and Related Work

There are several similarity measurements for time series data such as Euclidean distance measure, Dynamic Time Warping (DTW) distance measure [1], longest common subsequence (LCSS), etc. However, no conclusion has been drawn for the best similarity measurement since the accuracy for each similarity measurement depends on the type of data. For example, Query-by-Humming system [4] applies DTW distance for the similarity measurement because the great merit for DTW distance is that it is flexible and accurate for data that exhibit some shifting in the time axis which is a typical characteristic of the humming voice. However, DTW has a large time complexity of $O(n^2)$. Therefore, many researchers focus on speeding up the DTW calculation.

Since 1993, Agrawal et al. [7] has proposed an approach to speedup DTW on similarity search. They proposed an index structure using R*-Tree [5]. Although this method does not perform well because of the curse of dimensionality problem [8], it has been the first inspiration for many researchers to focus on indexing time series data.

Keogh and Ratanamahatana [3] proposed an improved index structure using R*-Tree [5] and a new lower bounding function for DTW distance using Piecewise Aggregate Approximation (PAA) [9]. Moreover, Zhu and Shasha [4] developed the lower bounding function using PAA which is tighter than Keogh and Ratanamahatana's method. In other words, Zhu and Shasha's lower bounding function gives results which are closer to the real DTW distance. Although this index structure can significantly reduce the number of data needed to be searched, each data retrieval causes large overheads on disk from random access on each node in tree-based index structure.

Sakurai et al. [6] proposed a new approach to speed up DTW using sequential access on disk. They proposed an efficient lower bounding function which can prune off most of the candidate sequences before DTW distance. Their experiments show that their method is much faster than Zhu and Shasha's calculation [4]. However, if an efficient index structure can be adapted to use with their lower bounding function, FTW will absolutely be able to speed up the search since it is obvious that there is no reason to retrieve all data in the dataset for the search.

Before moving on to explain our proposed work, we first provide background knowledge necessary for understanding our methodology.

2.1 Dynamic Time Warping Distance Measure

Dynamic Time Warping (DTW) distance measure [1] is a flexible and accurate distance measure for time series data. Since time series data is high-dimensional data with relationships among other dimensions, Euclidean distance measure, a straightforward distance measure using comparison of data from each dimension independently, is not quite accurate. As a result, DTW distance is developed for time series' domain. DTW distance measure allows comparisons among values from adjacent dimensions. Consider DTW distance calculation between two time series data. Let C be time series of length N, where $C = \{c_1, c_2, \ldots, c_i, \ldots, c_N\}$, and Q be time

series of length M, where $Q = \{q_1, q_2, \ldots, q_j, \ldots, q_M\}$. DTW distance $DTW(C,Q)$ is calculated according to eq. (1) below

$$DTW(C,Q) = \sqrt{f(N,M)}$$
$$f(i,j) = d(c_i, q_j) + \min \begin{cases} f(i-1,j-1) \\ f(i-1,j) \\ f(i,j-1) \end{cases} \quad (1)$$
$$f(0,0) = 0, f(i,0) = f(0,j) = \infty$$

where $1 \le i \le N$, $1 \le j \le M$ and $d(c_i, q_j)$ is a distance between data points c_i and q_j from time series C and Q, respectively. The distance is calculated by using the following Lp-norm equation (2).

$$d(x,y) = |x - y|^p \quad (2)$$

Since DTW distance allows comparisons among values from different dimensions, DTW calculation sometimes gives an inaccurate result because of undesirable or sometimes inappropriate alignment. Therefore, global constraint is assigned to limit the scope on the comparisons; it does not allow any comparisons of any pairs of values which have large distance in the time domain.

3. Proposed Method

In this section, we introduce a new similarity search method on time series data under Sequentially Indexed Structure (SIS). In our work, we propose a new lower bounding function on DTW distance for a group of time series data called Lower Bounding Function for a Group of Time Series (LBG). Furthermore, our new index structure, SIS can significantly speed up the search under DTW distance measure.

SIS consists of two parts which are indexing phase and query phase. In an indexing phase, SIS clusters all data in the dataset into groups and generates a representative for each group. Note that we can complete the indexing phase in advance before the search (offline). In a query process, we use these representatives to calculate the lower bounding distances using our new lower bounding function. Finally, we use these lower bounding distances to guide the search.

3.1 Indexing Phase

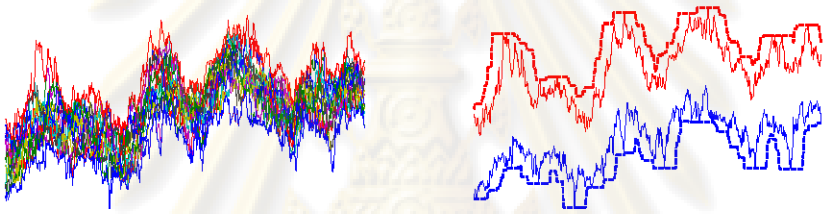
Initially, we cluster all the data sequences from a dataset using any existing clustering methods, using Euclidean distance as a distance measure. In our experiments, we use k-means clustering [10] in order to minimize the maximum distance between each pair of time series within the same group. Afterwards, we generate an index for each data group. The indexing method is described as follows.

After we obtain groups of data sequences from the clustering method, we assign a bound for each data group. Each bound consists of the upper bound U and the lower

bound L which are the maximum and the minimum of the values from each dimension among the data sequences in a group, respectively. Fig. 1a illustrates an example of the bound of a data group. The upper solid profile is the upper bound and the lower solid profile is the lower bound of the group.

Afterwards, we formulate an index of a group of time series data using its bound, as mentioned above, by transforming both upper and lower bounds into an envelope using the algorithm extended from LB_Keogh [3]. In other words, we transform each upper bound and lower bound into LB_Keogh's upper and lower envelope respectively. In Fig. 1, the upper and lower dashes represent the upper envelope and the lower envelope of the group, respectively.

Once we obtain k groups of time series and k envelopes generated for SIS in this phase, we will utilize it in the query phase.



(a) (b)

Fig. 1. (a) Illustration of a group of time series. (b) Illustration of the profile of the group of time series from (a), shown in solid, and the LB_Keogh's envelope, shown in dash.

3.2 Lower Bounding Function for a Group of Time Series

Lower Bounding Function for Group of Time Series (LBG) is a lower bound of the DTW distance for all time series data in a group. In other words, LBG distance between a query sequence and a group of sequences is a lower bound for the distance between the query sequence and the nearest candidate sequence among sequences in that group. We can simply use the LB_Keogh function [3] to calculate the lower bounding distance. The detail of the calculation is described as follows.

Let Q be a query sequence of length n where $Q = \{q_1, q_2, \ldots, q_i, \ldots, q_n\}$, and let E be the envelope of the group of time series which contains U and L as the upper and the lower envelopes, respectively, where $U = \{u_1, u_2, \ldots, u_i, \ldots, u_n\}$ and $L = \{l_1, l_2, \ldots, l_i, \ldots, l_n\}$. The LBG distance $LBG(Q, E)$ calculation is shown in eq. (3).

$$LBG(Q,E) = \sqrt{\sum_{i=1}^{n} d(q_i,E)}$$

$$d(q_i,E) = \begin{cases} (q_i - u_i) & ; \text{if } q_i > u_i \\ (q_i - l_i) & ; \text{if } q_i < l_i \\ 0 & ; \text{otherwise} \end{cases} \quad (3)$$

3.3 Query Phase

According to the indexing phase from Section 3.1, the dataset has been clustered and a representative (LB_Keogh's upper envelope and lower envelope) is generated from each cluster. In this query phase, we use the LBG distance to indicate the order of the search. Specifically, the search will sequentially scan in each cluster ordered in an ascending order of the LBG distance. Since LBG is a lower bounding distance, the search can also prune off data sequences from clusters which has larger LBG distance than the best-so-far distance.

4. Experiment

In this section, we demonstrate the efficiency of our proposed method in terms of speed performance with the best existing similarity search method, FTW [6]. Implementations are done in Java, running on an Intel Pentium 4 ® CPU 3.06GHz platform with 1GB of memory. We conduct experiments on one nearest neighbor search using both synthetic datasets and real world datasets. We emphasize on the search over a large dataset which makes FTW perform at its best efficiency. Accordingly, the experiment can demonstrate our superiority.

Eight real-world training datasets from the UCR Time Series Classification Clustering Page [11], which cover various characteristic of time series data, are used in our experiment. To build a large dataset, each dataset is z-normalized and filled with additional randomwalk sequences of the same length to create a total of 100,000-sequence dataset. The properties of the real-world datasets are shown in Table 1. According to the UCR Time Series Classification Clustering Page, we assign time warping's global constraints that give the most accurate results. In the indexing phase, we cluster each dataset into 1,000 groups using *k*-mean clustering method. Finally, we evaluate the efficiency of SIS, our proposed index structure, in terms of pruning power and time consumption. We conduct the one-nearest-neighbor search using each real-world test dataset as queries for each generated dataset.

Table 1. The details of training datasets

Name	*Size of training set*	*Size of test set*	*length*	*Global constraint*
Gun-Point	50	150	150	0%
Face (all)	560	1690	131	3%
Trace	100	100	275	3%
Wafer	1000	6174	152	1%
Face (four)	24	88	350	2%
Lightning-2	60	61	637	6%
Lightning-7	70	73	319	5%
Beef	30	30	470	0%

4.1 Pruning Power

Pruning power is a measure which can evaluate the efficiency of any indexing methods [3]. Note that pruning power is a fraction of the candidate sequences which are discarded before the search. Specifically, in this experiment, pruning power is the fraction of the total sequences from all groups which are discarded by SIS method. Fig. 2(a) demonstrates the pruning power of each test dataset. The Lightning2 dataset gives the best pruning power performance for SIS since the shapes of its training data are greatly distinguished from random-walk data. Consequently, the clustering method can efficiently distinguish the Lightning2 sequences from the random-walk sequences.

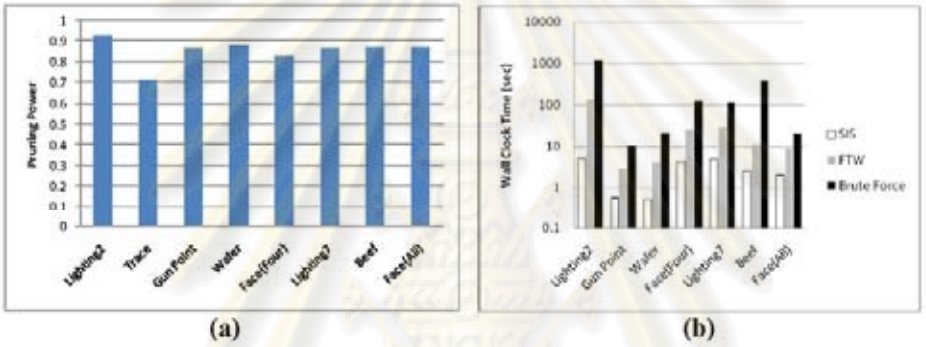


Fig. 2. (a) Pruning power of SIS on our test datasets. (b) Wall clock time, comparing FTW with our proposed SIS on various dataset

4.2 Time Consumption

We evaluate time consumption on similarity search under SIS by comparing with FTW, the fastest existing similarity search method, and the brute force method, sequential search using the original DTW distance measure. Fig. 2(b) demonstrates the comparison of time consumption between FTW and SIS in terms of average wall clock time per one query sequence. The result shows that our method outperforms FTW in every dataset, particularly in datasets which have high pruning power, by an order of magnitude. Note that both FTW and brute force method use sequential access on disk. As a result, they do not achieve any pruning power since accesses to all candidate sequences are required.

5. Conclusion

We propose a new index structure for similarity search on time series data called SIS, Sequentially Indexed Structure, which can guide the search to a more appropriate part of the database. SIS can also prune off a large number of candidate sequences, which are distinguishable from the query sequence without accessing most of the raw

candidate sequences in the database. Moreover, data retrieval applied in our method uses the sequential access on disk. Consequently, this can significantly reduce the I/O cost which can greatly speed up the search. Our experimental results show the superiority of our method, SIS, over the rival method, FTW [6], in terms of speed performance up to 26 times faster. For further works, we will focus on our preprocessing steps, clustering method since the cohesion of the clustered data leads to the efficiency on pruning power of SIS.

Acknowledgement

This work is partially supported by the Thailand Research Fund (Grant No. MRG5080246).

References

1. Ratanamahatana, C.A., Keogh, E.: Three Myths about Dynamic Time Warping. Proceedings of SIAM International Conference on Data Mining, Newport Beach, CA, USA (2005) 506-510
2. Euachongprasit, W., Ratanamahatana, C.A.: Accurate and Efficient Retrieval of Multimedia Time Series Data under Uniform Scaling and Time Warping. Proceedings of The Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Osaka, Japan (2008)
3. Keogh, E., Ratanamahatana, C.A.: Exact Indexing of Dynamic Time Warping. Knowledge and Information Systems (KAIS) 7 (2005) 358-386
4. Zhu, Y., Shasha, D.: Warping indexes with envelope transforms for query by humming. Proceedings of the 2003 ACM SIGMOD international conference on Management of data. ACM Press, San Diego, CA, USA (2003) 181-192
5. Beckmann, N., Kriegel, H.-P., Schneider, R., Seeger, B.: The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles. Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, Atlantic City, NJ, USA (1990) 322-331
6. Sakurai, Y., Yoshikawa, M., Faloutsos, C.: FTW: Fast Similarity Search under the Time Warping Distance. Proceedings of 24th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, ACM Press, Baltimore, MA, USA (2005) 326-337
7. Agrawal, R., Faloutsos, C., Swami, A.: Efficient Similarity Search In Sequence Databases. Proceedings of 4th International Conference of Foundations of Data Organization and Algorithms (FODO), Chicago, Illinois (1993) 69-84
8. Berchtold, S., Bohm, C., Kriegal, H.-P.: The pyramid-technique: towards breaking the curse of dimensionality. Proceedings of the 1998 ACM SIGMOD international conference on Management of data, ACM, Seattle, Washington, United States (1998)
9. Yi, B.-K., Faloutsos, C.: Fast Time Sequence Indexing for Arbitrary Lp Norms. Proceedings of 26th International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc., Cairo, Egypt (2000) 385-394
10. MacQueen, J.B.: Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability (1967) 281-297
11. Keogh, E., Xi, X., Wei, L., Ratanamahatana, C.A.: The UCR Time Series Classification Clustering Homepage. www.cs.ucr.edu/~eamonn/time_series_data

ประวัติผู้เขียนวิทยานิพนธ์

นายพงศกร เรืองรองหิรัญญา เกิดเมื่อวันที่ 23 มีนาคม พ.ศ. 2530 สำเร็จการศึกษาระดับมัธยมศึกษาที่โรงเรียนเซนต์คาเบรียล จากนั้นเข้าศึกษาต่อในคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2547 และในปีการศึกษา 2550 ก็ได้สำเร็จการศึกษาในระดับปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ หลังจากจบการศึกษาแล้วก็ได้เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2551