การสร้างแผ่นแบบโดยใช้การปรับแนวแบบไทม์วอร์ปปิง

นางสาวดารารัตน์ ศรีใส

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2552

TEMPLATE CONSTRUCTION USING TIME-WARPING ALIGNMENT

Miss Dararat Srisai

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 2009

หัวข้อวิทยานิพนธ์	การสร้างแผ่นแบบโดยใช้การปรับแนวแบบไทม์วอร์ปปิง
โดย	นางสาวดารารัตน์ ศรีใส
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.โชติรัตน์ รัตนามหัทธนะ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

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

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.โชติรัตน์ รัตนามหัทธนะ)

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

..... กรรมการภายนอกมหาวิทยาลัย
(อาจารย์ ดร.ทรงพล องค์วัฒนกุล)

ดารารัตน์ ศรีใส : การสร้างแผ่นแบบโดยใช้การปรับแนวแบบไทม์วอร์ปปิง. (TEMPLATE CONSTRUCTION USING TIME-WARPING ALIGNMENT) อ. ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ. ดร.โชติรัตน์ รัตนามหัทธนะ, 105 หน้า.

การจำแนกประเภทข้อมูลสำหรับข้อมูลอนุกรมเวลาเป็นเรื่องที่น่าสนใจสำหรับการวิจัย เนื่องจากสามารถนำไปใช้ได้กับหลาย ๆ โปรแกรมประยุกต์ในหลากหลายด้าน เช่น ด้านการแพทย์ ด้านการเงิน ด้านการบันเทิง และด้านอุตสาหรรมต่าง ๆ เพราะเหตุนี้จึงมีนักวิจัยเป็นจำนวนมากที่มุ่งศึกษาและพัฒนาวิธีที่จะนำมาแก้ปัญหาการจำแนกประเภทข้อมูล เช่น ต้นไม้การตัดสินใจ โครงข่ายประสาทเทียม เป็นต้น อย่างไรก็ตามวิธีที่ได้รับความนิยมและมีความแม่นยำสูงก็คือวิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้สุดอันดับที่หนึ่งด้วยการวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง แต่การใช้ไดนามิกไทม์วอร์ปปิงสำหรับการจำแนกประเภทข้อมูลนั้นต้องใช้เวลาในการคำนวณสูง นอกจากนั้นปริมาณของข้อมูลเรียนรู้ที่ต้องจัดเก็บในหน่วยเก็บข้อมูลถ้ามีจำนวนมากก็จะต้องมีพื้นที่ในการเก็บมาก ซึ่งในบางโปรแกรมประยุกต์ที่มีข้อจำกัดในด้านหน่วยเก็บข้อมูลก็จะไม่สามารถทำงานได้ ทำให้มีงานวิจัยเป็นจำนวนมากที่พยายามแก้ปัญหาโดยวิธีที่ใช้ส่วนมากก็คือการลดจำนวนข้อมูลเรียนรู้ จากประเด็นที่กล่าวมาจึงเป็นเหตุผลที่ทำให้เกิดงานวิจัยนี้ขึ้นมา โดยแนวคิดในงานวิจัยนี้คือการสร้างแผ่นแบบหรือตัวแทนกลุ่มที่สามารถแทนข้อมูลอนุกรมเวลาตัวอื่น ๆ ที่อยู่ในกลุ่มเดียวกันได้ สำหรับตัวแทนกลุ่มในงานวิจัยนี้ใช้การหาค่าเฉลี่ยรูปร่างของข้อมูลภายในกลุ่มโดยใช้การปรับแนวแบบไทม์วอร์ปปิงซึ่งเรียกวิธีการนี้ว่า ASA ซึ่งเป็นการลดจำนวนข้อมูลเรียนรู้ที่ต้องเก็บในหน่วยเก็บข้อมูลให้เหลือเพียงกลุ่มละหนึ่งอนุกรมเท่านั้น และยังเป็นการลดจำนวนครั้งในการคำนวณไดนามิกไทม์วอร์ปปิงเพราะการจำแนกประเภทข้อมูลอนุกรมเวลาหนึ่งอนุกรมก็ทำการเปรียบเทียบกับแผ่นแบบเท่านั้น ซึ่งในการทดลองนั้นวิธีการจำแนกประเภทข้อมูลโดยใช้ตัวแทนกลุ่มสามารถลดจำนวนข้อมูลที่ต้องจัดเก็บ และเวลาที่ใช้ในการจำแนกประเภทข้อมูลแต่ละตัวก็ลดลงหลายเท่ากว่าวิธีที่ใช้ในปัจจุบัน โดยที่ไม่ได้สูญเสียประสิทธิภาพในด้านความแม่นยำในการจำแนกประเภทข้อมูล

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต ดารารัตน์ ศรีใส
สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่ออ. ที่ปรึกษาวิทยานิพนธ์หลัก
ปีการศึกษา 2552

5170307721 : MAJOR COMPUTER ENGINEERING

KEYWORDS : TIME SERIES DATA / TEMPLATE MATCHING / TEMPLATE CONSTRUCTION / DYNAMIC TIME WARPING / SHAPE AVERAGING / CUBIC SPLINE APPROXIMATION

DARARAT SRISAI : TEMPLATE CONSTRUCTION USING TIME-WARPING ALIGNMENT. THESIS ADVISOR : ASST. PROF. CHOTIRAT RATANAMAHATANA, Ph.D., 105 pp.

Time series data classification is an interesting research topic because it can be applied to various fields such as medical, financial, entertainment, and industrial. For this reason, many researchers around the world research and develop various solutions for data classification such as decision trees, artificial neural networks, etc. However, one of the most popular and accurate methods is the One Nearest Neighbor classification using Dynamic Time Warping (DTW) distance measure, but utilizing DTW for data classification requires large computation time. In addition, the training data may require significant amount of storage. Therefore, some applications that have limited storage may not be suitable. Many researches try to solve this problem through the use of data reduction schemes to reduce the size of training data. From the reasons mentioned above, this research is originated on a template construction concept or a group representative which can represent all other sequences of the same class. The representative, in this research, is computed using a shape averaging technique with DTW alignment called Accurate Shape Average (ASA) on the training data of the same class. This technique reduces the training data storage requirement to only one sequence per class. In the experiment, template based data classification can significantly reduce data storage and computation time several times better than current methods without sacrificing classification accuracy.

Department : Computer Engineering Student's signature ดารารัตน์ ศรีสาย

Field of study : Computer Engineering Advisor's signature .....

Academic year : 2009

กิตติกรรมประกาศ

ตลอดระยะเวลาในการจัดทำวิทยานิพนธ์ฉบับนี้ ได้มีอุปสรรคต่าง ๆ เกิดขึ้นนานัปการ อันเป็นบทเรียนที่ทรงคุณค่ายิ่งแก่ผู้จัดทำ เพื่อที่จะได้ฝึกฝน เรียนรู้ และแก้ไขปัญหาตลอดจนได้เพิ่มพูนทักษะต่าง ๆ ที่จำเป็นสำหรับการทำวิจัย ซึ่งทั้งหมดนี้ล้วนเป็นปัจจัยที่ช่วยส่งเสริมและผลักดันศักยภาพให้แก่ผู้จัดทำเป็นอย่างมาก อย่างไรก็ตาม วิทยานิพนธ์ฉบับนี้ จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี ถ้าขาดแรงสนับสนุนจากบุคคลหลายฝ่าย ซึ่งผู้จัดทำซาบซึ้งในความกรุณาเหล่านี้อย่างล้นพ้น และใคร่ขอใช้เนื้อหาในกิตติกรรมประกาศของวิทยานิพนธ์ฉบับนี้ เป็นสื่อกลางในการแสดงความขอบพระคุณอย่างสุดซึ้งจากผู้จัดทำ

ประการแรก ขอบพระคุณอาจารย์ที่ปรึกษาวิทยานิพนธ์ฉบับนี้ ผู้ช่วยศาสตราจารย์ ดร.โชติรัตน์ รัตนามหัทธนะ ผู้ซึ่งอบรม สั่งสอน ชี้แนะ และแก้ไขศิษย์คนนี้ด้วยดีเสมอมา อันเป็นปัจจัยหลักที่ทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี

ขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ฉบับนี้ ที่ให้ข้อคิดและข้อเสนอแนะต่าง ๆ อันเป็นประโยชน์อย่างยิ่งในการพัฒนาคุณภาพของวิทยานิพนธ์ฉบับนี้ ซึ่งคณะกรรมการสอบวิทยานิพนธ์นั้น ประกอบไปด้วย ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ และอาจารย์ ดร.ทรงพล องค์วัฒนกุล

ขอบคุณเพื่อน ๆ และพี่ ๆ ในห้องปฏิบัติการทุกคนที่ช่วยให้ชีวิตในการทำวิจัยมีสีสันและมีความหมายมากยิ่งขึ้น รวมทั้งช่วยเสนอแนวคิดต่าง ๆ ในการแก้ไขปัญหา และให้ความร่วมมือในการเก็บตัวอย่างข้อมูลอนุกรมเวลาเพื่อใช้สำหรับการทดลองในงานวิทยานิพนธ์ฉบับนี้

สุดท้ายที่ขาดเสียมิได้ ขอบพระคุณครอบครัวที่น่ารักทุก ๆ คนของผู้จัดทำ ที่เป็นกำลังใจ และให้การสนับสนุนทุกสิ่งทุกอย่างด้วยดีเสมอมา

สารบัญ

หน้า

หน้า

สารบัญภาพ

หน้า

หน้า

หน้า

สารบัญตาราง

หน้า

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

การสร้างแผ่นแบบ (Template) สำหรับมนุษย์นั้นเป็นการสร้างความจำชนิดหนึ่งซึ่งถูกกำหนดขึ้นโดยประสบการณ์การได้รับรู้ซ้ำแล้วซ้ำเล่า ก่อให้เกิดกระบวนการกำหนดและรู้จำรูปแบบ ในมนุษย์กระบวนการนี้สามารถทำได้อย่างง่ายดายและเป็นส่วนหนึ่งของการเรียนรู้ตามธรรมชาติ ซึ่งนอกจากทำให้เกิดการอยู่รอดแล้วยังเป็นส่วนหนึ่งของการพัฒนาความฉลาด ในกระบวนการเรียนรู้นี้อาศัยความคล้ายกันเป็นเครื่องมือในการช่วยให้รับรู้ถึงประสบการณ์ที่เกิดขึ้นซ้ำ ๆ ได้ ดังนั้นหากต้องการสร้างความฉลาดให้เกิดขึ้นกับเครื่องจักรโดยอาศัยการเรียนรู้ของมนุษย์เป็นต้นแบบ จะต้องทำให้เครื่องจักรสามารถบอกถึงความคล้ายกันให้ได้ การบอกความคล้ายกันโดยเครื่องจักรนั้นเป็นเรื่องหนึ่งที่มีความยากและซับซ้อน ในขณะที่มนุษย์สามารถบอกได้ถึงความคล้ายกันของวัตถุได้อย่างรวดเร็วเสมือนไม่ต้องใช้ความพยายาม อีกทั้งความคล้ายกันหรือเหมือนกันสำหรับเครื่องจักรนั้นหากต้องการให้ง่าย จะต้องเป็นความเหมือนกันทุกประการจึงจะเรียกว่าเหมือน ซึ่งคุณสมบัตินี้ไม่ปรากฏโดยทั่วไปในธรรมชาติจึงเป็นความท้าทายอันหนึ่ง ที่จะให้เครื่องจักรเป็นตัวระบุความเหมือนหรือความคล้ายกันของวัตถุ

สำหรับคำว่า ความคล้ายกัน ตามพจนานุกรม ฉบับราชบัณฑิตยสถาน พ.ศ. 2542 มีความหมายว่า คล้ายคลึง ส่อแสดงให้รู้ได้ว่ามีลักษณะเหมือนสิ่งอื่นหรือเกือบเหมือน ซึ่งจากความหมายนี้ก็ไม่ได้ทำให้ทราบความหมายที่ชัดเจนมากขึ้น แต่ในความเป็นจริงแล้วก็เป็นที่ทราบกันดีว่าความคล้ายกันหมายถึงอะไร แต่ยากที่จะให้คำนิยาม เพราะความคล้ายกันเป็นเรื่องที่ขึ้นกับความคิดของแต่ละบุคคล (Subjectivity) ในการทำเหมืองข้อมูล (Data Mining) กับอนุกรมเวลา (Time Series) นั้นมีหัวใจหลักที่สำคัญที่สุดก็คือ การให้นิยามหรือให้ความหมายของความคล้ายกันระหว่างอนุกรมเวลา 2 อนุกรม ซึ่งเมื่อเราหาความคล้ายกันของอนุกรมเวลาได้แล้ว เราก็สามารถนำมาใช้ประโยชน์ได้หลากหลาย ตัวอย่างเช่น การจำแนกประเภทของข้อมูล (Classification) [1, 2] การทำดัชนีหรือการค้นข้อมูลจากเนื้อหา (Indexing / Query by Content) [3-5] การจัดกลุ่มข้อมูล (Clustering) [6] และการตรวจหาสิ่งผิดปกติหรือสิ่งที่น่าสนใจ (Anomaly/ Interestingness detection) [7] เป็นต้น ซึ่งทุก ๆ อย่างนี้ต้องการนิยามความคล้ายกันระหว่างอนุกรมเวลา และถ้าหากเราไม่สามารถให้นิยามของความคล้ายกันระหว่าง 2 อนุกรมเวลาที่ถูกต้องได้ ผลลัพธ์ที่ได้จากการทำเหมืองข้อมูลก็ไม่สามารถออกมาเป็นคำตอบที่ถูกต้องได้

สำหรับข้อมูลอนุกรมเวลา การวัดความคล้ายคลึง (Similarity Measurement) จึงเป็นส่วนที่สำคัญมาก เนื่องจากเป็นวิธีที่มีทั้งความแม่นยำและประสิทธิภาพสูง ตัวอย่างมาตรวัดระยะทาง (Distance Measure) ที่เป็นที่นิยมใช้ในการวัดความคล้ายคลึงของข้อมูลอนุกรมเวลา ได้แก่ ตัววัดความคล้ายแบบยุคลิด (Euclidean Distance Metric) ซึ่งเป็นวิธีวัดระยะทางแบบจุดต่อจุด (One-to-One) วิธีนี้เป็นวิธีที่คำนวณได้อย่างรวดเร็วและไม่ซับซ้อน เนื่องจากมีขีดจำกัดเชิงสัญกรณ์ (Asymptotic Limit) เท่ากับ $O(n)$ เมื่อ n คือความยาวของอนุกรมเวลา แต่วิธีการนี้มีข้อจำกัดคือ ไม่รองรับกับข้อมูลที่มีการแปรผันเชิงเวลา ทำให้การเปรียบเทียบความคล้ายคลึงด้วยวิธีนี้มีประสิทธิภาพด้านความแม่นยำไม่ค่อยสูงมาก ดังนั้นจึงมีวิธีเปรียบเทียบความคล้ายคลึงอีกวิธีหนึ่ง ซึ่งเป็นวิธีที่มีประสิทธิภาพด้านความแม่นยำสูงกว่าวิธียุคลิดเรียกว่าวิธีไดนามิกไทม์วอร์ปปิง (Dynamic Time Warping-DTW) [8] เนื่องจากการวัดระยะทางด้วยวิธีนี้รองรับกับอนุกรมเวลาที่มีการแปรผันเชิงเวลา ซึ่งอนุญาตให้เกิดการจับคู่ของจุดระหว่างอนุกรมเวลาไม่เป็นแบบหนึ่งต่อหนึ่ง หมายถึงมีการปรับแนวระหว่างจุดข้อมูลที่มีการเลื่อนทางแกนเวลาเพื่อให้สามารถคำนวณระยะทางสะสมน้อยที่สุดได้ อย่างไรก็ตามวิธีไดนามิกไทม์วอร์ปปิงนี้ยังมีข้อบกพร่องเรื่องความเร็วในการคำนวณระยะทาง เนื่องจากวิธีการนี้มีขีดจำกัดเชิงสัญกรณ์ในด้านเวลา เท่ากับ $O(n^2)$ ทำให้เมื่อต้องคำนวณหาระยะทางระหว่างอนุกรมเวลาที่มีความยาวและจำนวนมาก ๆ นั้น การเปรียบเทียบความคล้ายคลึงด้วยวิธีไดนามิกไทม์วอร์ปปิงก็จะเสียเวลาในการคำนวณมาก จากปัญหาดังกล่าว ได้มีงานวิจัย [9, 10] เป็นจำนวนมากที่มุ่งพัฒนาเพื่อเพิ่มความเร็วในการวัดความคล้ายคลึงของข้อมูลอนุกรมเวลา โดยใช้วิธีการวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงเป็นตัวกำหนดความคล้ายคลึง

การจำแนกประเภทข้อมูลของอนุกรมเวลา ซึ่งวิธีที่นิยมใช้คือวิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้สุดอันดับที่หนึ่ง (1-Nearest Neighbor) ด้วยการวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง คือถ้ามีข้อมูลสอบถาม (Query Sequence) เข้ามาในระบบ แล้วต้องการจำแนกว่าข้อมูลอนุกรมเวลาตัวนี้อยู่ในกลุ่มใด ระบบจะทำการเปรียบเทียบความคล้ายคลึงระหว่างข้อมูลสอบถามกับข้อมูลที่อยู่ในฐานข้อมูลทั้งหมด ด้วยวิธีการวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง เพื่อตรวจสอบว่าข้อมูลสอบถามคล้ายกับข้อมูลตัวใดมากที่สุด แล้วระบบก็จะจำแนกประเภทของข้อมูลสอบถามให้อยู่ในประเภทเดียวกับข้อมูลที่คล้ายกับข้อมูลสอบถามมากที่สุด ในการจำแนกประเภทดังกล่าวข้างต้นจะเห็นว่าต้องมีการเปรียบเทียบความคล้ายหรือมีการวัดระยะทางหลายครั้ง ทำให้การจำแนกประเภทข้อมูลอนุกรมเวลาใช้เวลานาน

ถ้าในฐานข้อมูลมีข้อมูลอนุกรมเวลาแบ่งออกเป็น 3 คลาส โดยที่แต่ละคลาสมีจำนวนอนุกรมเวลาเท่ากับหนึ่งแสนอนุกรม ปัญหาที่เกิดขึ้นก็คือ ถ้าต้องการจำแนกประเภทข้อมูลสอบถามหนึ่งอนุกรม จะต้องทำการคำนวณระยะทางด้วยวิธีไดนามิกไทม์วอร์ปปิงทั้งหมดสามแสนครั้ง จึงจะสามารถระบุได้ว่าข้อมูลสอบถามนี้อยู่ในคลาสใด นอกจากนี้จะเห็นว่าฐานข้อมูลนี้ถือเป็นฐานข้อมูลที่มีขนาดใหญ่มาก ในบางโปรแกรมประยุกต์ที่มีข้อจำกัดในด้าน

หน่วยเก็บข้อมูล อย่างเช่น งานที่เกี่ยวกับการรู้จำคำพูด (Speech Recognition) ที่ต้องมีการโหลดข้อมูลไปทำงานบนระบบฝังตัว (Embedded System) ซึ่งมีหน่วยเก็บข้อมูลอยู่อย่างจำกัด นั้น ถ้าฐานข้อมูลมีขนาดใหญ่มากก็จะไม่สามารถโหลดข้อมูลไปทำงานได้

จากปัญหาข้างต้น การสร้างแผ่นแบบจึงเป็นวิธีหนึ่งที่น่าสนใจในการนำมาแก้ปัญหาดังกล่าว เนื่องจากการสร้างแผ่นแบบก็คือการหาตัวแทนกลุ่มของข้อมูลอนุกรมเวลา โดยเป็นการลดจำนวนข้อมูลภายในกลุ่มข้อมูลเรียนรู้ ซึ่งก็จะสามารถแก้ปัญหาทั้งเรื่องความเร็วในการจำแนกประเภทข้อมูลและปริมาณของข้อมูลที่ต้องจัดเก็บในหน่วยเก็บข้อมูล เพราะถ้าสามารถหาแผ่นแบบหรือตัวแทนของข้อมูลทั้งกลุ่มได้ การคำนวณเพื่อจำแนกประเภทของข้อมูลจากที่ต้องเปรียบเทียบความคล้ายคลึงระหว่างข้อมูลสอบถามกับข้อมูลในฐานข้อมูลทั้งหมด ก็จะเหลือการเปรียบเทียบระหว่างข้อมูลสอบถามกับแผ่นแบบเท่านั้น อีกทั้งเมื่อเราหาแผ่นแบบที่สามารถแทนข้อมูลของทั้งกลุ่มได้แล้ว การจัดเก็บข้อมูลอนุกรมเวลาก็จะเก็บเฉพาะตัวแผ่นแบบเท่านั้น จากตัวอย่างข้างต้น หากไม่มีแผ่นแบบจะต้องเก็บข้อมูลอนุกรมในหน่วยเก็บข้อมูลทั้งหมดสามแสนอนุกรม แต่ถ้ามีแผ่นแบบแล้วจะเหลือข้อมูลอนุกรมเวลาที่ต้องจัดเก็บเพียงแค่สามอนุกรมเท่านั้น

งานวิจัยที่มุ่งพัฒนาเพื่อที่จะลดจำนวนข้อมูลในกลุ่มข้อมูลเรียนรู้มีอยู่เป็นจำนวนมาก แต่งานวิจัยที่เป็นการหาแผ่นแบบเพื่อเป็นตัวแทนกลุ่มสำหรับข้อมูลอนุกรมเวลานั้นได้แก่วิธี AWARD [11] เป็นวิธีการเรียงลำดับความสำคัญของข้อมูลในกลุ่มข้อมูลเรียนรู้ จากนั้นเลือกข้อมูลอนุกรมเวลาที่มีความสำคัญลำดับในลำดับต้น ๆ มาเป็นตัวแทนกลุ่มของข้อมูลเรียนรู้แต่ละกลุ่ม ซึ่งวิธีการนี้เป็นการนำข้อมูลอนุกรมเวลาเพียงอนุกรมเดียวมาแทนข้อมูลทั้งกลุ่มซึ่งเป็นการไม่สมเหตุสมผล เนื่องจากข้อมูลเพียงตัวเดียวที่ถูกหยิบออกมาจะไม่สามารถแทนข้อมูลทั้งกลุ่มได้ เมื่อนำแผ่นแบบที่ได้จากวิธี AWARD มาใช้ในการจำแนกประเภทข้อมูลก็จะทำให้ความแม่นยำในการจำแนกประเภทของข้อมูลนั้นต่ำ

นอกจากการเลือกข้อมูลมาเป็นตัวแทนกลุ่มแล้วยังมีอีกวิธีที่สามารถหาตัวแทนกลุ่มสำหรับข้อมูลอนุกรมเวลาได้ นั่นก็คือการหาค่าเฉลี่ยสำหรับข้อมูลอนุกรมเวลาทุกตัวที่อยู่ภายในกลุ่ม โดยการหาค่าเฉลี่ยของข้อมูลอนุกรมเวลานั้นจะต้องอาศัยการปรับแนว (Alignment) เพื่อเป็นตัวช่วยระบุว่าจุดข้อมูลคู่ใดระหว่างข้อมูลอนุกรมเวลาที่จะนำมาหาค่าเฉลี่ยคู่กัน การปรับแนวที่ง่ายที่สุดก็คือการปรับแนวแบบหนึ่งต่อหนึ่ง ซึ่งการหาค่าเฉลี่ยที่ใช้การปรับแนวแบบนี้เรียกว่า การเฉลี่ยแบบเลขคณิต (Arithmetic Averaging) วิธีการนี้เป็นวิธีที่ง่ายและคำนวณได้เร็วที่สุด แต่ก็ยังมีปัญหา เพราะแผ่นแบบที่ได้จากการหาค่าเฉลี่ยของอนุกรมเวลาด้วยวิธีนี้ เมื่อนำแผ่นแบบมาคำนวณหาความคล้ายคลึงด้วยวิธีไดนามิกไทม์วอร์ปปิงกับอนุกรมเวลาตัวอื่นแล้วจะมีค่าระยะทางที่สูงมาก นั่นก็หมายถึงประสิทธิภาพในด้านความแม่นยำลดลงเนื่องด้วยการจับคู่จุดที่ไม่เหมาะสม เพราะการจับคู่แบบหนึ่งต่อหนึ่งระหว่างข้อมูลที่มีการเลื่อนทางแกนเวลาจะทำให้สูญเสียบางลักษณะในแกนเวลาไป และปัญหาอีกอย่างหนึ่งก็คืออนุกรม

เวลาที่จะทำการหาค่าเฉลี่ยเลขคณิตนั้นต้องมีความยาวเท่ากันเพราะถ้าความยาวไม่เท่ากันก็จะไม่สามารถคำนวณแบบหนึ่งต่อหนึ่งได้

งานวิจัยต่อ ๆ มา [12, 13] จึงพยายามแก้ปัญหานี้โดยใช้วิธีการหาค่าเฉลี่ยข้อมูลอนุกรมเวลาโดยใช้การปรับแนวแบบไทม์วอร์ปปิง (Time-Warping Alignment) เพื่อเป็นการระบุคู่ที่ต้องหาค่าเฉลี่ยแทน ซึ่งการระบุคู่เพื่อหาค่าเฉลี่ยด้วยวิธีการนี้เรียกว่า การเฉลี่ยรูปร่างอนุกรมเวลา (Time Series Shape Averaging) การสร้างแผ่นแบบด้วยวิธีนี้ แผ่นแบบที่ได้ยังคงมีลักษณะในแกนเวลาเหมือนกับข้อมูลในกลุ่ม ทำให้เมื่อมีการวัดความคล้ายคลึง ค่าระยะทางที่ได้จากการวัดระยะทางระหว่างแผ่นแบบและอนุกรมเวลาตัวอื่นจะมีค่าน้อย อย่างไรก็ดี วิธีการหาค่าเฉลี่ยรูปร่างอนุกรมเวลาด้วยวิธีใช้การปรับแนวแบบไทม์วอร์ปปิงในหลาย ๆ งานวิจัย ก็ยังมีจุดบกพร่อง ได้แก่ การจับคู่ระหว่างอนุกรมเวลา 2 อนุกรมแบบไม่เป็นหนึ่งต่อหนึ่งนั้น จะเห็นว่าจำนวนจุดข้อมูลหลังจากทำการหาค่าเฉลี่ยจะเพิ่มมากขึ้น เมื่อต้องทำการเฉลี่ยอนุกรมเวลาหลาย ๆ ครั้งจะทำให้ความยาวของอนุกรมเวลาที่ได้จากการหาค่าเฉลี่ยนั้นมีความยาวเพิ่มมากขึ้น ซึ่งไม่เหมาะสมที่จะใช้เป็นตัวแทนกลุ่ม เพราะการวัดความคล้ายคลึงด้วยวิธีไดนามิกไทม์วอร์ปปิงที่ต้องนำอนุกรมเวลาตัวอื่นมาเปรียบเทียบกับแผ่นแบบ ถ้าแผ่นแบบมีความยาวมากๆ จะทำให้เวลาที่ใช้ในการคำนวณแต่ละครั้งนั้นนานมาก และข้อบกพร่องอีกประการหนึ่ง คือข้อมูลที่ได้จากการหาค่าเฉลี่ยจะไม่อยู่ในตำแหน่งกริด (Grid)

สำหรับงานวิจัยนี้ มีวัตถุประสงค์เพื่อนำเสนอวิธีการสร้างแผ่นแบบสำหรับใช้เป็นตัวแทนกลุ่มของข้อมูลอนุกรมเวลา เพื่อลดจำนวนของข้อมูลอนุกรมเวลาในกลุ่มข้อมูลเรียนรู้ให้เหลือเพียงอนุกรมเดียวต่อหนึ่งกลุ่ม ทั้งนี้แผ่นแบบที่ได้ยังสามารถแทนข้อมูลอนุกรมเวลาทั้งกลุ่มได้ สำหรับการสร้างแผ่นแบบนั้นจะใช้วิธีการหาค่าเฉลี่ยรูปร่างข้อมูลอนุกรมเวลาโดยใช้การปรับแบบแนวผสมระหว่างไทม์วอร์ปปิงกับไทม์วอร์ปปิงแบบอนุพันธ์ เพื่อเป็นตัวกำหนดว่าจุดข้อมูลคู่ใดจะทำการเฉลี่ยค่ากัน นอกจากนี้แผ่นแบบที่ได้จะสามารถกำหนดความยาวให้จำกัดได้ และแต่ละจุดข้อมูลของอนุกรมเวลาที่เป็นแผ่นแบบนั้นจะมีค่าในแกน X เป็นจำนวนเต็มหรืออยู่ในตำแหน่งกริดนั่นเอง สำหรับวิธีวัดความคล้ายคลึงของแผ่นแบบที่เป็นตัวแทนกลุ่มของอนุกรมเวลาจะใช้วิธีวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง โดยผลที่ได้รับจะมีประสิทธิภาพทั้งด้านความเร็วและความแม่นยำ ในส่วนของการทดลองใช้วิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้สุดอันดับที่หนึ่งซึ่งเป็นตัววัดประสิทธิภาพความคล้ายคลึงของข้อมูลอนุกรมเวลา

1.2 วัตถุประสงค์ของการวิจัย

1. นำเสนอวิธีการสร้างแผ่นแบบที่ใช้เป็นตัวแทนกลุ่มของข้อมูลอนุกรมเวลา
2. เพื่อลดจำนวนข้อมูลที่เก็บไว้ในหน่วยเก็บข้อมูล
3. เพื่อเพิ่มความเร็วในการจำแนกประเภทข้อมูลสำหรับข้อมูลอนุกรมเวลา

4. เพื่อเพิ่มประสิทธิภาพในด้านความแม่นยำในการจำแนกประเภทข้อมูลอนุกรมเวลาโดยใช้แผ่นแบบ

1.3 ขอบเขตของการวิจัย

1. พัฒนาวิธีการสร้างแผ่นแบบ เพื่อหาตัวแทนกลุ่มของข้อมูลอนุกรมเวลา
2. สร้างแผ่นแบบสำหรับกลุ่มข้อมูลอนุกรมเวลาโดยใช้วิธีการหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลาที่อยู่ภายในกลุ่มข้อมูลเรียนรู้
3. แผ่นแบบที่เป็นตัวแทนกลุ่มข้อมูลอนุกรมเวลา สามารถแทนข้อมูลอนุกรมเวลาตัวอื่น ๆ ในกลุ่มได้ด้วยข้อมูลอนุกรมเวลาเพียงอนุกรมเดียว
4. ประยุกต์ใช้การปรับแนวแบบไทม์วอร์ปปิง (Time-Warping Alightment) เพื่อเป็นตัวกำหนดคู่ของจุดข้อมูลที่จะนำมาทำการหาค่าเฉลี่ย
5. ประยุกต์ใช้ทฤษฎีการประมาณค่าด้วยวิธีกระดูกงูกำลังสาม (Cubic Spline Approximation) สำหรับประมาณค่าจุดข้อมูลของข้อมูลอนุกรมเวลาให้อยู่ในตำแหน่งที่ต้องการเพื่อเป็นการแก้ปัญหาข้อมูลอนุกรมเวลาไม่ลงกริด และสามารถกำหนดความยาวของข้อมูลอนุกรมเวลาตามที่ต้องการหลังจากที่ทำการเฉลี่ยรูปร่างข้อมูลอนุกรมเวลาแล้ว
6. ทดสอบความแม่นยำและความเร็วของวิธีที่นำเสนอ โดยใช้วิธีจำแนกข้อมูลโดยการประเมินผลวัดจากวิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้สุดอันดับที่หนึ่งโดยวิธีการวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงกับข้อมูลอนุกรมเวลา และเปรียบเทียบผลกับวิธีอื่น ๆ เพื่อที่จะแสดงให้เห็นว่าวิธีที่นำเสนอสามารถจำแนกประเภทข้อมูลอนุกรมเวลาได้อย่างแม่นยำและรวดเร็วมากกว่าวิธีในปัจจุบัน

1.4 ประโยชน์ที่ได้รับ

ได้วิธีการสร้างแผ่นแบบสำหรับเป็นตัวแทนกลุ่มของข้อมูลอนุกรมเวลา เพื่อลดจำนวนของข้อมูลที่ต้องเก็บไว้ในหน่วยเก็บข้อมูล และยังเป็นการลดจำนวนครั้งในการคำนวณเปรียบเทียบความคล้ายคลึงด้วยการวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงระหว่างข้อมูลอนุกรมเวลา โดยที่ยังคงสามารถจำแนกประเภทข้อมูลอนุกรมเวลาได้อย่างมีประสิทธิภาพทั้งในด้านความแม่นยำและความเร็ว

1.5 วิธีดำเนินการวิจัย

1. ศึกษาการทำเหมืองข้อมูลกับข้อมูลอนุกรมเวลา
2. ศึกษางานวิจัยที่เกี่ยวข้องกับการสร้างแผ่นแบบ (Template Construction)

3. ศึกษางานวิจัยที่เกี่ยวข้องกับการสร้างแผ่นแบบเพื่อเป็นตัวแทนสำหรับเป็นตัวแทนกลุ่มข้อมูลอนุกรมเวลา ทั้งวิธีการหาค่าเฉลี่ยแบบเลขคณิต (Arithmetic Averaging) โดยใช้การปรับแนวแบบยุคลิด (Euclidean Alignment) หรือที่เรียกว่า การปรับแนวแบบหนึ่งต่อหนึ่ง และวิธีการหาค่าเฉลี่ยรูปร่าง (Shape Averaging) โดยใช้วิธีการปรับแนวแบบไทม์วอร์ปปิง (Time-Warping Alignment) พร้อมทั้งวิเคราะห์ข้อดีและข้อเสียของงานวิจัยที่เกี่ยวข้อง
4. ศึกษาแนวทางเพื่อนำหลักการทางคณิตศาสตร์มาช่วยแก้ไขปัญหาเรื่องตำแหน่งของข้อมูลอนุกรมเวลาที่ไม่ลงกริดและจำนวนจุดข้อมูลที่เพิ่มมากขึ้น
5. ออกแบบและพัฒนาวิธีการสร้างแผ่นแบบที่เหมาะสมสำหรับทำเป็นตัวแทนกลุ่มข้อมูลอนุกรมเวลาโดยใช้การปรับแนวแบบไทม์วอร์ปปิง
6. ออกแบบและพัฒนาวิธีการหาค่าของข้อมูลในตำแหน่งกริดด้วยวิธีการประมาณด้วยวิธีกระดูกงูกำลังสาม
7. ทดสอบประสิทธิภาพความแม่นยำและความเร็วของวิธีที่นำเสนอ โดยการเปรียบเทียบผลการทดลองในการจำแนกข้อมูลกับวิธีอื่นๆ การประเมินผลวัดจากการจำแนกข้อมูลแบบเพื่อนบ้านใกล้ที่สุดอันดับที่หนึ่ง ด้วยวิธีการวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง
8. วิเคราะห์และสรุปผลการทดลอง
9. สรุป เรียบเรียง และจัดทำวิทยานิพนธ์

1.6 ผลงานตีพิมพ์จากงานวิจัย

ส่วนหนึ่งของงานวิทยานิพนธ์นี้ ได้รับการตีพิมพ์เป็นบทความทางวิชาการสองเรื่อง ดังนี้

- Time Series Shape Averaging Using Time-Warping Alignment with Re-Sampling” โดย ดารารัตน์ ศรีใส และโชติรัตน์ รัตนามหัทธนะ ในงานประชุมวิชาการ “6th International Joint Conference on Computer Science and Software Engineering” ซึ่งจัดขึ้น ณ เมืองภูเก็ต ประเทศไทย ระหว่างวันที่ 13 พฤษภาคม ถึง 15 พฤษภาคม 2552 ดังรายละเอียดใน ภาคผนวก ข
- “Efficient Time Series Classification under Template Matching using Time Warping Alignment” โดยดารารัตน์ ศรีใส และโชติรัตน์ รัตนามหัทธนะ ในงานประชุมวิชาการนานาชาติครั้งที่ 4 “The 2009 International Conference on Computer Sciences and Convergence Information Technology” ซึ่งจัดขึ้น ณ เมืองโซล ประเทศเกาหลีใต้ ระหว่างวันที่ 24 พฤศจิกายน ถึง 26 พฤศจิกายน 2552 ดังรายละเอียดในภาคผนวก ค

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

สำหรับทฤษฎีและงานวิจัยที่เกี่ยวข้อง จะมีการนำเสนอทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับการสร้างแผ่นแบบที่เหมาะสมสำหรับข้อมูลอนุกรมเวลาเพื่อใช้เป็นตัวแทนของกลุ่ม การหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลา การจัดแนวแบบไทม์วอร์ปปิง วิธีการหาค่าประมาณของข้อมูลอนุกรมเวลาในตำแหน่งที่ต้องการ ซึ่งทั้งหมดนี้เป็นพื้นฐานในการวิจัยและพัฒนาวิธีการหาตัวแทนกลุ่มข้อมูลอนุกรมเวลา สุดท้ายตามด้วยงานวิจัยที่เกี่ยวข้องกับการสร้างแผ่นแบบเพื่อเป็นตัวแทนกลุ่มของข้อมูลอนุกรมเวลาด้วยวิธีการลดจำนวนข้อมูลในกลุ่มข้อมูลเรียนรู้ และการหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลาแบบต่าง ๆ

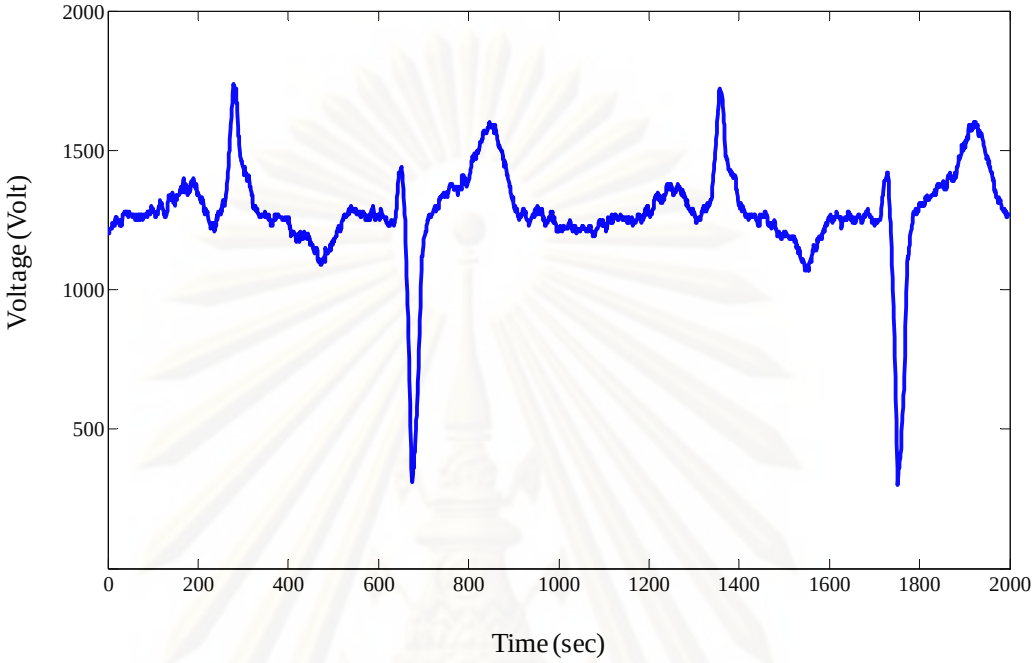
2.1 ทฤษฎีที่เกี่ยวข้อง

สำหรับหัวข้อทฤษฎีที่เกี่ยวข้องกับงานวิจัยนี้ จะเริ่มต้นนำเสนอจาก ความรู้เกี่ยวกับข้อมูลอนุกรมเวลาเพื่อให้ทราบว่าข้อมูลอนุกรมเวลาคืออะไร ตามด้วยขั้นตอนการวัดความคล้ายคลึง (Similarity Measure) แบบต่างๆ ซึ่งแต่ละวิธีจะให้การปรับแนวที่แตกต่างกันดังต่อไปนี้ ตัววัดความคล้ายแบบยุคลิด (Euclidean Distance Metric) ซึ่งเป็นวิธีที่ให้การปรับแนวแบบหนึ่งต่อหนึ่ง (One-to-One Alignment) การวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง (Dynamic Time Warping Distance Measure) จะให้การปรับแนวแบบไทม์วอร์ปปิง (Time-Warping Alignment) และการวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงของอนุพันธ์ (Derivative Dynamic Time Warping Distance Measure) [14] โดยที่การวัดระยะทางแบบนี้จะได้การปรับแนวแบบไทม์วอร์ปปิงของอนุพันธ์ (Derivative Time-Warping Alignment-DDTW) ต่อจากนั้นจะเป็นส่วนของการประมาณฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิงด้วยการวัดระยะทางแบบยุคลิด เพื่อเพิ่มประสิทธิภาพในด้านความเร็ว ในขั้นตอนการคำนวณไดนามิกไทม์วอร์ปปิง และในส่วนสุดท้ายจะเป็นส่วนของการประมาณค่าด้วยวิธีกระดูกงูกำลังสาม (Cubic Spline Approximation)

2.2 ข้อมูลอนุกรมเวลา (Time Series Data)

ข้อมูลอนุกรมเวลา เป็นข้อมูลที่สามารถพบได้ทั่วไปในชีวิตประจำวัน เนื่องจากมนุษย์เรามีการวัดค่าต่าง ๆ อยู่ตลอดเวลา ซึ่งค่าเหล่านั้นมักจะมีการเปลี่ยนแปลงไปเรื่อย ๆ ตามกาลเวลา เพราะฉะนั้นถ้าเรามีการบันทึกข้อมูลเหล่านั้นเอาไว้ ข้อมูลเหล่านั้นก็จะถือว่าเป็นข้อมูลอนุกรมเวลา หรืออาจกล่าวได้ว่า ข้อมูลอนุกรมเวลา คือข้อมูลที่มาจากการเก็บรวบรวมข้อมูลใด ๆ ก็ได้ที่ทำเป็นลำดับตามเวลาก่อนหลัง ตัวอย่างเช่น ข้อมูลของตลาดหุ้น ข้อมูลอัตรา

การเต้นของหัวใจ ข้อมูลความดันโลหิต ข้อมูลปริมาณน้ำฝนต่อปี ข้อมูลอุณหภูมิ ข้อมูลค่าความกดอากาศ เป็นต้น ดังแสดงในรูปที่ 2.1 ซึ่งเป็นกราฟแสดงข้อมูลคลื่นหัวใจของคนเรา ซึ่งข้อมูลได้มาจากการเก็บข้อมูลแรงดัน (Voltage) ของคลื่นหัวใจไปเรื่อย ๆ ตามลำดับเวลา



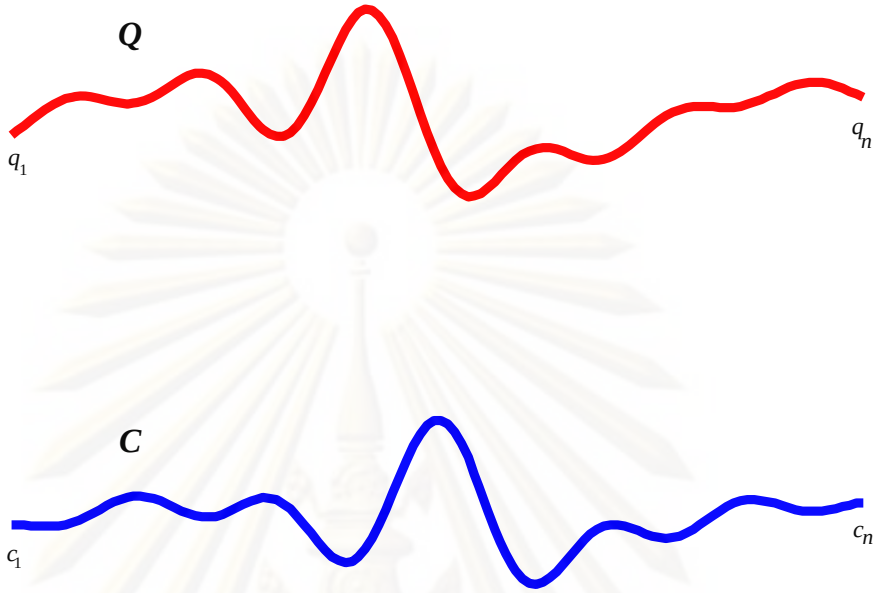
รูปที่ 2.1 ข้อมูลคลื่นหัวใจมนุษย์

2.3 ตัววัดความคล้ายแบบยุคลิด (Euclidean Distance Metric)

ตัววัดความคล้ายแบบยุคลิด เป็นวิธีการวัดระยะทางระหว่างข้อมูลอนุกรมเวลา 2 อนุกรม ซึ่งเป็นวิธีที่สามารถคำนวณได้ง่ายและเร็วที่สุด เนื่องจากมีขีดจำกัดเชิงสัญกรณ์เป็นเชิงเส้นหรือเท่ากับ $O(n)$ เท่านั้น การคำนวณระยะทางระหว่างข้อมูลอนุกรมเวลา 2 อนุกรมนั้นจะเป็นการจับคู่แบบหนึ่งต่อหนึ่ง กล่าวคือ จุดที่ 1 จะจับกับจุดที่ 1 ของอีกอนุกรมเวลา จุดที่ 7 จับกับจุดที่ 7 จุดที่ 20 จับกับจุดที่ 20 เป็นต้น โดยที่อนุกรมเวลาทั้งสองอนุกรมจะต้องยาวเท่ากัน มิฉะนั้นจะไม่สามารถจับแบบหนึ่งต่อหนึ่งได้

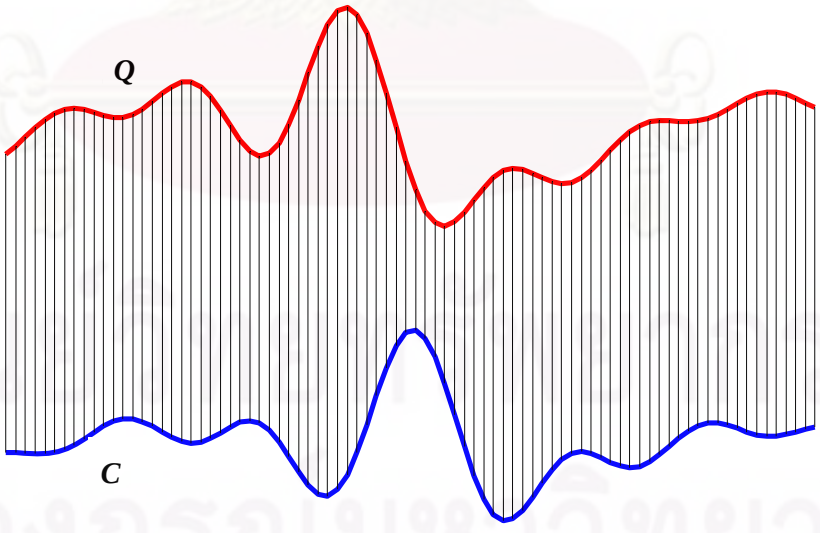
สำหรับนิยามของวิธียุคลิดนั้น สามารถอธิบายโดยละเอียดได้ดังนี้ ให้อนุกรมเวลา Q คือ ข้อมูลสอบถาม (Query Sequence) และ C คือ ข้อมูลที่จะใช้ในการเปรียบเทียบ (Candidate Sequence) ที่มีความยาว n เท่ากัน โดยที่ $Q = q_1, q_2, \ldots, q_n$ และ $C = c_1, c_2, \ldots, c_n$ ดังแสดงในรูปที่ 2.2 โดยการคำนวณหาระยะทางก็สามารถคำนวณจาก ค่าจากอนุกรมเวลาแต่ละอนุกรม ณ ตำแหน่งเดียวกัน นำมาลบกันแล้วยกกำลังสอง เช่น $(q_1 - c_1)^2$ $(q_{10} - c_{10})^2$ หรือ $(q_{45} - c_{45})^2$ เป็นต้น จากนั้นนำค่าที่ได้มารวมกันทั้งหมดแล้วถอดรากที่สองตามสมการที่ (2.1)

$$\text{Euclidean}\,(Q,C) = \sqrt{\sum_{i=1}^{n}(q_i - c_i)^2} \tag{2.1}$$



รูปที่ 2.2 อนุกรมเวลา Q และ C ที่มีความยาว n

เนื่องจากการวัดระยะทางแบบยุคลิดเป็นการจับคู่แบบหนึ่งต่อหนึ่ง ดังนั้นการปรับแนวของวิธีนี้ก็จะเป็นการปรับแนวแบบหนึ่งต่อหนึ่ง ดังรูปที่ 2.3

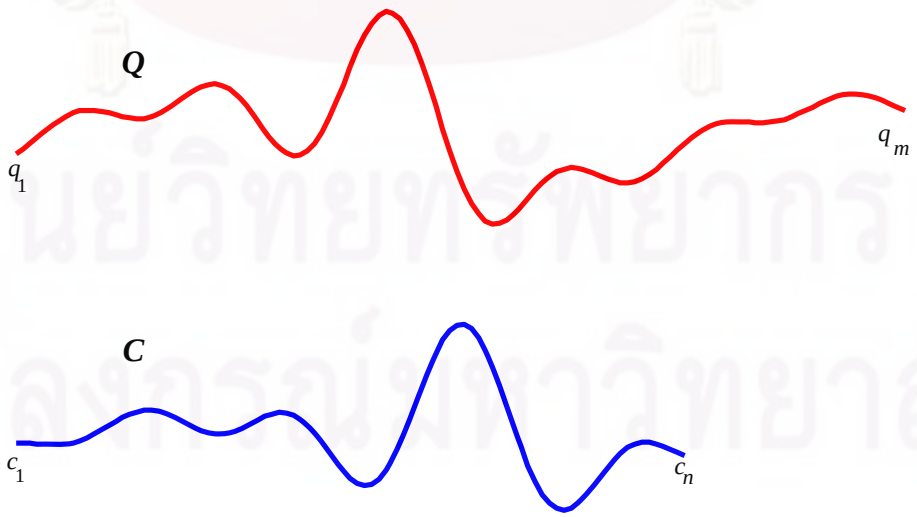


รูปที่ 2.3 การปรับแนวแบบหนึ่งต่อหนึ่ง

2.4 มาตรวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง (Dynamic Time Warping Distance Measure หรือ DTW)

ไดนามิกไทม์วอร์ปปิง เป็นวิธีการวัดความคล้ายคลึงกัน (Similarity Measure) ระหว่างข้อมูลอนุกรมเวลา 2 อนุกรม ซึ่งวิธีนี้เป็นวิธีวัดความคล้ายคลึงที่นิยมใช้กันมาก เนื่องจากไดนามิกไทม์วอร์ปปิงเป็นวิธีที่มีจุดเด่นคือ สามารถจับคู่แบบไม่เป็นหนึ่งต่อหนึ่งได้ หมายความว่า การวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงนี้ อนุกรมเวลา 2 อนุกรมที่นำมาคำนวณระยะทางไม่จำเป็นต้องมีความยาวเท่ากันก็ได้ และวิธีนี้ยังเหมาะสำหรับข้อมูลอนุกรมเวลาที่มีความแปรผันเฉพาะที่เชิงเวลา (Local Variation) ซึ่งผลลัพธ์ที่ได้จากการวัดความคล้ายคลึงดังกล่าว จะเป็นค่าระยะทางระหว่างข้อมูลอนุกรมเวลาทั้ง 2 อนุกรม สามารถหาค่าได้จากการคำนวณค่าระยะทางสะสม (Cumulative Distance) ระหว่างจุดข้อมูลในอนุกรมเวลาดังกล่าวที่มีการปรับแนวระหว่างกัน ซึ่งการปรับแนวนี้ทำเพื่อเป็นการรองรับความแปรผันเฉพาะที่เชิงเวลาที่เกิดขึ้นในข้อมูลอนุกรมเวลา และยังทำให้การคำนวณได้ค่าระยะทางสะสมที่น้อยที่สุด ยกตัวอย่างของข้อมูลที่มีความแปรผันเฉพาะที่เชิงเวลา เช่น ข้อมูลที่เป็นเสียงพูด เนื่องจากผู้พูดแต่ละคนมีลักษณะการพูดที่แตกต่างกัน บางคนพูดเร็ว บางคนพูดช้า หรือบางคนพูดติดขัด ทั้ง ๆ ที่เป็นข้อความเดียวกัน ก็จะมีความแตกต่างกันในแกนของเวลา ทำให้ถ้าเลือกใช้วิธีการวัดความคล้ายคลึงแบบไดนามิกไทม์วอร์ปปิง ซึ่งมีการปรับแนวระหว่างกันก็จะสามารถทำให้คำนวณหาค่าระยะทางที่น้อยที่สุดได้

การคำนวณหาค่าระยะทางโดยวิธีไดนามิกไทม์วอร์ปปิง สามารถอธิบายในรายละเอียดได้ดังนี้ กำหนดให้มีข้อมูลอนุกรมเวลา 2 อนุกรม ได้แก่ Q คือ ข้อมูลสอบถาม และ C คือ ข้อมูลที่จะใช้ในการเปรียบเทียบโดยมีความยาว m และ n ตามลำดับ เมื่อ $Q = q_1, q_2, \ldots, q_m$ และ $C = c_1, c_2, \ldots, c_n$ ดังแสดงใน รูปที่ 2.4



รูปที่ 2.4 อนุกรมเวลา Q และ C ที่มีความยาว m และ n ตามลำดับ

เมตริกซ์ระยะทางระหว่างอนุกรมทั้งสอง ($D = \{d\}_{m\times n}$) จะมีขนาดเท่ากับ $m\times n$ สามารถคำนวณจากสมการที่ (2.2) ดังนี้

$$d_{i,j} = (q_i - c_j)^2 \tag{2.2}$$

โดยที่ q_i และ c_j คือ ข้อมูลตำแหน่งที่ i และ j ในข้อมูลอนุกรมเวลา Q และ C ตามลำดับ วิถีของการวอร์ป (Warping Path, W) หาได้จากการหาค่าน้อยสุดของระยะทางสะสมระหว่างอนุกรมทั้งสอง องค์ประกอบของวิถีของการวอร์ป (w_k) มีค่าเท่ากับคู่ลำดับ $(i, j)_k$ โดยที่ i อยู่ระหว่าง 1 ถึง m $(1 \le i \le m)$ j อยู่ระหว่าง 1 ถึง n $(1 \le j \le n)$ k อยู่ระหว่าง 1 ถึง K $(1 \le k \le K)$ และ K อยู่ระหว่าง ค่าสูงสุดระหว่าง m กับ n และ $m+n-1$ $(\max(m, n) \le K \le m+n-1)$ วิถีของการวอร์ปที่เหมาะสมที่สุดจะเป็นวิถีที่มีต้นทุนการวอร์ปต่ำที่สุด ตามสมการที่ (2.3)

$$\mathrm{DTW}(Q, C) = \sqrt{\sum_{k=1}^{K} D(w_k)} \tag{2.3}$$

วิถี (Path) สามารถหาได้โดยการใช้วิธีกำหนดการพลวัต (Dynamic Programming) เพื่อคำนวณหาค่าระยะทางสะสม (Cumulative Distance, $\gamma_{i,j}$) น้อยที่สุดจากสามส่วนย่อยที่ประชิดกัน ดังนี้ $\gamma_{i,j} = d_{i,j} + \min\{\gamma_{i-1,j-1}, \gamma_{i-1,j}, \gamma_{i,j-1}\}$ โดยมีเงื่อนไขดังต่อไปนี้

- เงื่อนไขขอบ (Boundary Condition)

  วิถีจะต้องเริ่มต้นจากตำแหน่ง $w_1 = (1, 1)$ และสิ้นสุดที่ตำแหน่ง $w_k = (m, n)$

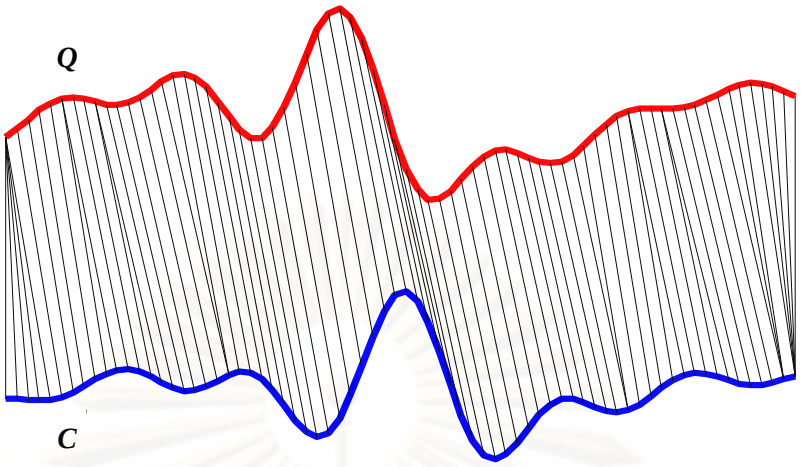
- เงื่อนไขภาวะต่อเนื่อง (Continuity Condition)

  ค่าดัชนี i และ j สามารถเพิ่มได้ไม่เกิน 1 ในแต่ละขั้นตลอดทั้งวิถี

- เงื่อนไขทางเดียว (Monotonic Condition)

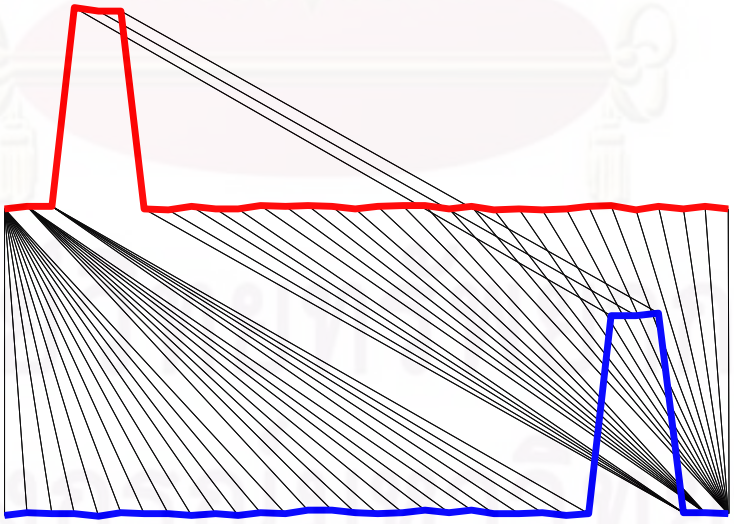
  วิถีของการวอร์ปสามารถเคลื่อนไปข้างหน้าตามแกนเวลาเท่านั้น

เนื่องจากวิธีไดนามิกไทม์วอร์ปปิง เป็นการจับคู่แบบไม่เป็นหนึ่งต่อหนึ่ง การปรับแนวที่เกิดขึ้นจึงเป็นการปรับแนวแบบไทม์วอร์ปปิง (Time-Warping Alignment) ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 การปรับแนวแบบไทม์วอร์ปปิง

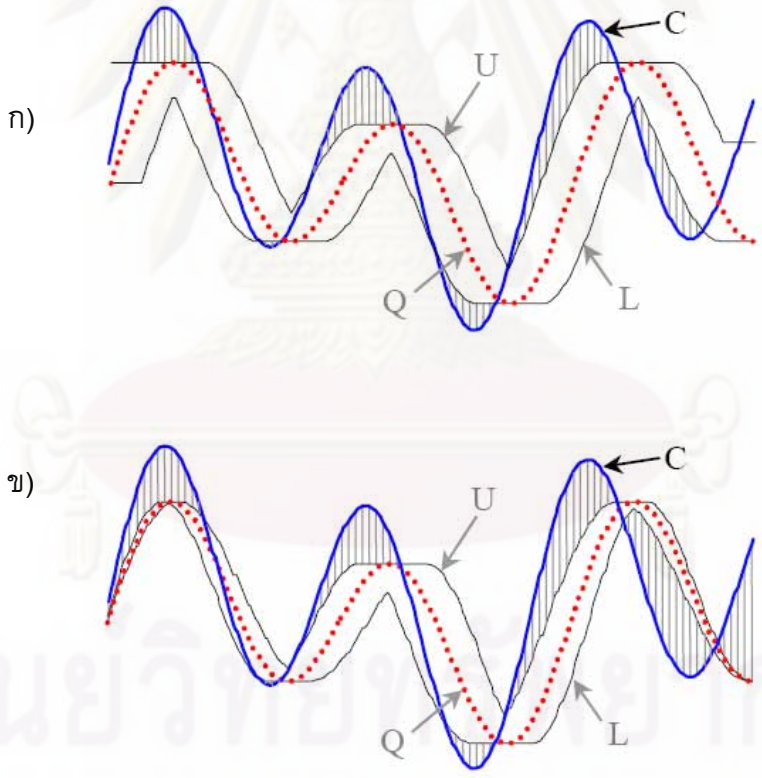
แม้ว่าวิธีไดนามิกไทม์วอร์ปปิงนี้ จะสามารถคำนวณค่าระยะทางระหว่างข้อมูลอนุกรมเวลาที่มีความแปรผันเชิงเวลาได้ป็นอย่างดี แต่ในการคำนวณค่าระยะทางด้วยวิธีไดนามิกไทม์วอร์ปปิงนั้น ซึ่งในบางกรณี วิธีดังกล่าวต้องการให้ค่าระยะทางสะสมที่ได้ออกมามีค่าน้อยที่สุด ทำให้วิธีไดนามิกไทม์วอร์ปปิงเลือกการปรับแนวบางคู่จุดไม่เหมาะสมดังแสดงในรูปที่ 2.6 กล่าวคือมีการปรับแนวให้มีการคำนวณค่าระยะทางระหว่างจุดยอดของข้อมูลอนุกรมเวลาทั้งสองที่อยู่ในส่วนต้นและส่วนปลายของข้อมูล โดยข้อมูลทั้งสองอาจเป็นข้อมูลต่างประเภทกัน นอกจากนั้นการคำนวณหาระยะทางของวิธีไดนามิกไทม์วอร์ปปิงเป็นการคำนวณหาค่าระยะทางสะสม ทำให้การคำนวณใช้เวลานาน เพราะมีขีดจำกัดเชิงสัญกรณ์เท่ากับ $O(n^2)$



รูปที่ 2.6 การปรับแนวในการหาระยะทางอย่างไม่เหมาะสมของวิธีไดนามิกไทม์วอร์ปปิง [15]

2.4.1 ฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปปิง (Lower Bounding Function of DTW)

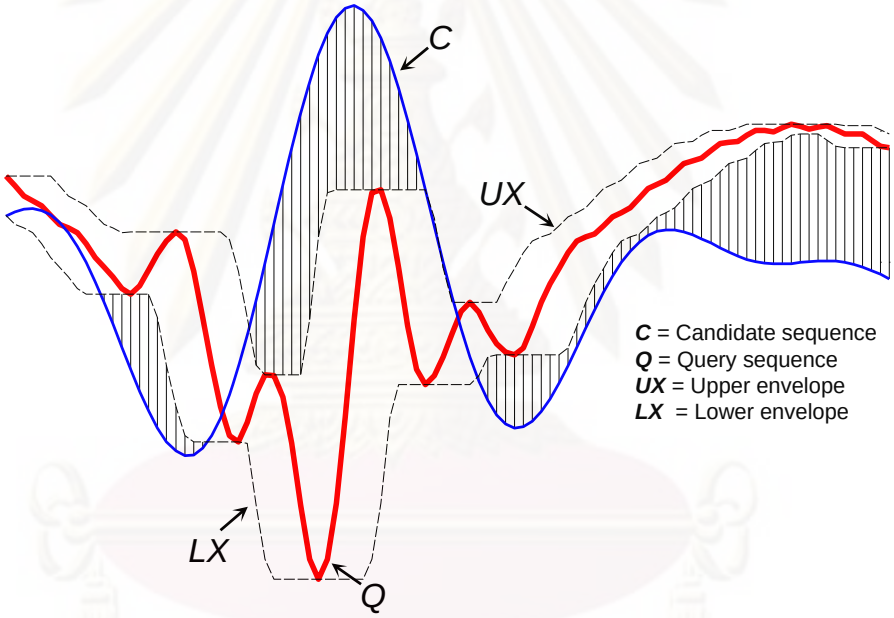
จากที่กล่าวมาข้างต้น การคำนวณด้วยวิธีไดนามิกไทม์วอร์ปปิงนั้นต้องใช้เวลาในการคำนวณสูง เพราะมีขีดจำกัดเชิงสัญกรณ์ในด้านเวลาเท่ากับ $O(n^2)$ นั่นคือใช้เวลาในการคำนวณเป็นฟังก์ชันพหุนามกับความยาวของข้อมูลขาเข้า ดังนั้นจึงได้มีผู้เสนอฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปปิงที่ใช้การคำนวณแบบยุคลิดขึ้น [3, 4, 9, 10] เพื่อใช้ในการประมาณค่าระยะทางระหว่างข้อมูลอนุกรมเวลา 2 อนุกรมอย่างมีประสิทธิภาพ ก่อนที่จะทำการคำนวณหาค่าระยะทางจริงด้วยวิธีไดนามิกไทม์วอร์ปปิง เพื่อเป็นการลดปริมาณข้อมูลที่ต้องคำนวณระยะทางด้วยวิธีไดนามิกไทม์วอร์ปปิงลง ค่าระยะทางที่ได้จากการคำนวณฟังก์ชันขอบเขตล่าง จะต้องมีค่าไม่เกินค่าระยะทางจริงที่ได้จากการคำนวณด้วยวิธีไดนามิกไทม์วอร์ปปิง จนกระทั่งมีงานวิจัยล่าสุดของ Keogh [9] ได้เสนอฟังก์ชันขอบเขตล่างสำหรับไดนามิกไทม์วอร์ปปิงด้วยการวัดระยะทางแบบยุคลิด เรียกว่า LB_Keogh ดังแสดงในรูปที่ 2.7



รูปที่ 2.7 ตัวอย่างการคำนวณฟังก์ชันขอบเขตล่างของระยะทางแบบไดนามิกไทม์วอร์ปปิง LB_Keogh ก) ฟังก์ชันขอบเขตล่างภายใต้การกำหนดเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะ ข) ฟังก์ชันขอบเขตล่างภายใต้การกำหนดเงื่อนไขบังคับโดยรวมอิตาคุระ (ที่มา : Keogh และ Ratanamahatana [9])

ในส่วนของรูปที่ 2.7 ก) เป็นการกำหนดเงื่อนไขบังคับโดยรวม (Global Constraint) ในรูปแบบของซาโก-ชิบะ [16] และรูปที่ 2.7 ข) จะเป็นการกำหนดเงื่อนไขบังคับโดยรวมในรูปแบบของอิตาคุระ [17] โดยการกำหนดขอบเขตช่วงบนของแต่ละจุดบนข้อมูลอนุกรมเวลา U และขอบเขตช่วงล่างของแต่ละจุดบนข้อมูลอนุกรมเวลา L จากค่าในการคำนวณกำหนดการพลวัตภายใต้เงื่อนไขบังคับโดยรวมสำหรับข้อมูลสอบถาม

วิธีใช้งานการสร้างฟังก์ชันขอบเขตล่างของไดนามิกไทม์วอร์ปปิงนั้น สามารถทำได้โดยสร้างเส้นขอบเขตบนและขอบเขตล่างจากเงื่อนไขบังคับโดยรวมของข้อมูลสอบถามเพื่อเป็นตัวแทนของข้อมูลสอบถามในการคำนวณหาค่าระยะทางขอบเขตล่างกับข้อมูลอนุกรมเวลาที่ใช้ในการเปรียบเทียบด้วยวิธียุคลิด ดังแสดงในรูปที่ 2.8 ซึ่งแสดงการคำนวณหาค่าระยะทางด้วยวิธียุคลิดระหว่างเส้นขอบเขตบน (UX) และเส้นขอบเขตล่าง (LX) กับข้อมูลอนุกรมเวลา C



รูปที่ 2.8 การคำนวณค่าระยะทางขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปปิง (ที่มา : Euachongprasit และ Ratanamahatana [18])

สำหรับฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปปิงนั้น สามารถอธิบายรายละเอียดได้ดังนี้ กำหนดข้อมูลอนุกรมเวลาสองอนุกรม ได้แก่ข้อมูลอนุกรมเวลา Q ซึ่งประกอบไปด้วยจุดข้อมูล $q_1, q_2, q_3, \ldots, q_m$ และข้อมูลอนุกรมเวลา C ซึ่งประกอบด้วยจุดข้อมูล $c_1, c_2, c_3, \ldots, c_n$ โดยที่ข้อมูลอนุกรมเวลา Q และ C มีความยาว m และ n ตามลำดับ โดยที่ $m = n$ และ $UX = ux_1, ux_2, \ldots, ux_m$ เป็นเส้นขอบเขตบนของข้อมูลอนุกรมเวลา Q ส่วน $LX = lx_1, lx_2, \ldots, lx_m$ เป็นเส้นขอบเขตล่างของข้อมูลอนุกรมเวลา Q โดยสามารถคำนวณค่า LX และ UX ของแต่ละจุดบนข้อมูลอนุกรมเวลาได้จากสมการที่ (2.4)

$$UX_i = \max(q_{\min(1,i-r)},...,q_{\max(i+r,m)})$$
$$LX_i = \max(q_{\min(1,i-r)},...,q_{\max(i+r,m)}) \quad (2.4)$$

กำหนดให้ $\mathrm{LBX}(Q,C)$ เป็นฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปปิง ระหว่างข้อมูลอนุกรมเวลา Q และ C โดยสามารถคำนวณได้ตามสมการที่ (2.5)

$$\mathrm{LBX}(Q,C) = \sum_{i=1}^{m} \begin{cases} (UX_i - c_i)^2 & \text{if } c_i > UX_i \\ (LX_i - c_i)^2 & \text{if } c_i < LX_i \\ 0 & \text{otherwise} \end{cases} \quad (2.5)$$

2.5 ไดนามิกไทม์วอร์ปปิงแบบอนุพันธ์ (Derivative Dynamic Time Warping หรือ DDTW)

ถ้าไดนามิกไทม์วอร์ปปิงใช้วิธีการปรับแนวระหว่างข้อมูลอนุกรมเวลา 2 อนุกรมที่มีความคล้ายคลึงกัน ยกเว้นส่วนที่เป็นความเร่งและความหน่วงเฉพาะที่ในแกนเวลา ขั้นตอนวิธีนี้จะมีปัญหาเมื่ออนุกรมเวลาทั้งสองมีความแตกต่างในแกน Y ด้วยเช่นกัน ความแตกต่างโดยรวม (Global Differences) มีผลกระทบตลอดทั้งอนุกรมเวลา เช่น มีค่าเฉลี่ยที่แตกต่างกัน มีมาตราส่วนที่แตกต่างกัน หรือมีแนวโน้มเชิงเส้นที่ถูกกำจัดได้ง่าย [19, 20] อย่างไรก็ดีอนุกรมเวลาทั้งสองยังอาจมีความแตกต่างเฉพาะที่ (Local Differences) ในแกน Y อีกด้วย

ซึ่งวิธีไดนามิกไทม์วอร์ปปิงนั้น ในการคำนวณแต่ละครั้งจะมีเฉพาะข้อมูลในแกน Y เพียงจุดเดียวเท่านั้นที่ถูกนำมาพิจารณา ตัวอย่างเช่น เมื่อพิจารณาจุดข้อมูล 2 จุด คือ q_i และ c_j ซึ่งมีค่าเท่ากันทุกประการ แต่ q_i อยู่ในส่วนที่มีแนวโน้มกำลังจะเพิ่มและ c_j อยู่ในส่วนที่มีแนวโน้มกำลังลดลง วิธีไดนามิกไทม์วอร์ปปิงจะพิจารณาการจับคู่ทั้ง 2 จุดนี้โดยไม่คำนึงถึงแนวโน้มที่แตกต่างกัน ซึ่งการที่วิธีไดนามิกไทม์วอร์ปปิงจับคู่จุดข้อมูลที่ไม่เหมาะสมเช่นนั้นอาจส่งผลให้ค่าระยะทางที่ได้ไม่ถูกต้อง เพื่อป้องกันปัญหานี้การปรับปรุงวิธีไดนามิกไทม์วอร์ปปิงที่ไม่ใช้ค่าในแกน Y แต่จะใช้คุณสมบัติในขั้นสูงกว่าของรูปทรง โดยที่คำว่ารูปทรงอาจหมายถึงอนุพันธ์อันดับหนึ่งของอนุกรมเวลา ดังนั้นจึงเรียกขั้นตอนวิธีนี้ว่า ไดนามิกไทม์วอร์ปปิงแบบอนุพันธ์ (Derivative Dynamic Time Warping)

ไดนามิกไทม์วอร์ปปิงตามปกติจะใช้เมตริกซ์ระยะทาง (Distance Metrix) ที่เป็นการคำนวณโดยใช้วิธีหาระยะทางแบบยุคลิด แต่สำหรับไดนามิกไทม์วอร์ปปิงแบบอนุพันธ์ การวัดระยะทางนั้นจะใช้ผลต่างกำลังสองของ q_i และ c_j เป็นค่าประมาณของอนุพันธ์ตามสมการที่ (2.6) ดังนี้

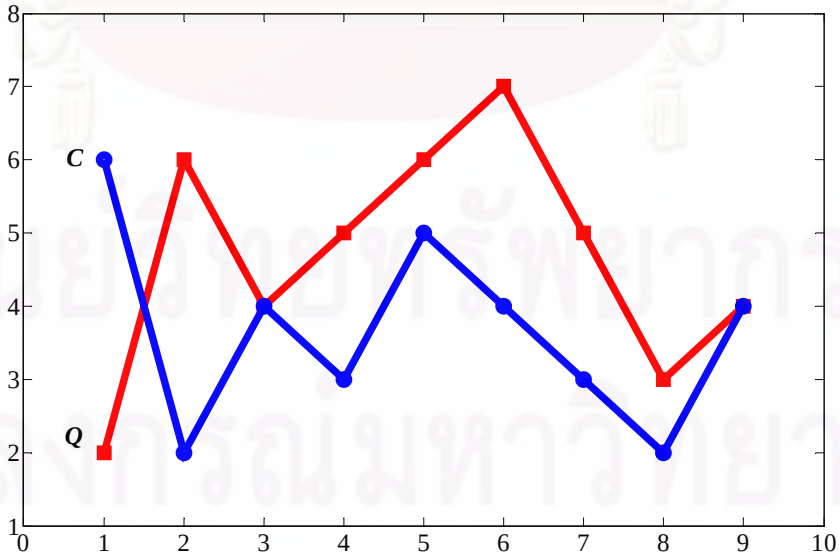
$$D_x[q] = \frac{(q_i - q_{i-1}) + ((q_{i+1} - q_{i-1})/2)}{2} \quad (2.6)$$

การประมาณนี้เป็นการประมาณค่าความชัน โดยการหาค่าเฉลี่ยของความชันระหว่างเส้นตรงที่ผ่านจุดที่ต้องการกับจุดเพื่อนบ้านทางซ้ายมือ และความชันของเส้นตรงจากเพื่อนบ้านซ้ายมือถึงเพื่อนบ้านขวามือของจุดที่ต้องการ การประมาณนี้มีความทนทานต่อข้อมูลแปลกแยก (Outlier) สำหรับกลุ่มข้อมูลที่มีสัญญาณรบกวน (Noise) อาจใช้การปรับเรียบด้วยเส้นโค้งเลขชี้กำลัง (Exponential Smoothing) ก่อนการประมาณค่าอนุพันธ์

ไดนามิกไทม์วอร์ปปิงแบบอนุพันธ์มีความซับซ้อนเท่ากับ $O(n^2)$ ซึ่งเทียบเท่ากับวิธีไดนามิกไทม์วอร์ปปิง เพราะการคำนวณของทั้งสองเหมือนกันทุกประการ ยกเว้นการคำนวณระยะทางเท่านั้น

2.6 ค่าเฉลี่ยรูปร่าง (Shape Averaging)

ค่าเฉลี่ยรูปร่าง คือการหาค่าเฉลี่ยระหว่างอนุกรมเวลาสองอนุกรม โดยในแต่ละคู่ของจุดข้อมูลที่นำมาเฉลี่ย สามารถจับคู่ได้จากการปรับแนวแบบไทม์วอร์ปปิง เพื่อให้ค่าเฉลี่ยรูปร่างที่คำนวณได้ยังคงมีลักษณะในแกนเวลาของข้อมูลต้นแบบทั้งสองอยู่ด้วย การหาค่าเฉลี่ยรูปร่างนั้นสามารถอธิบายรายละเอียดได้ดังนี้ กำหนดให้มีข้อมูลอนุกรมเวลา 2 อนุกรม ได้แก่ Q = [2 6 4 5 6 7 5 3 4] และ C = [6 2 4 3 5 4 3 2 4] ตัวเลขที่ปรากฏคือ ค่าของข้อมูลอนุกรมเวลาในแกน Y ส่วนค่าในแกนเวลา (แกน X) ก็เรียงตามลำดับ เช่น อนุกรม Q ในตำแหน่งแกนเวลาที่ 3 มีค่าเท่ากับ 4 หรือ อนุกรม C ในตำแหน่งแกนเวลาที่ 7 มีค่าเท่ากับ 3 เป็นต้น ดังแสดงในรูปที่ 2.9



รูปที่ 2.9 ข้อมูลอนุกรมเวลา Q และ C

วิถีการวอร์ปหรือการปรับแนวที่ได้จากการคำนวณไดนามิกไทม์วอร์ปปิงระหว่างอนุกรม Q และ C มีค่าดังนี้ W = {(1,1), (1,2), (2,3), (3,3), (3,4), (4,5), (5,5), (6,5), (7,5), (8,6), (8,7), (8,8), (9,9)} ดังแสดงในรูปที่ 2.10

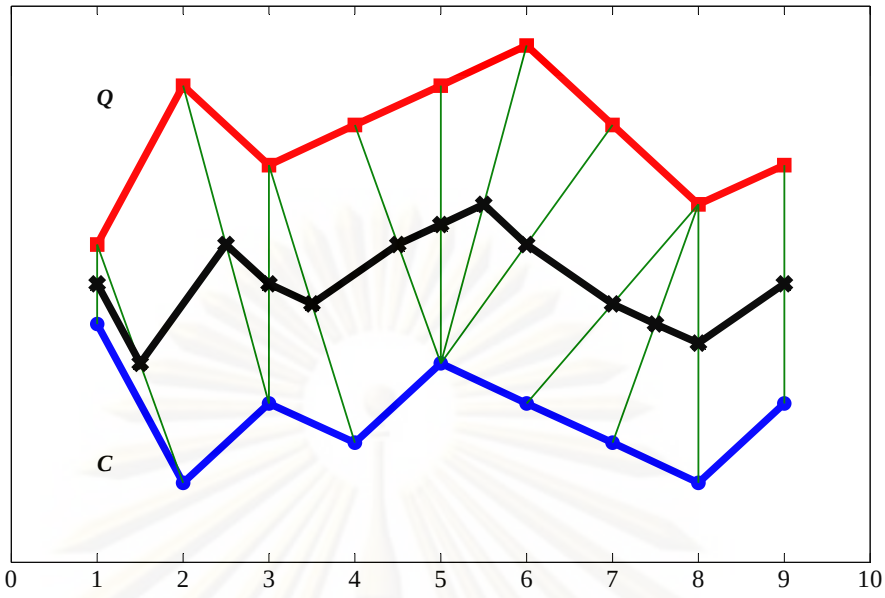
				$D_{i,j}$						Q_i	i
	36	28	24	25	26	26	27	31	**28**	4	9
	32	24	25	25	29	**27**	**27**	**28**	29	3	8
	23	31	32	36	**26**	27	31	40	41	5	7
	22	46	34	41	**26**	31	41	56	41	7	6
	21	37	25	30	**22**	25	31	42	32	6	5
	21	29	21	24	**21**	22	26	32	28	5	4
	20	20	**20**	**21**	22	22	23	27	27	4	3
	16	32	**20**	29	22	26	35	51	39	6	2
	16	**16**	20	21	30	34	35	35	39	2	1
C_j	6	2	4	3	5	4	3	2	4		
j	1	2	3	4	5	6	7	8	9		

รูปที่ 2.10 ระยะทางสะสม และวิถีการวอร์ปหรือการปรับแนว (แสดงในช่องที่เป็นสีเทา) ระหว่างอนุกรมเวลา Q และ C

การคำนวณค่าเฉลี่ยรูปร่างสำหรับข้อมูลอนุกรมเวลาสามารถคำนวณได้ตามสมการที่ (2.7)

$$s_k = \left(\frac{w_{k,1} + w_{k,2}}{2}, \frac{Q(w_{k,1}) + C(w_{k,2})}{2} \right) \tag{2.7}$$

โดยที่ $w_{k,1}$ และ $w_{k,2}$ เป็นดรรชนีบอกตำแหน่งที่ k ของการปรับแนวระหว่างคู่ของจุดข้อมูลอนุกรม โดยค่าเฉลี่ยรูปร่างแสดงในรูปที่ 2.10 ผลลัพธ์ที่ได้จากการคำนวณค่าเฉลี่ยรูปร่างเป็นดังนี้ {(1,4), (1.5,2), (2.5,3), (3,4), (3.5,3.5), (4.5,5), (5,5.5), (5.5,6), (6,5), (7,3.5), (7.5,3) , (8,2.5), (9,4)}



รูปที่ 2.11 ข้อมูลอนุกรมเวลาที่ได้จากการหาค่าเฉลี่ยรูปร่าง (เส้นสีดำที่มีเครื่องหมาย x) ระหว่างอนุกรมเวลา Q และ C

2.7 การประมาณค่าด้วยวิธีกระดูกงูกำลังสาม (Cubic Spline Approximation)

ฟังก์ชันกระดูกงู (Spline Function) เป็นฟังก์ชันที่มักถูกใช้ในการทำการประมาณค่าในช่วง (Interpolation) [21, 22] หรือ การปรับเรียบ (Smoothing) ข้อมูล [23, 24] สามารถใช้ได้ทั้งกับข้อมูล 1 มิติหรือหลายมิติ การใช้ฟังก์ชันกระดูกงูเพื่อการประมาณค่าในช่วงจะเป็นกุญแจสำคัญในการคำนวณค่าประมาณของสัญญาณเพื่อให้ข้อมูลนั้นมีคาบที่สม่ำเสมอและมีความยาวของข้อมูลคงที่ได้ โดยที่รูปทรงของสัญญาณไม่ถูกเปลี่ยนแปลงมากนัก

ทฤษฎีของฟังก์ชันกระดูกงูกำลังสาม เริ่มต้นด้วยการแบ่งข้อมูล $S(x)$ ออกเป็น n - 1 ส่วน ที่สามารถแทนที่ด้วยสมการ $s_i(x)$ จำนวน n - 1 สมการ ภายใต้จุดข้อมูลของเงื่อนไขขอบ n จุด ตามสมการที่ (2.8)

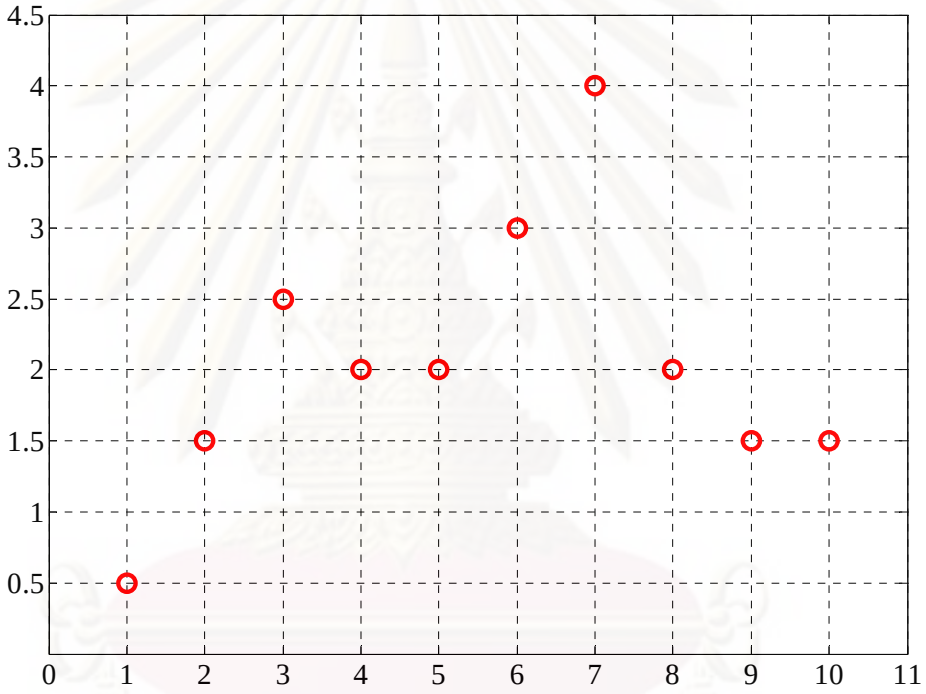
$$S(x) = \begin{cases} s_1(x) & \text{if } x_1 \le x < x_2 \\ s_2(x) & \text{if } x_2 \le x < x_3 \\ \vdots & \vdots \\ s_{n-1}(x) & \text{if } x_{n-1} \le x < x_n \end{cases} \tag{2.8}$$

โดยสมการแต่ละสมการสามารถจัดให้อยู่ในรูปของสมการพหุนามกำลังสามได้ตามสมการที่ (2.9)

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \tag{2.9}$$

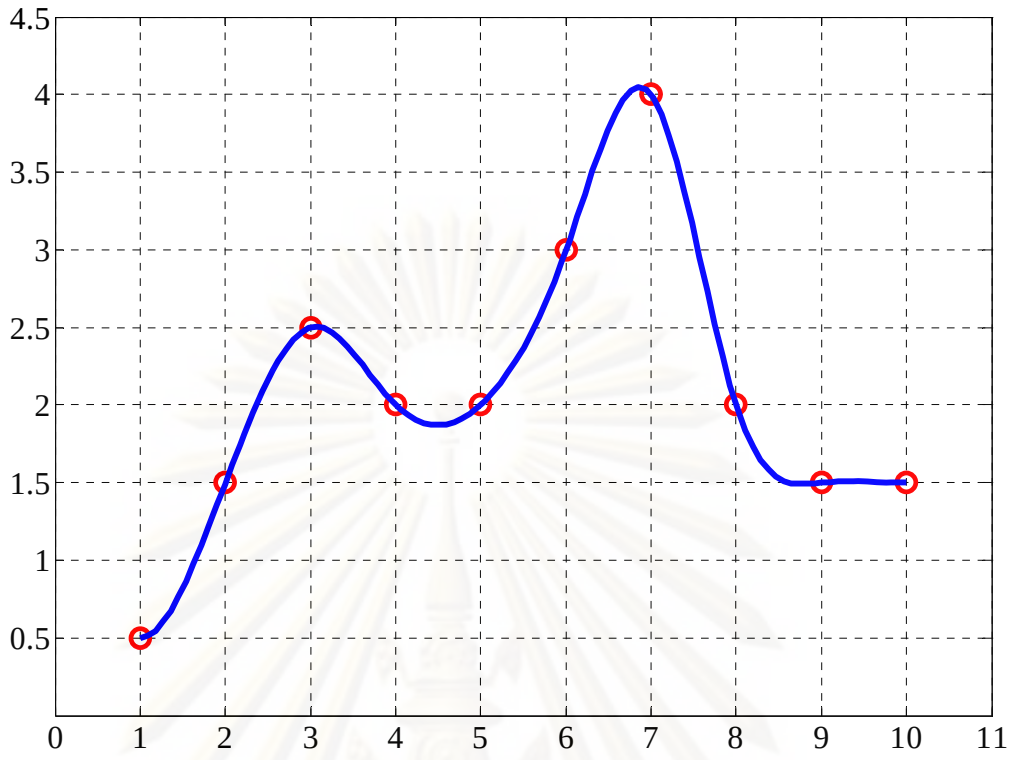
โดยที่ i จะมีค่าตั้งแต่ 1 ถึง n - 1 ก่อนการนำสมการนี้ไปใช้งานจะต้องหาผลเฉลยของ a_i, b_i, c_i และ d_i เสียก่อน ซึ่งขั้นตอนวิธีการหาผลเฉลยสามารถใช้การแก้สมการโดยใช้เมตริกซ์เข้ามาประยุกต์ได้

ในการนำฟังชันก์กระดูกงูกำลังสามไปใช้งานให้แทนค่า x ที่ต้องการลงในสมการข้างต้น ก็จะได้ค่า y ที่สอดคล้องกันเป็นผลลัพธ์ ซึ่งเป็นตัวแทนที่เกิดจากการประมาณในช่วงนั้นเอง ในรูปที่ 2.12 มีจุดข้อมูล 10 จุด โดยแต่ละจุดมีค่าดังนี้ 0.5 1.5 2.5 2 2 3 4 2 1.5 และ 1.5 ถ้าต้องการหาค่าประมาณในช่วงของจุดข้อมูลทั้ง 10 สามารถทำได้โดยใช้ฟังก์ชันกระดูกงูกำลังสาม ซึ่งเมื่อนำข้อมูลทั้ง 10 จุดไปผ่านฟังก์ชันกระดูกงูกำลังสามแล้วจะได้ค่าของข้อมูลในช่วงดังแสดงในรูปที่ 2.13



รูปที่ 2.12 จุดข้อมูล 10 จุด

ในรูปที่ 2.13 แสดงเห็นว่าการใช้ฟังก์ชันกระดูกงูกำลังสามสามารถหาค่าประมาณในช่วงได้ ยกตัวอย่างเช่น ข้อมูลในตำแหน่งที่ 1.5 มีค่าประมาณ 1 ข้อมูลในตำแหน่งที่ 5.7 มีค่าประมาณ 2.5 และข้อมูลในตำแหน่งที่ 7.5 มีค่าประมาณ 3 เป็นต้น



รูปที่ 2.13 การใช้ฟังก์ชันกระดูกงูเพื่อใช้ประมาณค่าในช่วง

2.8 งานวิจัยที่เกี่ยวข้อง

สำหรับข้อมูลอนุกรมเวลา การสร้างแผ่นแบบเพื่อเป็นตัวแทนกลุ่มของข้อมูลนั้นถือว่าเป็นงานวิจัยที่มีความสำคัญ เพราะถ้าสามารถสร้างหรือออกแบบแผ่นแบบที่มีความเหมาะสมสำหรับข้อมูลอนุกรมเวลาได้นั้นจะทำให้เกิดประโยชน์เป็นอย่างยิ่ง เนื่องจากการทำเหมืองข้อมูลสำหรับข้อมูลอนุกรมเวลาส่วนใหญ่แล้วจะเป็นการทำงานกับข้อมูลที่มีปริมาณมาก ซึ่งจะเกิดปัญหาในการจัดเก็บข้อมูลถ้ามีปริมาณหน่วยเก็บข้อมูลที่จำกัด และการทำงานที่เกี่ยวข้องกับข้อมูลอนุกรมเวลาจะทำงานแบบอนุกรมต่ออนุกรม ทำให้เมื่อต้องการทำเหมืองข้อมูลอนุกรมเวลาจำเป็นจะต้องใช้เวลาค่อนข้างนาน ถ้าแผ่นแบบที่สร้างขึ้นมานั้นสามารถเป็นตัวแทนของข้อมูลอนุกรมเวลาทั้งกลุ่มได้ ก็จะเป็นการลดปริมาณข้อมูลอนุกรมเวลาที่ต้องจัดเก็บและคำนวณได้ อีกทั้งประสิทธิภาพที่ได้จากการนำแผ่นแบบไปใช้งานในการจำแนกประเภทข้อมูลอนุกรมเวลาก็จะมีความแม่นยำอีกด้วย

ในหัวข้อนี้จะอธิบายถึงงานวิจัยที่เกี่ยวข้อง โดยจะแบ่งงานวิจัยที่ได้ทำการศึกษาออกเป็น 2 ส่วน ได้แก่ ส่วนที่หนึ่งเป็นงานวิจัยที่เกี่ยวกับการสร้างแผ่นแบบ และส่วนที่สองจะเป็นงานวิจัยที่เกี่ยวกับการหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลา โดยในแต่ละส่วนจะอธิบายวิธีการทำงาน พร้อมทั้งบอกข้อดีและข้อเสียของงานวิจัยแต่ละงาน

2.8.1 งานวิจัยเกี่ยวกับการสร้างแผ่นแบบ

การแก้ปัญหาเรื่องการสร้างแผ่นแบบหรือการหาตัวแทนกลุ่มของข้อมูลนั้น งานวิจัยส่วนใหญ่จะเน้นไปที่การลดจำนวนข้อมูลที่อยู่ในกลุ่มข้อมูลเรียนรู้ (Training Set) ซึ่งการลดจำนวนข้อมูลนั้น ทำได้โดยเลือกข้อมูลบางตัวที่อยู่ในกลุ่มข้อมูลเรียนรู้มาเป็นตัวแทนกลุ่ม โดยวิธีการเลือกข้อมูลนั้นก็มีหลายวิธี เช่น การสุ่มเลือกข้อมูล [25] การจัดลำดับความสำคัญของข้อมูล [26] โดยหลักการในการหาแผ่นแบบนี้ก็ได้ถูกนำไปใช้กับข้อมูลประเภทอื่น ๆ ที่ไม่ใช่ข้อมูลอนุกรมเวลา สำหรับงานวิจัยที่ศึกษาเกี่ยวกับการหาแผ่นแบบสำหรับข้อมูลอนุกรมเวลาเพื่อใช้ประโยชน์ในการจำแนกประเภทของข้อมูลด้วยวิธีการจำแนกข้อมูลแบบเพื่อนบ้านใกล้สุดอันดับที่หนึ่ง โดยใช้การวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงนั้น มีรายละเอียดดังต่อไปนี้

ในปี 2006 Keogh และคณะ [11] ได้เสนอวิธีการหาตัวแทนกลุ่มสำหรับข้อมูลอนุกรมเวลา ซึ่งวิธีที่นำเสนอนี้เรียกว่า Adaptive WARping winDow (AWARD) หลักการที่ใช้หาแผ่นแบบสำหรับวิธี AWARD คือนำข้อมูลอนุกรมเวลาที่อยู่ภายในกลุ่มข้อมูลเรียนรู้มาจัดลำดับความสำคัญ แล้วเลือกอนุกรมเวลาที่มีความสำคัญอันดับต้น ๆ มาเป็นตัวแทนของแต่ละกลุ่มของข้อมูล โดยวิธีที่นำมาใช้ในการจัดลำดับความสำคัญของข้อมูลนั้นเรียกว่า Naïve Rank Reduction โดยการเรียงลำดับความสำคัญนั้นจะนำข้อมูลอนุกรมเวลาที่อยู่ในกลุ่มข้อมูลเรียนรู้มาทำการจำแนกข้อมูลแบบเพื่อนบ้านใกล้ที่สุดอันดับที่หนึ่งด้วยวิธีทดสอบแบบการนำออกหนึ่ง (Leaving-one-out) โดยตัวที่สามารถจำแนกประเภทได้ถูกต้องจะถูกจัดอยู่ลำดับต้น ๆ ส่วนข้อมูลอนุกรมเวลาตัวที่จำแนกประเภทผิดก็จะถูกจัดให้อยู่ในลำดับท้าย จากนั้นนำข้อมูลที่จำแนกประเภทถูกต้องมาให้คะแนนโดยข้อมูลอนุกรมเวลาตัวใดที่ถูกเลือกให้เป็นเพื่อนบ้านใกล้ที่สุดอันดับหนึ่งของข้อมูลอนุกรมเวลาตัวอื่น ๆ มากที่สุดจะมีคะแนนสูง นั่นก็หมายความว่าอยู่ลำดับต้น ๆ เพราะสามารถแทนข้อมูลตัวอื่นได้หลายตัว อย่างไรก็ดีวิธีการนี้เป็นการเลือกข้อมูลอนุกรมเวลาหนึ่งอนุกรมมาเป็นตัวแทนกลุ่มของข้อมูลทั้งหมด โดยที่อนุกรมเวลาที่เลือกมานี้ถึงแม้จะสามารถแทนข้อมูลได้บางส่วน แต่จะไม่สามารถแทนข้อมูลอนุกรมเวลาทั้งกลุ่มได้ เมื่อนำตัวแทนหรือแผ่นแบบที่ได้จากวิธีนี้มาใช้ประโยชน์ในการจำแนกประเภทข้อมูลก็จะทำให้สามารถจำแนกได้เร็วขึ้น แต่ประสิทธิภาพด้านความแม่นยำจะลดลง

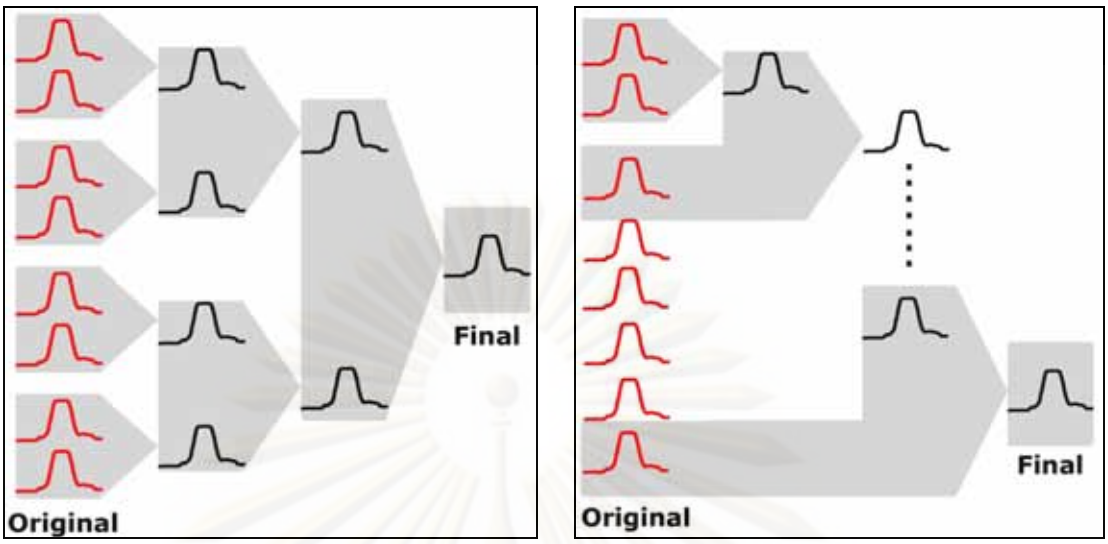
2.8.2 งานวิจัยเกี่ยวกับการหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลา

งานวิจัยที่ศึกษา ค้นคว้า และพัฒนาเกี่ยวกับการหาค่าเฉลี่ยรูปร่างเริ่มปรากฏในปี 1992 ส่วนใหญ่เป็นงานวิจัยในด้านการประมวลผลสัญญาณ (Singal Processing) [13, 27, 28] งานวิจัยในด้านนี้ส่วนมากจะหาค่าเฉลี่ยด้วยการวิเคราะห์เส้นโค้ง เพื่อดูว่าแนวโน้มของสัญญาณว่าจะมีการเปลี่ยนแปลงเส้นโค้งอย่างไรแล้วกำหนดฟังก์ชันทางคณิตศาสตร์เพื่อประมาณค่าเส้นโค้งจากแนวโน้มของสัญญาณที่ได้

ในปี 1996 Gupta และคณะ [12] ได้นำเสนอปัญหาที่เกี่ยวข้องกับการหาค่าเฉลี่ยของการตอบสนองของสมองที่มีการกระตุ้นซ้ำๆ ในการทำให้การประมาณค่าการตอบสนองนี้ดีขึ้น ได้มีการใช้วิธีการหาค่าเฉลี่ยของสัญญาณที่มีการรวมการปรับแนวแบบไม่เป็นเชิงเส้น (Nonlinear Alignment) หรือการปรับแนวแบบไทม์วอร์ปปิง วิธีนี้จะช่วยจับคู่ข้อมูลที่สอดคล้องกันเพราะเป็นการใช้การปรับแนวที่มีความยืดหยุ่นของไดนามิกไทม์วอร์ปปิง ทำให้ค่าระยะทางระหว่างคู่ที่นำมาเฉลี่ยมีค่าน้อย ผลลัพธ์ที่ได้ คือขั้นตอนวิธีในการหาค่าเฉลี่ยสัญญาณที่ได้รักษารูปของสัญญาณไว้ได้ และมีความทนทานที่ดี ขั้นตอนวิธีถูกเรียกว่า Non-linear Alignment and Averaging Filters (NLAAF) ซึ่งภายในประกอบด้วย NLAAF1และ NLAAF2 NLAAF1จะถูกใช้กับข้อมูลอนุกรมเวลาที่มีจำนวนเท่ากับจำนวนที่เป็นเลขสองยกกำลังใด ๆ เช่น 2 8 64 128 เป็นต้น และ NLAAF2 จะถูกใช้เมื่อจำนวนอนุกรมเวลาเป็นเลขจำนวนอื่น ๆ โดยวิธี NLAAF1 จะจับคู่หาค่าเฉลี่ยรูปร่างครั้งละสองอนุกรม ในการหาค่าเฉลี่ยรูปร่างแต่ละครั้งจำนวนของข้อมูลอนุกรมเวลาจะลดลงครึ่งหนึ่งเสมอ และขั้นตอนวิธีนี้จะสิ้นสุดเมื่อเหลือข้อมูลอนุกรมเวลาเพียงอนุกรมเดียวเท่านั้น ส่วนวิธี NLAAF2 จะทำการหาค่าเฉลี่ยรูปร่างข้อมูลอนุกรมเวลาคู่แรกก่อน หลังจากนั้นจะนำผลลัพธ์มาเฉลี่ยกับข้อมูลอนุกรมเวลาที่เหลือครั้งละอนุกรมต่อไปเรื่อย ๆ จนครบทุกข้อมูลอนุกรมเวลา ดังแสดงในรูปที่ 2.14

ในการคำนวณหาค่าเฉลี่ยรูปร่างด้วยวิธี NLAAF ยังมีข้อเสียได้แก่ การเพิ่มขึ้นของจำนวนจุดข้อมูลในผลลัพธ์ทำให้ข้อมูลอนุกรมเวลาที่เป็นผลลัพธ์ที่ได้มีความยาวเพิ่มขึ้น และการขยายจำนวนจุดข้อมูลไม่ได้ขยายแบบเอกรูป (Uniform) จากข้อเสียนี้ถ้านำวิธีการนี้มาสร้างเป็นแผ่นแบบแล้ว การเพิ่มประสิทธิภาพด้านความเร็ว ด้วยวิธีการวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงก็จะไม่เหมาะสม เพราะวิธีวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงไม่เหมาะสำหรับข้อมูลที่มีขนาดยาว จะทำให้การคำนวณช้ามาก

การคำนวณหาค่าเฉลี่ยรูปร่างด้วยวิธี NLAAF นั้น ไม่มีการเรียงลำดับข้อมูลอนุกรมเวลาก่อนที่จะทำการคำนวณ ทำให้เกิดผลเสียคือ หากข้อมูลข้อมูลอนุกรมเวลาคู่แรกเป็นข้อมูลแปลกแยกของกลุ่ม ค่าเฉลี่ยรูปร่างที่เป็นแผ่นแบบท้ายสุดที่ได้ก็จะมีรูปร่างไม่เหมือนกับข้อมูลอนุกรมเวลาตัวอื่น ๆ ที่อยู่ภายในกลุ่ม นอกจากนั้นยังมีงานวิจัยต่อ ๆ มา [29, 30] ที่ทำการหาค่าเฉลี่ยรูปร่างโดยใช้การปรับแนวแบบไทม์วอร์ปปิง และยังมีการเรียงลำดับข้อมูลอนุกรมที่จะนำมาหาค่าเฉลี่ยโดยใช้การจัดกลุ่มข้อมูลแบบขั้น (Hierarchical Clustering) ก่อนเพื่อเฉลี่ยอนุกรมที่มีความคล้ายคลึงกันมากที่สุดก่อน อย่างไรก็ดีแผ่นแบบที่ได้ยังมีความยาวเพิ่มขึ้น ซึ่งไม่เหมาะกับวิธีวัดความคล้ายคลึงแบบไดนามิกไทม์วอร์ปปิง



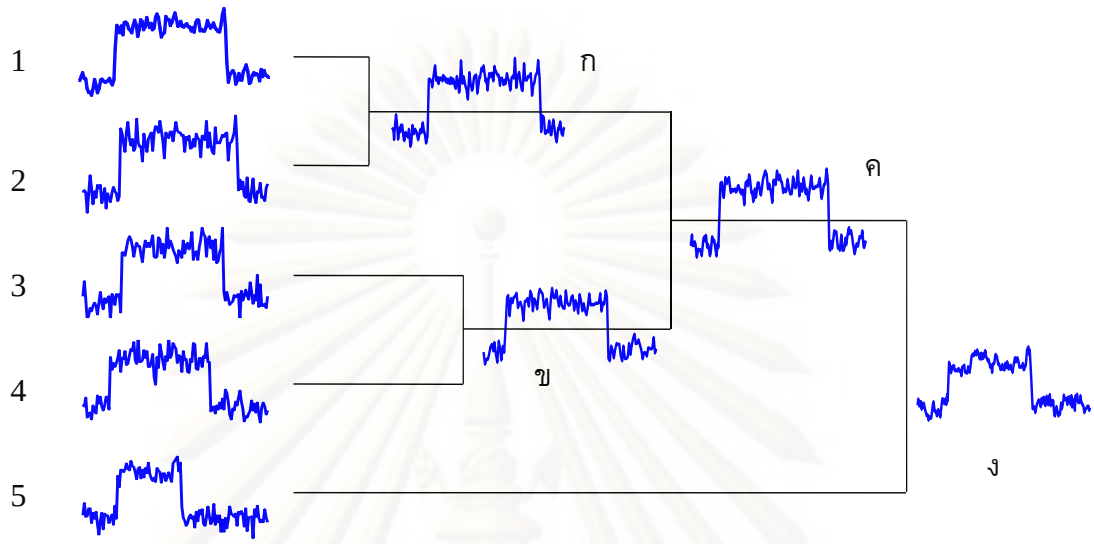
ก) ข)

รูปที่ 2.14 ขั้นตอนวิธี NLAAF ก) NLAAF 1 ข) NLAAF2 (ที่มา : Niennattrakul และ Ratanamahatana [29])

นอกจากการหาค่าเฉลี่ยโดยใช้การปรับแนวของคู่จุดแล้ว ในปี 1998 Keogh และคณะ [20] ได้เสนอวิธีการหาค่าเฉลี่ยข้อมูลอนุกรมเวลา โดยขั้นตอนวิธีที่เสนอเรียกว่า Enhanced Representation of Time Series วิธีนี้ถูกเสนอขึ้นเพื่อทำให้การคำนวณเร็วขึ้นและการจำแนกข้อมูลทำได้แม่นยำขึ้น การแทน (Representation) ประกอบด้วยชิ้นของข้อมูลแบบเชิงเส้นที่ใช้แทนรูปร่างและเวกเตอร์น้ำหนัก กล่าวคือขั้นตอนวิธีนี้เป็นการใช้เส้นตรงหลาย ๆ เส้นมาประกอบกันเป็นค่าเฉลี่ยของรูปร่างข้อมูลอนุกรมเวลา อย่างไรก็ดีการสร้างแผ่นแบบด้วยวิธีการนี้ เมื่อนำแผ่นแบบวัดระยะทางด้วยวิธีไดนามิกไทม์วอร์ปปิงเพื่อใช้ในการเปรียบเทียบความคล้ายคลึง ประสิทธิภาพในด้านความแม่นยำจะไม่สูง เนื่องจากข้อมูลอนุกรมเวลาจริงที่นำมาทดสอบ ไม่ได้มีลักษณะหรือรูปแบบเป็นเส้นตรงหลาย ๆ เส้นมาต่อกัน แต่เป็นจุดหลาย ๆ จุดมาต่อกัน

ในปี 2009 Niennattrakul และคณะ [31] เสนอวิธีการสร้างแผ่นแบบสำหรับข้อมูลอนุกรมเวลา เรียกว่า Prioritized Shape Averaging (PSA) งานวิจัยนี้สร้างแผ่นแบบโดยวิธีการหาค่าเฉลี่ยรูปร่าง เนื่องจากการหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลานั้นสามารถคำนวณได้ครั้งละสองอนุกรมเท่านั้น วิธี PSA ใช้การจับกลุ่มแบบลำดับขั้นเพื่อจัดลำดับว่าข้อมูลอนุกรมเวลาใดจะหาค่าเฉลี่ยก่อนหลัง ดังแสดงในรูปที่ 2.15 แล้วจึงทำการหาค่าเฉลี่ยรูปร่างระหว่างข้อมูลที่ทำการเรียงลำดับขั้นแล้ว จากรูปที่ 2.15 เริ่มต้นจากการหาค่าเฉลี่ยรูปร่างระหว่างข้อมูลอนุกรมเวลาตัวที่ 1 กับตัวที่ 2 ก่อนจะได้เป็นข้อมูลอนุกรมเวลา ก จากนั้นก็หาค่าเฉลี่ยรูปร่างระหว่างข้อมูลอนุกรมเวลาตัวที่ 3 กับ 4 จะได้เป็นข้อมูลอนุกรมเวลา ข แล้วนำ

ข้อมูลอนุกรมเวลา ก และข้อมูลอนุกรมเวลา ข มาหาค่าเฉลี่ยรูปร่างได้เป็นข้อมูลอนุกรมเวลา ค สุดท้ายจึงนำข้อมูลอนุกรมเวลา ค มาเฉลี่ยรูปร่างกับข้อมูลอนุกรมเวลาตัวที่ 5 จะได้เป็นข้อมูลอนุกรมเวลา ง ซึ่งก็คือตัวแทนหรือแผ่นแบบสำหรับข้อมูลอนุกรมเวลา 1-5 นี้



รูปที่ 2.15 ลำดับในการหาค่าเฉลี่ยรูปร่างด้วยวิธี PSA

แต่เมื่อทำการหาค่าเฉลี่ยรูปร่างในแต่ละครั้งแล้ว ข้อมูลอนุกรมเวลาที่เกิดจากการหาค่าเฉลี่ยรูปร่างนั้นมีจำนวนจุดข้อมูลเพิ่มมากขึ้นและระยะห่างระหว่างจุดข้อมูลไม่เป็นแบบเอกรูป วิธี PSA นี้ใช้วิธีการยืดข้อมูลออกให้แต่ละจุดมีระยะห่างที่เท่ากันหรือทำให้จุดข้อมูลขยายออกแบบเอกรูปแล้วจึงตัดบางจุดข้อมูลทิ้งไปและเลือกจุดข้อมูลบางจุดมาใช้เลย ซึ่งการทำเช่นนี้จะทำให้ผลลัพธ์ที่ได้หลังจากการหาค่าเฉลี่ยนั้นสูญเสียคุณลักษณะบางประการของข้อมูลต้นแบบได้ ทำให้เมื่อนำแผ่นแบบที่ได้จากวิธีการนี้มาใช้งาน ประสิทธิภาพในด้านความแม่นยำที่ได้จะต่ำ

บทที่ 3

การสร้างแผ่นแบบเพื่อแทนกลุ่มข้อมูลด้วยการหาค่าเฉลี่ยรูปร่าง

แนวคิดที่ผู้วิจัยได้นำเสนอในงานวิจัยนี้ เป็นการนำเสนอวิธีการสร้างแผ่นแบบเพื่อใช้เป็นตัวแทนกลุ่มสำหรับข้อมูลอนุกรมเวลา ซึ่งงานวิจัยนี้จะมุ่งเน้นไปที่วิธีการหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลาโดยอาศัยการปรับแนวแบบไทม์วอร์ปปิง เพื่อผลลัพธ์ที่ได้นั้นจะถูกนำมาเป็นแผ่นแบบของกลุ่มข้อมูลอนุกรมเวลา โดยที่จุดประสงค์สำคัญของการสร้างแผ่นแบบนั้นเพื่อเป็นการลดจำนวนของข้อมูลอนุกรมเวลาในกลุ่มข้อมูลเรียนรู้ที่ต้องจัดเก็บลงในหน่วยเก็บข้อมูลและเพื่อลดจำนวนครั้งในการคำนวณวัดความคล้ายคลึงเพื่อจำแนกประเภทข้อมูล เมื่อมีแผ่นแบบสำหรับแต่ละกลุ่มข้อมูลอนุกรมเวลาแล้ว ถ้าต้องการเปรียบเทียบความคล้ายคลึงของข้อมูลด้วยการวัดระยะทางด้วยวิธีไดนามิกไทม์วอร์ปปิง ก็สามารถวัดความคล้ายคลึงกับตัวแผ่นแบบเท่านั้น ทำให้เป็นการเพิ่มประสิทธิภาพในการจำแนกประเภทข้อมูลอนุกรมเวลาในด้านความเร็ว

สำหรับในบทที่ 3 ผู้วิจัยจะนำเสนอกรอบงานในการสร้างแผ่นแบบหรือตัวแทนของกลุ่มข้อมูลอนุกรมเวลา เริ่มตั้งแต่ที่มาของข้อมูลอนุกรมเวลาซึ่งได้จากการสกัดลักษณะสำคัญของข้อมูล (Feature Extraction) การแปลงข้อมูลอนุกรมเวลาให้เป็นบรรทัดฐาน (Normalization) และในส่วนสุดท้ายจะเสนอขั้นตอนวิธีในการสร้างแผ่นแบบเพื่อเป็นตัวแทนสำหรับกลุ่มข้อมูลอนุกรมเวลา

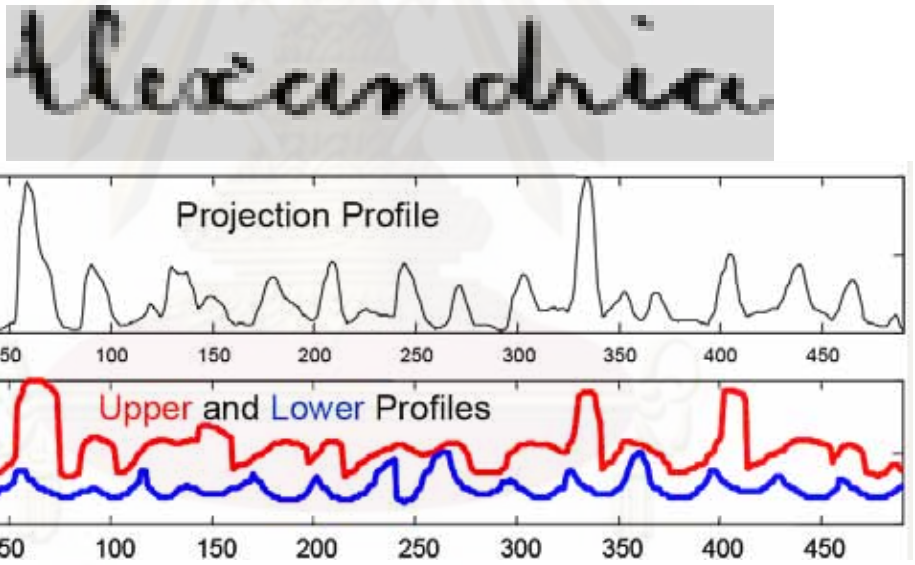
3.1 การสกัดลักษณะสำคัญของข้อมูล

งานประยุกต์ต่าง ๆ ที่ทำเกี่ยวกับการจำแนกประเภทข้อมูลนั้นมักมีให้พบเห็นได้ทั่วไป และเนื่องจากในปัจจุบันเทคโนโลยีในด้านการจัดเก็บข้อมูลนั้นได้มีการพัฒนาอย่างต่อเนื่อง ทำให้ปัญหาในการจัดเก็บข้อมูลที่มีความซับซ้อนสูงจึงไม่เป็นอุปสรรคอีกต่อไป ตัวอย่างของข้อมูลที่มีความซับซ้อน เช่น การจัดเก็บข้อมูลในรูปแบบของข้อมูลภาพ ข้อมูลเสียง อีกทั้งยังรวมถึงข้อมูลในรูปแบบของสื่อประสมต่าง ๆ อย่างไรก็ดีการจำแนกประเภทข้อมูลในรูปแบบข้อมูลที่ซับซ้อนดังกล่าวนั้นมีความยุ่งยากและซับซ้อนมากเมื่อเทียบกับการจำแนกประเภทของข้อมูลบนฐานข้อมูลทั่วไป

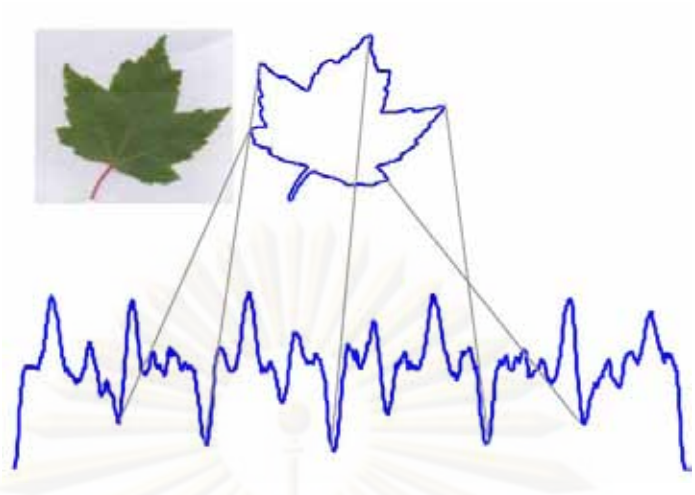
การเปรียบเทียบข้อมูลที่มีความซับซ้อนสูงนั้น ในบางกรณีที่ต้องทำการเปรียบเทียบข้อมูลเหล่านั้นโดยตรงตามรูปแบบของการจัดเก็บข้อมูลแต่ละประเภทนั้น ๆ อาจให้ผลของการเปรียบเทียบที่ไม่ดี ยกตัวอย่างเช่น การเปรียบเทียบความคล้ายคลึงกันของรูปภาพ โดยทั่วไปข้อมูลรูปภาพจะถูกจัดเก็บในรูปแบบของตาราง 2 มิติ ซึ่งเก็บเป็นค่าของสีในแต่ละจุดภาพ ในการเปรียบเทียบความคล้ายคลึงกันโดยตรงนั้นใช้การเปรียบเทียบความ

คล้ายคลึงกันของค่าสีในแต่ละจุดภาพ ซึ่งจะเห็นว่าการเปรียบเทียบเช่นนั้นเป็นการไม่สมเหตุสมผลในการบอกถึงความคล้ายกันของรูปภาพ เพราะในบางกรณีที่ค่าสีในแต่ละจุดของรูปภาพสองรูปอาจให้ค่าความคล้ายคลึงกัน แต่รูปภาพทั้งสองรูปนั้นอาจจะเป็นคนละรูปเลยก็ได้ ดังนั้นงานประยุกต์ด้านการประมวลผลรูปภาพจึงมักจะใช้วิธีการสกัดลักษณะสำคัญของข้อมูล ซึ่งในการทำการสกัดเฉพาะคุณลักษณะที่บ่งบอกถึงเอกลักษณ์ของข้อมูลเหล่านั้นได้ โดยในงานประยุกต์หลาย ๆ งานจะสกัดลักษณะสำคัญของข้อมูลที่มีความซับซ้อนออกมาในรูปแบบของข้อมูลอนุกรมเวลา ดังแสดงในรูปที่ 3.1 และรูปที่ 3.2

โดยที่รูปที่ 3.1 เป็นรูปตัวอย่างในการสกัดลักษณะสำคัญจากข้อมูลภาพถ่ายตัวหนังสือที่เป็นลายมือให้เปลี่ยนมาอยู่ในรูปของข้อมูลอนุกรมเวลา ซึ่งสามารถสกัดคุณลักษณะออกมาได้ 2 รูปแบบ ได้แก่ การสกัดลักษณะสำคัญจากโพรไฟล์ของภาพฉาย (Projection Profile) ดังแสดงในรูปที่ 3.1 ก) และการสกัดลักษณะสำคัญจากขอบบนและขอบล่างของภาพถ่ายดังแสดงในรูปที่ 3.1 ข) ในส่วนของรูปที่ 3.2 แสดงตัวอย่างการสกัดลักษณะสำคัญจากภาพถ่ายใบไม้ให้อยู่ในรูปของข้อมูลอนุกรมเวลา

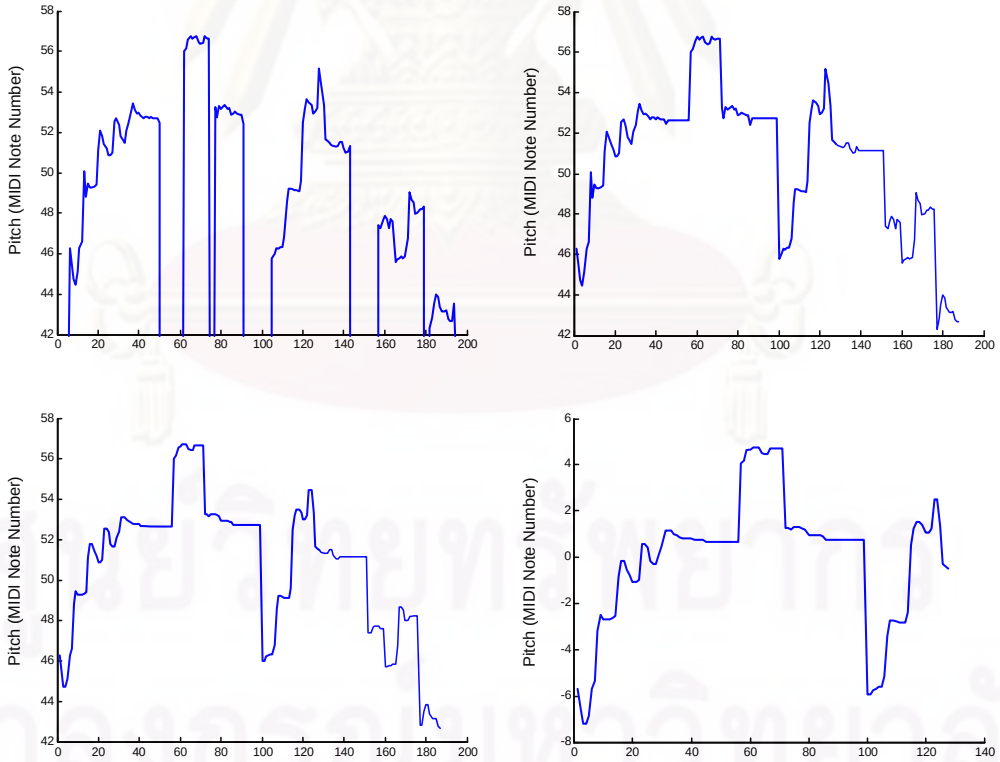


รูปที่ 3.1 ตัวอย่างการสกัดลักษณะสำคัญจากข้อมูลภาพถ่ายลายมือ ก) การสกัดลักษณะสำคัญจากภาพถ่ายลายมือด้วยโพรไฟล์ของภาพฉาย ข) การสกัดลักษณะสำคัญจากขอบบนและล่างของภาพถ่ายลายมือ (ที่มา : Ratanamahatana และ Keogh [15])



รูปที่ 3.2 ตัวอย่างการสกัดลักษณะสำคัญจากภาพถ่ายใบไม้ (ที่มา : Ratanamahatana และ Keogh [15])

นอกจากข้อมูลรูปภาพแล้ว ยังสามารถสกัดคุณลักษณะสำคัญของข้อมูลเสียงร้องทำนองได้อีกด้วย ดังแสดงในรูปที่ 3.3 เป็นการสกัดคอนทัวร์ระดับเสียงออกจากข้อมูลเสียงร้องทำนอง

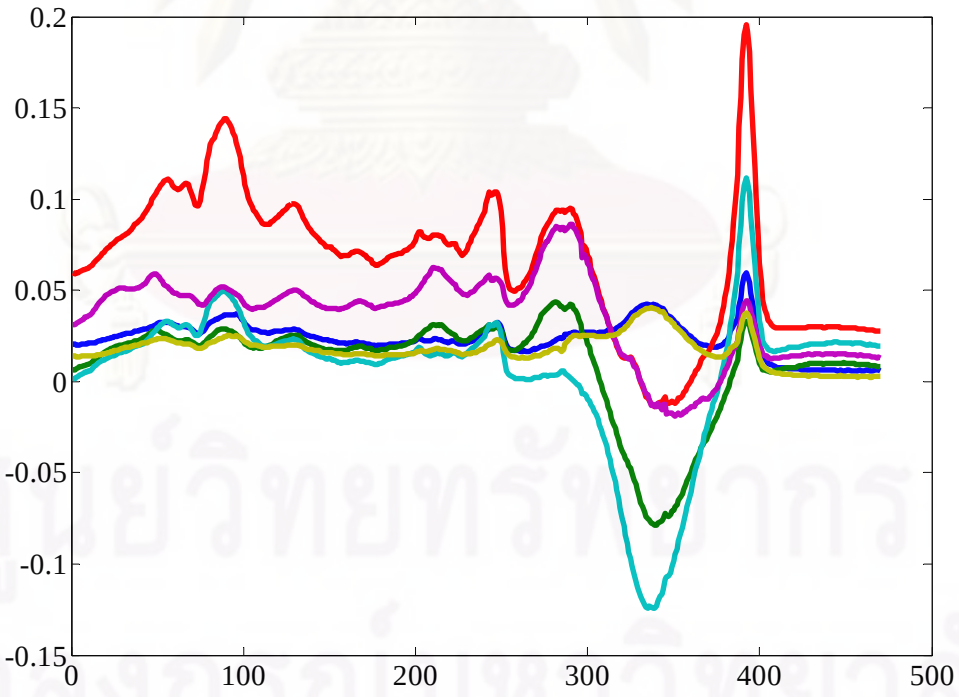


รูปที่ 3.3 คอนทัวร์ระดับเสียงที่ได้จากขั้นตอนต่าง ๆ ในการสกัดคุณลักษณะ ก) คอนทัวร์ระดับเสียงที่ได้จากขั้นตอนการสกัดคุณลักษณะออกจากเสียงร้องทำนอง ข) คอนทัวร์ระดับเสียงหลังจากผ่านขั้นตอนการเติมเต็มช่วงที่ไม่มีเสียง ค) คอนทัวร์ระดับเสียงที่ได้หลังจากผ่าน

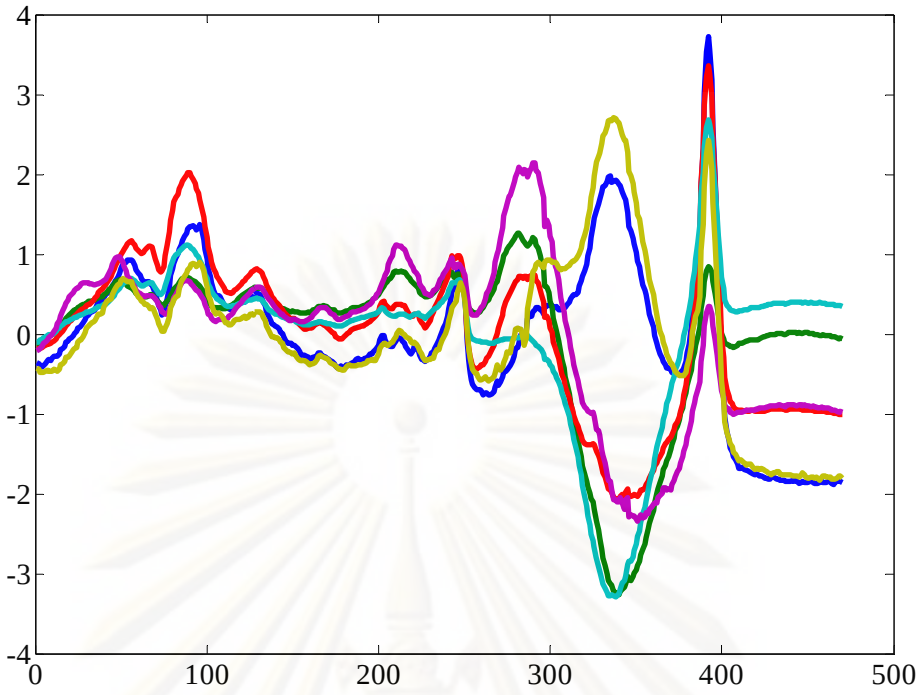
กระบวนการปรับเรียบ ง) คอนทัวร์ระดับเสียงที่ผ่านขั้นตอนการตัดขนาดและแปลงข้อมูลดังกล่าวให้เป็นบรรทัดฐาน (ที่มา : Euachongprasit และ Ratanamahatana [18])

3.2 การแปลงข้อมูลอนุกรมเวลาให้เป็นบรรทัดฐาน

การเปรียบเทียบความคล้ายคลึงระหว่างข้อมูลอนุกรมเวลาด้วยวิธีการวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง มีข้อดีกว่าวิธีการวัดระยะทางแบบยุคลิดเพราะสามารถเปรียบเทียบความคล้ายคลึงกันในเชิงรูปร่างของข้อมูลอนุกรมเวลาได้ จึงเป็นผลให้การเปรียบเทียบความคล้ายคลึงด้วยวิธีไดนามิกไทม์วอร์ปปิงให้ผลความแม่นยำมากกว่า อย่างไรก็ตามถึงแม้วิธีนี้จะสามารถเปรียบเทียบความคล้ายคลึงในเชิงรูปร่างของข้อมูลอนุกรมเวลาได้ แต่ก็ยังมีปัญหาคือ ถ้าข้อมูลอนุกรมเวลามีรูปร่างคล้ายกัน แต่มีมาตราส่วน (Scale) ที่แตกต่างกัน ดังแสดงในรูปที่ 3.4 อาจทำให้ผลลัพธ์ที่ได้ในการเปรียบเทียบเชิงรูปร่างของข้อมูลอนุกรมเวลาผิดพลาดได้ ดังนั้นก่อนที่จะทำการเปรียบเทียบความคล้ายคลึงกันระหว่างข้อมูลอนุกรมเวลาจะต้องทำให้ข้อมูลทั้งสองอนุกรมเป็นบรรทัดฐานเดียวกัน ซึ่งวิธีการที่ทำให้ข้อมูลอนุกรมเวลาเป็นบรรทัดฐานเดียวกันนั้นทำได้โดยการปรับมาตราส่วนและแอมพลิจูดของข้อมูลอนุกรมเวลาให้อยู่ในระดับเดียวกัน ในงานวิจัยทั่วไปมักใช้วิธีการแปลงข้อมูลอนุกรมเวลาให้เป็นบรรทัดฐานด้วยวิธีการใช้คะแนน Z (Z-score Normalization) ดังแสดงในรูปที่ 3.5



รูปที่ 3.4 กลุ่มข้อมูลอนุกรมเวลาที่มีมาตราส่วนที่แตกต่างกัน



รูปที่ 3.5 กลุ่มข้อมูลอนุกรมเวลา หลังจากทำให้เป็นมาตรฐานเดียวกันโดยวิธีการใช้คะแนน Z

การแปลงข้อมูลอนุกรมเวลาให้เป็นบรรทัดฐานเดียวกันด้วยวิธีการใช้คะแนน Z นั้น สามารถอธิบายรายละเอียดได้ดังนี้ กำหนดให้มีข้อมูลอนุกรมเวลา Q มีความยาว n โดยที่ $Q = q_1, q_2, \ldots, q_n$ โดยวิธีการแปลงข้อมูลอนุกรมเวลาให้เป็นบรรทัดฐานนั้นจะแทนที่จุดข้อมูลเดิมด้วยจุดข้อมูลใหม่ที่เป็นค่าคะแนน Z ของแต่ละจุดข้อมูล ซึ่ง Qz เป็นข้อมูลอนุกรมเวลาที่ได้จากการแปลงข้อมูลอนุกรมเวลา Q ให้เป็นบรรทัดฐาน โดยที่ $Qz = qz_1, qz_2, \ldots, qz_n$ และสามารถคำนวณได้ตามสมการที่ (3.1)

$$qz_i = \frac{q_i - \bar{q}}{\sigma} \tag{3.1}$$

โดยที่ $\bar{q}$ และ SD เป็นค่าเฉลี่ยเลขคณิตของทุกจุดข้อมูลของข้อมูลอนุกรมเวลา Q และส่วนเบี่ยงเบนมาตรฐานของทุกจุดข้อมูลของข้อมูลอนุกรมเวลา Q ตามลำดับ ซึ่งสามารถคำนวณได้จากสมการที่ (3.2)

$$\bar{q} = \frac{\sum_{i=1}^{n} q_i}{n} \tag{3.2}$$

$$\sigma = \sum_{i=1}^{n} \sqrt{\frac{(q_i - \bar{q})^2}{n}}$$

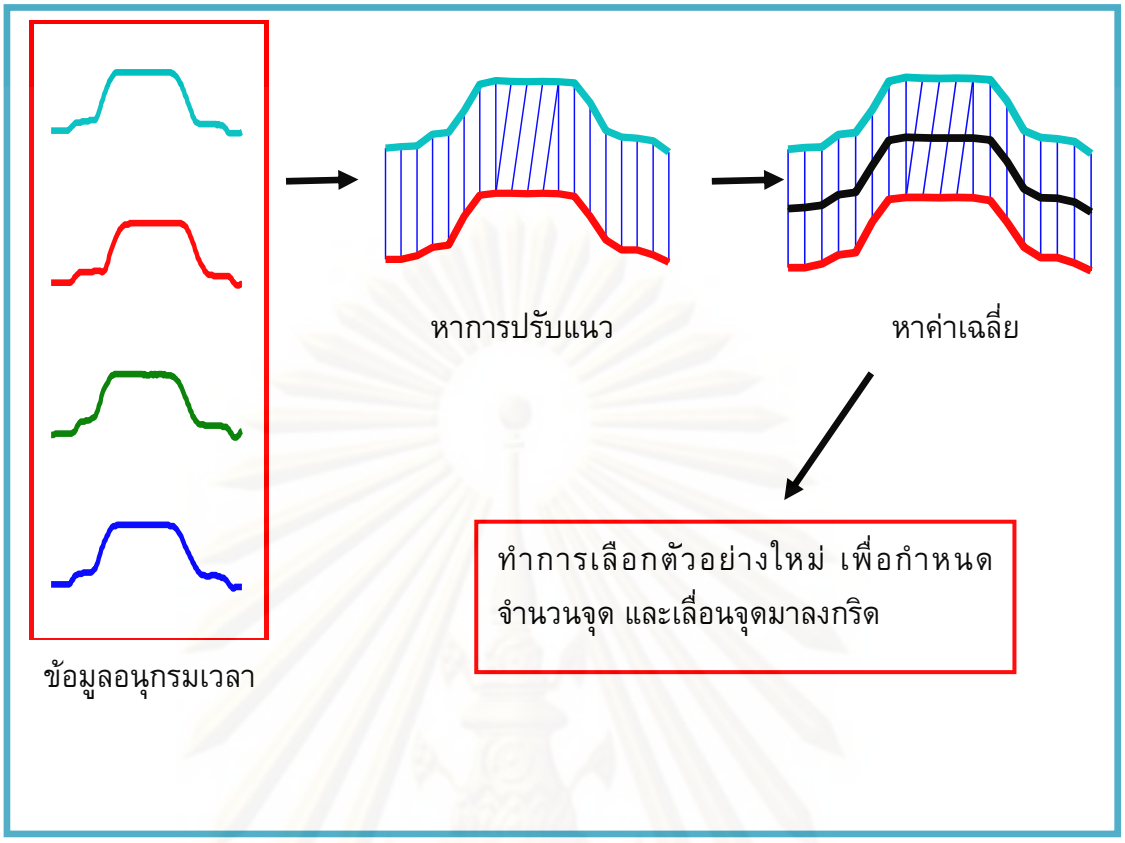
3.3 ขั้นตอนในการสร้างแผ่นแบบสำหรับกลุ่มข้อมูลอนุกรมเวลา

ในภาพรวมของงานวิจัยนี้ได้เตรียมการก่อนทำการจำแนกประเภทข้อมูลสำหรับข้อมูลอนุกรมเวลา ถ้าปริมาณข้อมูลในกลุ่มเรียนรู้มีเป็นจำนวนมาก จะส่งผลให้เกิดปัญหาในเรื่องข้อจำกัดของหน่วยเก็บข้อมูลและความเร็วในการจำแนกประเภทของข้อมูลอนุกรมเวลา งานวิจัยนี้จึงเสนอแนวทางในการลดจำนวนข้อมูลในกลุ่มข้อมูลเรียนรู้ด้วยการหาตัวแทนของกลุ่มข้อมูลเรียนรู้ โดยที่วิธีการที่จะได้มาซึ่งตัวแผ่นแบบหรือตัวแทนกลุ่มของข้อมูลอนุกรมเวลานั้นใช้หลักการหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลาโดยอาศัยการปรับแนวแบบไทม์วอร์ปปิงเรียกวิธีการนี้ว่าวิธี ASA (Accurate Shape Averaging) ซึ่งตัวแทนกลุ่มข้อมูลอนุกรมเวลาหรือแผ่นแบบที่ได้จากงานวิจัยนี้จะสามารถเป็นตัวแทนสำหรับข้อมูลอนุกรมเวลาตัวอื่น ๆ ที่อยู่ภายในกลุ่มเดียวกันได้เป็นอย่างดี โดยงานวิจัยนี้ได้มุ่งเน้นไปที่ปัญหาในการจำแนกประเภทของข้อมูลอนุกรมเวลาเป็นหลัก ดังนั้นแผ่นแบบหรือตัวแทนกลุ่มที่ได้จากงานวิจัยนี้จะต้องสามารถใช้จำแนกประเภทข้อมูลที่อยู่ในกลุ่มเดียวกันได้อย่างรวดเร็วและมีความถูกต้องแม่นยำอีกด้วย

จากนี้จะกล่าวถึงรายละเอียดของขั้นตอนวิธี ASA ที่ใช้ในการสร้างแผ่นแบบสำหรับข้อมูลอนุกรมเวลาในงานวิจัยนี้ ซึ่งประกอบไปด้วย 4 ขั้นตอน ดังนี้

1. *ขั้นที่หนึ่ง* เป็นขั้นตอนของการหาว่าข้อมูลอนุกรมเวลาคู่ใดจะทำการคำนวณค่าเฉลี่ยรูปร่างก่อน
2. *ขั้นที่สอง* เป็นขั้นตอนในการสร้าง การปรับแนว เพื่อเป็นตัวกำหนดคู่จุดข้อมูลที่จะทำการหาค่าเฉลี่ย
3. *ขั้นที่สาม* ในขั้นตอนนี้เป็นขั้นตอนที่ทำการคำนวณค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลา
4. *ขั้นที่สี่* เป็นขั้นตอนที่ทำการปรับให้แผ่นแบบที่ได้มีจุดข้อมูลทุกจุดมีระยะห่างเท่า ๆ กันหรืออยู่ในตำแหน่งกริด พร้อมทั้งกำหนดความยาวของข้อมูลอนุกรมเวลาที่เป็นแผ่นแบบให้เท่ากับความยาวของข้อมูลอนุกรมเวลาต้นแบบ ขั้นตอนนี้เรียกว่าการเลือกตัวอย่างใหม่ (Re-Sample)

โดยขั้นตอนวิธี ASA ทั้ง 4 ขั้นตอนนั้นแสดงในรูปที่ 3.6



รูปที่ 3.6 ภาพรวมของขั้นตอนวิธีในการสร้างแผ่นแบบด้วยวิธี ASA

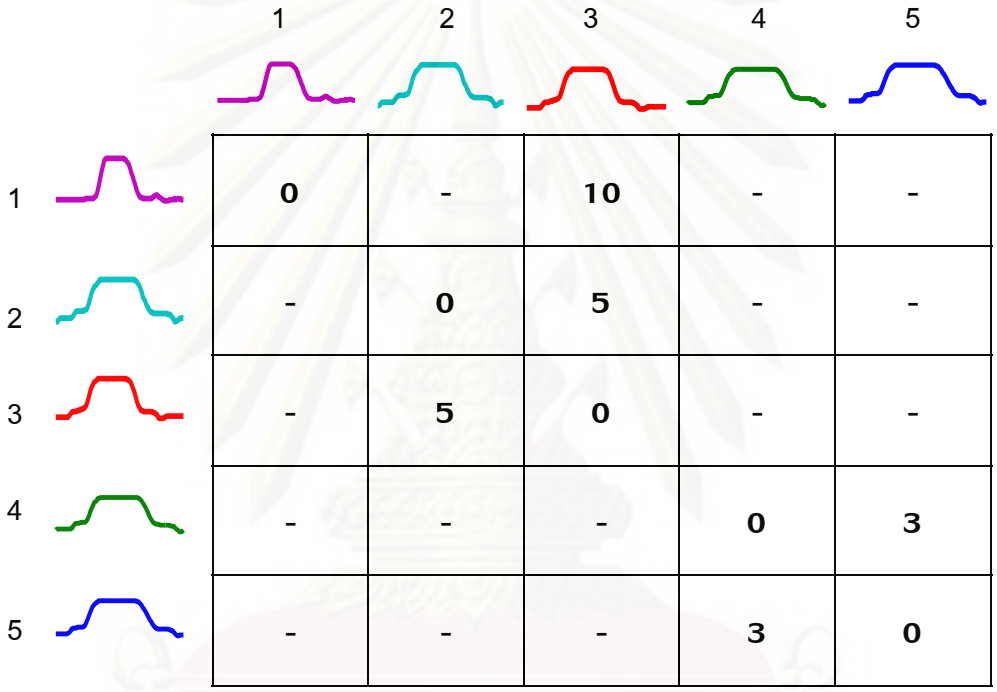
3.3.1 การจัดลำดับข้อมูลอนุกรมเวลา

งานวิจัยนี้เสนอวิธีการสร้างแผ่นแบบด้วยการคำนวณหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลา ซึ่งวิธีการนี้สามารถคำนวณค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลาได้เพียงครั้งละสองอนุกรมเท่านั้น ทำให้ต้องมีการจัดลำดับในการคำนวณค่าเฉลี่ยรูปร่างว่าข้อมูลอนุกรมเวลาคู่ใดจะถูกนำมาเฉลี่ยก่อน เพื่อให้แผ่นแบบที่ได้สามารถแทนข้อมูลอนุกรมเวลาตัวอื่น ๆ ที่อยู่ภายในกลุ่มทั้งหมดได้อย่างมีประสิทธิภาพ

งานวิจัยนี้จึงได้เสนอวิธีในการจัดลำดับในการหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลา โดยข้อมูลที่จะถูกนำมาคำนวณหาค่าเฉลี่ยรูปร่างก่อนนั้นจะต้องเป็นคู่ของข้อมูลอนุกรมเวลาที่มีความคล้ายคลึงกันมากที่สุด วิธีที่จะหาว่าข้อมูลอนุกรมเวลาคู่ใดมีความคล้ายคลึงกันมากที่สุดนั้น งานวิจัยนี้ใช้การเปรียบเทียบความคล้ายคลึงระหว่างข้อมูลอนุกรมเวลาทุกคู่ที่เป็นไปได้ที่อยู่ในกลุ่มข้อมูลเรียนรู้เดียวกัน โดยใช้วิธีวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง แต่เนื่องจากในขั้นตอนนี้ต้องการหาเฉพาะคู่ของข้อมูลอนุกรมเวลาที่มีค่าระยะทางน้อยที่สุดเท่านั้น ถ้าใช้การคำนวณแบบไดนามิกไทม์วอร์ปปิงทั้งหมด จะทำให้การทำงานในขั้นตอนนี้ช้ามากเนื่องจากต้องคำนวณระยะทางทั้งหมดถึง $(n)\times(n\text{-}1)$ ครั้ง เมื่อ n คือจำนวนข้อมูลอนุกรมเวลาที่อยู่ในกลุ่มข้อมูลเรียนรู้ ดังนั้นในงานวิจัยนี้จึงใช้ฟังก์ชันขอบเขตล่างของวิธีได

นามิกไทม์วอร์ปปิงที่ใช้การคำนวณแบบยุคลิด [9] เพื่อเป็นการลดปริมาณจำนวนข้อมูลที่ต้องทำการคำนวณระยะทางด้วยวิธีไดนามิกไทม์วอร์ปปิง

เนื่องจากในทุก ๆ รอบของการหาค่าเฉลี่ยรูปร่างของข้อมูลจะต้องหาข้อมูลอนุกรมเวลาคู่ที่มีระยะทางน้อยที่สุด ดังนั้นเพื่อให้การหาคู่ลำดับในรอบต่อ ๆ ไปไม่ต้องทำการคำนวณใหม่ทั้งหมด ในรอบแรกของการหาคู่ของข้อมูลอนุกรมเวลานั้น ข้อมูลอนุกรมเวลาแต่ละอนุกรมจะต้องบันทึกค่าของข้อมูล 2 ค่า ได้แก่ บันทึกว่าข้อมูลอนุกรมเวลาตัวใดมีระยะทางเมื่อเทียบกับตัวมันเองน้อยที่สุด และระยะทางมีค่าเท่าไร ดังแสดงในรูปที่ 3.7



รูปที่ 3.7 แผนภาพแสดงภาพรวมการเก็บระยะทางน้อยที่สุดของข้อมูลอนุกรมเวลาแต่ละอนุกรม

ในรูปที่ 3.7 เป็นรูปที่แสดงระยะทางที่น้อยที่สุดระหว่างข้อมูลอนุกรมเวลาแต่ละอนุกรมดังนี้ ข้อมูลอนุกรมเวลาตัวที่ 1 มีระยะทางน้อยที่สุดเมื่อเทียบกับข้อมูลอนุกรมเวลาตัวที่ 3 และมีระยะทางเท่ากับ 10 ข้อมูลอนุกรมเวลาตัวที่ 2 มีระยะทางน้อยที่สุดเมื่อเทียบกับข้อมูลอนุกรมเวลาตัวที่ 3 และมีระยะทางเท่ากับ 5 ข้อมูลอนุกรมเวลาตัวที่ 3 มีระยะทางน้อยที่สุดเมื่อเทียบกับข้อมูลอนุกรมเวลาตัวที่ 2 และมีระยะทางเท่ากับ 5 ข้อมูลอนุกรมเวลาตัวที่ 4 มีระยะทางน้อยที่สุดเมื่อเทียบกับข้อมูลอนุกรมเวลาตัวที่ 5 และมีระยะทางเท่ากับ 3 ข้อมูลอนุกรมเวลาตัวที่ 5 มีระยะทางน้อยที่สุดเมื่อเทียบกับข้อมูลอนุกรมเวลาตัวที่ 4 และมีระยะทางเท่ากับ 3 นอกจากนี้เลข 0 หมายถึงระยะทางระหว่างข้อมูลอนุกรมเวลาตัวนั้นๆ เทียบกับตัวเองจะเท่ากับ

0 เนื่องจากเป็นข้อมูลอนุกรมเวลาตัวเดียวกัน และเครื่องหมายขีด (-) แสดงถึงระยะทางอื่น ๆ ที่ไม่ได้มีค่าน้อยที่สุดเมื่อเทียบกับข้อมูลอนุกรมเวลาแต่ละตัว

จากรูปที่ 3.7 ข้อมูลอนุกรมเวลาคู่ที่มีระยะทางน้อยที่สุดคือข้อมูลอนุกรมเวลาตัวที่ 4 และตัวที่ 5 ดังนั้นข้อมูลอนุกรมเวลาคู่นี้จะถูกคำนวณหาค่าเฉลี่ยรูปร่างเป็นคู่แรก จากนั้นอนุกรมเวลาที่เกิดจากการหาค่าเฉลี่ยนั้นจะถูกนำมาแทนที่ข้อมูลอนุกรมเวลาตัวที่ 4 และ 5 ในรอบต่อมาจึงไม่จำเป็นต้องคำนวณหาระยะทางน้อยที่สุดใหม่ทั้งหมด เพียงแค่คำนวณว่าข้อมูลอนุกรมเวลาตัวใหม่นี้มีระยะทางน้อยที่สุดเมื่อเทียบกับข้อมูลอนุกรมเวลาตัวใดและมีระยะทางเท่าไร ก็จะทำให้ทราบว่าในรอบนั้นข้อมุลอนุกรมเวลาคู่ใดที่มีระยะทางน้อยที่สุด โดยในรอบต่อ ๆ มานั้นไม่จำเป็นว่าข้อมูลอนุกรมเวลาที่เป็นค่าเฉลี่ยของรอบก่อนหน้าจะต้องเป็นอนุกรมที่จะต้องนำมาคำนวณหาค่าเฉลี่ยในรอบต่อไปก็ได้

3.3.1.1 การหาลำดับในการหาค่าเฉลี่ยในแต่ละกลุ่มข้อมูล

ในหัวข้อนี้จะอธิบายขั้นตอนวิธีในการหาคู่ของข้อมูลอนุกรมเวลาที่จะนำมาหาค่าเฉลี่ยรูปร่างโดยละเอียด ในส่วนแรกจะอธิบายการทำงานของฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปปิงภายใต้การกำหนดเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะ และในส่วนสุดท้ายจะอธิบายรายละเอียดของฟังก์ชันสำหรับหาคู่ของอนุกรมเวลาที่มีระยะทางน้อยที่สุด

Algorithm 1 : *Best_so_far* = LB_Keogh(*C*, *Q*, *r*)

```
 1:   Best_so_far <= ∞
 2:   Foreach c_i in C do
 3:     d <= lowerbound(c_i, Q, r)
 4:     If d < Best_so_far then
 5:        New_dist <= dtw(c_i, Q)
 6:        If New_dist < Best_so_far then
 7:           Best_so_far <= New_dist
 8:        EndIf
 9:     EndIf
10:   EndFor
```

รูปที่ 3.8 รหัสเทียมของฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปปิงภายใต้การกำหนดเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะ ในการเปรียบเทียบความคล้ายคลึงของข้อมูลอนุกรมเวลา

จากที่ได้อธิบายในหัวข้อที่ 2.4.1 ถึงประโยชน์ของการใช้ฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปปิงว่าสามารถลดการคำนวณไดนามิกไทม์วอร์ปปิงได้ ในรูปที่ 3.8 แสดงรหัสเทียมของฟังก์ชัน LB_Keogh(*C*, *Q*, *r*) ซึ่งอธิบายการทำงานของฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปปิงภายใต้การกำหนดเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะ โดยกำหนดให้

มีพารามิเตอร์ตัวแรก C แทนข้อมูลที่ใช้ในการเปรียบเทียบ พารามิเตอร์ตัวที่สอง Q แทนข้อมูลสอบถาม และให้พารามิเตอร์ตัวสุดท้าย r แทนความกว้างของเงื่อนไขบังคับโดยรวม เริ่มต้นบรรทัดที่ 1 กำหนดให้ค่า *best_so_far* มีค่าเป็นอนันต์ ในบรรทัดที่ 2-7 ทำการคำนวณค่า lowerbound(c_i, Q, r) ทำได้โดยหาค่าขอบเขตบนและขอบเขตล่างของข้อมูลอนุกรมเวลา Q ตามสมการที่ (2.4) และนำข้อมูลอนุกรมเวลา C มาคำนวณระยะทางแบบยุคลิด d ตามสมการที่ (2.5) ถ้าระยะทาง d มีค่าน้อยกว่าค่า *best_so_far* จะทำการคำนวณระยะทางด้วยวิธีไดนามิกไทม์วอร์ปปิง *New_dist*(c_i, Q) ระหว่างข้อมูลอนุกรมเวลา Q และ c_i ถ้าค่า *New_dist* ที่ได้มีค่าน้อยกว่า *best_so_far* จะทำการปรับค่า *best_so_far* ให้มีค่าเท่ากับค่า *New_dist* แล้วทำการคำนวณจนกระทั่งทำการเปรียบเทียบระหว่าง Q กับ c_i ครบทุกตัว

ในส่วนต่อมาจะอธิบายถึงการหาคู่ข้อมูลอนุกรมเวลาที่มีระยะทางน้อยที่สุดในกลุ่มข้อมูลเรียนรู้ ดังแสดงในรูปที่ 3.9

Algorithm 2 : (*a*, *b*) = *Min_Distance_Pair*(*S*)

```
1:   Foreach s_i in S do
2:     Q <= s_i
3:     C <= S - s_i
4:     (Dist_i, min_c_i) <= LB_Keogh(C, Q, r)
5:   EndFor
6:   New_S <= Sort S with respect to Dist
7:   min_c <= Sort min_c with respect to Dist
8:   a <= New_S(1)
9:   b <= min_c(1)
```

รูปที่ 3.9 รหัสเทียมสำหรับการหาคู่ของอนุกรมเวลาที่มีระยะทางน้อยที่สุด

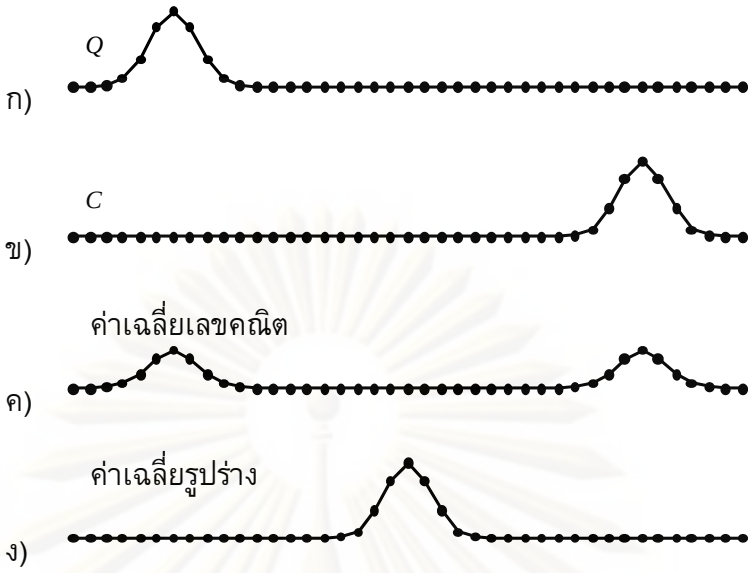
ในส่วนการคำนวณระยะทางระหว่างข้อมูลอนุกรมเวลาทุกอนุกรมที่อยู่ในกลุ่มข้อมูลเรียนรู้ดังที่ได้อธิบายในหัวข้อที่ 3.3.1 นั้น งานวิจัยนี้ได้เสนอวิธีการหาค่าระยะทางของคู่อนุกรมเวลาน้อยที่สุดโดยการใช้ฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปปิงภายใต้การกำหนดเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะ ในรูปที่ 3.9 แสดงรหัสเทียมของฟังก์ชัน *Min_Distance_Pair(S)* ซึ่งอธิบายการทำงานของฟังก์ชันการหาคู่ข้อมูลอนุกรมเวลาคู่ที่มีระยะทางน้อยที่สุด โดยกำหนดให้มีพารามิเตอร์ S แทนกลุ่มข้อมูลเรียนรู้สำหรับการสร้างแผ่นแบบ

จากรหัสเทียมในรูปที่ 3.9 บรรทัดที่ 1-5 แสดงวิธีการคำนวณระยะทางระหว่างข้อมูลอนุกรมเวลาแต่ละอนุกรมเปรียบเทียบกับข้อมูลอนุกรมเวลาที่เหลือที่อยู่ในกลุ่มข้อมูล

เรียนรู้ เริ่มจากนำข้อมูลอนุกรมเวลาตัวแรกในกลุ่มข้อมูลเรียนรู้ Q มาสร้างค่าขอบเขตบน และขอบเขตล่างของแต่ละจุดข้อมูล ดังที่อธิบายรายละเอียดในหัวข้อที่ 2.4.1 แล้วนำข้อมูลอนุกรมเวลา C ที่เหลือทุกตัวมาเปรียบเทียบความคล้ายคลึงโดยการเรียกใช้ฟังก์ชัน LB_Keogh ดังแสดงรายละเอียดในรูปที่ 3.8 เพื่อลดจำนวนครั้งในการคำนวณระยะทางด้วยวิธีไดนามิกไทม์วอร์ปปิง เมื่อทำการวัดระยะทางระหว่างข้อมูลอนุกรมเวลา Q กับข้อมูลอนุกรมเวลา C จนครบแล้ว จะคืนค่าระยะทางที่น้อยที่สุด $Dist_i$ และข้อมูลอนุกรมเวลาที่ให้ค่าระยะทางน้อยที่สุด min_c_i เมื่อเทียบกับ Q จากนั้นในบรรทัดที่ 6 จะได้กลุ่มข้อมูลเรียนรู้กลุ่มใหม่ New_S ที่เรียงลำดับของข้อมูลอนุกรมเวลาตามค่าระยะทางที่คำนวณได้ และในบรรทัดที่ 7 ทำการเรียงลำดับข้อมูลอนุกรมเวลาที่มีระยะทางน้อยที่สุด min_c สุดท้ายบรรทัดที่ 8 และ 9 จะเรียกคืนค่าของคู่อนุกรมเวลาที่มีระยะทางน้อยที่สุด a, b

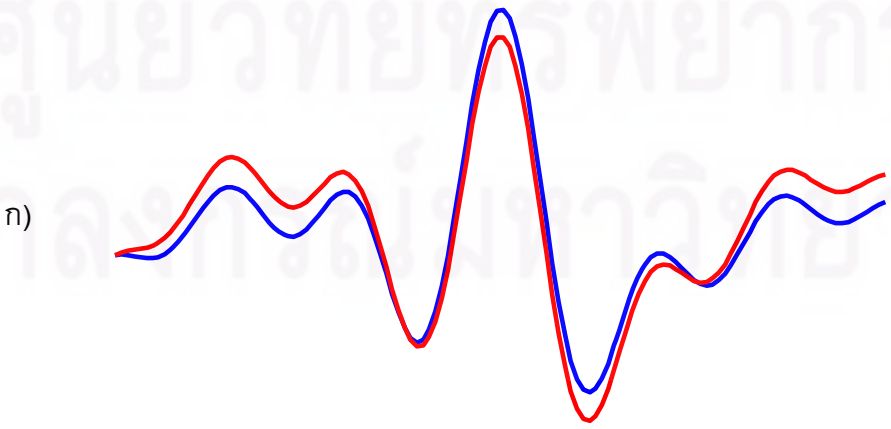
3.3.2 การปรับแนวแบบไทม์วอร์ปปิง (Time-Warping Alignment)

งานวิจัยนี้ได้นำเสนอวิธีการคำนวณค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลาโดยใช้การปรับแนวแบบไทม์วอร์ปปิงเพื่อเป็นตัวกำหนดคู่ของจุดข้อมูลที่จะนำเฉลี่ยกัน เนื่องจากการปรับแนวแบบไทม์วอร์ปปิงได้มาจากการคำนวณหาระยะทางสะสมที่น้อยที่สุดในการเปรียบเทียบความคล้ายคลึงระหว่างข้อมูลอนุกรมเวลาด้วยวิธีการวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง ซึ่งเป็นวิธีที่สามารถเปรียบเทียบข้อมูลในเชิงรูปร่างได้ ดังนั้นจะทำให้ค่าระยะทางระหว่างคู่ที่นำมาเฉลี่ยก็จะมีค่าน้อยและข้อมูลอนุกรมเวลาที่เกิดจาการเฉลี่ยรูปร่างก็จะยังคงลักษณะรูปร่างคล้ายกับข้อมูลอนุกรมเวลาต้นแบบ ดังแสดงในรูปที่ 3.10 การคำนวณค่าเฉลี่ยแบบเลขคณิตระหว่างข้อมูลอนุกรมเวลาสองอนุกรม ดังแสดงในรูปที่ 3.10 ค) รูปร่างของข้อมูลอนุกรมเวลาที่ได้หลังการหาค่าเฉลี่ยนั้นมีรูปร่างที่ไม่เหมือนกับข้อมูลอนุกรมเวลาต้นแบบ Q และ C เพราะข้อมูลอนุกรมเวลา Q และ C มีส่วนสูงสุด (Peak) เพียงตำแหน่งเดียว แต่ข้อมูลอนุกรมเวลาที่ได้หลังจากหาค่าเฉลี่ยเลขคณิตนั้นมีส่วนสูงสุดสองตำแหน่งซึ่งเป็นการไม่สมเหตุสมผล ส่วนข้อมูลอนุกรมเวลาที่ได้จากการหาค่าเฉลี่ยรูปร่างนั้นยังคงมีลักษณะเหมือนข้อมูลอนุกรมเวลาที่เป็นต้นแบบ ดังแสดงในรูปที่ 3.10 ง)

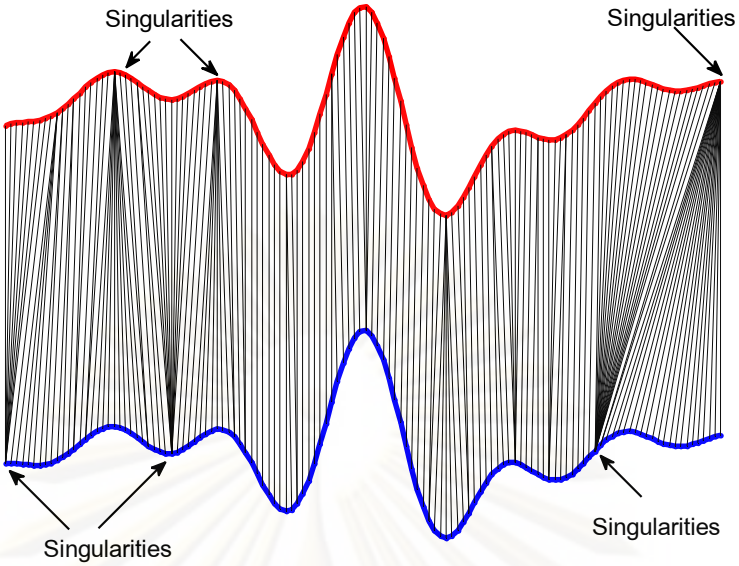


รูปที่ 3.10 ค่าเฉลี่ยของข้อมูลอนุกรมเวลาระหว่างอนุกรมเวลา Q ก) และ C ข) โดยใช้ค่าเฉลี่ยเลขคณิต ค) และค่าเฉลี่ยรูปร่าง ง)

อย่างไรก็ตาม การปรับแนวที่ได้จาการคำนวณระยะทางด้วยวิธีไดนามิกไทม์วอร์ปปิงก็ยังมีจุดอ่อน คือวิธีดังกล่าวต้องการคำนวณให้ค่าระยะทางสะสมที่ได้มีค่าน้อยที่สุด ทำให้ไดนามิกไทม์วอร์ปปิงเลือกการปรับแนวบางคู่จุดไม่เหมาะสม และถ้าข้อมูลอนุกรมเวลามีการเปลี่ยนแปลงเฉพาะในแกน Y เท่านั้นการปรับแนวแบบไทม์วอร์ปปิงก็จะทำให้เกิดภาวะเอกฐาน (Singularities) ดังแสดงในรูปที่ 3.11 จึงได้เสนอการปรับแนวที่ได้จาการคำนวณระยะทางด้วยวิธีไดนามิกไทม์วอร์ปปิงแบบอนุพันธ์ที่แก้ปัญหาการปรับแนวที่ไม่เหมาะสมของวิธีไดนามิกไทม์วอร์ปปิงดังที่ได้อธิบายในหัวข้อที่ 2.5 แต่วิธีการนี้ก็ยังให้การปรับแนวที่ไม่เหมาะสมในบางกรณี เช่น เมื่อพิจารณาจุดข้อมูล 2 จุด คือ q_i และ c_j ซึ่งมีค่าไม่เท่ากัน แต่ q_i อยู่ในส่วนที่มีแนวโน้มกำลังจะเพิ่มและ c_j อยู่ในส่วนที่มีแนวโน้มกำลังเพิ่มเช่นเดียวกัน วิธีไดนามิกไทม์วอร์ปปิงแบบอนุพันธ์จะพิจารณาการจับคู่ทั้ง 2 จุดนี้โดยไม่คำนึงถึงค่าของข้อมูลที่แตกต่างกัน จะทำให้การปรับแนวแบบไทม์วอร์ปปิงแบบอนุพันธ์จับคู่จุดข้อมูลที่ไม่ถูกต้องได้



ข)



รูปที่ 3.11 ข้อมูลอนุกรมเวลาสองอนุกรม ก) การปรับแนวแบบไทม์วอร์ปปิงทำให้เกิดภาวะเอกฐาน ข)

ในรูปที่ 3.11 ข) การปรับแนวแบบไทม์วอร์ปปิงทำให้เกิดภาวะเอกฐาน ซึ่งภาวะเอกฐานนี้หมายถึงจุดข้อมูลหนึ่งจุดบนข้อมูลอนุกรมเวลาหนึ่ง จับกับอีกหลาย ๆ จุดของอนุกรมหนึ่ง

งานวิจัยนี้จึงเสนอวิธีการปรับแนวแบบใหม่ ที่นำทั้งค่าของจุดข้อมูลและค่าประมาณอนุพันธ์ของจุดข้อมูลมาคำนวณเรียกว่า การปรับแนวแบบผสมระหว่างไดนามิกไทม์วอร์ปปิงกับไดนามิกไทม์วอร์ปปิงแบบอนุพันธ์

3.3.2.1 การปรับแนวแบบผสมระหว่างไดนามิกไทม์วอร์ปปิงกับไดนามิกไทม์วอร์ปปิงแบบอนุพันธ์ (Hybrid DTW-DDTW)

การปรับแนวแบบผสมระหว่างไดนามิกไทม์วอร์ปปิงกับไดนามิกไทม์วอร์ปปิงแบบอนุพันธ์ ในการคำนวณระยะทางระหว่างข้อมูลอนุกรมเวลาสองอนุกรมจะใช้ทั้งค่าของจุดข้อมูลและค่าประมาณอนุพันธ์ในแต่ละจุดข้อมูล ในส่วนของการคำนวณอนุพันธ์ของจุดข้อมูลนั้นงานวิจัยนี้ใช้วิธีที่เรียกว่า Five-point Stencil ในการหาค่าประมาณ ซึ่งเป็นการนำเพื่อนบ้าน 4 จุดข้อมูล คือก่อนหน้า 2 จุดและตามหลังอีก 2 จุด เพื่อมาคำนวณหาค่าประมาณของอนุพันธ์

การปรับแนวแบบผสมระหว่างไดนามิกไทม์วอร์ปปิงกับไดนามิกไทม์วอร์ปปิงแบบอนุพันธ์ สามารถอธิบายโดยละเอียดได้ดังนี้ กำหนดให้มีข้อมูลอนุกรมเวลา 2 อนุกรม ได้แก่อนุกรมเวลา Q และอนุกรมเวลา C โดยมีความยาว m และ n ตามลำดับ โดยที่ $Q = q_1, q_2, \ldots, q_m$ และ $C = c_1, c_2, \ldots, c_n$ เมื่อ $q_1, q_2, \ldots, q_m$ และ $c_1, c_2, \ldots, c_n$ คือค่าของข้อมูลในแต่

ละจุดข้อมูล จากข้อมูลอนุกรมเวลา Q และ C สามารถหาค่าประมาณอนุพันธ์ของข้อมูลอนุกรมเวลา $Q' = q_1', q_2', \ldots, q_m'$ และอนุกรมเวลา $C' = c_1', c_2', \ldots, c_n'$ ได้ตามสมการที่ (3.3)

$$q_i' \approx \frac{-q_{(i+2)} + 8q_{(i+1)} - 8q_{(i-1)} + q_{(i-2)}}{12}$$
$$c_i' \approx \frac{-c_{(i+2)} + 8c_{(i+1)} - 8c_{(i-1)} + c_{(i-2)}}{12} \quad (3.3)$$

โดยที่ $q_{(i+2)}$, $q_{(i+1)}$, $q_{(i-1)}$ และ $q_{(i-2)}$ เป็นเพื่อนบ้านของจุดข้อมูล $q_{(i)}$ ส่วน $c_{(i+2)}$, $c_{(i+1)}$, $c_{(i-1)}$ และ $c_{(i-2)}$ เป็นเพื่อนบ้านของจุดข้อมูล $c_{(i)}$ เมื่อได้ค่าประมาณอนุพันธ์ของแต่ละจุดข้อมูลแล้ว ก็สามารถนำมาคำนวณเมตริกซ์ระยะทางระหว่างอนุกรมทั้งสอง ($D = \{d\}_{m\times n}$) ที่มีขนาดเท่ากับ $m\times n$ โดยที่ระยะทางของค่าจุดข้อมูลและอนุพันธ์ของจุดข้อมูลจะต้องมีการคำนวณค่าใหม่โดยใช้ค่าคะแนน Z เพื่อให้ทั้งค่าจุดข้อมูลและอนุพันธ์นั้นมีบรรทัดฐานเดียวกันตามสมการที่ (3.4) ดังนี้

$$d_{i,j} = \frac{d_{i,j}^{(0)} - \mu_0}{\sigma_0} + \frac{d_{i,j}^{(1)} - \mu_1}{\sigma_1} \quad (3.4)$$

โดยที่ $d^{(0)}$ และ $d^{(1)}$ คือระยะทางที่คำนวณจากค่าของจุดข้อมูลและค่าประมาณอนุพันธ์อันดับหนึ่งของจุดข้อมูล ตามลำดับ เมื่อ μ_0, μ_1, σ_0 และ σ_1 คือค่าเฉลี่ยและส่วนเบี่ยงเบนมาตรฐานของระยะทาง $d^{(0)}$ และ $d^{(1)}$ ตามลำดับ ในการหาวิถีการวอร์ปนั้นก็เลือกจากค่าระยะทางสะสมน้อยที่สุดแช่นเดียวกับวิธีไดนามิกไทม์วอร์ปปิง

3.3.3 ค่าเฉลี่ยรูปร่าง (Shape Averaging)

งานวิจัยนี้ใช้การหาค่าเฉลี่ยรูปร่างโดยอาศัยการปรับแนวแบบผสมระหว่างไดนามิกไทม์วอร์ปปิงกับไดนามิกไทม์วอร์ปปิงแบบอนุพันธ์เป็นตัวระบุว่าจะนำจุดข้อมูลใดมาเฉลี่ยกันตามสมการที่ (2.7) คือนำค่าในแกน X และแกน Y มาเฉลี่ยกันตามการปรับแนวที่ได้ โดยในส่วนนี้จะต้องมีการนำค่าถ่วงน้ำหนัก (Weight) มาคำนวณด้วยเนื่องจากในแต่ละรอบของการหาค่าเฉลี่ยรูปร่างนั้น ค่าเฉลี่ยที่ได้จะมีลักษณะคล้ายกับตัวต้นแบบทั้งสอง ถ้าไม่มีการนำค่าถ่วงน้ำหนักมาคำนวณรูปร่างของแผ่นแบบที่ได้จะมีรูปร่างคล้ายกับข้อมูลอนุกรมเวลาตัวหลัง ๆ ที่ถูกนำมาคำนวณซึ่งดูไม่สมเหตุสมผล เพราะถ้าตัวใดถูกนำมาคำนวณหาค่าเฉลี่ยบ่อย ๆ รูปร่างท้ายสุดควรจะใกล้เคียงกับอนุกรมเวลาตัวนั้น

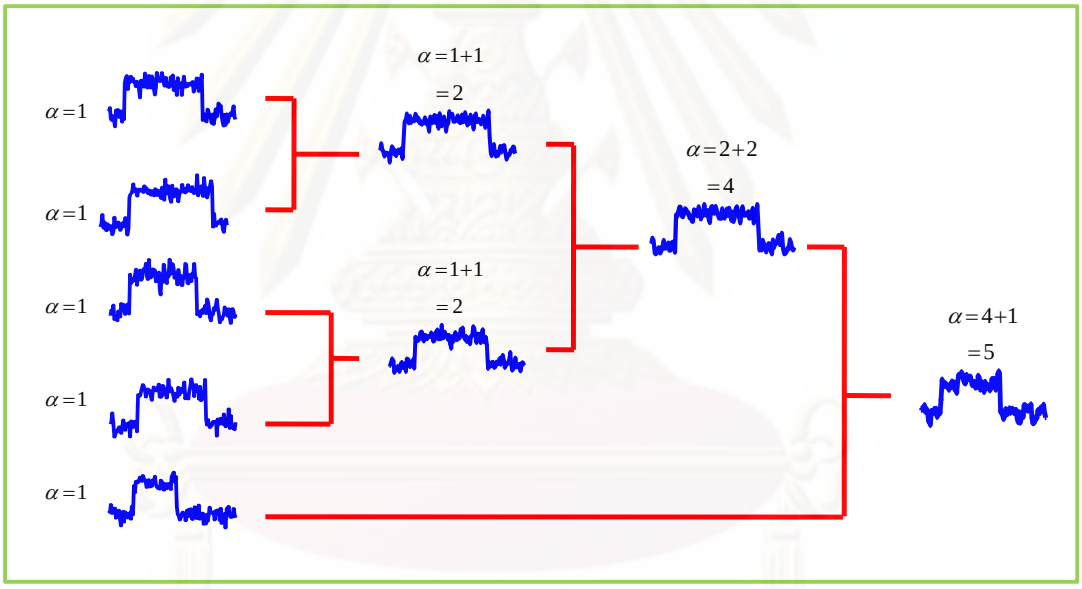
ในการหาค่าเฉลี่ยรูปร่างด้วยวิธี ASA สามารถอธิบายรายละเอียดได้ดังนี้ กำหนดให้กลุ่มข้อมูลเรียนรู้ S ซึ่งประกอบไปด้วยข้อมูลอนุกรมเวลา n อนุกรม ดังนี้ $S = S_1, S_2,$

…, S_n เมื่อต้องนำข้อมูลอนุกรมเวลาสองอนุกรมที่อยู่ในกลุ่มข้อมูลเรียนรู้มาหาค่าเฉลี่ยรูปร่าง $A = A_1, A_2, \ldots, A_k$ จะสามารถคำนวณได้จากสมการที่ (3.5)

$$A_k = \left(\frac{\alpha_x \cdot w_{k,1} + \alpha_y \cdot w_{k,2}}{\alpha_x + \alpha_y}, \frac{\alpha_x \cdot S_x(w_{k,1}) + \alpha_y \cdot S_y(w_{k,2})}{\alpha_x + \alpha_y} \right) \quad (3.5)$$

โดยที่ α_x และ α_y เป็นค่าถ่วงน้ำหนักของข้อมูลอนุกรมเวลา S_x และ S_y ตามลำดับ เมื่อ $w_{k,1}$ และ $w_{k,2}$ เป็นดรรชนีบอกตำแหน่งที่ k ของการปรับแนวระหว่างคู่อนุกรม

สำหรับค่าถ่วงน้ำหนักในการคำนวณค่าเฉลี่ยรูปร่างนั้น งานวิจัยนี้ได้กำหนดให้ข้อมูลอนุกรมเวลาในตอนเริ่มต้นทุกอนุกรมมีค่าถ่วงน้ำหนักเท่ากันโดยมีค่าเท่ากับหนึ่ง เมื่อมีการคำนวณค่าเฉลี่ยรูปร่างในแต่ละรอบ ข้อมูลอนุกรมเวลาที่เป็นค่าเฉลี่ยจะมีค่าถ่วงน้ำหนักเท่ากับค่าถ่งวน้ำหนักของต้นแบบทั้งสองบวกกัน ซึ่งการคำนวณค่าถ่วงน้ำหนักแสดงในรูปที่ 3.12



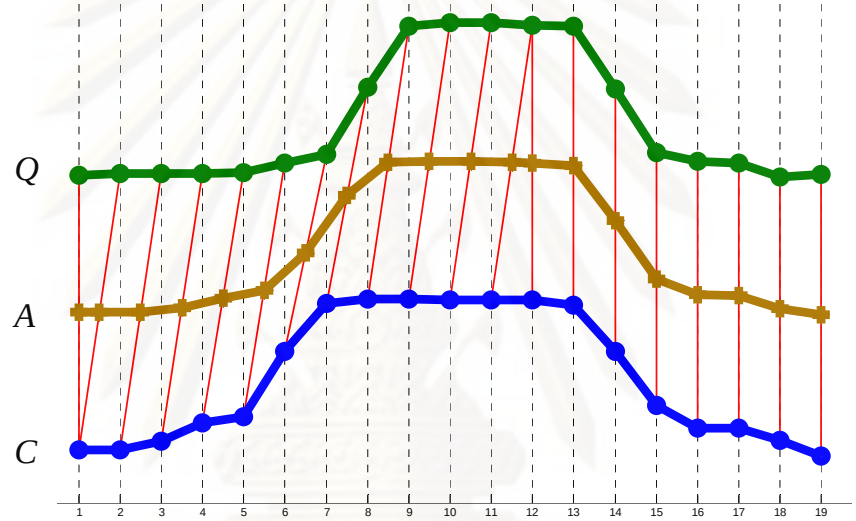
รูปที่ 3.12 การคำนวณค่าถ่วงน้ำหนัก

3.3.4 การเลือกตัวอย่างใหม่ (Re-sampling)

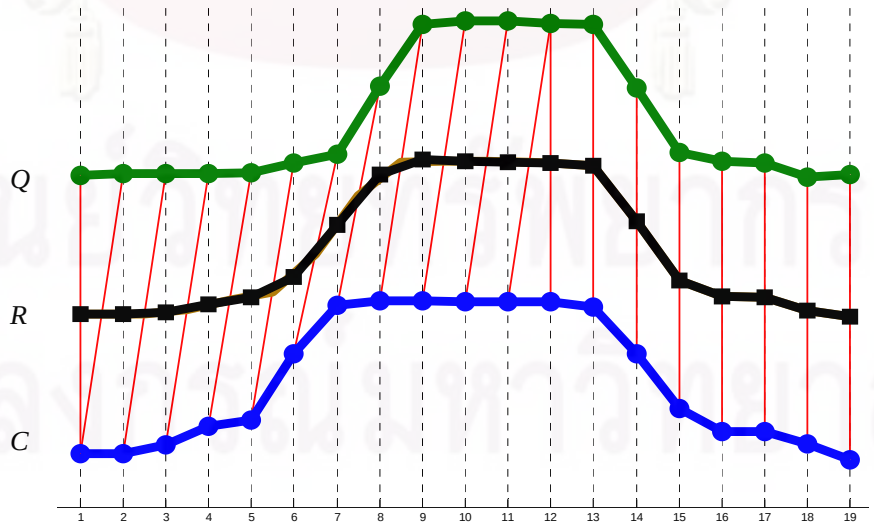
เนื่องจากการหาค่าเฉลี่ยรูปร่างที่อาศัยการปรับแนวแบบผสมระหว่างไดนามิกไทม์วอร์ปปิงกับไดนามิกไทม์วอร์ปปิงแบบอนุพันธ์มีการปรับแนวไม่เป็นแบบหนึ่งต่อหนึ่ง ทำให้หลังจากที่ทำการหาค่าเฉลี่ยรูปร่างแล้วข้อมูลอนุกรมเวลาที่เกิดจากการเฉลี่ยรูปร่างนั้นมีจำนวนจุดข้อมูลที่มากกว่าข้อมูลอนุกรมเวลาต้นแบบ จากรูปที่ 3.13 แสดงข้อมูลอนุกรมเวลาต้นแบบ Q และ C มีจำนวนจุดข้อมูล 19 จุดข้อมูล เมื่อหาค่าเฉลี่ยรูปร่างโดยอาศัยการปรับแนวที่ไม่เป็นแบบหนึ่งต่อหนึ่งทำให้ข้อมูลอนุกรมเวลา A มีจำนวนจุดข้อมูลเพิ่มขึ้นเป็น 20 จุดข้อมูล

นอกจากนั้นยังมีจำนวนจุดข้อมูลถึง 11 จุดข้อมูลได้แก่จุดข้อมูลที่ 2 3 4 5 6 7 8 9 10 11 และ 12 ที่มีค่าในแกน X ไม่อยู่ในตำแหน่งจำนวนเต็มหรือลงไม่ลงกริด

งานวิจัยนี้จึงเสนอวิธีการที่จะทำให้จำนวนจุดข้อมูลหลังการหาค่าเฉลี่ยมีจำนวนเท่ากับข้อมูลอนุกรมเวลาต้นแบบและทุกจุดข้อมูลยังมีค่าในแกน X เป็นจำนวนเต็มหรือลงกริด ดังแสดงในรูปที่ 3.14 โดยใช้การประมาณค่าด้วยฟังก์ชันกระดูกงูกำลังสามซึ่งเป็นการประมาณค่าในช่วงดังที่ได้อธิบายรายละเอียดในหัวข้อที่ 2.7 มาแก้ปัญหาดังกล่าว จากรูปที่ 3.14 ข้อมูลอนุกรมเวลา R หลังจากผ่านการเลือกตัวอย่างใหม่โดยใช้ฟังก์ชันกระดูกงูกำลังสามจะมีจำนวนจุดข้อมูล 19 จุดเท่ากับข้อมูลอนุกรมเวลาต้นแบบ Q และ C นอกจากนั้นทุก ๆ จุดข้อมูลยังมีค่าในแกน X เป็นจำนวนเต็มทุกจุดอีกด้วย



รูปที่ 3.13 ค่าเฉลี่ยรูปร่าง A ระหว่างข้อมูลอนุกรมเวลา Q และ C



รูปที่ 3.14 ข้อมูลอนุกรมเวลา R หลังจากผ่านการเลือกตัวอย่างใหม่

3.3.5 ขั้นตอนวิธี ASA (Accurate Shape Averaging หรือ ASA)

งานวิจัยนี้เสนอวิธีการสร้างแผ่นแบบสำหรับข้อมูลอนุกรมเวลา โดยใช้วิธี ASA ซึ่งขั้นตอนการสร้างแผ่นแบบในแต่ละขั้นตอนนั้นได้กล่าวไว้ในหัวข้อที่ 3.3 ในส่วนนี้จะอธิบายถึงการทำงานโดยรวมของวิธี ASA เมื่อต้องการสร้างแผ่นแบบสำหรับกลุ่มข้อมูลอนุกรมเวลา ดังแสดงในรูปที่ 3.15

Algorithm 3 : *TEMPLATE = ASA(S)*

```
 1:   While count(S) > 1 do
 2:     (a, b) = Min_Dist_Pair(S)
 3:     (dist_a,b, pth_a,b) = hybrid_dtw_ddtw(s_a, s_b);
 4:     x_1 <= {1, 2, …, size(s_a)}
 5:     x_2 <= {1, 2, …, size(s_b)}
 6:     New_x <= (W_a*x_1(pth_a,b(1)) + W_b*x_2(pth_a,b(2)))/(W_a + W_b)
 7:     New_y <= (W_a*s_a(pth_a,b(1)) + W_b*s_b(pth_a,b(2)))/(W_a + W_b)
 8:     Resampled_S <= Cubic_Spline(New_x, New_y, x_1)
 9:     S_a <= Resampled_S
10:     W_a <= W_a + W_b
11:     Remove s_b from S
12:     Remove W_b from W
13:   EndWhile
14:   TEMPLATE <= S
```

รูปที่ 3.15 รหัสเทียมของฟังก์ชันในการสร้างแผ่นแบบด้วยวิธี ASA

ในรูปที่ 3.15 แสดงรหัสเทียมของฟังก์ชัน ASA ที่ใช้ในการสร้างแผ่นแบบสำหรับข้อมูลอนุกรมเวลา โดยสามารถอธิบายรายละเอียดของฟังก์ชัน $ASA(S)$ ได้ดังนี้ กำหนดพารามิเตอร์ S เป็นกลุ่มข้อมูลอนุกรมเวลาที่จะนำมาสร้างแผ่นแบบ ในบรรทัดที่ 1-2 ทำการเรียกใช้ฟังก์ชัน $Min\_Dist\_Pair(S)$ เพื่อที่จะหาคู่ของอนุกรมเวลาที่จะนำมาทำการเฉลี่ยรูปร่างตามที่ได้อธิบายรายละเอียดในการคำนวณจากรูปที่ 3.9 เมื่อได้ข้อมูลอนุกรมเวลาในกลุ่มข้อมูล S ตัวที่ a และ b แล้ว บรรทัดที่ 3 นำข้อมูลอนุกรมเวลาทั้งสองมาคำนวณระยะทางด้วยวิธีไดนามิกไทม์วอร์ปปิงแบบผสมระหว่างไทม์วอร์ปปิงกับไทม์วอร์ปปิงแบบอนุพันธ์ด้วยฟังก์ชัน $hybrid\_dtw\_ddtw(s_a, s_b)$ เพื่อสร้างการปรับแนวระหว่างข้อมูลอนุกรมเวลาทั้งสอง $pth_{a,b}$ ตามที่ได้อธิบายรายละเอียดในหัวข้อที่ 3.3.2.1 ต่อมาในบรรทัดที่ 4 และ 5 ทำการเก็บค่าในแกน X ของข้อมูลอนุกรมเวลา s_a และ s_b ในบรรทัดที่ 6 และ 7 เป็นการหาค่าเฉลี่ยระหว่างจุดข้อมูลที่ได้จากการปรับแนวตามสมการที่ (3.5) ในบรรทัดที่ 8 นำข้อมูลอนุกรมเวลาที่ได้จากการหาค่าเฉลี่ยรูปร่างระหว่างข้อมูลอนุกรมเวลา s_a และ s_b มาทำการเลือกตัวอย่างใหม่

โดยใช้ฟังก์ชัน *Cubic_Spline*(*New_x*, *New_y*, x_1) เมื่อ *New_x* และ *New_y* คือค่าในแกน X และแกน Y ของแต่ละจุดข้อมูลหลังจากการหาค่าเฉลี่ย และ x_1 คือความยาวของข้อมูลต้นแบบที่นำมาหาค่าเฉลี่ยรูปร่าง ในบรรทัดที่ 9 ทำการปรับค่า s_a เดิมให้เป็นค่าของข้อมูลอนุกรมเวลา *Resampled_S* และบรรทัดที่ 10 ก็ทำการปรับค่าถ่วงน้ำหนัก w_a ของข้อมูลอนุกรมเวลา *Resampled_S* โดยนำค่าถ่วงน้ำหนักของข้อมูลอนุกรมเวลาต้นแบบทั้งสองมาบวกกัน แล้วทำการลบข้อมูลอนุกรมเวลา s_b ออกจากกลุ่มข้อมูลเรียนรู้ ในบบรทัดที่ 12 แล้วทำซ้ำจนกระทั่งเหลือข้อมูลอนุกรมเวลาเพียงอนุกรมเดียว

บทที่ 4

การทดลองและวิเคราะห์ผล

ในหัวข้อนี้เพื่อที่จะสามารถทำการประเมินคุณภาพและประสิทธิภาพของแนวคิดทั้งหมดที่ผู้วิจัยนำเสนอได้อย่างครบถ้วน ดังนั้นจะแบ่งการทดลองทั้งหมดออกเป็นสามส่วนหลัก ๆ ด้วยกัน คือ 1. การทดลองเพื่อวิเคราะห์คุณภาพของการหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลาด้วยวิธีที่นำเสนอ 2. การทดลองเพื่อวิเคราะห์ประสิทธิภาพทั้งในด้านความแม่นยำและความเร็วในการจำแนกประเภทของข้อมูลอนุกรมเวลา และ 3. การทดลองเกี่ยวกับประสิทธิภาพในการแบ่งกลุ่มย่อยภายในกลุ่มข้อมูลก่อนสร้างแผ่นแบบ

4.1 รูปแบบของข้อมูลที่ใช้ทดลองในงานวิจัย

ในงานวิจัยนี้ได้แบ่งชนิดของข้อมูลที่ใช้ในการทดลองออกเป็น 2 ประเภท ได้แก่ ข้อมูลจริงที่ใช้กันทั่วไปในงานวิจัยด้านการทำเหมืองข้อมูลอนุกรมเวลา และข้อมูลที่ได้จากการสังเคราะห์ขึ้นเอง โดยสังเคราะห์ตามวิธีที่ใช้กันทั่วไป

4.1.1 ข้อมูลจริงที่ใช้กันทั่วไปในงานวิจัยด้านการทำเหมืองข้อมูลอนุกรมเวลา

ในงานวิจัยนี้ได้นำชุดข้อมูลอนุกรมเวลาที่ได้รับการเปิดเผยสำหรับงานวิจัยที่สนใจศึกษาเกี่ยวกับข้อมูลอนุกรมเวลามาใช้ในการทดสอบประสิทธิภาพของแนวคิดที่ได้นำเสนอ ซึ่งรายละเอียดของชุดข้อมูลทั้งหมดสามารถหาได้ในเว็บไซต์ Archive ของ University of California, Riverside [32] โดยในตารางที่ 4.1 แสดงคุณลักษณะของชุดข้อมูลจริงทั้งหมดที่งานวิจัยนี้ได้นำมาเป็นข้อมูลในการทดลอง

สำหรับงานวิจัยนี้ได้มุ่งเน้นไปที่การสร้างแผ่นแบบเพื่อเป็นตัวแทนกลุ่มของข้อมูลอนุกรมเวลา ดังนั้นชุดข้อมูลจริงที่นำมาทดสอบนั้นจึงจำเป็นจะต้องเป็นชุดข้อมูลที่มีความหลากหลายของข้อมูลทั้งความยาวของข้อมูลอนุกรมเวลา และจำนวนคลาสในแต่ละชุดข้อมูล เพื่อที่จะทดสอบว่าแนวคิดในการสร้างแผ่นแบบในงานวิจัยนี้จะสามารถนำไปใช้กับชุดข้อมูลอนุกรมเวลาได้หลากหลายหรือไม่ จากตารางที่ 4.1 ชุดข้อมูลที่นำมาทดสอบนั้นมีความหลากหลายทั้งในเรื่องของจำนวนคลาสซึ่งมีจำนวนคลาสน้อย ๆ ตั้งแต่ 2 คลาสซึ่งได้แก่ ชุดข้อมูล Coffee ชุดข้อมูล ECG ชุดข้อมูล Gun-Point และชุดข้อมูล Lightning2 จนกระทั่งชุดข้อมูลที่มีจำนวนคลาสมากถึง 50 คลาส ซึ่งก็คือชุดข้อมูล 50Words นอกจากนั้นชุดข้อมูลที่นำมาทดสอบยังมีความยาวที่แตกต่างกันตั้งแต่ชุดข้อมูลที่มีความยาวเพียง 60 จุดข้อมูล คือชุดข้อมูล Synthetic Control จนกระทั่งชุดข้อมูลที่มีความยาวถึง 637 จุดข้อมูลซึ่งก็คือ ชุดข้อมูล Lightning2

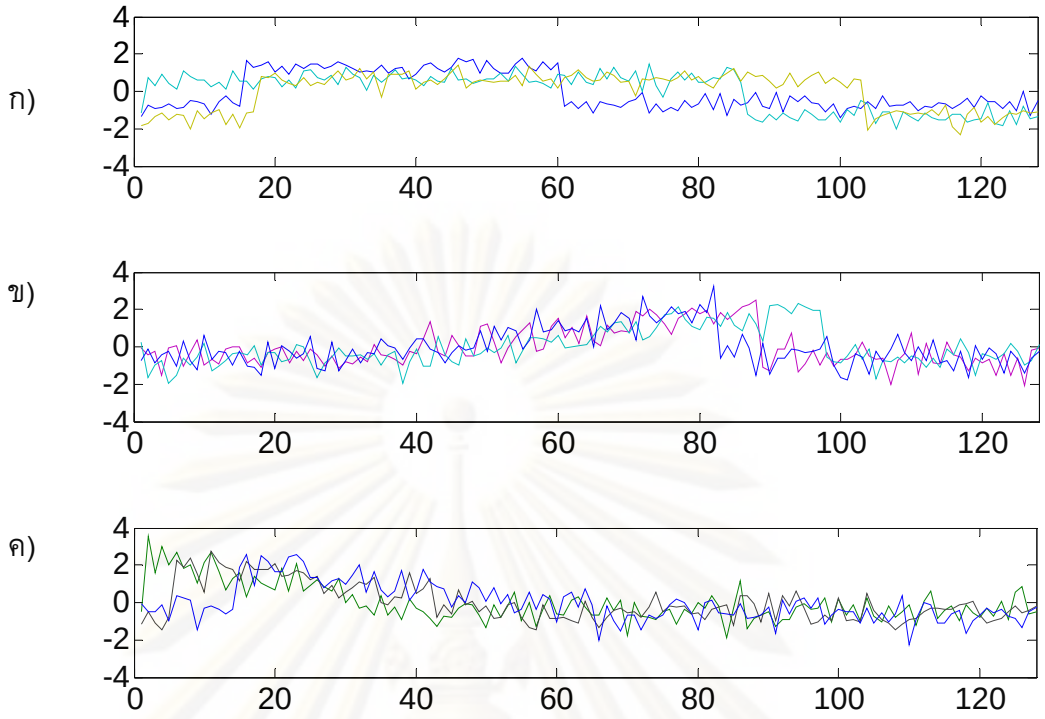
ตารางที่ 4.1 คุณลักษณะของชุดข้อมูลจริงที่ใช้ในการทดลอง

ชุดข้อมูล	*จำนวนคลาส*	*จำนวนข้อมูลในชุดข้อมูล*	*จำนวนข้อมูลสอบถาม*	*ความยาว (มิติ)*
50Words	50	450	455	270
Adiac	37	390	391	176
Beef	5	30	30	470
CBF	3	30	900	128
Coffee	2	28	28	286
ECG	2	100	100	96
Face (all)	14	560	1690	131
Face (four)	4	24	88	350
Gun-Point	2	50	150	150
Lightning2	2	60	61	637
Lightning7	7	70	73	319
Oliveoil	4	30	30	570
OSU Leaf	6	200	242	427
Swedish Leaf	15	500	625	128
Synthetic Control	6	300	300	60
Trace	4	100	100	275
Two Patterns	4	1000	4000	128

4.1.2 ข้อมูลที่ได้จากการสังเคราะห์ขึ้น

4.1.2.1 การสังเคราะห์ข้อมูลซีบีเอฟ (CBF: Cylinder Bell Funnel)

ข้อมูลซีบีเอฟ [33] เป็นข้อมูลสังเคราะห์ที่สามารถแบ่งออกได้เป็น 3 คลาส ได้แก่ คลาสกระบอก (Cylinder) ดังแสดงรูปที่ 4.1 ก) คลาสระฆัง (Bell) ดังแสดงในรูปที่ 4.1 ข) และคลาสกรวย (Funnel) ดังแสดงในรูปที่ 4.1 ค) โดยข้อมูลซีบีเอฟทุกตัวจะมีความยาวเท่ากับ 128 จุดข้อมูล รายละเอียดในการสังเคราะห์ข้อมูลซีบีเอฟนั้นมีดังต่อไปนี้



รูปที่ 4.1 ตัวอย่างข้อมูลซีบีเอฟทั้ง 3 คลาส ก) คลาสกระบอก ข) คลาสระฆัง ค) คลาสกรวย

ในการสังเคราะห์ข้อมูลซีบีเอฟ สามารถอธิบายรายละเอียดได้ดังต่อไปนี้ แต่ละอนุกรมจะต้องทำการสุ่มค่าตัวแปร α และ β โดยที่ α นั้นมีค่าอยู่ในช่วง 16 ถึง 32 และค่า $\beta-\alpha$ จะอยู่ในช่วง 32 ถึง 96 จากนั้นกำหนดให้ข้อมูลอนุกรมเวลาในคลาสกระบอก C ข้อมูลอนุกรมเวลาในคลาสระฆัง B ข้อมูลอนุกรมเวลาในคลาสกรวย F ตามสมการที่ (4.1)

$$\begin{aligned} C &= \{C_1, C_2, \ldots, C_{128}\} \\ B &= \{B_1, B_2, \ldots, B_{128}\} \\ F &= \{F_1, F_2, \ldots, F_{128}\} \end{aligned} \tag{4.1}$$

และสามารถสร้างข้อมูลอนุกรมเวลาทั้ง 3 คลาสได้จากสมการที่ (4.2)

$$\begin{aligned} C_i &= \eta(6,1) * \chi_{[\alpha,\beta]}(i) + \eta(0,1) \\ B_i &= \eta(6,1) * \chi_{[\alpha,\beta]}(i) * (i-\alpha)/(\beta-\alpha) + \eta(0,1) \\ F_i &= \eta(6,1) * \chi_{[\alpha,\beta]}(i) * (\beta-i)/(\beta-\alpha) + \eta(0,1) \\ \chi_{[\alpha,\beta]}(i) &= \begin{cases} 1 & \text{if } \alpha \le i \le \beta \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{4.2}$$

4.2 การทดลองเพื่อวิเคราะห์คุณภาพการหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลา

ในหัวข้อนี้จะนำเสนอวิธีการทดลอง ผลการทดลองเพื่อวิเคราะห์คุณภาพการหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลาด้วยวิธีที่นำเสนอ ดังต่อไปนี้

4.2.1 ทดลองเพื่อวิเคราะห์คุณภาพของการจัดลำดับในการหาค่าเฉลี่ยรูปร่างด้วยวิธีที่นำเสนอ

สำหรับการทดลองในหัวข้อนี้มีจุดประสงค์เพื่อวิเคราะห์ว่าลำดับในการหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลาที่อยู่ในคลาสเดียวกันนั้นมีผลหรือไม่ต่อผลลัพธ์ที่ได้ โดยการเรียงลำดับในการหาค่าเฉลี่ยที่จะนำมาเปรียบเทียบกับวิธีที่นำเสนอ ได้แก่ การหาค่าเฉลี่ยรูปร่างของคู่ข้อมูลอนุกรมเวลาที่มีระยะทางมากที่สุดก่อน และการหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลาโดยการสุ่ม (Random) โดยมาตรวัดที่จะนำมาใช้ทดสอบคุณภาพของลำดับในการหาค่เฉลี่ยรูปร่างคือระยะทางภายในคลาส (Intra-class Distance) โดยสามารถคำนวณได้จากค่าเฉลี่ยระยะทางระหว่างข้อมูลอนุกรมเวลาทุกตัวที่อยู่ในคลาสเทียบกับข้อมูลอนุกรมเวลาที่ได้จากการเฉลี่ยข้อมูล

การคำนวณค่าระยะทางภายในคลาส สามารถอธิบายรายละเอียดได้ดังนี้ กำหนดให้กลุ่มข้อมูลอนุกรมเวลา $S = S_1, S_2, \ldots, S_n$ ประกอบไปด้วยข้อมูลอนุกรมเวลา n อนุกรม เมื่อนำข้อมูลในกลุ่มข้อมูล S มาคำนวณค่าเฉลี่ยรูปร่างจะได้ข้อมูลอนุกรมเวลาที่เป็นค่าเฉลี่ยของข้อมูลทั้งกลุ่ม $A = a_1, a_2, \ldots, a_n$ โดยระยะทางภายในคลาสสามารถคำนวณได้ตามสมการที่ (4.3)

$$Intra-class\_Dist = \frac{\sum_{i=1}^{n} dtw(A, s_i)}{n} \tag{4.3}$$

ระยะทางภายในคลาสนั้น เป็นระยะทางที่บอกถึงระยะทางเฉลี่ยระหว่างข้อมูลอนุกรมเวลาทุกตัวที่อยู่ในคลาสเปรียบเทียบข้อมูลอนุกรมเวลาที่เป็นค่าเฉลี่ยรูปร่าง ซึ่งระยะทางภายในคลาสที่ได้นั้นควรจะมีค่าน้อย ๆ แสดงว่าข้อมูลอนุกรมเวลาที่เป็นค่าเฉลี่ยรูปร่างนั้นสามารถแทนข้อมูลอนุกรมเวลาตัวอื่น ๆ ในกลุ่มได้เป็นอย่างดี ตารางที่ 4.2 แสดงผลการทดลองเปรียบเทียบระยะทางภายในคลาสระหว่างข้อมูลอนุกรมเวลาทุกตัวที่อยู่ในคลาสกับข้อมูลอนุกรมเวลาที่เป็นค่าเฉลี่ยรูปร่าง โดยทำการปรับลำดับในการคำนวณค่าเฉลี่ยรูปร่างดังนี้ คำนวณคู่ของข้อมูลอนุกรมเวลาที่มีระยะทางมากที่สุดในแต่ละรอบ คำนวณคู่ของข้อมูลอนุกรมเวลาด้วยการสุ่มโดยไม่สนใจระยะทางระหว่างแต่ละข้อมูล และคู่ของข้อมูลอนุกรมเวลาที่มีระยะทางน้อยที่สุดในแต่ละรอบดังที่ได้นำเสนอในหัวข้อที่ 3.3.1 การทดลองนี้วัดผลด้วยระยะทางภายในคลาสด้วยชุดข้อมูลจริงในตารางที่ 4.1 โดยผลการทดลองดังกล่าวแสดงไว้ในตารางที่ 4.2

ตารางที่ 4.2 ผลการทดลองเปรียบเทียบระยะทางภายในคลาสของค่าเฉลี่ยรูปร่างข้อมูลอนุกรมเวลากับข้อมูลอนุกรมเวลาในคลาส ด้วยการปรับลำดับในการหาค่าเฉลี่ย

ชุดข้อมูล	***ระยะทางภายในคลาส***		
	ASA-Min	***ASA-Max***	***ASA-Random***
50Words	**2.67**	3.28	2.89
Adiac	**0.36**	2.45	0.76
Beef	**0.35**	1.26	0.43
CBF	**1.24**	2.22	1.73
Coffee	**0.65**	1.25	0.96
ECG	**2.59**	5.10	3.27
Face (all)	**3.89**	5.79	4.39
Face (four)	**5.58**	6.23	5.78
Gun-Point	**2.80**	3.28	2.89
Lightning2	**9.32**	12.02	9.85
Lightning7	**5.95**	6.98	6.37
Oliveoil	**0.05**	1.12	0.09
OSU Leaf	**5.95**	6.54	7.69
Swedish Leaf	**1.29**	2.16	1.70
Synthetic Control	**3.46**	3.96	3.72
Trace	**1.35**	1.78	1.46
Two Patterns	**3.98**	5.09	4.87

จากผลการทดลองในตารางที่ 4.2 แสดงให้เห็นว่าการเรียงลำดับในการหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลานั้น จะมีประสิทธิภาพมากขึ้นเมื่อมีการเรียงลำดับข้อมูลอนุกรมเวลาในการเฉลี่ยข้อมูลอนุกรมเวลาที่ดี โดยจากผลการทดลองการเรียงลำดับที่ให้ประสิทธิภาพในการหาค่าเฉลี่ยรูปร่างดีที่สุดคือเฉลี่ยข้อมูลที่มีความคล้ายคลึงกันมากที่สุดหรือมีระยะทางน้อยที่สุดก่อน เพราะสามารถให้ค่าระยะทางภายในคลาสน้อยที่สุดเมื่อเปรียบเทียบกับการเรียงลำดับในการหาค่าเฉลี่ยแบบอื่น ๆ เหตุผลที่ทำให้การเฉลี่ยข้อมูลที่คล้ายกันมากที่สุด

ก่อนดีกว่าการเรียงลำดับแบบอื่น ๆ นั้น เพราะการนำอนุกรมเวลาที่มีความคล้ายคลึงกันมาหาการปรับแนวก่อนทำการหาค่าเฉลี่ยรูปร่างนั้น จะได้การปรับแนวที่เกือบจะเป็นแบบหนึ่งต่อหนึ่ง ทำให้เมื่อทำการหาค่าเฉลี่ยแล้วจำนวนจุดก็ไม่ได้เพิ่มมากขึ้นจากต้นแบบ และเมื่อผ่านการประมาณค่าด้วยฟังก์ชันกระดูกงูกำลังสามแล้วค่าประมาณที่ได้ก็จะแตกต่างจากค่าเฉลี่ยจริงเพียงเล็กน้อยเท่านั้น

4.2.2 ทดลองเพื่อวิเคราะห์คุณภาพการปรับแนวแบบผสมระหว่างไทม์วอร์ปปิงกับไทม์วอร์ปปิงแบบอนุพันธ์

สำหรับการทดลองในหัวข้อนี้เพื่อที่จะวิเคราะห์คุณภาพของการปรับแนวด้วยวิธีที่นำเสนอนั้นจะสามารถทำให้ได้ค่าเฉลี่ยรูปร่างในเชิงเวลาที่แท้จริงหรือไม่ เนื่องจากการหาค่าเฉลี่ยระหว่างข้อมูลอนุกรมเวลาในงานวิจัยนี้ได้ใช้การปรับแนวแบบผสมระหว่างไทม์วอร์ปปิงกับไทม์วอร์ปปิงแบบอนุพันธ์เพื่อเป็นตัวกำหนดคู่ของจุดข้อมูลที่จะนำค่าของจุดข้อมูลมาเฉลี่ยกันเพื่อให้ผลลัพธ์ที่ได้เป็นผลลัพธ์จากการเฉลี่ยค่าในเชิงรูปร่างของข้อมูล โดยการปรับแนวที่นำมาทดลองนั้นได้แก่ การปรับแนวแบบไทม์วอร์ปปิง การปรับแนวแบบไทม์วอร์ปปิงแบบอนุพันธ์ และการปรับแนวแบบผสมระหว่างไทม์วอร์ปปิงกับไทม์วอร์ปปิงแบบอนุพันธ์ซึ่งได้นำเสนอไว้ในหัวข้อที่ 3.3.2.1 โดยมาตรวัดที่ใช้ในการทดสอบประสิทธิภาพของการปรับแนวแบบต่าง ๆ เพื่อจับคู่จุดข้อมูลนั้นจะวัดจากระยะทางภายในคลาส เช่นเดียวกับการทดลองที่ 4.2.1 การทดลองนี้ทดสอบด้วยการหาค่าเฉลี่ยรูปร่างข้อมูลอนุกรมเวลาด้วยการปรับแนวที่แตกต่างกัน ด้วยชุดข้อมูลตามตารางที่ 4.1 โดยผลการทดลองดังกล่าวแสดงไว้ในตารางที่ 4.3

ตารางที่ 4.3 ผลการทดลองการเปรียบเทียบระยะทางภายในคลาสระหว่างการหาค่าเฉลี่ยรูปร่างโดยอาศัยการปรับแนวที่แตกต่างกันได้แก่ การปรับแนวแบบไทม์วอร์ปปิง (ASA-DTW) การปรับแนวแบบไทม์วอร์ปปิงแบบอนุพันธ์ (ASA-DDTW) และการปรับแนวแบบผสมระหว่างไทม์วอร์ปปิงกับไทม์วอร์ปปิงแบบอนุพันธ์ (ASA-HDTW)

ชุดข้อมูล	***ระยะทางภายในคลาส***		
	ASA-DTW	***ASA-DDTW***	***ASA-HDTW***
50Words	2.78	2.86	**2.67**
Adiac	0.39	0.39	**0.36**
Beef	0.42	0.39	**0.35**
CBF	4.01	4.29	**1.24**
Coffee	2.12	0.72	**0.65**

ชุดข้อมูล	ระยะทางภายในคลาส		
	ASA-DTW	ASA-DDTW	ASA-HDTW
ECG	2.78	2.62	**2.59**
Face (all)	3.99	3.97	**3.89**
Face (four)	5.86	6.18	**5.58**
Gun-Point	2.87	2.97	**2.80**
Lightning2	9.42	9.47	**9.32**
Lightning7	5.98	6.17	**5.95**
Oliveoil	0.09	0.08	**0.05**
OSU Leaf	6.47	7.69	**5.95**
Swedish Leaf	1.30	1.32	**1.29**
Synthetic Control	3.58	3.53	**3.46**
Trace	1.36	1.37	**1.35**
Two Patterns	4.26	4.17	**3.98**

จากผลการทดลองในตารางที่ 4.3 แสดงให้เห็นว่าค่าเฉลี่ยรูปร่างที่ได้จากการปรับแนวแบบผสมระหว่างไทม์วอร์ปปิงกับไทม์วอร์ปปิงแบบอนุพันธ์มีค่าระยะทางภายในคลาสน้อยที่สุดเมื่อเทียบกับค่าเฉลี่ยรูปร่างที่ใช้การปรับแนวแบบอื่น ๆ นั่นก็หมายความว่าค่าเฉลี่ยรูปร่างที่ได้จากการปรับแนวด้วยวิธีที่นำเสนอนั้นสามารถให้ค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมภายในคลาสได้ดีที่สุด เนื่องจากการปรับแนวแบบผสมนั้นมีการนำทั้งค่าของจุดข้อมูลและค่าประมาณอนุพันธ์ของแต่ละจุดข้อมูลมาคำนวณ ทำให้ได้คู่ของจุดข้อมูลที่มีค่าใกล้เคียงกันหรือคล้ายกันมากที่สุดเพื่อนำมาคำนวณค่าเฉลี่ย

4.2.3 ทดลองเพื่อวิเคราะห์คุณภาพการหาค่าเฉลี่ยรูปร่างด้วยวิธีที่นำเสนอเมื่อเปรียบเทียบกับวิธีอื่น ๆ

สำหรับการทดลองในหัวข้อนี้มีจุดประสงค์เพื่อวิเคราะห์คุณภาพของวิธีในการหาค่าเฉลี่ยรูปร่างจากวิธีที่นำเสนอนั้น จะสามารถหาข้อมูลอนุกรมเวลาที่เป็นค่าเฉลี่ยของข้อมูลอนุกรมเวลาต้นแบบทั้งสองได้อย่างมีประสิทธิภาพมากน้อยเพียงใด โดยจะทำการทดลองกับชุดข้อมูลที่แสดงในตารางที่ 4.1 มาตรวัดที่ใช้ทดสอบในหัวข้อนี้คือระยะทางความคลาดเคลื่อน (Discrepancy Distance) โดยจะเปรียบเทียบระยะทางความคลาดเคลื่อนระหว่างวิธีที่นำเสนอ

กับวิธีอื่น ๆ สำหรับรายละเอียดของวิธีที่จะนำมาใช้ในการเปรียบเทียบนั้น สามารถแสดงได้ดังต่อไปนี้

1. *การหาค่าเฉลี่ยรูปร่างโดยใช้การปรับแนวแบบไทม์วอร์ปปิงด้วยวิธี NLAAF* การหาค่าเฉลี่ยรูปร่างด้วยวิธีนี้จะใช้การปรับแนวแบบไทม์วอร์ปปิง ซึ่งได้มาจากวิถีของการวอร์ปในการคำนวณระยะทางแบบไดนามิกไทม์วอร์ปปิง โดยเมื่อได้คู่ของจุดข้อมูลที่จะนำมาเฉลี่ยแล้วจะทำการหาค่าเฉลี่ยเฉพาะค่าในแนวแกน Y โดยที่ค่าในแนวแกน X จะทำการเลื่อนเพิ่มไปเรื่อย ๆ โดยมีระยะทางที่เท่ากัน เช่น มีข้อมูลอนุกรมเวลา Q และ C เมื่อ $Q = q_1, q_2, \ldots, q_n$ และ $C = c_1, c_2, \ldots, c_n$ ถ้าจุด q_1 จับกับจุด c_1 และ c_2 จะนำค่าในแกน Y ของจุด q_1 กับจุด c_1 มาเฉลี่ยกันได้เป็นจุดข้อมูลแรกของค่าเฉลี่ย จากนั้นก็เฉลี่ยจุด q_1 กับจุด c_2 ซึ่งจะได้เป็นจุดข้อมูลที่สอง โดยไม่มีการนำค่าในแกน X มาเฉลี่ยค่ากัน
2. *การหาค่าเฉลี่ยรูปร่างโดยใช้การปรับแนวแบบไทม์วอร์ปปิงด้วยวิธี PSA* การหาค่าเฉลี่ยรูปร่างด้วยวิธีนี้นั้น ใช้การปรับแนวแบบไทม์วอร์ปปิงที่ได้จากการคำนวณระยะทางแบบไดนามิกไทม์วอร์ปปิงเช่นเดียวกับวิธี NLAAF ต่างกันตรงที่วิธี PSA นั้น ในการหาค่าเฉลี่ยแต่ละรอบจะมีการนำค่าในแกน X มาคำนวณด้วย นอกจากนั้นค่าในแกน X ที่นำมาคำนวณนั้นจะคำนวณค่าถ่วงน้ำหนักเช่นเดียวกับค่าในแกน Y ตามสมการที่ (3.5) และเมื่อหาค่าเฉลี่ยรูปร่างแล้ว จะทำการตัดบางจุดข้อมูลทิ้งเพื่อให้ข้อมูลอนุกรมเวลาที่ได้มีความยาวเท่ากับข้อมูลต้นแบบ

เนื่องจากนิยามของค่าเฉลี่ยคือ ระยะทางระหว่างค่าเฉลี่ยใด ๆ กับข้อมูลต้นแบบจะต้องมีค่าเท่ากัน กล่าวคือถ้ามีข้อมูลอนุกรมเวลา Q และ C เมื่อหาค่าเฉลี่ยได้เป็นอนุกรมเวลา A ทำการวัดระยะทางระหว่างข้อมูลอนุกรมเวลา Q กับ A จะต้องมีค่าเท่ากับระยะทางระหว่าง C กับ A ดังนั้นถ้าหากค่าเฉลี่ยที่ได้มีความคลาดเคลื่อน ระยะทางความคลาดเคลื่อนที่ได้ค่าต่ำแสดงว่าความคลาดเคลื่อนที่เกิดจากวิธีหาค่าเฉลี่ยนั้นมีน้อย

การคำนวณระยะทางความคลาดเคลื่อน สามารถอธิบายรายละเอียดได้ดังนี้ กำหนดให้มีข้อมูลอนุกรมเวลาสองอนุกรม $Q = q_1, q_2, \ldots, q_m$ และ $C = c_1, c_2, \ldots, c_n$ เมื่อนำข้อมูลอนุกรมเวลาทั้งสองมาคำนวณค่าเฉลี่ยรูปร่างตามสมการที่ (3.5) จะได้ข้อมูลอนุกรมเวลาที่เป็นค่าเฉลี่ยรูปร่าง $A = a_1, a_2, \ldots, a_k$ เมื่อนำมาคำนวณหาระยะทางเทียบกับข้อมูลอนุกรมเวลาต้นแบบทั้งสองควรจะได้ระยะทางที่เท่ากัน ตามสมการที่ (4.4)

$$dtw(Q, A) = dtw(C, A) \tag{4.4}$$

โดยระยะทางความคลาดเคลื่อนนั้นสามารถคำนวณได้ตามสมการที่ (4.5)

$$Discrepancy = \frac{\left|dtw(Q,A) - dtw(C,A)\right|}{\min(dtw(Q,A), dtw(C,A))} \quad (4.5)$$

การทดลองนี้ทดสอบด้วยการหาค่าเฉลี่ยรูปร่างข้อมูลอนุกรมเวลาด้วยวิธีที่นำเสนอ เปรียบเทียบกับวิธี PSA และวิธี NLAAF ด้วยชุดข้อมูลในตารางที่ 4.1 โดยผลการทดลองดังกล่าวแสดงไว้ในตารางที่ 4.4

ตารางที่ 4.4 ผลการทดลองค่าระยะทางความคลาดเคลื่อนระหว่างข้อมูลอนุกรมเวลาภายในกลุ่มเปรียบเทียบกับค่าเฉลี่ยรูปร่างที่ได้จากแต่ละวิธี ได้แก่ วิธี ASA วิธี PSA และวิธี NLAAF

ชุดข้อมูล	***ระยะทางความคลาดเคลื่อน***		
	ASA	***PSA***	***NLAAF***
50Words	**0.08**	0.45	0.56
Adiac	**0.03**	0.24	0.26
Beef	**0.07**	0.12	0.13
CBF	**0.07**	0.17	0.22
Coffee	**0.03**	0.07	0.27
ECG	**0.12**	0.29	0.34
Face (all)	**0.08**	0.27	0.39
Face (four)	0.07	**0.04**	0.25
Gun-Point	**0.08**	0.16	0.19
Lightning2	0.13	**0.09**	0.17
Lightning7	**0.16**	0.22	0.29
Oliveoil	**0.00**	0.32	0.38
OSU Leaf	**0.06**	0.48	0.52
Swedish Leaf	**0.07**	0.62	0.77
Synthetic Control	**0.14**	0.18	0.37
Trace	**0.07**	0.09	0.16

ชุดข้อมูล	*ระยะทางความคลาดเคลื่อน*		
	ASA	***PSA***	***NLAAF***
Two Patterns	**0.09**	0.14	0.31

จากผลการทดลองในตารางที่ 4.4 แสดงให้เห็นว่าการหาค่าเฉลี่ยด้วยวิธีที่นำเสนอสามารถให้ค่าระยะทางความคลาดเคลื่อนต่ำกว่าวิธีอื่น ๆ ที่นำมาเปรียบเทียบ ซึ่งแสดงว่าการหาค่าเฉลี่ยรูปร่างด้วยวิธีที่นำเสนอนั้นให้ค่าเฉลี่ยรูปร่างที่ดีที่สุดเมื่อเทียบกับทั้งสองวิธี เพราะมีระยะทางความคลาดเคลื่อนน้อยมาก เนื่องจากการหาค่าเฉลี่ยด้วยวิธีที่นำเสนอนั้นมีการใช้ฟังก์ชันกระดูกงูกำลังสามเพื่อทำการประมาณค่าจุดข้อมูลในตำแหน่งแกน X ที่ต้องการ ซึ่งฟังก์ชันดังกล่าวให้ค่าของแต่ละจุดข้อมูลที่ใกล้เคียงกับค่าเฉลี่ยจริง ทำให้เมื่อวัดระยะทางความคลาดเคลื่อนจึงทำให้ได้ค่าที่ต่ำกว่าวิธีอื่น

4.3 ทดลองเพื่อวิเคราะห์คุณภาพของแผ่นแบบที่ได้จากการหาค่าเฉลี่ยรูปร่างของวิธีที่นำเสนอเมื่อเปรียบเทียบกับวิธีอื่น ๆ

ในหัวข้อนี้จะอธิบายถึงขั้นตอนการทำงานของแต่ละวิธีที่งานวิจัยนี้ได้นำมาเปรียบเทียบกับการหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลา ได้แก่ การหาค่าเฉลี่ยรูปร่างด้วยวิธี NLAAF และการหาค่าเฉลี่ยรูปร่างด้วยวิธี PSA ซึ่งทั้งสองงานวิจัยมีแนวทางในการหาค่าเฉลี่ยรูปร่างโดยใช้การปรับแนวแบบไทม์วอร์ปปิงคล้ายกับการหาค่าเฉลี่ยแบบ ASA ที่นำเสนอ โดยจะเปรียบเทียบค่าระยะทางภายในคลาสของข้อมูลอนุกรมเวลาในแต่ละชุดข้อมูล สำหรับรายละเอียดของวิธีที่จะนำมาใช้ในการเปรียบเทียบนั้น สามารถแสดงได้ดังต่อไปนี้

1. *การหาค่าเฉลี่ยรูปร่างด้วยวิธี NLAAF* เป็นการหาค่าเฉลี่ยรูปร่างโดยใช้การปรับแนวแบบไทม์วอร์ปปิง ซึ่งแบ่งการทำงานออกเป็น 2 ส่วนคือ NLAAF1 และ NLAAF2 วิธีการนี้จะไม่มีการจัดเรียงลำดับการหาค่าเฉลี่ยของข้อมูลอนุกรมเวลา แต่ใช้การสุ่มเลือกข้อมูลเพื่อนำมาทำการหาค่าเฉลี่ยรูปร่างไปเรื่อย ๆ จนกระทั่งเหลือข้อมูลอนุกรมเวลาเพียงอนุกรมเดียว หลังจากการหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลาในแต่ละรอบด้วยวิธีนี้ เมื่อมีจุดข้อมูลที่มีค่าในแกน X ไม่เป็นจำนวนเต็มหรือไม่ลงกริด จะทำการยืดจุดข้อมูลเหล่านั้น โดยให้ทุก ๆ จุดข้อมูลมีระยะห่างเท่า ๆ กัน และไม่ทำการลดจำนวนจุดข้อมูล คือปล่อยให้ข้อมูลอนุกรมเวลามีจำนวนจุดข้อมูลเพิ่มมากขึ้นหรือมีความยาวเพิ่มมากขึ้น
2. *การหาค่าเฉลี่ยรูปร่างด้วยวิธี PSA* เป็นการหาค่าเฉลี่ยรูปร่างโดยใช้การปรับแนวแบบไทม์วอร์ปปิง วิธี PSA ใช้การจับกลุ่มแบบลำดับขั้นเพื่อเรียงลำดับของข้อมูลอนุกรมเวลาในการหาค่าเฉลี่ยรูปร่าง หลังจากการหาค่าเฉลี่ยรูปร่างในแต่ละรอบถ้ามีจุดข้อมูลที่มีค่าในแกน X ไม่เป็นจำนวนเต็มจะทำการยืดจุดข้อมูล

ออกเพื่อให้ทุก ๆ จุดมีระยะห่างเท่ากัน และจะเลือกหยิบหรือตัดบางจุดข้อมูลมาใช้เพื่อให้ข้อมูลอนุกรมเวลาที่ได้จากการหาค่าเฉลี่ยรูปร่างมีจำนวนจุดข้อมูลเท่ากับข้อมูลอนุกรมเวลาต้นแบบ เช่น ถ้าข้อมูลอนุกรมเวลาต้นแบบมีความยาว 10 จุด หลังจากหาค่าเฉลี่ยรูปร่างมีความยาว 30 จุด วิธี PSA จะใช้วิธีเลือกหนึ่งจุดตัดทิ้งสองจุด กล่าวคือ จุดที่ 1 จะถูกนำมาใช้ อีกสองจุดต่อมานั่นคือจุดที่ 2 และ 3 จะถูกตัดทิ้ง จุดที่ 4 จะถูกนำมาใช้และจุดที่ 5, 6 จะถูกตัดทิ้ง จะเห็นว่าท้ายที่สุดข้อมูลอนุกรมเวลาที่เกิดจากการหาค่าเฉลี่ยรูปร่างด้วยวิธี PSA จะมีจำนวนจุดข้อมูลเท่ากับ 10 จุดเช่นเดียวกับข้อมูลอนุกรมเวลาต้นแบบ

การทดลองนี้ทดสอบด้วยการคำนวณระยะทางภายในคลาสจากวิธีหาค่าเฉลี่ยรูปร่างข้อมูลอนุกรมเวลาด้วยวิธีที่นำเสนอ เปรียบเทียบกับวิธี PSA และวิธี NLAAF ด้วยชุดข้อมูลจริงในตารางที่ 4.1 โดยผลการทดลองดังกล่าวแสดงไว้ในตารางที่ 4.5

ตารางที่ 4.5 ผลการทดลองระยะทางภายในคลาสที่ได้จากการหาค่าเฉลี่ยของกลุ่มข้อมูลอนุกรมเวลาแต่ละวิธี ได้แก่ วิธี ASA วิธี PSA และวิธี NLAAF

ชุดข้อมูล	***ระยะทางภายในคลาส***		
	ASA	***PSA***	***NLAAF***
50Words	**2.67**	2.75	3.31
Adiac	**0.36**	0.59	1.02
Beef	**0.35**	3.81	5.78
CBF	**3.64**	3.87	4.48
Coffee	**0.89**	1.09	4.21
ECG	**2.59**	2.90	4.45
Face (all)	**3.89**	6.01	11.12
Face (four)	**5.28**	5.35	6.41
Gun-Point	**2.70**	2.72	4.17
Lightning2	**9.32**	12.40	16.60
Lightning7	**5.95**	6.94	8.94
Oliveoil	**0.05**	0.95	2.01
OSU Leaf	**6.47**	6.82	11.01

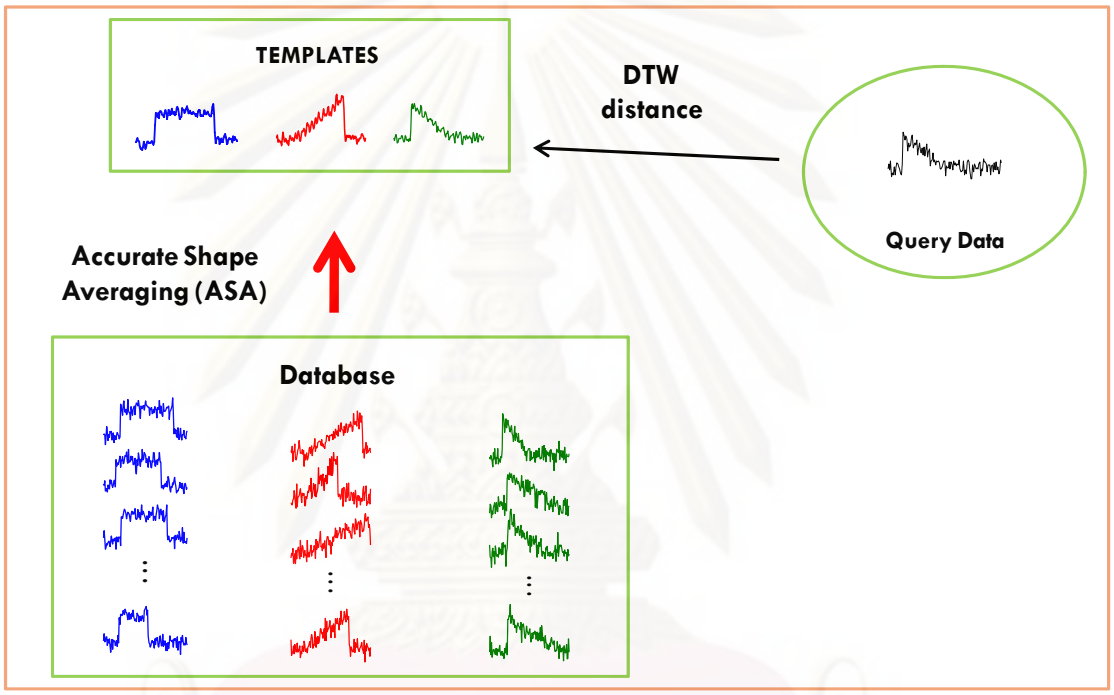
ชุดข้อมูล	*ระยะทางภายในคลาส*		
	ASA	***PSA***	***NLAAF***
Swedish Leaf	**1.29**	2.08	4.12
Synthetic Control	**0.84**	0.96	4.97
Trace	**1.35**	1.53	1.80
Two Patterns	**3.78**	3.85	6.33

จากผลการทดลองในตารางที่ 4.5 แสดงให้เห็นว่าค่าระยะทางภายในคลาสที่ได้จากวิธีที่นำเสนอนั้นมีค่าน้อยที่สุด สำหรับเหตุผลหลัก ๆ ที่ทำให้วิธีที่นำเสนอนั้น มีค่าระยะทางภายในคลาสน้อยกว่าวิธีอื่น ๆ นั้นก็คือการจัดเรียงลำดับในการหาค่าเฉลี่ย ซึ่งจากการทดลองจะเห็นว่าวิธีที่นำเสนอ ใช้วิธีการหาคู่ของข้อมูลอนุกรมเวลาที่คล้ายคลึงกันมากที่สุดมาเฉลี่ยกันในทุก ๆ รอบนั้นจะให้ผลในการหาค่าระยะทางภายในคลาสที่ดีกว่าวิธี PSA ที่ใช้การจัดกลุ่มแบบลำดับขั้น เพราะวิธีที่นำเสนอจะทำการคำนวณหาคู่ที่เหมาะสมในทุก ๆ รอบ แต่วิธี PSA นั้นคำนวณเพียงครั้งเดียว ซึ่งในบางกรณีผลลัพธ์ที่ได้จากการหาค่าเฉลี่ยรูปร่างในรอบก่อนหน้าอาจไม่ได้มีความคล้ายคลึงกับข้อมูลอนุกรมเวลาตามการจัดกลุ่มแบบลำดับขั้นนั้นก็ได้ และที่วิธีที่นำเสนอนั้นดีกว่าวิธี NLAAF ที่ไม่มีการจัดลำดับเลยเพราะในการสุ่มข้อมูลอนุกรมเวลามาหาค่าเฉลี่ยรูปร่างในรอบแรกของวิธี NLAAF ถ้าข้อมูลอนุกรมเวลาที่ถูกนำมาคำนวณค่าเฉลี่ยรูปร่างเป็นข้อมูลแปลกแยกแต่ถูกนำมาคำนวณในรอบแรก ๆ จะทำให้แนวโน้มของรูปร่างที่ได้จากการคำนวณค่าเฉลี่ยรูปร่างของข้อมูลทั้งคลาสไม่เหมือนกับข้อมูลตัวอื่น ๆ ภายในคลาส แต่จะไปเหมือนกับข้อมูลแปลกแยกที่ถูกนำมาคำนวณตั้งแต่ต้น

และอีกเหตุผลที่ทำให้ผลการทดลองด้วยวิธีที่นำเสนอ ดีกว่าอีกสองวิธีที่นำมาเปรียบเทียบ คือวิธีที่นำเสนอใช้ฟังก์ชันกระดูกงูกำลังสามในการประมาณค่าในช่วงในทุก ๆ รอบของการหาค่าเฉลี่ยรูปร่าง ซึ่งทำให้ผลลัพธ์ที่ได้ค่อนข้างใกล้เคียงกับค่าที่ได้จากการหาค่าเฉลี่ยรูปร่างจริง แต่วิธี PSA ที่ใช้การเลือกบางจุดมาใช้แล้วตัดบางจุดทิ้ง ทำให้ในทุก ๆ รอบมีการสูญเสียคุณลักษณะบางประการของข้อมูลต้นแบบไป ส่วนวิธี NLAAF นั้นไม่มีการลดจำนวนจุดข้อมูล เช่น ถ้าข้อมูลในคลาสมีความยาว 128 จุด แต่เมื่อหาค่าเฉลี่ยรูปร่างด้วยวิธี NLAAF แล้วข้อมูลอนุกรมเวลาเฉลี่ยสุดท้ายมีความยาวจุดเป็น 1,000 จุด ก็เป็นการไม่สมเหตุสมผล และเมื่อนำมาวัดระยะทางด้วยวิธีไดนามิกไทม์วอร์ปปิงเทียบกับข้อมูลตัวอื่น ๆ ที่อยู่ในคลาสเดียวกันก็จะได้ค่าระยะทางที่สูงกว่าวิธีอื่น

4.4 การทดลองเพื่อวิเคราะห์ประสิทธิภาพในด้านความเร็วและความแม่นยำของวิธีที่นำเสนอเมื่อเปรียบเทียบกับวิธีอื่น ๆ

การจำแนกประเภทข้อมูลสำหรับข้อมูลอนุกรมเวลานั้น สิ่งที่นักวิจัยส่วนใหญ่มุ่งเน้นที่จะศึกษาและพัฒนาก็คือความเร็วและความแม่นยำในการจำแนกประเภทข้อมูล ในหัวข้อนี้จะนำเสนอผลการทดลองเพื่อวิเคราะห์ว่าการสร้างแผ่นแบบที่ใช้เป็นตัวแทนกลุ่มของข้อมูลอนุกรมเวลาด้วยวิธีที่นำเสนอนั้นสามารถทำงานได้อย่างมีประสิทธิภาพทั้งในด้านความเร็วและความแม่นยำหรือไม่ โดยที่การทดลองสำหรับการจำแนกประเภทข้อมูลสำหรับข้อมูลอนุกรมเวลาโดยใช้แผ่นแบบนั้นเมื่อต้องจำแนกประเภทข้อมูล ข้อมูลที่ต้องการจำแนกประเภทจะถูกเปรียบเทียบความคล้ายคลึงด้วยการวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงเปรียบเทียบกับแผ่นแบบสำหรับข้อมูลแต่ละคลาสเท่านั้น ดังแสดงในรูปที่ 4.2



รูปที่ 4.2 การจำแนกประเภทข้อมูลอนุกรมเวลาโดยวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงเปรียบเทียบกับข้อมูลที่เป็นแผ่นแบบเท่านั้น

ในรูปที่ 4.2 แสดงถึงวิธีการในการจำแนกประเภทของข้อมูลอนุกรมเวลา โดยที่มีข้อมูล 3 คลาส ซึ่งข้อมูลอนุกรมเวลาในแต่ละคลาสจะถูกนำมาสร้างแผ่นแบบ โดยที่ข้อมูลหนึ่งคลาสจะมีแผ่นแบบที่เป็นข้อมูลอนุกรมเวลาเพียงหนึ่งอนุกรม จากนั้นเมื่อต้องการจำแนกประเภทข้อมูลสอบถาม ก็จะทำการวัดระยะทางด้วยวิธีไดนามิกไทม์วอร์ปปิงระหว่างข้อมูลสอบถามกับข้อมูลที่เป็นแผ่นแบบของแต่ละคลาส เมื่อคำนวณเปรียบครบทุกคลาสแล้วค่าระยะทางระหว่างข้อมูลสอบถามกับแผ่นแบบของข้อมูลคลาสใดมีค่าน้อยที่สุด ก็จะจำแนกให้ข้อมูลสอบถามอยู่ในคลาสเดียวกับข้อมูลแผ่นแบบนั้น

ในส่วนนี้จะอธิบายถึงขั้นตอนการทำงานของแต่ละวิธีที่งานวิจัยนี้ได้นำมาเปรียบเทียบกับการสร้างแผ่นแบบด้วยวิธีที่นำเสนอ โดยจะเปรียบเทียบประสิทธิภาพในด้าน

การลดจำนวนของข้อมูลในกลุ่มข้อมูลเรียนรู้ ได้แก่ วิธี PSA และวิธี AWARD ซึ่งงานวิจัยทั้งสองมีแนวทางเพื่อลดจำนวนของข้อมูลในกลุ่มข้อมูลเหมือนกับการสร้างแผ่นแบบด้วยวิธีที่นำเสนอ รวมทั้งเปรียบเทียบกับการจำแนกประเภทข้อมูลด้วยการวัดระยะทางแบบไดนามิกไทม์วอร์ปปิง ได้แก่ แผ่นแบบที่ได้จากวิธี PSA แผ่นแบบที่ได้จากวิธี AWARD และการใช้ข้อมูลเรียนรู้ทุกตัวเป็นแผ่นแบบ โดยรายละเอียดของวิธีที่จะนำมาเปรียบเทียบมีดังนี้

1. *การสร้างแผ่นแบบด้วยวิธี PSA* วิธีนี้เป็นการหาค่าเฉลี่ยรูปร่างของข้อมูลในกลุ่มข้อมูลเรียนรู้ โดยใช้การปรับแนวแบบไทม์วอร์ปปิงเพื่อเป็นตัวระบุว่าจุดข้อมูลใดจะนำมาเฉลี่ยกัน นอกจากนั้นในการหาค่าเฉลี่ยยังมีการเรียงลำดับในการเฉลี่ยใช้การจับกลุ่มแบบลำดับขั้น โดยแผ่นแบบที่ได้จากวิธีนี้หนึ่งคลาสจะมีหนึ่งแผ่นแบบ
2. *การสร้างแผ่นแบบด้วยวิธี AWARD* การสร้างแผ่นแบบด้วยวิธี AWARD นั้น จะเป็นการนำข้อมูลอนุกรมเวลาบางอนุกรมที่อยู่ในกลุ่มข้อมูลเรียนรู้มาเป็นแผ่นแบบ โดยใช้การเรียงลำดับความสำคัญของข้อมูลอนุกรมเวลา ว่าข้อมูลอนุกรมเวลาตัวใดถูกเลือกให้เป็นเพื่อนบ้านใกล้สุดอันดับที่หนึ่งของข้อมูลอนุกรมเวลาตัวอื่น ๆ มากที่สุดในการทำการจำแนกข้อมูลแบบเพื่อนบ้านใกล้ที่สุดอันดับที่หนึ่งด้วยวิธีทดสอบแบบการนำออกหนึ่ง เพราะถือว่าข้อมูลตัวนั้นจะสามารถแทนข้อมูลตัวอื่น ๆ ในกลุ่มข้อมูลได้
3. *ข้อมูลอนุกรมเวลาทุกตัวที่อยู่ในกลุ่มข้อมูลเรียนรู้เป็นแผ่นแบบ* วิธีนี้เป็นวิธีที่ใช้กันมากที่สุดในปัจจุบัน คือให้เก็บข้อมูลในกลุ่มข้อมูลเรียนรู้ทุกตัวไว้เป็นแผ่นแบบ เมื่อจะทำการจำแนกประเภทข้อมูลของข้อมูลสอบถามก็จะทำการเปรียบเทียบระยะทางระหว่างข้อมูลสอบถามกับข้อมูลเรียนรู้ทุกตัว

โดยในส่วนของการทดลองในหัวข้อนี้จะวัดประสิทธิภาพด้านความแม่นยำในการจำแนกประเภทข้อมูล เปรียบเทียบระหว่างวิธีที่นำเสนอกับวิธีอื่น ๆ มาตรวัดที่ใช้ในการทดลองนี้คือค่าความแม่นยำ (Accuracy) ซึ่งสามารถคำนวณได้ตามสมการที่ (4.6)

$$\eta = \frac{n_p}{n_p + n_n} \times 100 \tag{4.6}$$

โดยที่ n_p และ n_n คือจำนวนข้อมูลอนุกรมเวลาที่จำแนกประเภทถูกต้อง และจำนวนข้อมูลอนุกรมเวลาที่จำแนกประเภทผิด ตามลำดับ

ตารางที่ 4.6 ผลการทดลองการเปรียบเทียบความแม่นยำในการจำแนกประเภทข้อมูลอนุกรมเวลาโดยใช้แผ่นแบบที่ได้จากวิธี ASA วิธี PSA วิธี AWARD และวิธี 1-NN ดั้งเดิมซึ่งใช้กลุ่มข้อมูลเรียนรู้เป็นแผ่นแบบ

ชุดข้อมูล	ความแม่นยำ (%)			
	ASA	PSA	AWARD	ข้อมูลเรียนรู้
50Words	**71.21**	40.66	0.44	69.00
Adiac	**70.84**	33.50	3.84	60.00
Beef	**50.00**	43.33	23.33	**50.00**
CBF	96.33	95.11	21.33	**99.70**
Coffee	**100.00**	**100.00**	46.43	82.00
ECG	**80.00**	61.00	76.00	77.00
Face (all)	**87.22**	62.19	3.20	80.80
Face (four)	76.14	67.50	18.18	**83.00**
Gun-Point	77.33	70.00	48.00	**90.70**
Lightning-2	67.38	65.67	39.34	**86.90**
Lightning-7	71.23	63.01	5.48	**72.60**
Oliveoil	**86.67**	76.67	10.00	**86.67**
OSU Leaf	58.26	25.51	16.53	**59.10**
Swedish Leaf	**88.96**	37.60	8.00	79.00
Synthetic Control	**97.33**	95.67	13.33	99.30
Trace	**100.00**	**100.00**	39.00	**100.00**
Two Patterns	**99.40**	93.00	32.20	**100.00**

จากผลการทดลองในตารางที่ 4.6 แสดงให้เห็นว่าการจำแนกประเภทข้อมูลอนุกรมเวลาโดยใช้แผ่นแบบที่ได้จากวิธีที่นำเสนอนั้นให้ค่าความถูกต้องแม่นยำสูงกว่าวิธีอื่น ๆ ซึ่งก็หมายถึงแผ่นแบบที่ได้จากวิธีที่นำเสนอนั้นสามารถแทนข้อมูลอนุกรมเวลาตัวอื่น ๆ ภายในกลุ่มข้อมูลเรียนรู้ได้ดีกว่าวิธีอื่น ๆ และเหตุผลที่ทำให้แผ่นแบบที่ได้จากวิธีที่นำเสนอดีกว่าวิธีอื่นที่นำมาเปรียบเทียบนั้น สามารถอธิบายได้ดังนี้ เนื่องจากแผ่นแบบที่ได้จากวิธี PSA นั้นมีข้อเสียในการลดจำนวนจุดข้อมูลให้กลับมามีจำนวนเท่ากับข้อมูลต้นแบบ ซึ่งไม่ได้ใช้วิธีการนำจุดข้อมูลอื่น ๆ มาคำนวณแต่เป็นการตัดจุดข้อมูลบางจุดทิ้งไป ซึ่งทำให้สูญเสียคุณลักษณะบางประการของข้อมูลอนุกรมเวลาต้นแบบไป และแผ่นแบบที่ได้จากวิธี AWARD นั้นเป็นเพียงการนำข้อมูลอนุกรมเวลาเพียงอนุกรมเดียวมาแทนข้อมูลอนุกรมเวลาทุกตัวในคลาส ซึ่งในความเป็นจริงแล้ว

การหยิบข้อมูลหนึ่งตัวเพื่อนำมาเป็นตัวแทนของคลาสโดยไม่ได้มีการนำข้อมูลตัวอื่น ๆ ในคลาสมาวิเคราะห์และคำนวณด้วยนั้น ข้อมูลอนุกรมเวลาเพียงตัวเดียวที่หยิบมาใช้นั้นก็ไม่สามารถแสดงลักษณะทั้งหมดของข้อมูลภายในคลาสตัวอื่น ๆ ได้ทั้งหมด และจากผลการทดลองจะเห็นว่าการจำแนกประเภทข้อมูลโดยใช้แผ่นแบบด้วยวิธีที่นำเสนอนั้นได้ผลของความถูกต้องแม่นยำเท่ากับหรือสูงกว่าการใช้ข้อมูลทั้งหมดในกลุ่มข้อมูลเรียนรู้เป็นแผ่นแบบถึง 10 ชุดข้อมูล ได้แก่ ชุดข้อมูล 50Words ชุดข้อมูล Adiac ชุดข้อมูล Beef ชุดข้อมูล Coffee ชุดข้อมูล ECG ชุดข้อมูล Face(all) ชุดข้อมูล Lightning7 ชุดข้อมูล Oliveoil ชุดข้อมูล Swedish Leaf และชุดข้อมูล Trace ซึ่งการใช้ข้อมูลทั้งหมดในกลุ่มการเรียนรู้เป็นแผ่นแบบนั้นน่าจะเป็นวิธีที่ให้ค่าความแม่นยำสูงที่สุด แต่การเก็บข้อมูลทุกตัวให้เป็นแผ่นแบบอาจทำให้เกิดการพอดีเกินไป (Overfitting) ของข้อมูลซึ่งเป็นผลทำให้เกิดการจำแนกประเภทข้อมูลผิด

ในส่วนต่อมาจะทำการทดลองเพื่อวัดประสิทธิภาพในด้านความเร็วจากการจำแนกประเภทข้อมูลอนุกรมเวลาโดยใช้แผ่นแบบด้วยวิธีที่ได้นำเสนอ โดยการทดลองทั้งหมดในหัวข้อนี้จะทำการดำเนินงานบนเครื่องคอมพิวเตอร์ที่ใช้ซีพียู AMD Athlon™ ความเร็ว 2.71 กิกะเฮิรตซ์ และใช้แรมขนาด 2 กิกะไบต์ งานทั้งหมดดำเนินงานภายใต้ระบบปฏิบัติการ Windows XP โดยใช้ชุดคำสั่งภาษา Matlab ทั้งหมด โดยเวลาที่ทำการวัดในการทดลองนี้คือเริ่มตั้งแต่การสร้างแผ่นแบบของแต่ละคลาสในชุดข้อมูลแล้วทำการจำแนกประเภทข้อมูลอนุกรมเวลาโดยเปรียบเทียบกับแผ่นแบบที่สร้างในแต่ละวิธี

ตารางที่ 4.7 ผลการทดลองการเปรียบเทียบเวลาในการจำแนกประเภทข้อมูลอนุกรมเวลา โดยใช้แผ่นแบบที่ได้จากวิธี ASA วิธี PSA วิธี AWARD และวิธี 1-NN ดั้งเดิมซึ่งใช้ข้อมูลทุกตัวในกลุ่มข้อมูลเรียนรู้เป็นแผ่นแบบ

ชุดข้อมูล	***เวลา (วินาที)***			
	ASA	***PSA***	***AWARD***	***ข้อมูลเรียนรู้***
50Words	**381.14**	4,585.90	567.06	4,167.87
Adiac	**83.29**	747.80	495.46	682.18
Beef	**10.55**	33.86	11.00	29.86
CBF	**9.55**	9.64	11.21	72.29
Coffee	5.91	10.36	**4.49**	10.00
ECG	**12.44**	16.07	12.81	15.91
Face (all)	**147.78**	912.74	809.54	2,565.54

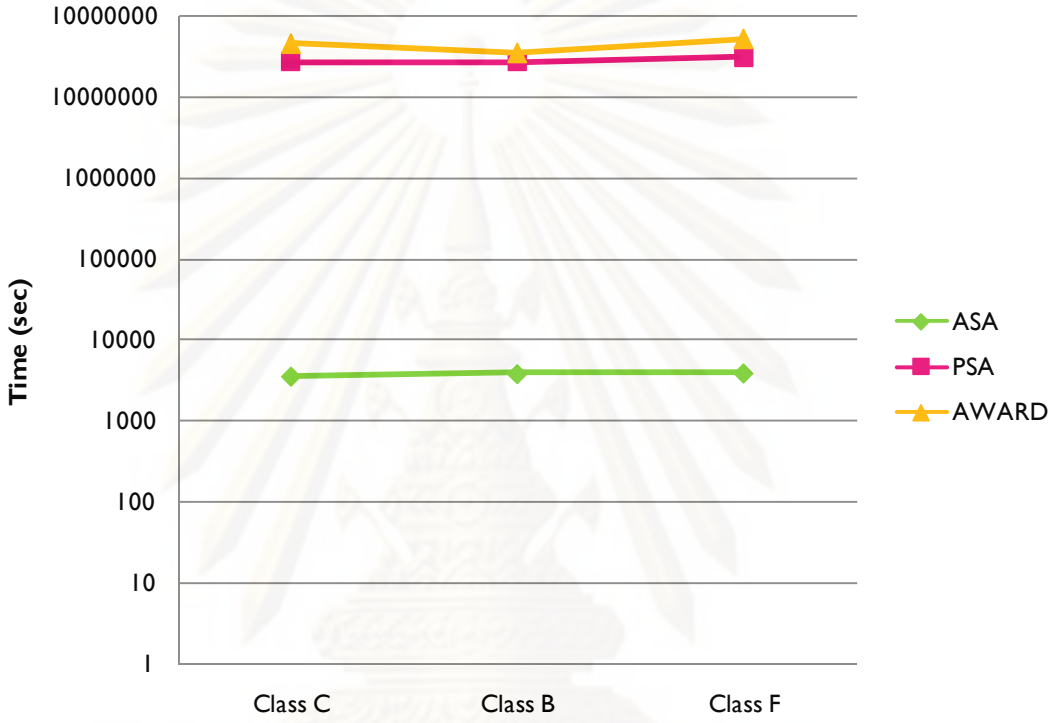
ชุดข้อมูล	*เวลา (วินาที)*			
	ASA	***PSA***	***AWARD***	***ข้อมูลเรียนรู้***
Face (four)	**9.79**	16.77	14.15	39.17
Gun-Point	6.52	9.68	**4.99**	26.40
Lightning-2	135.29	220.90	**87.77**	220.78
Lightning-7	**21.84**	84.19	35.37	80.55
Oliveoil	**17.98**	45.39	48.07	41.26
OSU Leaf	258.69	1,104.44	**225.74**	1,295.79
Swedish Leaf	**81.99**	655.05	342.99	802.81
Synthetic Control	**17.23**	67.26	82.91	66.16
Trace	37.67	123.79	**31.16**	120.18
Two Patterns	**903.49**	2659.73	1,041.60	10,481.66

จากผลการทดลองในตารางที่ 4.7 แสดงให้เห็นว่าในการจำแนกประเภทข้อมูลอนุกรมเวลาโดยการใช้แผ่นแบบที่สร้างจากวิธีที่นำเสนอนั้นมีประสิทธิภาพด้านเวลาสูงกว่าวิธีอื่น ๆ มีเพียงแค่ 5 ชุดข้อมูลเท่านั้นที่วิธี AWARD สามารถเอาชนะวิธีที่นำเสนอได้ เนื่องจากวิธี AWARD นั้นเป็นวิธีที่ผู้นำเสนอต้องการเพิ่มประสิทธิภาพในด้านความเร็วของการจำแนกประเภทข้อมูลเท่านั้น จากผลการทดลองด้านความแม่นยำในตารางที่ 4.6 จะเห็นว่าวิธี AWARD มีความแม่นยำต่ำที่สุดเมื่อเทียบกับวิธีอื่น ๆ ถ้าต้องการเพิ่มความแม่นยำให้กับแผ่นแบบที่ได้จากวิธีนี้ข้อมูลหนึ่งคลาสจะไม่สามารถแทนด้วยแผ่นแบบเพียงตัวเดียวได้ ส่วนสาเหตุหลักที่ทำให้การจำแนกประเภทข้อมูลโดยใช้วิธีที่นำเสนอนั้นเร็วกว่าวิธี PSA นั้นก็เพราะเวลาในการสร้างแผ่นแบบด้วยวิธีนี้นานมาก เพราะต้องนำข้อมูลอนุกรมเวลาในกลุ่มข้อมูลเรียนรู้มาทำการจัดกลุ่มข้อมูลแบบขั้นก่อนทำการหาค่าเฉลี่ยรูปร่าง ส่วนวิธีใช้ข้อมูลอนุกรมเวลาในกลุ่มข้อมูลเรียนรู้เป็นแผ่นแบบนั้นช้ากว่าวิธีที่นำเสนอเป็นเพราะจำนวนข้อมูลที่ต้องเปรียบเทียบระยะทางในแต่ละคลาสนั้นมากกว่าวิธีใช้แผ่นแบบที่นำเสนอ

4.5 การทดลองเพื่อวิเคราะห์ประสิทธิภาพด้านเวลาในการสร้างแผ่นแบบด้วยวิธีที่นำเสนอเปรียบเทียบกับวิธีอื่น ๆ

ในหัวข้อนี้จะทำการทดลองเพื่อวัดประสิทธิภาพในด้านความเร็วในการสร้างแผ่นแบบด้วยวิธีที่ได้นำเสนอ โดยการทดลองทั้งหมดในหัวข้อนี้จะทำการดำเนินงานบนเครื่องคอมพิวเตอร์ที่ใช้ซีพียู AMD Athlon™ ความเร็ว 2.71 กิกะเฮิรตซ์ และใช้แรมขนาด 2 กิกะไบต์

งานทั้งหมดดำเนินงานภายใต้ระบบปฏิบัติการ Windows XP โดยใช้ชุดคำสั่งภาษา Matlab ทั้งหมด โดยทำการทดลองกับชุดข้อมูลที่ได้จากการสังเคราะห์ เนื่องจากต้องการชุดข้อมูลที่มีขนาดใหญ่คือมีจำนวนข้อมูลในกลุ่มข้อมูลเรียนรู้เท่ากับ 300,000 อนุกรม โดยข้อมูลที่นำมาทดสอบนี้ได้แสดงรายละเอียดไว้ในหัวข้อที่ 4.1.2 การทดลองนี้จะทำการเปรียบเทียบเวลาที่ใช้ในการสร้างแผ่นแบบของแต่ละคลาสของชุดข้อมูลซีบีเอฟที่ได้ทำการสังเคราะห์ขึ้นมา โดยเปรียบเทียบวิธีที่นำเสนอกับวิธี PSA และวิธี AWARD โดยผลการทดลองแสดงในรูปที่ 4.3



รูปที่ 4.3 ผลการทดลองเปรียบเทียบเวลาในการสร้างแผ่นแบบของข้อมูลสามแสนอนุกรมด้วยวิธี ASA วิธี PSA และวิธี AWARD

จากผลการทดลองในรูปที่ 4.3 แสดงให้เห็นว่าการสร้างแผ่นแบบด้วยวิธีที่นำเสนอนั้นใช้เวลาในการสร้างแผ่นแบบน้อยที่สุด โดยใช้เวลาในการสร้างน้อยกว่าวิธีอื่น ๆ ถึง 100,000 เท่า เนื่องจากการสร้างแผ่นแบบด้วยวิธีที่นำเสนอนั้นใช้ฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปปิงที่ใช้การคำนวณแบบยุคลิดเพื่อจัดลำดับในการหาค่าเฉลี่ยรูปร่าง ทำให้สามารถลดปริมาณการคำนวณไดนามิกไทม์วอร์ปปิงได้ ในขณะที่วิธี PSA นั้นต้องคำนวณไดนามิกไทม์วอร์ปปิงทั้งหมดถึง 9,999,900,000 ครั้งต่อหนึ่งคลาสเพื่อทำการจัดกลุ่มข้อมูลแบบขั้นในการเรียงลำดับการคำนวณค่าเฉลี่ยรูปร่าง ส่วนวิธี AWARD นั้นต้องทำการจำแนกข้อมูลแบบเพื่อนบ้านใกล้ที่สุดอันดับที่หนึ่งด้วยวิธีทดสอบแบบการนำออกหนึ่งและยังต้องคำนวณหาความกว้างของเงื่อนไขบังคับโดยรวมที่เหมาะสมกับชุดของข้อมูลมากที่สุดทำให้ใช้เวลาในการสร้างแผ่นแบบนานกว่าวิธี ASA ที่ได้นำเสนอในงานวิจัยนี้

4.6 การวิเคราะห์ข้อมูลเพื่อแบ่งคลาสก่อนทำการสร้างแผ่นแบบ

แผ่นแบบที่ได้จากงานวิจัยนี้ ข้อมูลอนุกรมเวลาหนึ่งคลาสจะถูกแทนด้วยแผ่นแบบหรือตัวแทนกลุ่มเพียงอนุกรมเดียวเท่านั้น อย่างไรก็ตามชุดข้อมูลอนุกรมเวลาจริงที่พบเห็นได้ทั่วไปนั้นอาจมีข้อมูลอนุกรมเวลาบางประเภทที่ภายในหนึ่งคลาสไม่ได้มีข้อมูลอนุกรมเวลาที่มีรูปร่างเพียงแบบเดียว ทำให้เมื่อสร้างแผ่นแบบโดยลดจำนวนข้อมูลอนุกรมเวลาลงเหลือเพียงอนุกรมเดียว แผ่นแบบที่ได้อาจจะไม่มีประสิทธิภาพ ดังนั้นเพื่อเป็นการแก้ปัญหาดังกล่าวผู้วิจัยจึงได้เสนอวิธีการแบ่งคลาสย่อยก่อนทำการสร้างแผ่นแบบด้วยวิธีที่นำเสนอ ในส่วนของการแบ่งคลาสย่อยก่อนทำการสร้างแผ่นแบบด้วยวิธีที่นำเสนอนั้นประกอบด้วย 2 ขั้นตอนดังนี้

4.6.1 การแบ่งข้อมูลอนุกรมเวลาออกเป็นคลาสย่อย

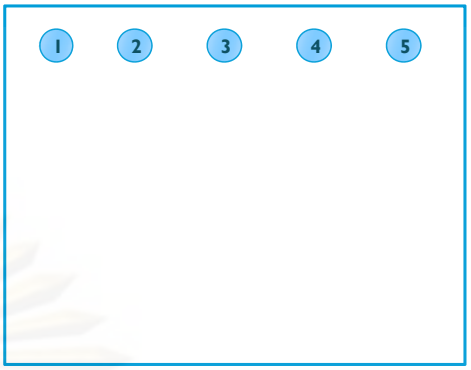
การแบ่งข้อมูลอนุกรมเวลาที่อยู่ในคลาสออกเป็นคลาสย่อย ๆ นั้น ในงานวิจัยนี้จะใช้ความสัมพันธ์ของระยะทางระหว่างข้อมูลอนุกรมเวลาแต่ละอนุกรมเป็นตัวแบ่งคลาสย่อยของข้อมูล เนื่องจากวิธีการสร้างแผ่นแบบตามที่ได้เสนอนั้นการหาลำดับในการหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลานั้นได้มีการคำนวณหาระยะทางระหว่างอนุกรมเวลาที่น้อยที่สุดทุกตัวไว้แล้ว วิธีการแบ่งคลาสย่อยนั้นจึงจะใช้ระยะทางที่ได้คำนวณไว้แล้วนั้นมาใช้ในการแบ่งข้อมูล ดังแสดงในรูปที่ 4.4

ในรูปที่ 4.4 แสดงขั้นตอนวิธีในการแบ่งคลาสย่อยด้วยวิธีที่นำเสนอ รูปที่ 4.4 ก) แสดงตารางข้อมูลอนุกรมเวลาแต่ละอนุกรมว่ามีระยะทางใกล้กับข้อมูลอนุกรมเวลาตัวใดมากที่สุดและมีระยะทางห่างกันเท่าใด โดยมีรายละเอียดดังนี้ ข้อมูลอนุกรมเวลาตัวที่ 1 มีระยะทางใกล้กับตัวที่ 3 มากที่สุดโดยมีระยะทางเท่ากับ 10 ข้อมูลอนุกรมเวลาตัวที่ 2 มีระยะทางใกล้กับตัวที่ 3 มากที่สุดโดยมีระยะทางเท่ากับ 5 ข้อมูลอนุกรมเวลาตัวที่ 3 มีระยะทางใกล้กับตัวที่ 2 มากที่สุดโดยมีระยะทางเท่ากับ 5 ข้อมูลอนุกรมเวลาตัวที่ 4 มีระยะทางใกล้กับตัวที่ 5 มากที่สุดโดยมีระยะทางเท่ากับ 3 และข้อมูลอนุกรมเวลาตัวที่ 5 มีระยะทางใกล้กับตัวที่ 4 มากที่สุดโดยมีระยะทางเท่ากับ 3 รูปที่ 4.4 ข) แสดงข้อมูลอนุกรมเวลาตัวที่ 1 ถึง 5 รูปที่ 4.4 ค) นำข้อมูลอนุกรมเวลาคู่ที่มีระยะทางน้อยที่สุดคือข้อมูลอนุกรมเวลาตัวที่ 4 และตัวที่ 5 ให้อยู่ในคลาสย่อยเดียวกัน จากนั้นนำคู่ของข้อมูลอนุกรมเวลาที่มีระยะทางน้อยรองลงมาคือข้อมูลอนุกรมเวลาตัวที่ 2 และตัวที่ 3 ให้อยู่ในกลุ่มเดียวกัน ดังแสดงในรูปที่ 4.4 ง) ต่อมานำข้อมูลอนุกรมเวลาคู่ที่มีระยะทางน้อยลงมาคือข้อมูลอนุกรมเวลาตัวที่ 1 และตัวที่ 3 มาอยู่ในคลาสย่อยเดียวกัน ซึ่งจะเห็นว่าข้อมูลอนุกรมเวลาตัวที่ 3 ถูกจัดให้อยู่ในคลาสย่อยกับตัวที่ 2 แล้ว ดังนั้นข้อมูลอนุกรมเวลาตัวที่ 1 จะถูกจัดให้อยู่ในคลาสย่อยเดียวกับข้อมูลอนุกรมเวลาตัวที่ 2 และตัวที่ 3 ดังแสดงในรูปที่ 4.4 จ) ข้อมูลในคลาสเดิมสามารถแบ่งออกเป็น 2 คลาสย่อยโดยข้อมูลอนุกรมเวลาตัวที่ 4 และ 5 อยู่ในคลาสย่อยเดียวกัน ส่วนข้อมูลอนุกรมเวลาตัวที่ 1 2 และ 3 อยู่ในคลาสย่อยเดียวกัน ดังแสดงในรูปที่ 4.4 ช)

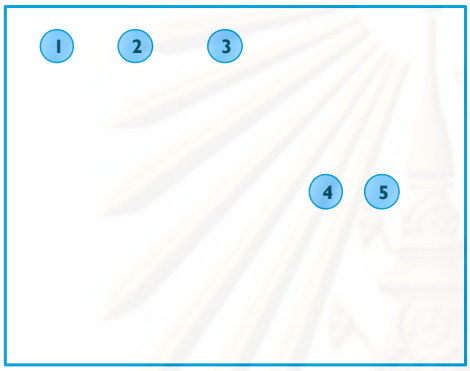
ก)

i	j	distance
1	3	10
2	3	5
3	2	5
4	5	3
5	4	3

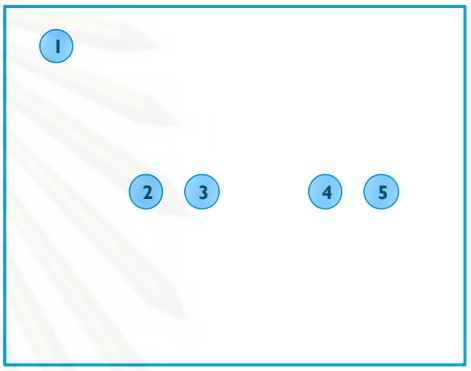
ข)



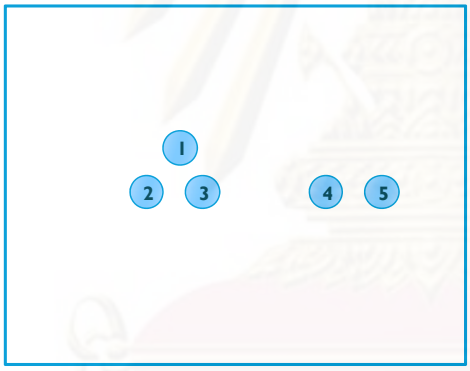
ค)



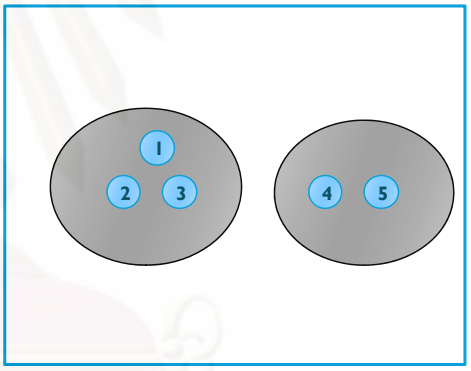
ง)



จ)



ช)



รูปที่ 4.4 ภาพรวมขั้นตอนวิธีในการแบ่งคลาสย่อย

4.6.2 การรวมคลาสย่อย

เมื่อสามารถแยกข้อมูลอนุกรมเวลาที่อยู่ในคลาสออกเป็นคลาสย่อยตามที่อธิบายรายละเอียดในหัวข้อที่ 4.6.1 ในหัวข้อนี้จะทำการพิจารณาว่าคลาสย่อยที่แบ่งได้ควรจะนำมารวมกันหรือไม่ โดยสิ่งที่นำมาพิจารณาในการรวมกันของคลาสย่อยนั้นก็คือค่าความแปรปรวน (Variance-SD^2) ของระยะทางระหว่างข้อมูลอนุกรมเวลาภายในคลาส

ความแปรปรวนเป็นการวัดการกระจายของข้อมูล คือถ้าค่าการกระจายมากแสดงว่าข้อมูลมีความแตกต่างกันมาก โดยค่าความแปรปรวน SD^2 สามารถคำนวณได้ตามสมการที่ (4.7)

$$SD^2 = \sqrt{\frac{\sum_{i=1}^{n}(d_i - \bar{d})^2}{n}} \tag{4.7}$$

โดยที่ d_i และ $\bar{d}$ คือระยะทางระหว่างข้อมูลอนุกรมเวลาทุกคู่ที่เป็นไปได้และค่าเฉลี่ยเลขคณิตของระยะทางทั้งหมด ตามลำดับ และ n คือจำนวนระยะทางที่เป็นไปได้ทั้งหมด

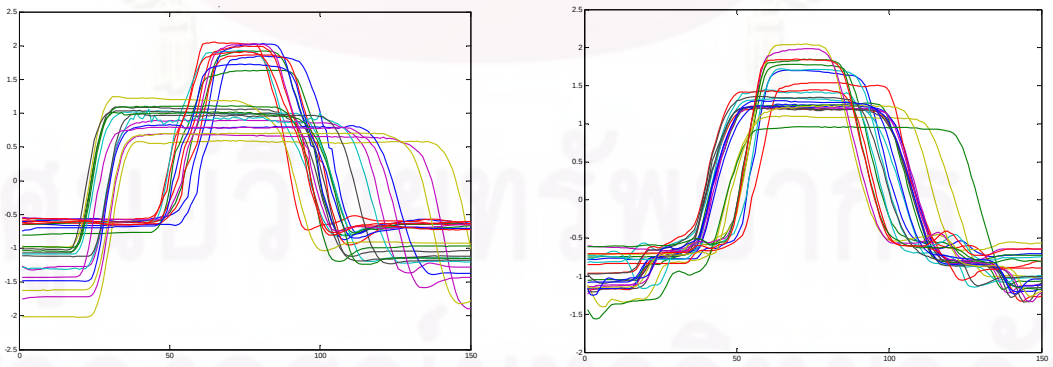
วิธีการรวมคลาสย่อย คือคำนวณระยะทางระหว่างข้อมูลอนุกรมเวลาที่อยู่ภายในคลาสย่อยเดียวกันด้วยวิธีไดนามิกไทม์วอร์ปปิง จากนั้นนำค่าระยะทางที่ได้มาคำนวณหาความแปรปรวนของระยะทางแต่ละคลาสย่อย โดยที่ในการนำข้อมูลแต่ละคลาสย่อยมารวมกันนั้นค่าความแปรปรวนที่ได้หลังจากรวมคลาสย่อยจะต้องไม่ทำให้ค่าความแปรปรวนเฉลี่ยของคลาสย่อยที่นำมารวมกันเพิ่มขึ้น

4.6.3 การทดลองเพื่อวิเคราะห์การแบ่งคลาสย่อยของข้อมูลด้วยวิธีที่นำเสนอ

ในส่วนนี้ทำการทดลองเพื่อวิเคราะห์ประสิทธิภาพของวิธีการแบ่งคลาสย่อยด้วยวิธีที่นำเสนอ โดยวิธีการทดสอบนำข้อมูลจริงที่ภายในหนึ่งคลาสไม่ได้มีข้อมูลเพียงรูปแบบเดียวเพื่อทำการแบ่งคลาสย่อย จากนั้นสร้างแผ่นแบบแต่ละคลาสย่อยเพื่อนำแผ่นแบบที่ได้ไปทดสอบความแม่นยำในการจำแนกประเภทข้อมูลอนุกรมเวลา โดยที่จะทำการทดลองกับชุดข้อมูลจริง 2 ชุด ดังนี้

4.6.3.1 ชุดข้อมูล Gun-Point

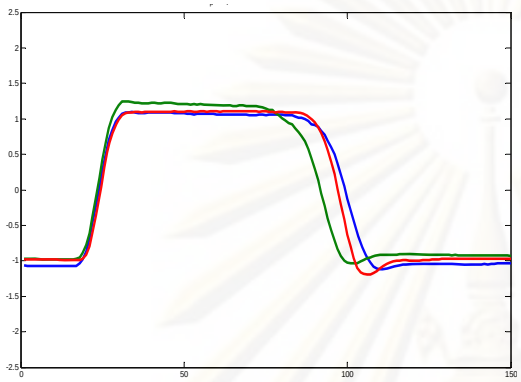
ชุดข้อมูล Gun-Point ประกอบด้วยข้อมูลอนุกรมเวลา 2 คลาส ดังแสดงในรูปที่ 4.5



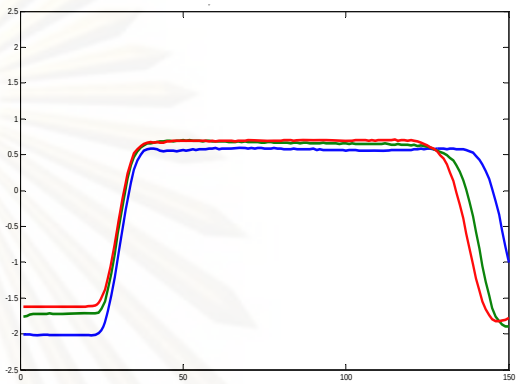
คลาสที่ 1 คลาสที่ 2

รูปที่ 4.5 ชุดข้อมูล Gun-Point แบ่งเป็น 2 คลาส

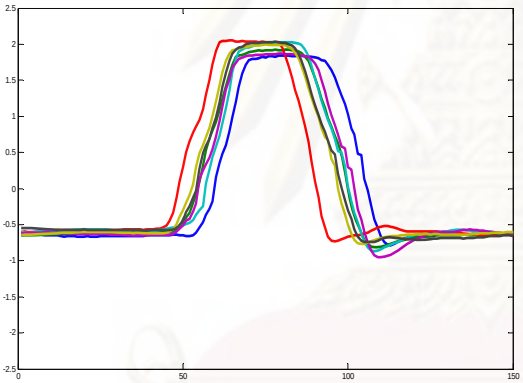
ในรูปที่ 4.6 แสดงให้เห็นว่าวิธีการแบ่งคลาสย่อยด้วยวิธีที่นำเสนอสามารถแบ่งคลาสย่อยของชุดข้อมูล Gun-Point ออกได้เป็น 6 คลาสย่อย ดังนี้ คลาสที่ 1 สามารถแบ่งได้เป็น 2 คลาสย่อย และคลาสที่ 2 แบ่งได้ 4 คลาสย่อยหลังจากทำการแบ่งคลาสย่อยด้วยวิธีที่นำเสนอแล้ว ก็สามารถสร้างแผ่นแบบด้วยวิธี ASA โดยให้หนึ่งคลาสย่อยมีหนึ่งแผ่นแบบ เมื่อนำแผ่นแบบของแต่ละคลาสย่อยมาใช้เป็นตัวแทนในการจำแนกประเภทข้อมูลอนุกรมเวลาเพื่อวัดประสิทธิภาพด้านความแม่นยำของแผ่นแบบที่ได้



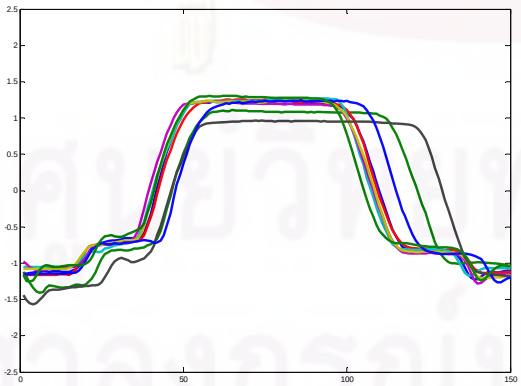
คลาสที่ 1 - คลาสย่อย 1



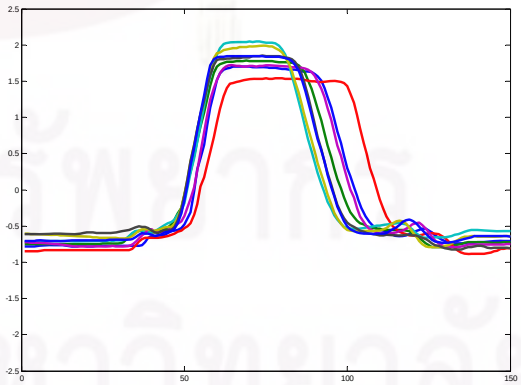
คลาสที่ 1 - คลาสย่อย 2



คลาสที่ 1 - คลาสย่อย 3



คลาสที่ 2 - คลาสย่อย 1



คลาสที่ 2 - คลาสย่อย 1

รูปที่ 4.6 ผลการทดลองในการแบ่งข้อมูลในคลาสออกเป็นคลาสย่อยของชุดข้อมูล Gun-Point

การทดสอบประสิทธิภาพด้านความแม่นยำของแผ่นแบบที่ได้จากการแบ่งคลาสย่อยของชุดข้อมูล Gun-Point เพื่อใช้เป็นตัวแทนกลุ่มในการจำแนกประเภทข้อมูล โดยผลลัพธ์ที่ได้แสดงในตารางที่ 4.8

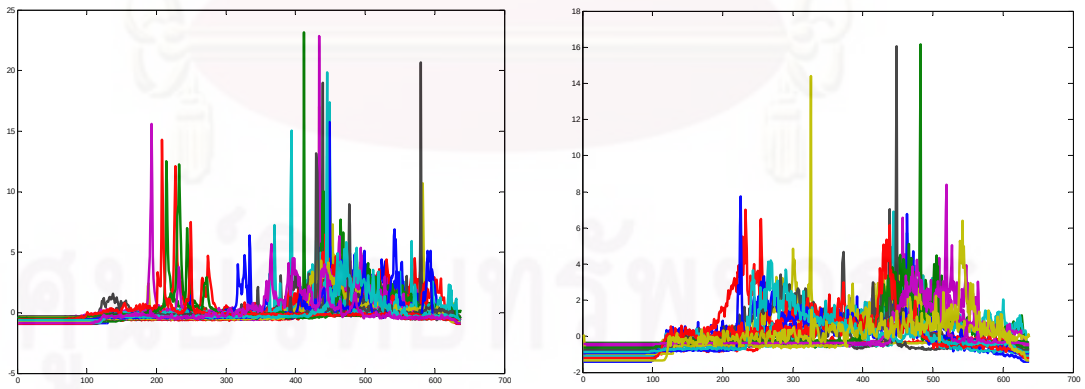
ตารางที่ 4.8 ผลการทดลองเปรียบเทียบความแม่นยำในการจำแนกประเภทข้อมูลระหว่างแผ่นแบบด้วยวิธีที่นำเสนอ โดยเปรียบเทียบระหว่างแบบที่แบ่งเป็นคลาสย่อยและแบบที่หนึ่งคลาสแทนด้วยแผ่นแบบเพียงตัวเดียว

ชุดข้อมูล	***ความแม่นยำ (%)***	
	คลาสย่อย	***หนึ่งแผ่นแบบ/คลาส***
Gun-Point	**85.33**	72.67

จากผลการทดลองในรูปที่ 4.6 และในตารางที่ 4.8 แสดงให้เห็นว่าวิธีในการแบ่งคลาสย่อยด้วยวิธีที่นำเสนอสามารถแบ่งข้อมูลอนุกรมเวลาที่อยู่ในคลาสเดียวกันออกเป็นคลาสย่อย ๆ ได้ โดยที่รูปร่างลักษณะของข้อมูลอนุกรมเวลาแต่ละคลาสย่อยนั้นมีเพียงลักษณะเดียวเท่านั้น และเมื่อแบ่งคลาสย่อยแล้วสร้างแผ่นแบบด้วยวิธีที่ได้นำเสนอในหัวข้อที่ 3.3 สามารถใช้แทนข้อมูลอนุกรมเวลาแต่ละคลาสย่อยในการจำแนกประเภทข้อมูลได้เป็นอย่างดี

4.6.3.2 ชุดข้อมูล Lightning2

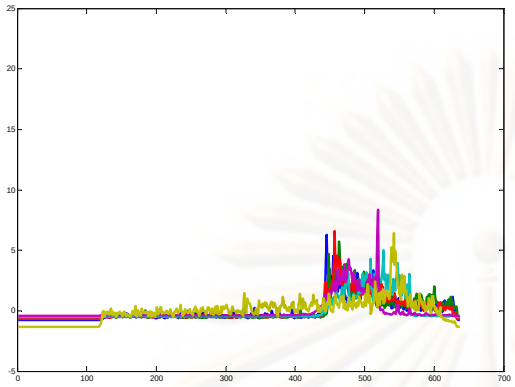
ชุดข้อมูล Lightning2 ประกอบด้วยข้อมูลอนุกรมเวลา 2 คลาส ดังแสดงในรูปที่ 4.7



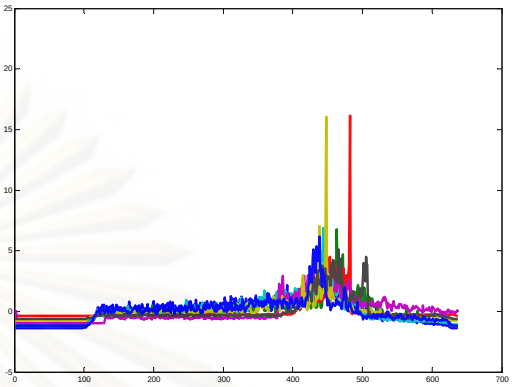
คลาสที่ 1 คลาสที่ 2

รูปที่ 4.7 ชุดข้อมูล Lightning2 แบ่งเป็น 2 คลาส

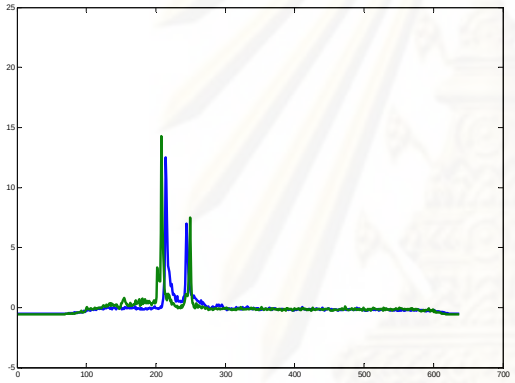
ในรูปที่ 4.8 แสดงให้เห็นว่าวิธีการแบ่งคลาสย่อยด้วยวิธีที่นำเสนอสามารถแบ่งคลาสย่อยของชุดข้อมูล Lightning2 ออกได้เป็น 6 คลาสย่อย ดังนี้ คลาสที่ 1 สามารถแบ่งได้เป็น 2 คลาสย่อย และคลาสที่ 2 แบ่งได้ 4 คลาสย่อย



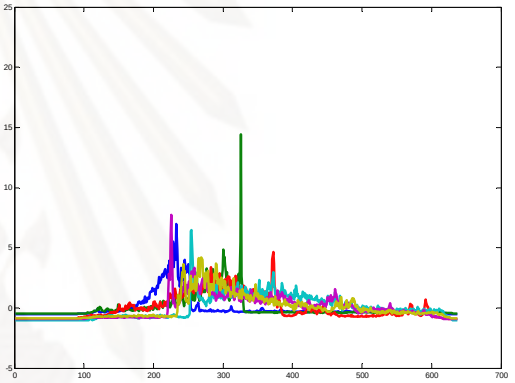
คลาสที่ 1 - คลาสย่อย 1



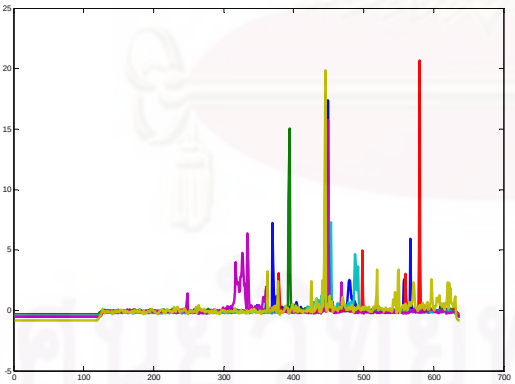
คลาสที่ 1 - คลาสย่อย 2



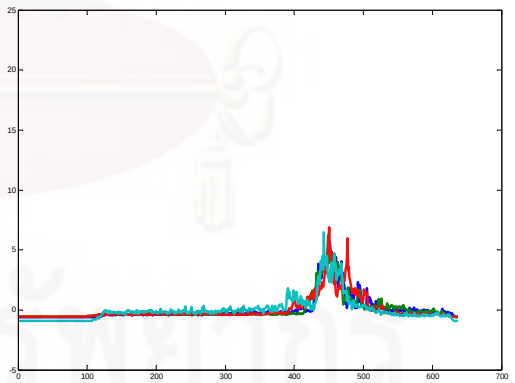
คลาสที่ 2 - คลาสย่อย 1



คลาสที่ 2 - คลาสย่อย 2



คลาสที่ 2 - คลาสย่อย 3



คลาสที่ 2 - คลาสย่อย 4

รูปที่ 4.8 ผลการทดลองในการแบ่งข้อมูลในคลาสออกเป็นคลาสย่อยของชุดข้อมูล Lightning2

การทดสอบประสิทธิภาพด้านความแม่นยำของแผ่นแบบที่ได้จากการแบ่งคลาสย่อยของชุดข้อมูล Lightning2 เพื่อใช้เป็นตัวแทนกลุ่มในการจำแนกประเภทข้อมูล โดยผลลัพธ์ที่ได้แสดงในตารางที่ 4.9

ตารางที่ 4.9 ผลการทดลองเปรียบเทียบความแม่นยำในการจำแนกประเภทข้อมูลระหว่างแผ่นแบบด้วยวิธีที่นำเสนอ โดยเปรียบเทียบระหว่างแบบที่แบ่งเป็นคลาสย่อยและแบบที่หนึ่งคลาสแทนด้วยแผ่นแบบเพียงตัวเดียว

ชุดข้อมูล	***ความแม่นยำ (%)***	
	คลาสย่อย	***หนึ่งแผ่นแบบ/คลาส***
Lightning2	**65.57**	59.02

จากผลการทดลองในตารางที่ 4.9 แสดงให้เห็นว่าแผ่นแบบที่ได้จากการแบ่งคลาสย่อยด้วยวิธีที่นำเสนอ สามารถให้ค่าความแม่นยำเอาชนะแผ่นแบบที่มีเพียงหนึ่งตัวต่อหนึ่งคลาส เนื่องจากในรูปที่ 4.7 จะเห็นว่าข้อมูลอนุกรมเวลาในแต่ละคลาสมีรูปแบบที่หลากหลาย เมื่อหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลาทั้งกลุ่มให้เหลือเพียงตัวเดียวทำให้ข้อมูลที่เป็นแผ่นแบบทั้งสองคลาสอาจมีความคล้ายคลึงกันมาก เมื่อนำแผ่นแบบที่ได้มาเป็นตัวแทนเพื่อใช้ในการจำแนกประเภทข้อมูลทำให้ได้ค่าความแม่นยำต่ำ แต่เมื่อทำการวิเคราะห์ข้อมูลเพื่อแบ่งคลาสย่อยก่อนทำการสร้างแผ่นแบบ ข้อมูลอนุกรมเวลาที่มีรูปร่างต่างกันนั้นจะถูกแยกให้อยู่คนละคลาสย่อย ทำให้เมื่อสร้างแผ่นแบบเพื่อเป็นตัวแทนกลุ่มในการจำแนกประเภทข้อมูลก็จะสามารถหาตัวแทนกลุ่มที่มีประสิทธิภาพและสามารถแทนข้อมูลที่อยู่ในคลาสย่อยได้เป็นอย่างดี

4.6.3.3 การทดลองเพื่อวิเคราะห์ประสิทธิภาพในด้านความแม่นยำของวิธีที่นำเสนอระหว่างแบบหลาย ๆ แผ่นแบบต่อหนึ่งคลาสกับแบบหนึ่งแผ่นแบบต่อหนึ่งคลาส

ในส่วนนี้จะทำการทดลองเพื่อวิเคราะห์ว่าการสร้างแผ่นแบบสำหรับกลุ่มข้อมูลอนุกรมเวลาด้วยแผ่นแบบหลาย ๆ แผ่นแบบต่อหนึ่งคลาสจะสามารถเพิ่มความแม่นยำในการจำแนกประเภทข้อมูลได้หรือไม่ เมื่อเปรียบเทียบกับหนึ่งคลาสแทนด้วยแผ่นแบบหนึ่งอันเท่านั้น โดยจะทดลองกับชุดข้อมูลจริงในตารางที่ 4.1 ซึ่งผลการทดลองแสดงในตารางที่ 4.10

ตารางที่ 4.10 ผลการทดลองเปรียบเทียบความแม่นยำในการจำแนกประเภทข้อมูลระหว่างแผ่นแบบด้วยวิธีที่นำเสนอ โดยเปรียบเทียบระหว่างแบบที่หนึ่งคลาสมีหลายแผ่นแบบและแบบที่หนึ่งคลาสแทนด้วยแผ่นแบบเพียงตัวเดียว

ชุดข้อมูล	***ความแม่นยำ (%)***	
	หลายแผ่นแบบ/คลาส	***หนึ่งแผ่นแบบ/คลาส***
50Words	**72.00**	71.21

ชุดข้อมูล	*ความแม่นยำ (%)*	
	หลายแผ่นแบบ/คลาส	*หนึ่งแผ่นแบบ/คลาส*
Adiac	**74.10**	70.84
Beef	**50.00**	**50.00**
CBF	**99.70**	96.33
Coffee	**100.00**	**100.00**
ECG	76.00	**80.00**
Face (all)	**89.35**	87.22
Face (four)	**79.50**	76.14
Gun-Point	**85.33**	77.33
Lightning-2	**68.57**	67.38
Lightning-7	**80.82**	79.45
Oliveoil	76.67	**86.67**
OSU Leaf	57.85	**58.26**
Swedish Leaf	84.80	**88.96**
Synthetic Control	**99.67**	97.33
Trace	**100.00**	**100.00**
Two Patterns	**100.00**	99.40

จากผลการทดลองในตารางที่ 4.10 แสดงให้เห็นว่าการสร้างแผ่นแบบด้วยวิธีที่นำเสนอ โดยการสร้างแผ่นแบบมากกว่าหนึ่งแผ่นแบบต่อหนึ่งคลาสทำให้ความแม่นยำในการจำแนกประเภทข้อมูลสูงขึ้น อย่างไรก็ดียังมีบางชุดข้อมูล เช่น ชุดข้อมูล ECG ชุดข้อมูล Oliveoil ชุดข้อมูล OSU Leaf และชุดข้อมูล Swedish Leaf ที่มีจำนวนแผ่นแบบหนึ่งแผ่นแบบต่อหนึ่งคลาสมีความแม่นยำสูงกว่าแผ่นแบบหลาย ๆ แผ่นแบบต่อหนึ่งคลาส สาเหตุก็เนื่องมาจากการที่มีจำนวนแผ่นแบบมากเกินไปในบางชุดข้อมูลทำให้เกิดการพอดีเกินไปของข้อมูลซึ่งเป็นผลทำให้เกิดการจำแนกประเภทข้อมูลผิด

บทที่ 5

สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

งานวิจัยนี้ ได้นำเสนอวิธีการสร้างแผ่นแบบสำหรับกลุ่มข้อมูลอนุกรมเวลา ที่มีการประยุกต์ใช้เทคนิคการหาค่าเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลาโดยอาศัยการปรับแนวแบบผสมระหว่างไทม์วอร์ปปิงกับไทม์วอร์ปปิงแบบอนุพันธ์ และฟังก์ชันกระดูกงูกำลังสาม เพื่อที่จะให้วิธีการที่นำเสนอสามารถสร้างแผ่นแบบสำหรับกลุ่มข้อมูลอนุกรมเวลาเพื่อใช้เป็นตัวแทนในการจำแนกประเภทข้อมูลอนุกรมเวลาได้อย่างรวดเร็วและถูกต้องแม่นยำ โดยได้มีการทดลองและวิเคราะห์ผลไว้อย่างละเอียดดังที่ได้นำเสนอในบทที่ 4 ซึ่งจะเห็นได้ว่าวิธีการที่นำเสนอนั้นสามารถสร้างแผ่นแบบที่เป็นตัวแทนกลุ่มของข้อมูลอนุกรมเวลาได้ และเมื่อนำมาใช้เป็นตัวแทนในการจำแนกประเภทข้อมูลก็สามารถจำแนกประเภทข้อมูลได้อย่างรวดเร็วและแม่นยำ โดยผลจากการวิจัยทั้งหมดที่ได้นำเสนอไปนั้น สามารถสรุปได้ดังนี้

5.1 สรุปผลการวิจัย

ในการจำแนกประเภทข้อมูลอนุกรมเวลาโดยใช้วิธีวัดระยะทางแบบไดนามิกไทม์วอร์ปปิงนั้น ปัญหาเรื่องจำนวนของข้อมูลอนุกรมเวลาในกลุ่มข้อมูลเรียนรู้ถือว่าเป็นปัญหาที่สำคัญมาก เนื่องจากถ้ามีข้อมูลอนุกรมเวลาในกลุ่มข้อมูลเรียนรู้เป็นจำนวนมาก การจำแนกประเภทข้อมูลอนุกรมเวลาจะต้องเสียเวลามากไปกับการคำนวณไดนามิกไทม์วอร์ปปิง เนื่องจากมีขีดจำกัดเชิงสัญกรณ์ในการคำนวณระยะทางสูงถึง $O(n^2)$ แต่ด้วยการแทนที่การคำนวณไดนามิกไทม์วอร์ปปิงด้วยค่าระยะทางขอบเขตล่างจากวิธีที่มีอยู่ในปัจจุบันนั้น อาจกล่าวได้ว่าขีดจำกัดเชิงสัญกรณ์นั้นแทบจะลดเหลือเพียงฟังก์ชันเชิงเส้นเท่านั้น อย่างไรก็ดีแม้จะสามารถคำนวณระยะทางได้อย่างรวดเร็ว แต่ก็ยังมีปัญหาเรื่องหน่วยเก็บข้อมูลเพราะถ้ากลุ่มข้อมูลมีปริมาณใหญ่มากอาจทำให้ไม่สามารถทำงานได้ เนื่องจากพื้นที่ในการเก็บข้อมูลไม่เพียงพอ ดังนั้นวิธีการลดจำนวนข้อมูลในกลุ่มข้อมูลเรียนรู้จึงเป็นวิธีที่เหมาะสมในการแก้ปัญหาทั้งในด้านความเร็วในการจำแนกประเภทข้อมูลและพื้นที่ในการเก็บข้อมูล โดยวิธีการลดจำนวนข้อมูลอนุกรมเวลาในงานวิจัยนี้คือ การสร้างแผ่นแบบสำหรับข้อมูลแต่ละคลาส

วิธีการสร้างแผ่นแบบที่ได้พัฒนาให้เหมาะสำหรับข้อมูลอนุกรมเวลาที่ได้กล่าวไว้ทั้งหมด จะเห็นได้ว่า งานวิจัยนี้สามารถให้ผลของประสิทธิภาพความแม่นยำและความเร็วในการจำแนกประเภทข้อมูลอนุกรมเวลาได้อย่างมีประสิทธิภาพกับทุกชุดข้อมูลที่ทดสอบ แม้ว่าในบางชุดข้อมูลจะมีประสิทธิภาพของความแม่นยำด้อยกว่าการใช้ข้อมูลอนุกรมเวลาทุกตัวในกลุ่มข้อมูลเรียนรู้เป็นแผ่นแบบได้ แต่การใช้แผ่นแบบเพียงตัวเดียวเพื่อแทนข้อมูลอนุกรมเวลาทั้งกลุ่มก็ถือเป็นการลดจำนวนข้อมูลลงมากซึ่งอาจทำให้สูญเสียคุณลักษณะบางประการของข้อมูล

อนุกรมเวลาไปบ้าง การนำไปเปรียบเทียบกับวิธีที่ใช้ข้อมูลทุกตัวเป็นแผ่นแบบนั้นถือว่าเป็นงานที่ยากและท้าทาย แต่จะพบว่าโดยส่วนมากความแม่นยำที่ได้จากการใช้แผ่นแบบด้วยวิธีที่นำเสนอเพื่อเป็นตัวแทนในการจำแนกประเภทของข้อมูลอนุกรมเวลาก็ได้ผลลัพธ์ใกล้เคียงกับการใช้ข้อมูลอนุกรมเวลาทุกตัวเป็นแผ่นแบบ และในบางผลการทดลองแผ่นแบบที่สร้างจากวิธีที่นำเสนอก็สามารถเอาชนะทุกวิธีที่นำมาเปรียบเทียบได้

ในส่วนของงานวิจัยอื่น ๆ ที่นำมาเปรียบเทียบ เช่น การสร้างแผ่นแบบด้วยวิธี PSA และแผ่นแบบที่ได้จากวิธี AWARD งานวิจัยที่นำเสนอนี้สามารถเอาชนะทั้งในด้านความแม่นยำและความเร็วได้ทั้งหมด ถึงแม้ว่าแผ่นแบบที่ได้จากทั้งสองวิธีจะมีเพียงตัวเดียวต่อหนึ่งคลาสเช่นเดียวกับวิธีที่ได้นำเสนอ แต่ในการสร้างแผ่นแบบใช้เวลานานมาก เนื่องจากทั้งสองวิธีต้องคำนวณไดนามิกไทม์วอร์ปปิงถึง $(n)\times(n\text{-}1)$ ครั้ง เมื่อ n คือข้อมูลอนุกรมเวลาในกลุ่มข้อมูลเรียนรู้ ซึ่งแสดงได้ดังผลการทดลองที่ 4.5 ซึ่งจากการทดลองทั้งหมดพอจะสรุปได้ว่า การสร้างแผ่นแบบด้วยวิธี ASA สามารถแทนข้อมูลอนุกรมเวลาในกลุ่มข้อมูลเรียนรู้ได้ดี ซึ่งทำให้ผลลัพธ์ในการจำแนกประเภทข้อมูลทั้งด้านความแม่นยำและความเร็วมีประสิทธิภาพเมื่อเทียบกับงานวิจัยอื่น ๆ

นอกจากนั้นเพื่อเป็นการเพิ่มประสิทธิภาพของแผ่นแบบด้วยวิธีที่นำเสนอยังสามารถนำข้อมูลอนุกรมเวลามาวิเคราะห์ก่อนการสร้างแผ่นแบบ เพื่อรองรับกับชุดข้อมูลอนุกรมเวลาที่หนึ่งคลาสไม่ได้มีข้อมูลเพียงรูปแบบเดียว โดยทำการแบ่งคลาสย่อยก่อนทำการสร้างแผ่นแบบ งานวิจัยนี้ได้นำเสนอการแบ่งคลาสย่อยด้วยค่าระยะทางระหว่างข้อมูลอนุกรมเวลาและรวมคลาสย่อยโดยใช้ค่าความแปรปรวนของระยะทางในแต่ละคลาสย่อยเป็นเกณฑ์ ซึ่งวิธีการดังกล่าวนั้นสามารถทำก่อนการสร้างแผ่นแบบด้วยวิธีที่นำเสนอได้เลย โดยมิได้มีความยุ่งยากซับซ้อน อีกทั้งผลการทดลองในการแบ่งคลาสย่อยด้วยวิธีที่นำเสนอนั้นก็ยังสามารถเพิ่มประสิทธิภาพในด้านความแม่นยำในการจำแนกประเภทข้อมูลอนุกรมเวลาได้อีกด้วย

5.2 ข้อเสนอแนะ

ข้อเสนอแนะสำหรับแนวทางต่อไปในงานวิจัยเพื่อพัฒนาวิธีการสร้างแผ่นแบบสำหรับข้อมูลอนุกรมเวลา โดยสิ่งที่ต้องการพัฒนาสำหรับงานวิจัยนี้คือการหาค่าเฉลี่ยสำหรับข้อมูลอนุกรมเวลาทั้งกลุ่ม เนื่องจากระยะทางแบบไดนามิกไทม์วอร์ปปิงไม่มีคุณสมบัติของความเป็นมาตรวัดเมตริก กล่าวคือไม่มีคุณสมบัติของอสมการสามเหลี่ยม (Triangular Inequality) ทำให้ค่าเฉลี่ยรูปร่างที่ได้ไม่เป็นค่าเฉลี่ยของกลุ่มข้อมูลที่แท้จริง ดังนั้นถ้าหากสามารถแก้ปัญหาดังกล่าวโดยการหาวิธีวัดระยะทางแบบไทม์วอร์ปปิงที่มีคุณสมบัติของอสมการสามเหลี่ยมได้แล้ว อาจส่งผลให้สามารถหาค่าเฉลี่ยรูปร่างของข้อมูลในเชิงเวลาที่แท้จริงได้ ปัญหาอีกประการหนึ่งของการสร้างแผ่นแบบด้วยวิธีที่นำเสนอก็คือ การหาค่าเฉลี่ยรูปร่างโดยอาศัยการปรับแนวแบบไทม์วอร์ปปิงขึ้นอยู่กับลำดับในการหาค่าเฉลี่ย ซึ่งถ้าลำดับเปลี่ยนไปแผ่นแบบที่ได้ก็จะมี

รูปร่างที่เปลี่ยนไปเป็นซึ่งอาจดูไม่สมเหตุสมผล ดังนั้นในการแก้ปัญหานี้คือหาวิธีที่ค่าเฉลี่ยรูปร่างไม่ขึ้นกับลำดับในการคำนวณหรือหาวิธีการเฉลี่ยรูปร่างของข้อมูลอนุกรมเวลาทั้งกลุ่มในครั้งเดียว และปัญหาอีกประการหนึ่งที่สำคัญในการสร้างแผ่นแบบก็คือ การหาวิธีที่จะประมาณค่าจากค่าเฉลี่ยรูปร่างแล้วให้ข้อมูลอนุกรมเวลามีจำนวนจุดข้อมูลเท่าเดิม ซึ่งอาจมีอีกหลายวิธีที่สามารถแก้ไขปัญหานี้ได้ ถ้าสามารถหาวิธีการประมาณค่าที่ดีที่สุดได้แผ่นแบบที่ได้ก็จะสามารถแทนข้อมูลทุกตัวในกลุ่มข้อมูลได้ดีที่สุด

สำหรับการพัฒนาอีกแนวทางหนึ่งคือ การแบ่งจำนวนคลาสย่อยที่เหมาะสมและทำงานรวดเร็ว ซึ่งมีหลายประด็นที่น่าสนใจในการทดลองต่อไป เช่น ค่าตัวแปรต่าง ๆ ที่งานวิจัยนี้ได้นำเสนอไว้ตามสมการที่กล่าวไว้ คือ ค่าความแปรปรวนที่จะทำการรวมกลุ่มข้อมูลหรือแยกกัน ซึ่งถ้าสามารถหาค่าพารามิเตอร์ที่เหมาะที่สุดได้ จะทำให้แผ่นแบบที่ได้มีประสิทธิภาพมากขึ้น เพราะถ้าสามารถวิเคราะห์ข้อมูลว่าในแต่ละคลาสควรแทนด้วยแผ่นแบบกี่ตัว และสามารถแบ่งข้อมูลเหล่านั้นได้อย่างชัดเจนจะทำให้เมื่อนำแผ่นแบบไปใช้งานจะมีประสิทธิภาพด้านความแม่นยำมากขึ้นกว่าเดิม

รายการอ้างอิง

[1] Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., and Keogh, E. (2008). Querying and mining of time series data: Experimental comparison of representations and distance measures. Proceedings of 34th International Conference on Very Large Data Bases (VLDB 2008), pp. 1542–1552. Auckland, New Zealand.

[2] Ratanamahatana, C.A., and Keogh, E. (2004). Making Time-Series Classification More Accurate Using Learned Constraints. Proceedings of SIAM International Conference on Data Mining (SDM), pp. 11–22. Lake Buena Vista, FL, USA.

[3] 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.

[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.

[5] 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.

[6] Chu, S., Keogh, E., Hart, D., and Pazzani, M. (2002). Iterative Deepening Dynamic Time Warping for Time Series. Proceedings of 2nd SIAM International Conference on Data Mining, Maebashi City, Japan.

[7] Keogh, E., Lonardi, S., and Chiu, W. (2002). Finding surprising patterns in a time series database in linear time and space. Proceedings of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 550-556. Edmonton, Alberta, Canada.

[8] 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.

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

[10] 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.

[11] Xi, X., Keogh, E., Shelton, C., Wei, L., and Ratanamahatana, C.A. (2006). Fast Time Series Classification Using Numerosity Reduction. Proceedings of 23rd International Conference on Machine Learning, pp. 1033–1040. Pittsburgh, PA, USA.

[12] Gupta, L., Molfese, D.L., Tammana, R., and Simos, P.G. (1996). Nonlinear Alignment and Averaging for Estimating the Evoked Potential. IEEE Transactions on Biomedical Engineering 43: 348–356.

[13] Wang, K., and Gasser, T. (1999). Synchronizing sample curves nonparametrically. The Annals of Statistics.

[14] Keogh, E., and Pazzani, M.J. (2001). Derivative Dynamic Time Warping. Proceeding of the First SIAM International Conference on Data Mining (SDM 2001), pp. 5–7. Chicago, USA.

[15] 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.

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

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

[18] Euachongprasit, W., and Ratanamahatana, C.A. (2007). Accurate Music Search by Humming using Uniform Scaling, Dynamic Time Warping, and Lower Bounding Function. Master's degree. Department of Computer Engineering, Chulalongkorn University.

[19] Agrawal, R., Lin, K.-I., Sawhney, H.S., and Shim, K. (1995). Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases. Proceedings of 21th International Conference on Very Large Data Bases, San Francisco, CA, USA.

[20] Keogh, E., and Pazzani, M.J. (1998). An enhanced representation of time series which allows fast and accurate classification, clustering and relevance

feedback. Proceedings of 4th International Conference of Knowledge Discovery and Data Mining, New York, NY, USA.

[21] Costantini, P., and Morandi, R. (1984). Monotone and convex cubic spline interpolation. Calcolo 21: 281–294.

[22] Cuche, E., Marquet, P., and Depeursinge, C. (2000). Aperture apodization using cubic spline interpolation: application in digital holographic microscopy. Optics communications 182: 59–69.

[23] White, I., Thompson, R., and Brotherstone, S. (1999). Genetic and environmental smoothing of lactation curves with cubic splines. Journal of Dairy Science 82: 632–638.

[24] Reinsch, C.H. (1967). Smoothing by spline functions. Numerische Mathematik 10: 177–183.

[25] Skalak, D.B. (1994). Prototype and feature selection by sampling and random mutation hill climbing algorithms. Proceedings of the Eleventh International Conference on Machine Learning (ML94), pp. 293–301. Morgan Kaufmann.

[26] Wilson, D.R., and Martinez, T.R. (1997). Instance pruning techniques. Proceedings of the Fourteenth International Conference (ICML'97), pp. 403–411. San Francisco, CA.

[27] Kneip, A., and Gasser, T. (1992). Statistical tools to analyze data representing a sample of curves. The Annals of Statistics 20: 1266–1305.

[28] Ramsay, J., and Li, X. (1998). Curve Registration. Journal of the Royal Statistical Society. Series B, Statistical Methodology: 351–363.

[29] Niennattrakul, V., and Ratanamahatana, C.A. (2007). Inaccuracies of Shape Averaging Method Using Dynamic Time Warping for Time Series Data. Proceedings of 7th International Conference on Computational Science: Advancing Science and Society through Computation, pp. 513–520.

[30] Keogh, E., and Lin, J. (2005). Clustering of time-series subsequences is meaningless: implications for previous and future research. Knowledge and Information Systems 8: 154–177.

[31] Niennattrakul, V., and Ratanamahatana, C.A. (2009). Shape Averaging under Time Warping. 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology (ECTI-CON 2009), Pattaya, Thailand.

[32] Keogh, E., Xi, X., Wei, L., and Ratanamahatana, C.A. (2008). The UCR Time Series Classification/Clustering Homepage [Online]. Available from: www.cs.ucr.edu/~eamonn/time_series_data/ [1 January 2008]

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

[34] Roverso, D. (2000). Multivariate temporal classification by windowed wavelet decomposition and recurrent networks. In 3rd ANS International Topical Meeting on Nuclear Plant Instrumentation, Control and Human-Machine Interface.

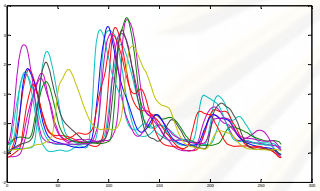
ภาคผนวก

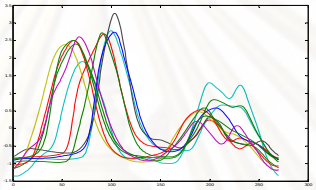
ภาคผนวก ก

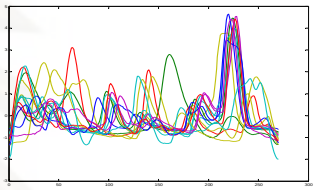
สำหรับตัวอย่างประเภทของชุดข้อมูลที่ใช้ในการทดลองซึ่งกล่าวไว้ในหัวข้อที่ 4.1.1 จะนำมาแสดงอยู่ในส่วนนี้ โดยจะแจกแจงตัวอย่างในแต่ละชุดข้อมูล ซึ่งแบ่งข้อมูลอนุกรมเวลาออกเป็นคลาส

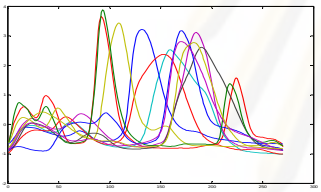
1. ชุดข้อมูล 50Words

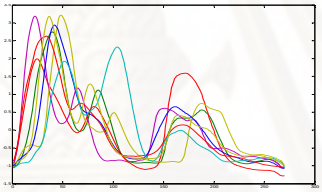
ชุดข้อมูล 50Words ประกอบด้วยข้อมูล 450 อนุกรม และมีจำนวนคลาสของข้อมูลอนุกรมเวลาเท่ากับ 50 คลาส ดังแสดงในรูปที่ ก. 1

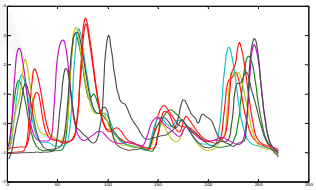

คลาส 1

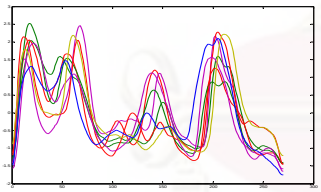

คลาส 2

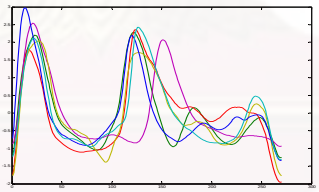

คลาส 3

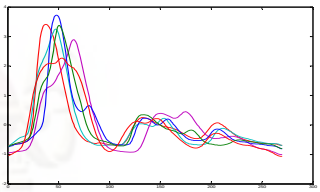

คลาส 4

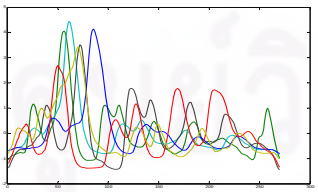

คลาส 5

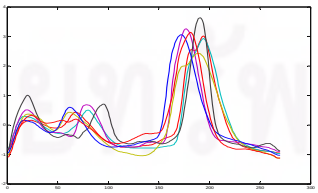

คลาส 6

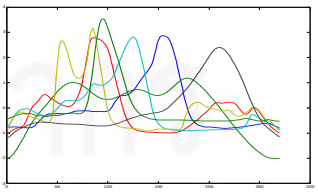

คลาส 7

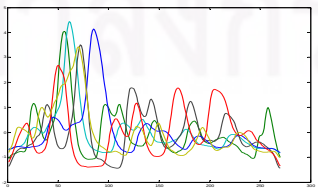

คลาส 8

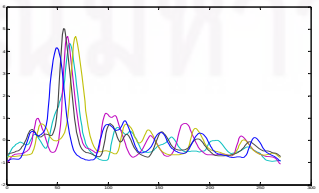

คลาส 9

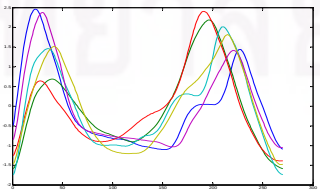

คลาส 10

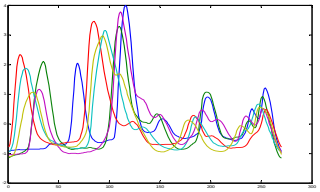

คลาส 11


คลาส 12

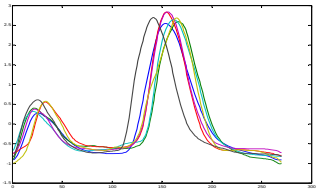

คลาส 13


คลาส 14

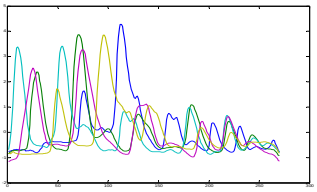

คลาส 15



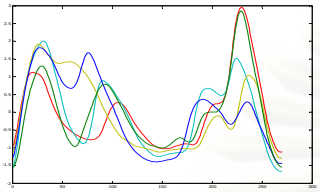
คลาส 16



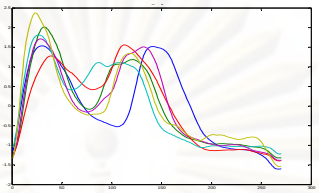
คลาส 17



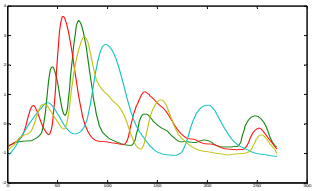
คลาส 18



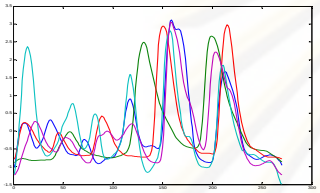
คลาส 19



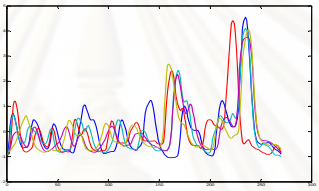
คลาส 20



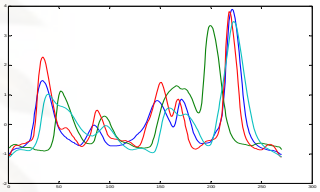
คลาส 21



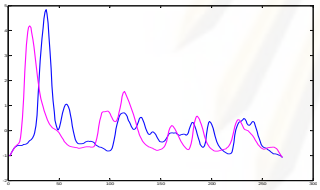
คลาส 22



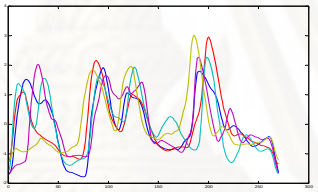
คลาส 23



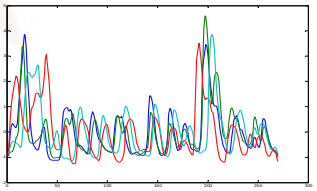
คลาส 24



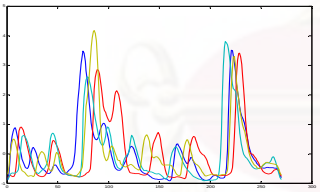
คลาส 25



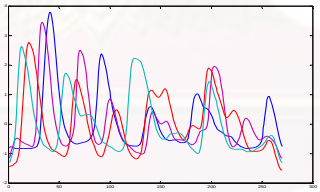
คลาส 26



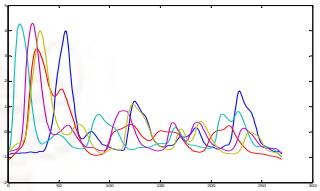
คลาส 27



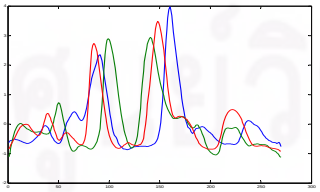
คลาส 28



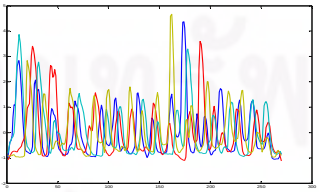
คลาส 29



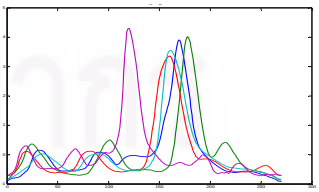
คลาส 30



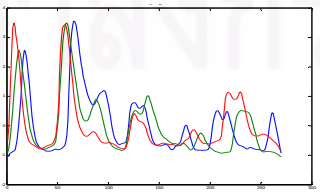
คลาส 31



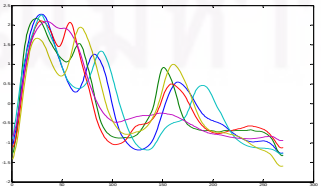
คลาส 32



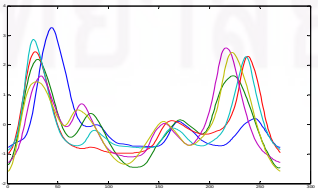
คลาส 33



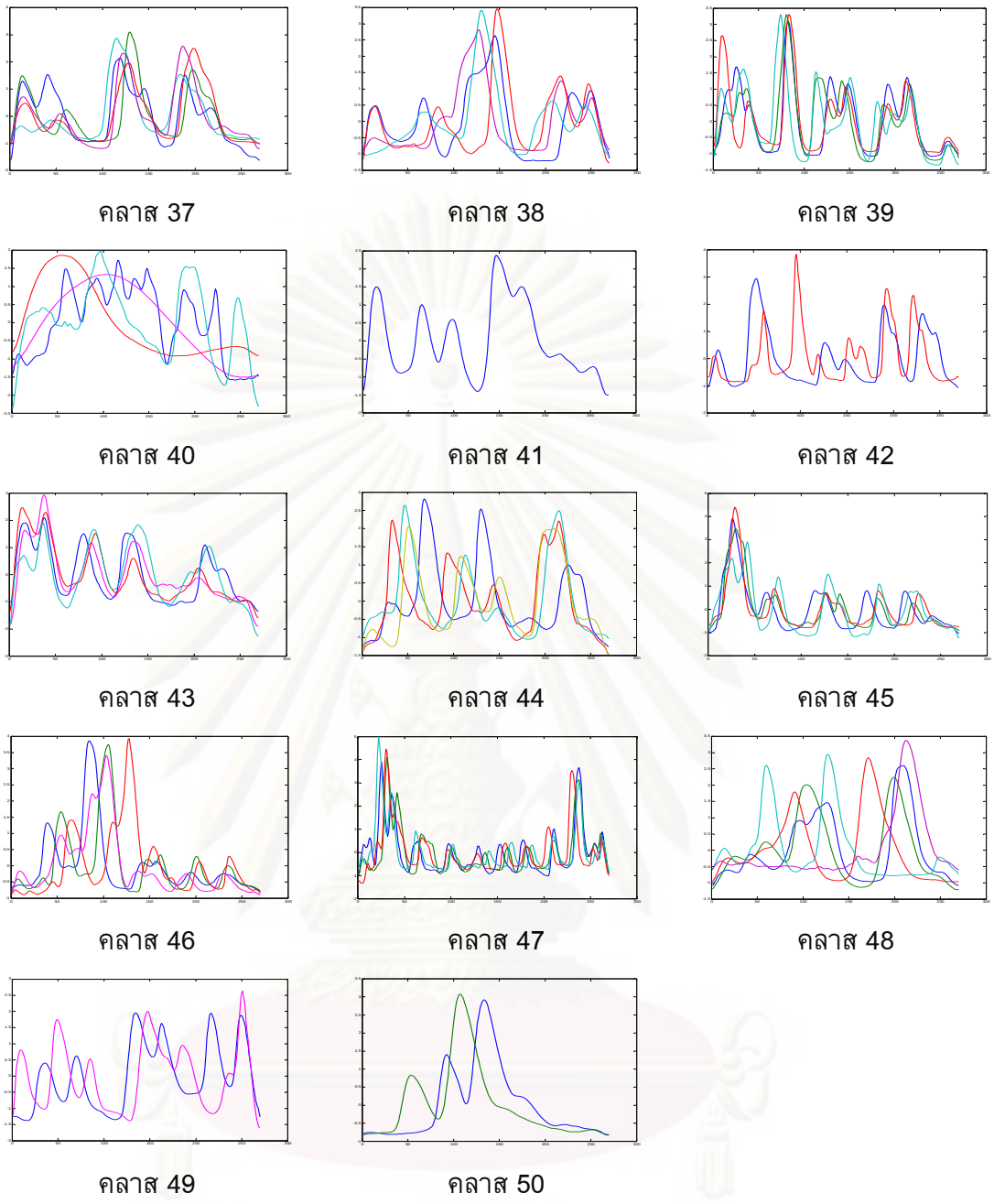
คลาส 34



คลาส 35



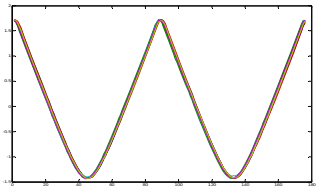
คลาส 36



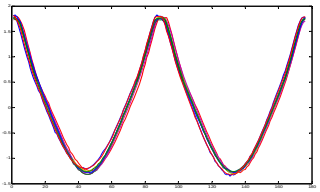
รูปที่ ก. 1 ข้อมูลอนุกรมเวลาแต่ละคลาสของชุดข้อมูล 50Words

2. ชุดข้อมูล Adiac

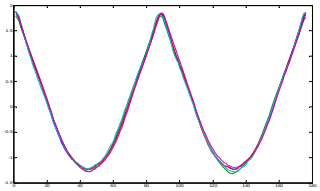
ชุดข้อมูล Adiac มีจำนวนคลาสของข้อมูลอนุกรมเวลาเท่ากับ 37 คลาส โดยมีข้อมูลทั้งหมด 390 อนุกรม และแต่ละอนุกรมมีความยาวเท่ากับ 176 จุดข้อมูลดังแสดงในรูปที่ ก. 2



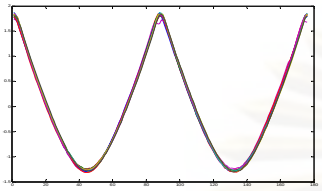
คลาส 1



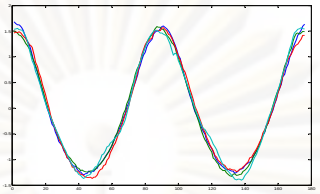
คลาส 2



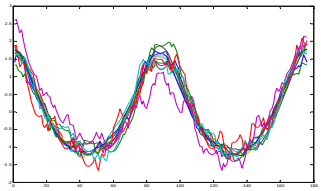
คลาส 3



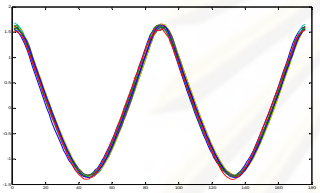
คลาส 4



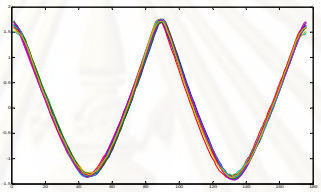
คลาส 5



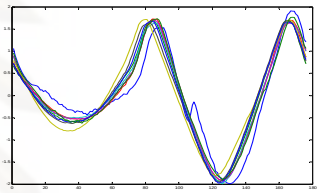
คลาส 6



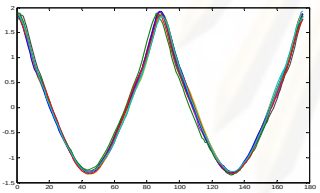
คลาส 7



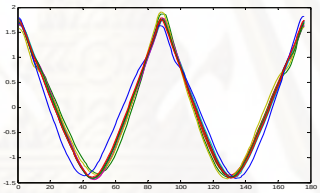
คลาส 8



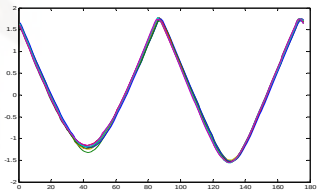
คลาส 9



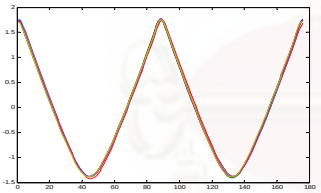
คลาส 10



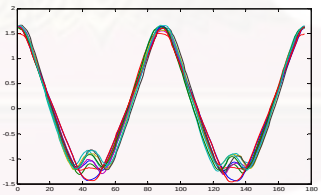
คลาส 11



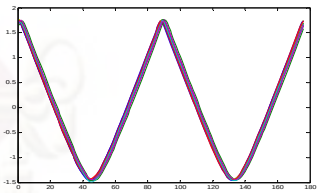
คลาส 12



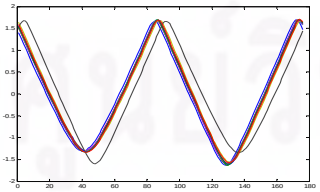
คลาส 13



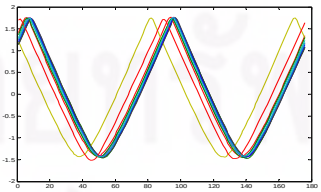
คลาส 14



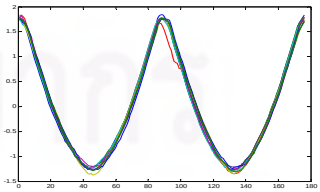
คลาส 15



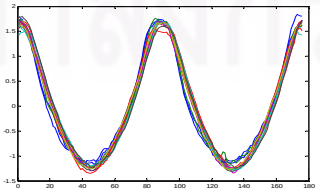
คลาส 16



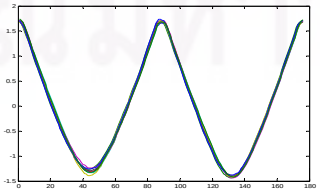
คลาส 17



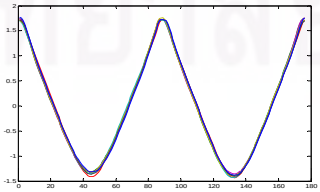
คลาส 18



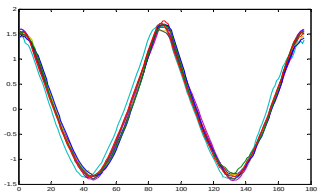
คลาส 19

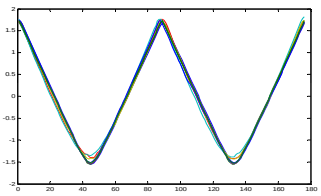


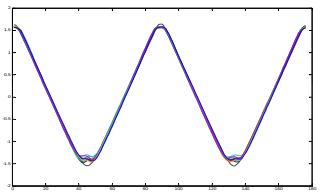
คลาส 20

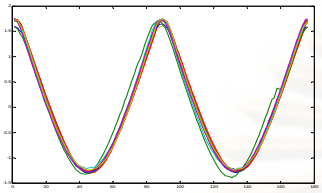


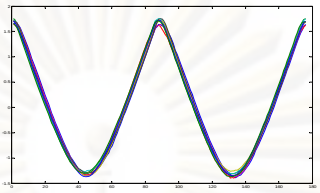
คลาส 21

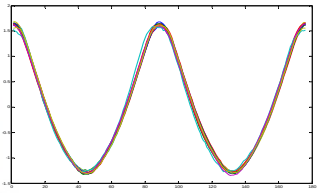

คลาส 22

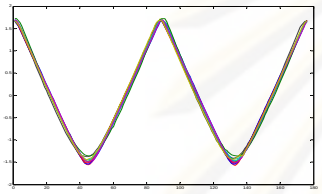

คลาส 23

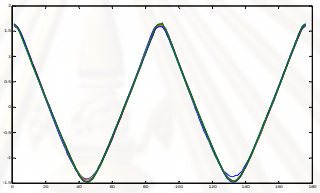

คลาส 24

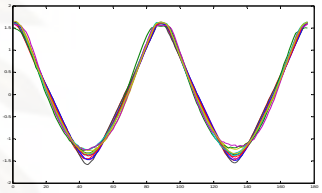

คลาส 25

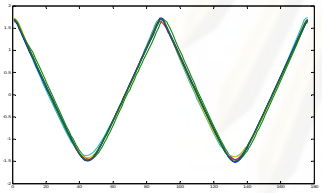

คลาส 26

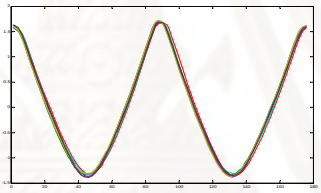

คลาส 27

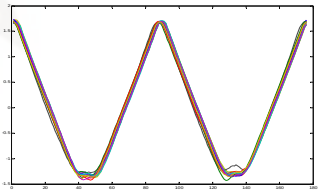

คลาส 28

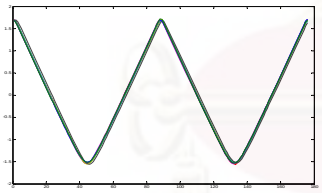

คลาส 29

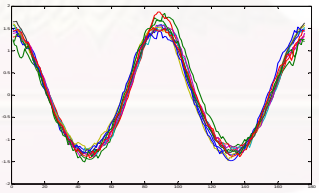

คลาส 30

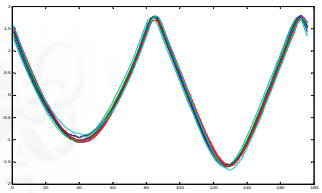

คลาส 31

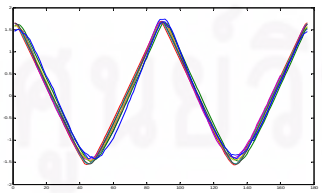

คลาส 32


คลาส 33


คลาส 34


คลาส 35

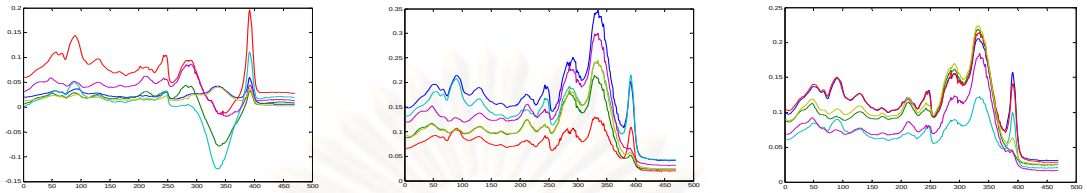

คลาส 36


คลาส 37

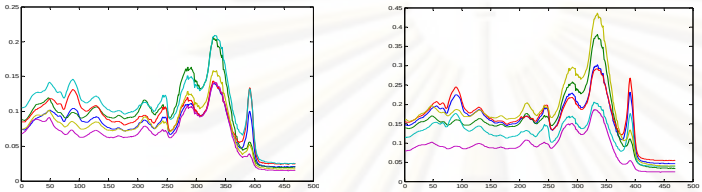
รูปที่ ก. 2 ข้อมูลอนุกรมเวลาแต่ละคลาสของชุดข้อมูล Adiac

3. ชุดข้อมูล Beef

ชุดข้อมูล Beef ประกอบด้วยข้อมูลอนุกรมเวลา 30 อนุกรม โดยที่แต่ละอนุกรมมีความยาวเท่ากับ 470 จุดข้อมูล และมีจำนวนคลาสของข้อมูลอนุกรมเวลาเท่ากับ 5 คลาส ดังแสดงในรูปที่ ก. 3



คลาส 1 คลาส 2 คลาส 3

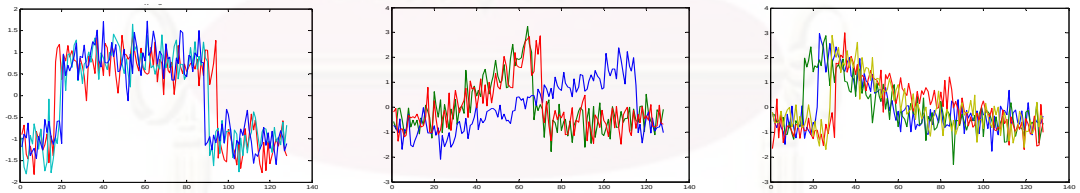


คลาส 4 คลาส 5

รูปที่ ก. 3 ข้อมูลอนุกรมเวลาแต่ละคลาสของชุดข้อมูล Beef

4. ชุดข้อมูล CBF

ชุดข้อมูล CBF ประกอบด้วยข้อมูลอนุกรมเวลา 30 อนุกรม โดยที่แต่ละอนุกรมมีความยาวเท่ากับ 128 จุดข้อมูล และมีจำนวนคลาสของข้อมูลอนุกรมเวลาเท่ากับ 3 คลาส ดังแสดงในรูปที่ ก. 4

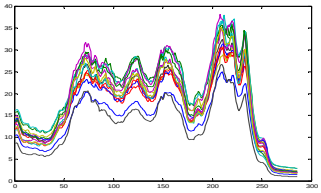


คลาส 1 คลาส 2 คลาส 3

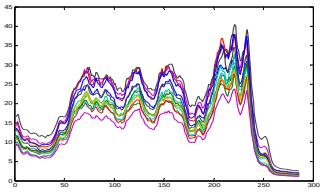
รูปที่ ก. 4 ข้อมูลอนุกรมเวลาแต่ละคลาสของชุดข้อมูล CBF

5. ชุกข้อมูล Coffee

ชุดข้อมูล Coffee ประกอบด้วยข้อมูลอนุกรมเวลา 28 อนุกรม โดยที่แต่ละอนุกรมมีความยาวเท่ากับ 286 จุดข้อมูล และที่มีจำนวนคลาสของข้อมูลอนุกรมเวลาเท่ากับ 2 คลาส ดังแสดงในรูปที่ ก. 5



คลาส 1

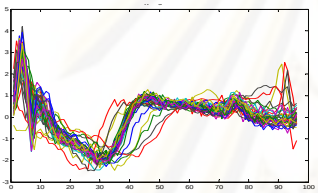


คลาส 2

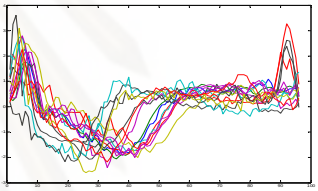
รูปที่ ก. 5 ข้อมูลอนุกรมเวลาแต่ละคลาสของชุดข้อมูล Coffee

6. ชุดข้อมูล ECG

ชุดข้อมูล ECG ประกอบด้วยข้อมูลอนุกรมเวลา 100 อนุกรม โดยที่แต่ละอนุกรมมีความยาวเท่ากับ 96 จุดข้อมูล และมีจำนวนคลาสของข้อมูลอนุกรมเวลาเท่ากับ 2 คลาส ดังแสดงในรูปที่ ก. 6



คลาส 1

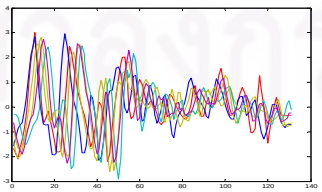


คลาส 2

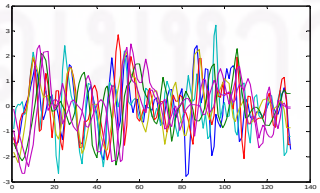
รูปที่ ก. 6 ข้อมูลอนุกรมเวลาแต่ละคลาสของชุดข้อมูล ECG

7. ชุดข้อมูล Face(all)

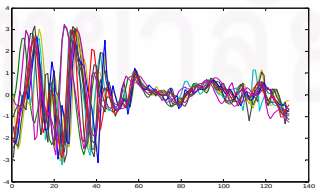
ชุดข้อมูล Face เป็นข้อมูลที่ได้จากโจทย์ของการจำแนกหน้าโดยใช้ภาพฉายของศีรษะ ซึ่งได้จากการถ่ายภาพจำนวน 20-35 ภาพของคน 4 คน ซึ่งประกอบด้วยหญิง 1 คน และชาย 3 คน ที่กำลังแสดงท่าทางต่าง ๆ กัน เช่น พูดคุย ยิ้ม หัวเราะ เป็นต้น หลังจากนั้นทำการแปลงภาพฉายของศีรษะ โดยเริ่มจากส่วนคอให้เป็นอนุกรมเวลาด้วยการวัดค่ามุมท้องถิ่นตามแนวของเส้นรอบวง ข้อมูลชุดข้อมูล Face(all) ประกอบด้วยข้อมูลอนุกรมเวลา 560 อนุกรม โดยที่แต่ละอนุกรมมีความยาวเท่ากับ 131 จุดข้อมูล และมีจำนวนคลาสของข้อมูลอนุกรมเวลาเท่ากับ 14 คลาส ดังแสดงในรูปที่ ก. 7



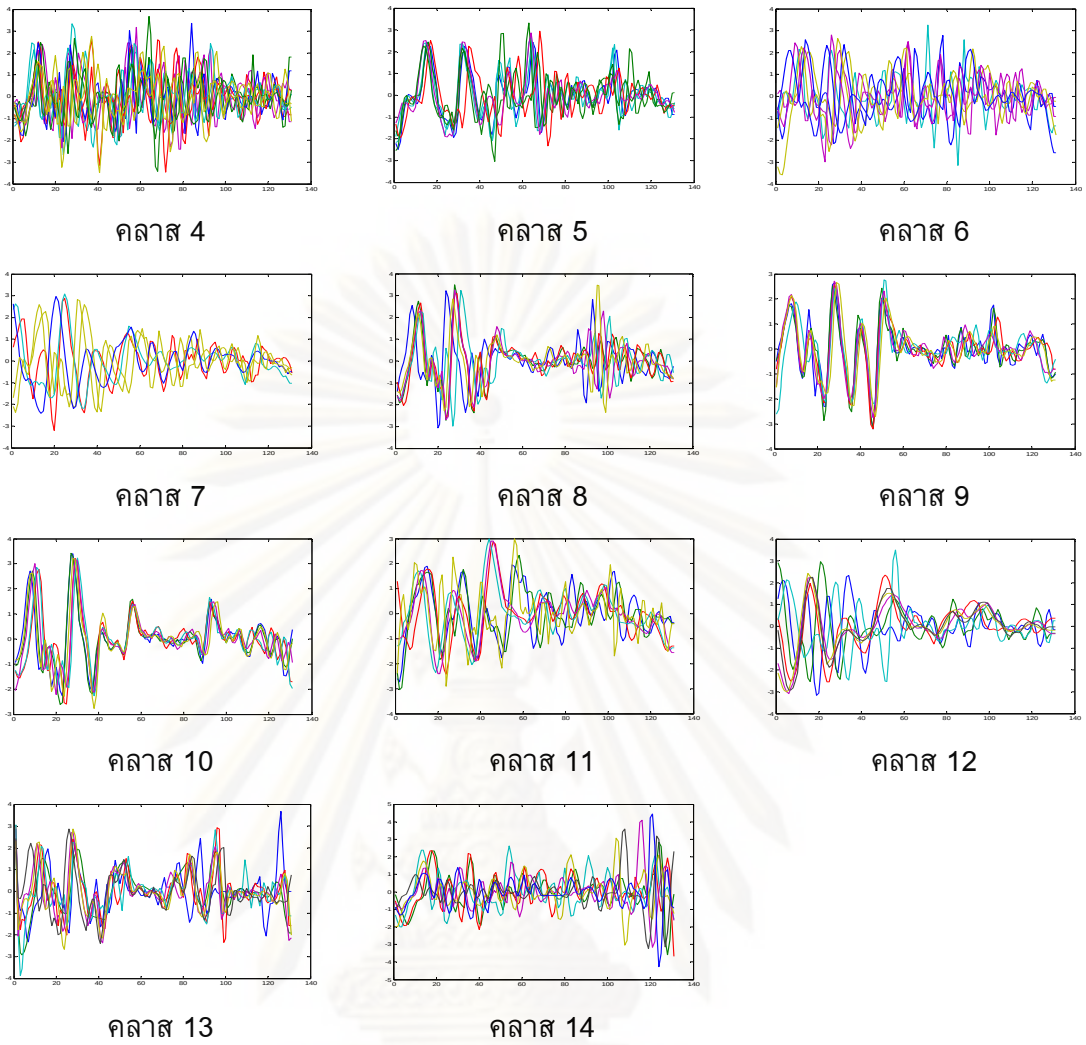
คลาส 1



คลาส 2



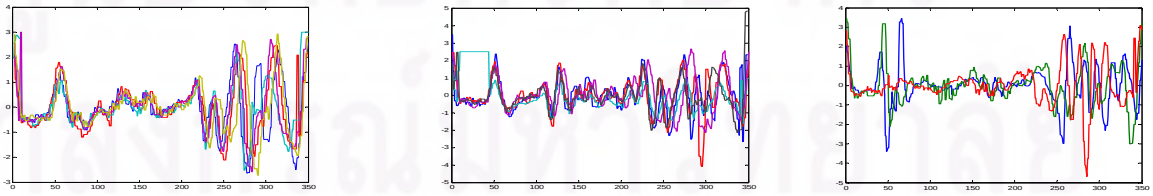
คลาส 3

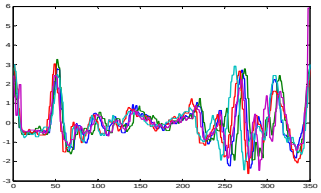


รูปที่ ก. 7 ข้อมูลอนุกรมเวลาแต่ละคลาสของชุดข้อมูล Face(all)

8. ชุดข้อมูล Face(four)

ชุดข้อมูล Face(four) ประกอบด้วยข้อมูลอนุกรมเวลา 24 อนุกรม โดยที่แต่ละอนุกรมมีความยาวเท่ากับ 350 จุดข้อมูล และมีจำนวนคลาสของข้อมูลอนุกรมเวลาเท่ากับ 4 คลาส ดังแสดงในรูปที่ ก. 8



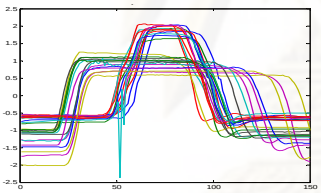


คลาส 4

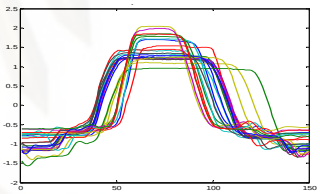
รูปที่ ก. 8 ข้อมูลอนุกรมเวลาแต่ละคลาสของชุดข้อมูล Face(four)

9. ชุดข้อมูล Gun-Point

ชุดข้อมูล Gun-Point เป็นชุดข้อมูลที่ได้จากการตรวจตราภาพวีดีทัศน์ ซึ่งชุดข้อมูลนี้ประกอบไปด้วยข้อมูลสองส่วนได้แก่ Gun เป็นส่วนที่มือของนักแสดงจะอยู่ที่ข้างลำตัว แล้วทำการชักปืนจำลองจากซองปืนที่ติดอยู่ที่เอวและชี้ปลายกระบอกปืนไปที่เป้าหมายประมาณ 1 วินาที แล้วเก็บปืนกลับเข้าซองและวางมือไว้ข้างลำตัว และ Point คือส่วนที่นักแสดงมีปืนอยู่ข้างลำตัว ชี้นิ้วชี้ไปที่เป้าหมายประมาณ 1 วินาที แล้ววางมือไว้ข้างลำตัว ชุดข้อมูล Gun-Point ประกอบด้วยข้อมูลอนุกรมเวลา 50 อนุกรม โดยที่แต่ละอนุกรมมีความยาวเท่ากับ 150 จุดข้อมูล และมีจำนวนคลาสของข้อมูลอนุกรมเวลาเท่ากับ 2 คลาส ดังแสดงในรูปที่ ก. 9



คลาส 1

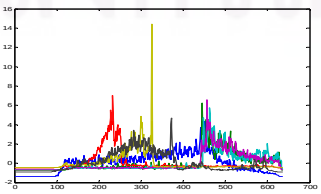


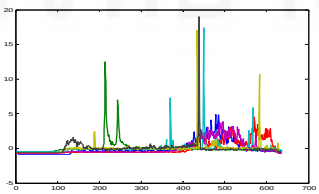
คลาส 2

รูปที่ ก. 9 ข้อมูลอนุกรมเวลาแต่ละคลาสของชุดข้อมูล Gun-Point

10. ชุดข้อมูล Lightning2

ชุดข้อมูล Loghting2 ประกอบด้วยข้อมูลอนุกรมเวลา 60 อนุกรม โดยที่แต่ละอนุกรมมีความยาวเท่ากับ 637 จุดข้อมูล และมีจำนวนคลาสของข้อมูลอนุกรมเวลาเท่ากับ 2 คลาส ดังแสดงในรูปที่ ก. 10



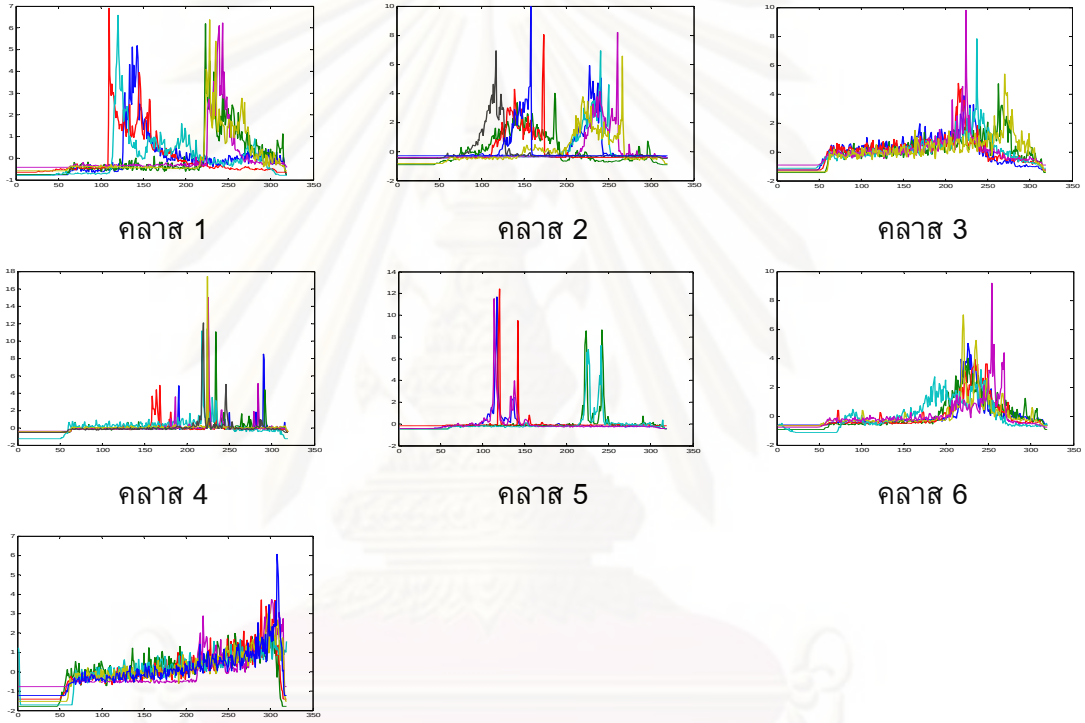


คลาส 1 คลาส 2

รูปที่ ก. 10 ข้อมูลอนุกรมเวลาแต่ละคลาสของชุดข้อมูล Lightning2

11. ชุดข้อมูล Lightning7

ชุดข้อมูล Loghting7 ประกอบด้วยข้อมูลอนุกรมเวลา 70 อนุกรม โดยที่แต่ละอนุกรมมีความยาวเท่ากับ 319 จุดข้อมูล และมีจำนวนคลาสของข้อมูลอนุกรมเวลาเท่ากับ 7 คลาส ดังแสดงในรูปที่ ก. 11



คลาส 1 คลาส 2 คลาส 3

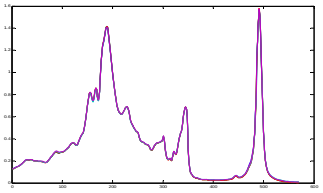
คลาส 4 คลาส 5 คลาส 6

คลาส 7

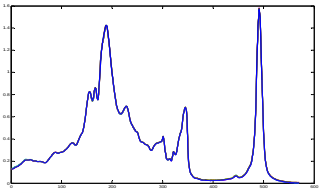
รูปที่ ก. 11 ข้อมูลอนุกรมเวลาแต่ละคลาสของชุดข้อมูล Lightning7

12. ชุดข้อมูล Oliveoil

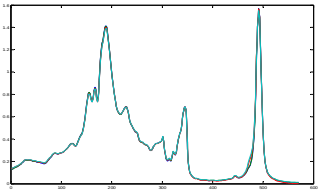
ชุดข้อมูล Oliveoil ประกอบด้วยข้อมูลอนุกรมเวลา 30 อนุกรม โดยที่แต่ละอนุกรมมีความยาวเท่ากับ 570 จุดข้อมูล และมีจำนวนคลาสของข้อมูลอนุกรมเวลาเท่ากับ 6 คลาส ดังแสดงในรูปที่ ก. 13



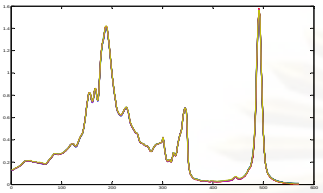
คลาส 1



คลาส 2



คลาส 3

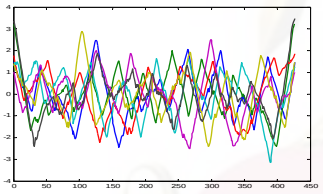


คลาส 4

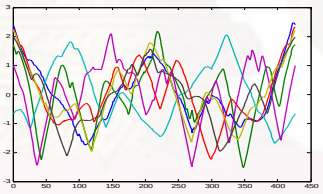
รูปที่ ก. 12 ข้อมูลอนุกรมเวลาแต่ละคลาสของชุดข้อมูล Oliveoil

13. ชุดข้อมูล OSU Leaf

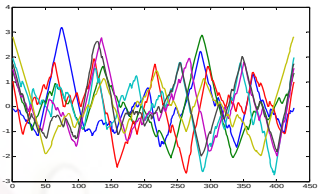
ชุดข้อมูล OSU Leaf เป็นชุดข้อมูลที่ได้จากการแปลงภาพฉายของใบไม้ ให้เป็นอนุกรมเวลาด้วยการวัดค่ามุมท้องถิ่นตามแนวของเส้นรอบวง และมีจำนวนคลาสของข้อมูลอนุกรมเวลาเท่ากับ 6 คลาส ดังแสดงในรูปที่ ก. 13



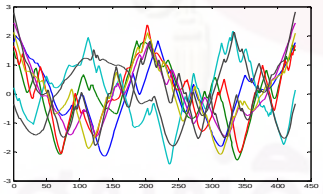
คลาส 1



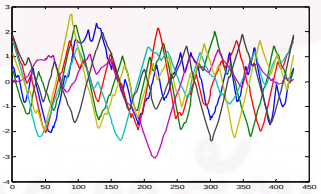
คลาส 2



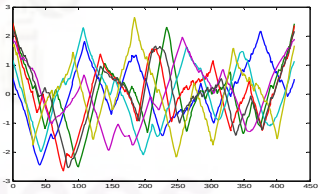
คลาส 3



คลาส 4



คลาส 5

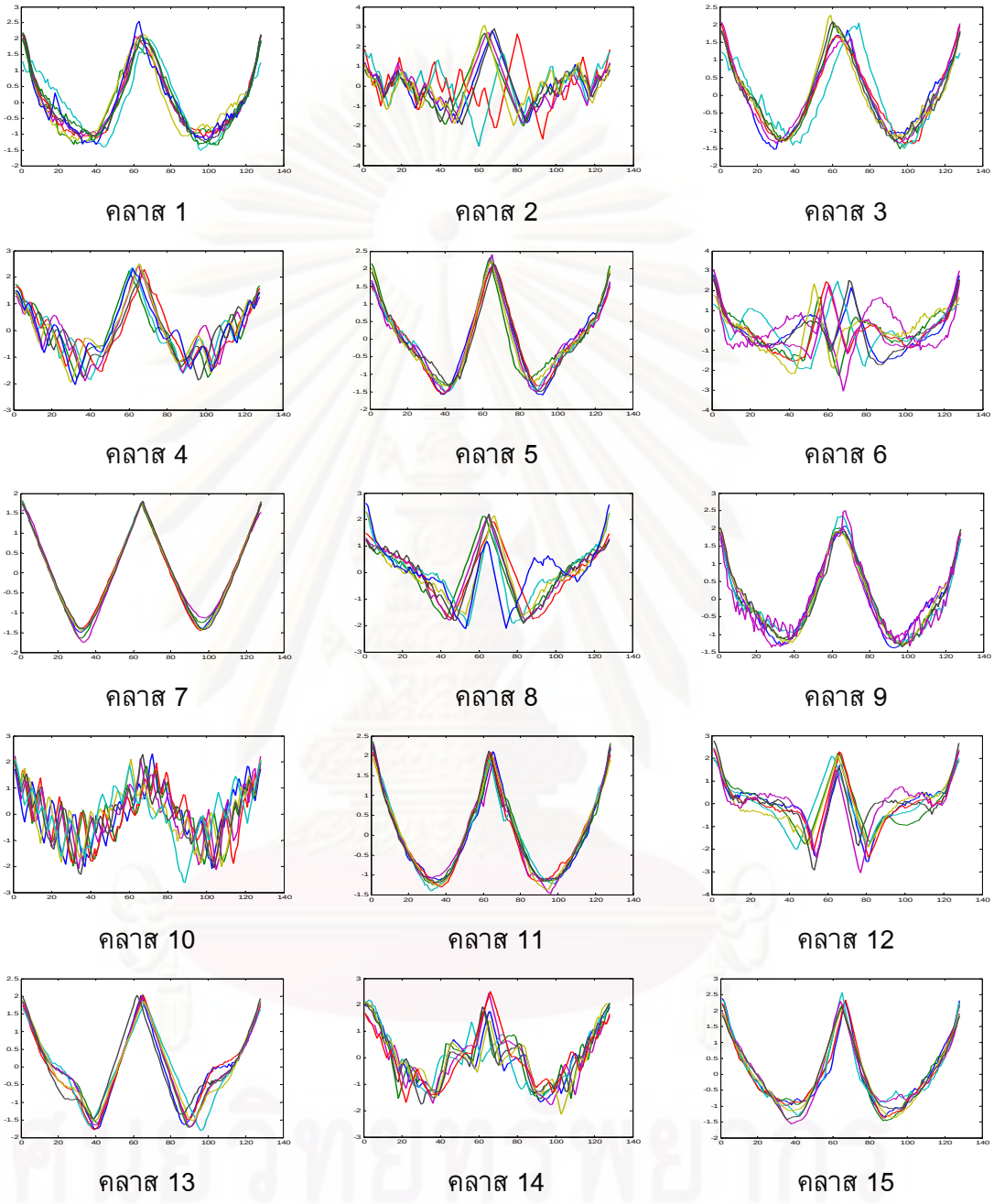


คลาส 6

รูปที่ ก. 13 ข้อมูลอนุกรมเวลาแต่ละคลาสของชุดข้อมูล OSU Leaf

14. ชุดข้อมูล Swedish Leaf

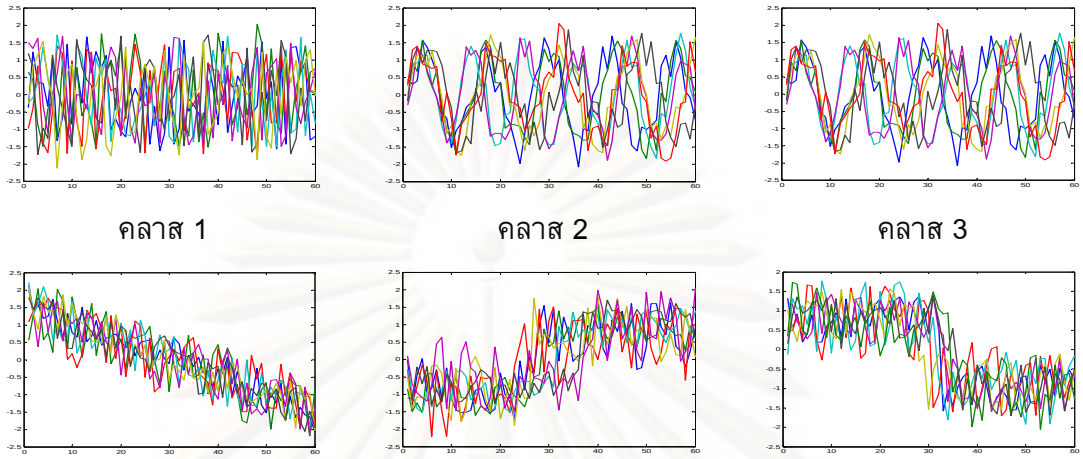
ชุดข้อมูล Swedish Leaf ประกอบด้วยข้อมูลอนุกรมเวลา 500 อนุกรม โดยที่แต่ละอนุกรมมีความยาวเท่ากับ 128 จุดข้อมูล และมีจำนวนคลาสของข้อมูลอนุกรมเวลาเท่ากับ 15 คลาส ดังแสดงในรูปที่ ก. 14



รูปที่ ก. 14 ข้อมูลอนุกรมเวลาแต่ละคลาสของชุดข้อมูล Swedish Leaf

15. ชุดข้อมูล Synthetic Control

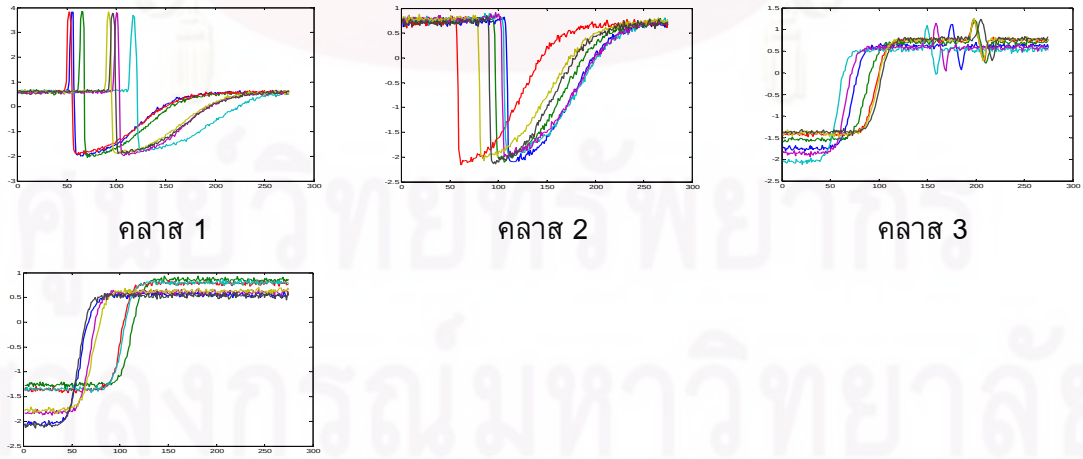
ชุดข้อมูล Synthetic Control ประกอบด้วยข้อมูลอนุกรมเวลา 300 อนุกรม โดยที่แต่ละอนุกรมมีความยาวเท่ากับ 60 จุดข้อมูล และมีจำนวนคลาสของข้อมูลอนุกรมเวลาเท่ากับ 6 คลาส ดังแสดงในรูปที่ ก. 15



รูปที่ ก. 15 ข้อมูลอนุกรมเวลาแต่ละคลาสของชุดข้อมูล Synthetic Control

16. ชุดข้อมูล Trace

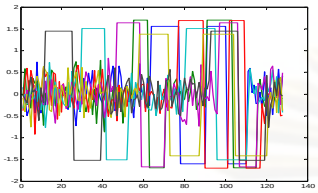
ชุดข้อมูล Trace เป็นชุดข้อมูลย่อยของชุดข้อมูล Transient Classification Benchmark ของ Davide Roverso [34] ข้อมูลสังเคราะห์ชุดนี้ถูกออกแบบมาเพื่อจำลองความบกพร่องของเครื่องมือในโรงงานไฟฟ้านิวเคลียร์ ชุดข้อมูลเต็มจะมีทั้งสิ้น 16 คลาส คลาสละ 50 อนุกรม แต่ชุดข้อมูล Trace ชุดนี้มีจำนวนคลาสของข้อมูลอนุกรมเวลาเท่ากับ 4 คลาส ดังแสดงในรูปที่ ก. 16



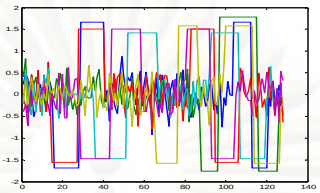
รูปที่ ก. 16 ข้อมูลอนุกรมเวลาแต่ละคลาสของชุดข้อมูล Trace

17. ชุดข้อมูล Two Patterns

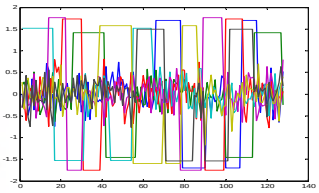
ชุดข้อมูล Two Patterns มีจำนวนคลาสของข้อมูลอนุกรมเวลาเท่ากับ 4 คลาส ซึ่งแต่ละคลาสจะถูกกำหนดโดยรูปแบบ 2 รูปที่มีลำดับแน่นอน คือ ลง-ลง ขึ้น-ลง ลง-ขึ้น และขึ้น-ขึ้น ดังแสดงในรูปที่ ก. 17



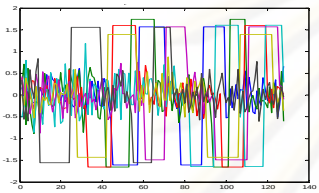
คลาส 1



คลาส 2



คลาส 3



คลาส 4

รูปที่ ก. 17 ข้อมูลอนุกรมเวลาแต่ละคลาสของชุดข้อมูล Two Patterns

ภาคผนวก ข

บทความทางวิชาการเรื่อง “Time Series Shape Averaging Using Time-Warping Alignment with Re-Sampling” โดย ดารารัตน์ ศรีใส และโชติรัตน์ รัตนามหัทธนะ ในงานประชุมวิชาการนานาชาติ “6th International Joint Conference on Computer Science and Software Engineering” ซึ่งจัดขึ้น ณ เมืองภูเก็ต ประเทศไทย ระหว่างวันที่ 13 พฤษภาคม ถึง 15 พฤษภาคม 2552

JCSSE2009, May 13-15, 2009 (Vol. 1)

Time Series Shape Averaging Using Time-Warping Alignment with Re-sampling

Dararat Srisai and Chotirat Ann Ratanamahatana
Department of Computer Engineering, Chulalongkorn University
254 Phayathai Road, Pathumwan, Bangkok Thailand. 10330
{g51dsr, ann}@cp.eng.chula.ac.th

Abstract

Signal averaging is one of the common tasks in finding a template or representative of a set of time series signals. In early development, arithmetic averaging method is typically used to find an average among a set of similar class signals. The main drawback of such method is that it is highly susceptible to both global and local shifting. Therefore, the averaged result may not be accurate. As a result, some efforts have been made to determine an average shape of the signals based on Dynamic Time Warping (DTW) distance measure. However, the challenge is that the averaged signal gets longer every time an average function call is made, and the result may not align with an integer grid, making subsequent processing becomes problematic. We propose a novel shape averaging algorithm that properly re-samples the averaged sequence to a designated sample size that always guarantees grid alignment. Our proposed Re-sampled Shape Averaging method (RSA) is validated on classification problems with various datasets on a wide range of domains. The accuracy, time, and memory requirement of the classification are benchmarked against the Non-Linear Alignment and Averaging Filters (NLAAF) and the Traditional One Nearest Neighbor (T-1NN) with DTW distance algorithms. The results show that the accuracy is slightly better than NLAAF, but the time and memory requirement are significantly improved over both algorithms.

Keywords: Shape Averaging, Dynamic Time Warping, Spline Approximation, Time Series Data

1. Introduction

Sequence Averaging is one of the typical routines to determine a template of time series signals. These templates are crucial in several data mining applications [1-3] such as clustering [4, 5] and classification [4]. Data classification typically is a computationally intensive operation, especially when the dataset contains a large number of classes. Some classification algorithms such as one nearest neighbor (1NN) with DTW distance classification simply take the entire training dataset for the computation. Therefore, it takes maximum computation time, but establishes the upper bound for accuracy. To reduce the computational complexity, many classification algorithms may use a representative sequence for each class. Instead, its computational complexity becomes a function of the number of classes, not the number of samples in the training data. As a result, classification operations can be executed much faster, but in most cases, the accuracy is sacrificed. To maximize accuracy under this circumstance, the best representative sequence for each class must be obtained. There are a variety of techniques to compute a good representative sequence. Shape averaging technique is one of them.

In early development, the averaging routine relies solely on arithmetic averaging to compute an averaged sequence among a set of signals. However, the main problem of using arithmetic averaging is that it is highly susceptible to both local and global shifting. Therefore, the averaged sequence may not be as accurate as it should, as illustrated in Figure 1(c). Given two signals, Q and C, with similar shape, their arithmetic averaging is computed by the following equation,

$$Y_i = \frac{Q_i + C_i}{2}, \qquad (1)$$

where Y_i is i^{th} arithmetic average, while Q_i and C_i are i^{th} data points of the signals Q and C, respectively. Figure 1(c) shows that an arithmetic averaging would result in a double-peak signal that does not reflect the shape of either Q or C. A more desirable shape is shown in Figure 1(d).

In addition, original time series sequences may come with various lengths. And since an arithmetic averaging technique (one-to-one alignment) [6]

requires both averaging sequences to be of the same length, it cannot be applied directly in such case.

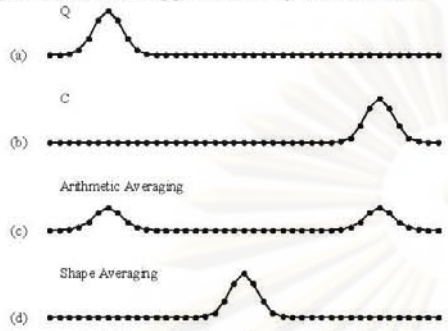


Figure 1. Time Series Averaging between sequences *Q* (a) and *C* (b) using arithmetic averaging (c) and the shape averaging (d).

Sequence averaging based on Dynamic Time Warping (DTW) alignment is one of the potential solutions, but with some limitation. This sequence alignment technique can only process two sequences at a time. Therefore, for a large number of sequences, this must be done iteratively. Gupta et al. [7] have proposed Non-Linear Alignment and Averaging Filters algorithm (NLAAF), but the length of the averaging results is inevitably longer in every averaging computation. The original sequences are sampled uniformly, but the NLAAF average no longer reflects a uniform sampling. Therefore, the shape of the NLAAF average will be somewhat distorted.

In an effort to obtain an undistorted shape averaging using DTW, one of the challenges is that the averaged result may not align with the original time grid. This is from the fact that some averaging algorithms do not only average the values, but also the time. As a result, the time average may no longer align with the original time grid. A Continuous Dynamic Time Warping (CDTW)[8] may be used to overcome this problem, but it would introduce much greater computational complexity.

To resolve these problems, we propose a novel shape averaging algorithm that utilizes DTW to determine the corresponding alignment, along with a re-sampling technique to find the template or the average sequences both accurately and efficiently.

2. Background

2.1. Dynamic Time Warping

Dynamic Time Warping (DTW) distance measure[9, 10] is an algorithm that provides similarity measure between two time series. Suppose there are two times series Q and C of length m and n, respectively. An $m \times n$ distance matrix, $D = \{d\}_{m \times n}$, can be constructed using the following equation,

$$d_{i,j} = (q_i - c_j)^2, \qquad (2)$$

where q_i and c_j are i^{th} and j^{th} elements in Q and C, respectively. A warping path, W, through the matrix that minimizes the cumulative distance provides an optimal alignment between the two sequences. The warping path's element, w_k, equals to $(i, j)_k$ where $1 \le i \le m$, $1 \le j \le n$, $1 \le k \le K$, and $\max(m, n) \le K \le m+n-1$. The optimal path, W_o, is the path that minimizes the warping cost,

$$DTW(Q,C) = \sqrt{\sum_{k=1}^{K} D(w_k)}, \qquad (3)$$

This path is discovered by using dynamic programming to evaluate the minimum cumulative distance, γ_{ij}, from three adjacent elements, $\gamma_{i,j} = d_{i,j} + \min\{\gamma_{i-1,\,j-1},\ \gamma_{i-1,\,j},\ \gamma_{i,\,j-1}\}$ with additional conditions as follows.

Boundary conditions: The path must start on $w_1 = (1, 1)$ and end on $w_K = (m, n)$.

Continuity condition: The i and j indices can only increase by 0 or 1 on each step along the path.

Monotonic condition: The warping path can only move forward in time (i and j monotonically increase). Given $w_k = (q, c)$ and $w_{k-1} = (q', c')$. The $q - q'$ and $c - c'$ must be greater than zero.

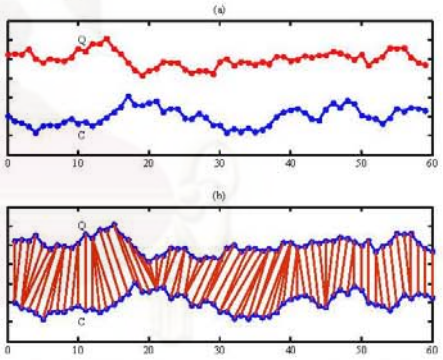


Figure 2. (a) Two time series signals; *Q* and *C*. (b) The corresponding DTW alignment between the two signals; *Q* and *C*.

In general, the results of DTW composes of the path and cost between the two time series. The path contains an array of data point pairs while the cost provides the similarity measure. For example, two given time series, Q and C., are shown in Figure 2(a). After the DTW computation, the resulting path provides the data point pairs between them as shown in

Figure 2(b). The DTW cost, which is a summation of all differences in the data point pairs along the path, provides the similarity measure between the two time series. The smaller the cost implies the greater the similarity between them.

2.2. Spline Approximation

Cubic spline is used in many computer graphic applications [11, 12]. It is a piecewise polynomial function that can provide smooth curve fitting through a set of points [13-15]. The cubic spline approximation is highly accurate and has interpolation error of $O(h^4)$ where h is the step size of the approximated derivative.

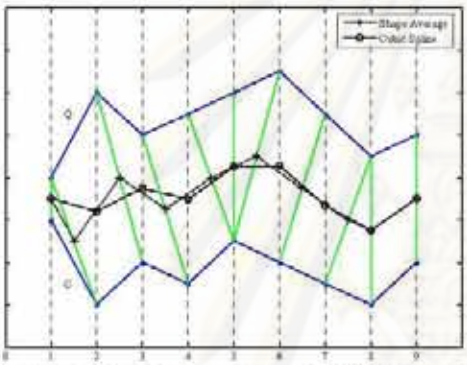


Figure 3. The shape average (dotted line with "+" markers) between two time series; Q and C. The cubic spline approximation (solid line with "o" markers) re-samples the shape average back to a uniform space grid.

Let the curve, S, be defined over a range $[a, b]$ that is divided into k pieces. A polynomial, P_i, defines the approximation value of i^{th} piece curve in these k subintervals. Therefore, S can be smoothed across the entire range $[a, b]$. Figure 3 shows an approximation from the shape average section. The previously proposed shape average, S, is plotted with "o" markers and the spline approximation is shown with "+" markers, while it is forced to align with the original signals, Q and C. The spline approximation, from the observation, can well preserve the original shape of both signals.

3. Shape Averaging

In this section, we will show that traditional shape averaging using arithmetic mean evidently does not provide an accurate result because the two signals are not properly aligned (as illustrated in Figure 1). Shape averaging based on DTW has recently been introduced to determine an optimal alignment. Given two time series Q = [2 6 4 5 6 7 5 3 4] and C = [6 2 4 3 5 4 3 2 4], the cumulative distance, D, and the warping path, W = {(1,1), (1,2), (2,3), (3,4), (4,5), (5,5), (6,5), (7,6), (8,7), (8,8), (9,9)}, are shown in Figure 4. The shape average, S, is computed by using the following formula,

$$S_k = \left(\frac{w_{k,1} + w_{k,2}}{2}, \frac{Q(w_{k,1}) + C(w_{k,2})}{2} \right), \quad (4)$$

where $w_{k,1}$ and $w_{k,2}$ are k^{th} index pair of the warping path, W. The average result, S, is a sequence {(1,4), (1.5,2), (2.5,4.5), (3.5,3.5), (4.5,5), (5,5.5), (5.5,6), (6.5,4.5), (7.5,3), (8,2.5), (9,4)}.

The DTW provides alignments where the corresponding samples between the two sequences are identified. Therefore, shape averaging can be performed with better accuracy.

D_{ij}									Q_i	i
14	12	10	11	12	11	12	13	**12**	4	9
12	10	11	11	12	11	**11**	**12**	13	3	8
9	11	12	14	10	**11**	13	16	17	5	7
8	12	12	13	**10**	11	13	16	17	7	6
7	11	9	10	**8**	9	11	14	14	6	5
7	9	7	8	**7**	8	10	12	12	5	4
6	6	6	**7**	8	8	9	11	11	4	3
4	8	**6**	9	8	10	13	17	15	6	2
4	**4**	6	7	10	12	13	13	15	2	1
C_j	6	2	4	3	5	4	3	2	4	
j	1	2	3	4	5	6	7	8	9	

Figure 4. The cumulative distance, *D*, and the warping path (shown in boldface within gray cells) between the time series *Q* and *C*.

4. Re-Sampled Shape Averaging (RSA)

Shape averaging computation requires equation (4) for the signal averaging with iterative computation of DTW. A problem arises after performing averaging that causes some x-coordinates to change from an integer to a real number. It implies that some of the x-coordinates on the shape average signal are not aligned with the integer grid coordinates. This imposes a challenge to the subsequent DTW computations.

The original DTW requires that the coordinate must align with an integer grid, such that the distance matrix can be constructed. Although there is a special type of DTW called continuous DTW [8] that allows real-value coordinates, the algorithm incurs significant

amount of computational cost without showing superior performance over the original DTW.

We propose a method that uses cubic spline approximation to shift those off-the-grid samples back to the original integer grid while the original shape of both signals can be well preserved. This re-sampling technique also helps fix the length of the signal to a designated size. Therefore, the length of the signal can be bounded.

Let S be a set of signals of similar class. In Figure 5, the algorithm starts from the construction of a cost matrix, D, from all possible pairs of signals in S (line 1) and a weight vector, α, which gives equal weight to all signals in S. The cost, D_{ij}, is computed from the cumulative distance of the DTW computation between S_i and S_j where $i \neq j$ (lines 2-10). The minimum cost location (x,y) is located inside the cost matrix, D. This implies that S_x and S_y are the most similar signal pair. Therefore, they are the best candidates to be averaged (lines 13-14). The next step is to compute the shape average signal, A, using the following equation,

$$A_k = \left(\frac{\alpha_x \cdot w_{k,1} + \alpha_y \cdot w_{k,2}}{\alpha_x + \alpha_y}, \quad \frac{\alpha_x \cdot S_x(w_{k,1}) + \alpha_y \cdot S_y(w_{k,1})}{\alpha_x + \alpha_y} \right), \quad (5)$$

where α_x and α_y are the corresponding weights of S_x and S_y, respectively. The $w_{k,1}$ and $w_{k,2}$ are k^{th} warping pair inside the warping path, W_{xy}, between S_x and S_y (lines 17-18). The averaged sequence, A, is re-sampled to the designated size (uniform space between two adjacent data points) by using a cubic spline approximation (line 21). The next step is to replace S_x with A and remove S_y from S. The weight vector, α, is updated by $\alpha_x = \alpha_x + \alpha_y$ and set $\alpha_y = 0$ (lines 22-29). The DTW cost matrix, $D(x, *)$ and $D(*, x)$ where $*$ indicates all the signals inside S, is also updated (lines 31-36). The process repeats from the minimum cost location, averaging, re-sampling, and update until there is only one signal left inside S. This signal becomes the shape average for the given set of sequences.

```
Function RSA(NS, S)
   NS: Number of data points
   S: Training sequences {s1, s2, s3, …, sn}

1    // compute D for all possible pairs inside
2    S
3    for (i = 1; i ≤ n; i++)
4       αi = 1;
5       for (j = 1; j ≤ n; j++)
6          if i == j then Di,j = 0
7          else Di,j = DTW_cost(si, sj)
8          end
9       end
10   end
11
12   while (count(S) > 1)
13      // locate the minimum DTW cost in D
14      [min_value, x, y] = min(D)
15      W = DTW_path(sx, sy)
16
17      // compute the average using (6)
18      ave_t = (αx·W*,1 + αy·W*,2)/(αx + αx)
19      ave_value = (αx·Sx(W*,1) + αy·Sy(W*,2))/(αx
20   + αx)
21
22      // resample with spline into NS samples
23      snew = spline(ave_t, ave_value, [1:NS])
24      // update S, α, and D
25      S = remove(S, sx)
26      S = remove(S, sy)
27      S = add(S, snew)
28      αnew = αx + αy
29      α = remove(α, αx);
30      α = remove(α, αy);
31      α = add(α, αnew);
32      for (i = 1; i ≤ count(s); i++)
33         if i == new then Di,i = 0;
34         else Di,new = DTW_cost(si, snew)
35              Dnew,i = DTW_cost(snew, si)
36         end
37      end
   end
   return S
```

Figure 5. Pseudocode of our proposed DTW shape averaging algorithm.

5. Experiments

We conduct two experiments to test the accuracy/effectiveness of the re-sampled shape averaging on various datasets from the UCR classification/clustering archive [16] (shown in Table 1). The first experiment is to observe the accuracy, execution time, and memory requirement of the proposed method on the UCR datasets against NLAAF and T-1NN algorithms. The second experiment is to observe the accuracy and execution time on a larger synthetic dataset (≥ 3,000 sequences) among the RSA, NLAAF and T-1NN classification algorithms.

Table 1. Characteristics of the datasets used in our experiments

Dataset	Number of classes	Size of training set	Size of test set	Time series Length
Beef	5	30	30	470
Coffee	2	28	28	286
ECG	2	100	100	96
Face(four)	4	24	88	350
Oliveoil	4	30	30	570
Synthetic control	6	300	300	60
Trace	4	100	100	275

5.1. Experiment 1

We conduct an experiment to compare the accuracy of our proposed RSA method, against NLAAF and T-1NN classification algorithms. The RSA and NLAAF classification are performed in a similar manner, by

measuring the DTW distances between a test sequence against all class representatives. The predicted class belongs to the representative with the smallest distance. The only difference between RSA and NLAFF is the way the class representatives are created. The T-1NN classification takes a slightly different approach in the classification. This traditional 1NN method uses the entire training set as individual class representatives. Therefore, each test sequence is simply compared against every single time series sequence in the database using DTW. The class of the nearest training data becomes the predicted class. The accuracy is determined by the percentage of the number of positive (correct) prediction divided by a total number of the test sequences. The execution time measures the time in second that each algorithm takes to perform classification for the entire test dataset. The results are shown in Tables 2 and 3. In addition, the comparison of memory requirement, the amount of memory resource needed to store all class representatives, is shown in Table 4.

Table 2. The classification accuracy comparison among RSA, NLAAF, and DTW based T-1NN

Dataset	Accuracy (%)		
	RSA	NLAAF	T-1NN
Beef	**50**	43.33	50
Coffee	60.71	**64.29**	82.14
ECG	69	**70**	80
Face(four)	**78.41**	77.27	84.09
Oliveoil	**83.33**	80	86.67
Synthetic control	**88**	72.33	98.67
Trace	93	93	99

Table 3. The performance comparison among RSA, NLAAF, and DTW based T-1NN

Dataset	Execution Time (sec)		
	RSA	NLAAF	T-1NN
Beef	**1,369**	7,367	8,140
Coffee	**135**	1,645	1,738
ECG	**32**	508	1,517
Face(four)	**1,343**	3,242	7,048
Oliveoil	**1,497**	2,859	9,711
Synthetic control	**99**	700	4,736
Trace	**884**	18,134	18,413

Table 4. The storage requirement comparison among RSA, NLAAF, and DTW based T-1NN

Dataset	Memory Requirement (KB)		
	RSA	NLAAF	T-1NN
Beef	**3.67**	52.66	110.16
Coffee	**2.23**	22.11	62.56
ECG	**0.75**	10.33	75.00
Face(four)	**2.73**	19.89	65.63
Oliveoil	**4.45**	26.79	133.59
Synthetic control	**0.47**	12.84	140.63
Trace	**2.15**	52.45	214.84

This experiment shows the accuracy, execution time, and memory requirement for classification problem among RSA, NLAAF, and T-1NN algorithms. Table 2 shows the accuracy achievable for the three algorithms on each dataset. Note that the T-1NN algorithm is our gold standard on the accuracy assessment; it is from the fact that the T-1NN classification simply uses all training samples during the classification, which undoubtedly will give the highest achievable for each dataset. On the other hand, the RSA and NLAAF algorithms have only one representative per class. Hence, their accuracies are expectedly inferior to those of T-1NN. On Beef dataset, the proposed RSA classification is able to attain similar accuracy as T-1NN, while only two datasets, Coffee and ECG, are outperformed by NLAAF. This could come from the spline approximation that causes the shape average to lose some accuracy.

The execution time, in Table 3, shows the time it takes to execute the same datasets among the three algorithms. The results show that RSA classification outperforms both NLAAF and 1-TNN classifications in all cases. This is contributed from the smaller number of data points when compared with NLAAF, and the smaller number of class representatives when compared with 1-TNN algorithm. The larger the number of data points results in the longer the DTW computation. The more class representatives results in the larger number of DTW computations.

In Table 4, we compare the memory requirement of each algorithm. The memory requirement shows the amount of memory (in KB or 1,024 Bytes unit) each algorithm needs to store the class representatives for classification. The results show that our proposed RSA algorithm uses the least amount of memory in all cases (shown in bold) while the NLAAF and T-1NN classification uses more memory, on average, 14.73 and 87.43 times, respectively. The amount of RSA samples is bounded to a designated value while the NLAAF is unbounded and becomes longer in every averaging operation. The T-1NN method, on the other hand, is also bounded, but contains too many representatives (all training sequences), which would be quickly untenable for massive datasets.

5.2. Experiment 2

In this experiment, we perform the proposed RSA, NLAAF, and T-1NN algorithm on a large synthetic

dataset to observe accuracy and execution time. A synthetic 3-class dataset called CBF is used in this experiment. It composes of 3,000 training sequences and 100 test sequences. Each sequence contains 128 data points. For the RSA and NLAAF algorithms, the class representatives can be computed offline, before accuracy and execution time assessments begin. The results are shown in Table 5.

Table 5. The CBF classification comparison among RSA, NLAAF, and DTW based T-1NN

Assessment	CBF Dataset Classification		
	RSA	NLAAF	T-1NN
Accuracy (%)	**100**	98	100
Time (sec)	**28.73**	3,495.6	28,724.4

The results show that our proposed RSA can achieve the same accuracy as the T-1NN method for the CBF dataset classification, but the RSA execution time is significantly faster than both NLAAF and T-1NN algorithms. In particular, our RSA can achieve a 3-order-of-magnitude speedup over the T-1NN. Both RSA and T-1NN methods have the same amount of data points in their representatives, but RSA only have 3 representatives compared to 3,000 representatives used in the T-1NN algorithm. In the RSA based classification, each test sequence is measured against the three representative using DTW as a similarity measure. In the CBF test set, there are 100 sequences. Therefore, only 300 DTW computations are required for the entire execution. On the other hand, the T-1NN classification has 3,000 representatives (from the training dataset). As a result, with the similar 100 test sequences, 300,000 DTW computations are required. This implies that the RSA is theoretically 300,000 ÷ 300 = 1,000 times faster than T-1NN algorithm.

6. Conclusion

We have presented a novel RSA based on DTW. Our experiments demonstrate that the accuracy of our proposed RSA outperforms that of NLAAF, but mostly less accurate than the T-1NN method as expected. The main reason comes from the fact that the DTW based T-1NN method has multiple class representatives while the RSA has only one representative per class. However, the execution time of RSA is significantly less than both NLAAF and T-1NN methods. The memory requirement for RSA is the smallest among the three algorithms. It is not only bounded, but also manageable to an optimal amount of data points depending on the application. In fact, if we would like to improve the accuracy, we can always increase the number of class representatives, which is sometimes desirable especially when there are slight variety of signals within the same class. Our second experiment concludes that, with large dataset, RSA method can always outperform both NLAAF and T-1NN methods in terms of speed.

References

[1] K. Wang, and T. Gasser, "Synchronizing sample curves nonparametrically," *The Annals of Statistics*, vol. 27, 1999, pp. 439-460.

[2] S. Boudaoud, H. Rix, and O. Meste, "Integral shape averaging and structural average estimation: a comparative study," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, 2005, pp. 3644-3650.

[3] J.O. Ramsay, and X. Li, "Curve registration," *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, vol. 60, no. 2, 1998, pp. 351-363.

[4] E. Keogh, and M. Pazzani, "An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback," *Proceedings of the 4th International Conference of Knowledge Discovery and Data Mining*, pp. 239-241.

[5] L. Gupta, and R. Tammana, "A discrepancy measure for improved clusterin," *Pattern Recognition*, vol. 28, no. 10, 1995, pp. 1627-1634.

[6] A.E. Kunst, C.W.N. Looman, and J.P. Mackenbach, "Outdoor Air Temperature and Mortality in the Netherlands: A Time-Series Analysis," vol. 137, no. 3, 1993, pp. 331-341.

[7] L. Gupta, D.L. Molfese, R. Tammana, and P.G. Simos, "Nonlinear Alignment and Averaging for Estimating the Evoked Potential," *IEEE Transactions on Biomedical Engineering*, vol. 43, no. 4, 1996, pp. 348–356.

[8] M.E. Munich, and P. Perona, "Continuous Dynamic Time Warping for Translation-Invariant Curve Alignment with Applications to Signature Verification," *Proceedings of 7th IEEE International Conference on Computer Vision*, pp. 108–115.

[9] D. Berndt, and J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series," *Proceedings of AAAI Workshop on Knowledge Discovery in Databases*, pp. 359-370.

[10] C.A. Ratanamahatana, and E. Keogh, "Three Myths about Dynamic Time Warping," *Proceedings of SIAM International Conference on Data Mining*, pp. 506–510.

[11] H. Hou, and H. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, 1978, pp. 508-517.

[12] R. Keys, "Cubic Convolution Interpolation for Digital Image Processing," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, 1981, pp. 1153-1160.

[13] I. White, R. Thompson, and S. Brotherstone, "Genetic and Environmental Smoothing of Lactation Curves with Cubic Splines," *Journal of Dairy Science*, vol. 82, 1999, pp. 632-638.

[14] C.H. Reinsch, "Smoothing by spline functions," *Numerische Mathematik*, vol. 10, 1967, pp. 177-183.

[15] J. HAYES, and J. HALLIDAY, "The Least-squares Fitting of Cubic Spline Surfaces to General Data Sets," *IMA Journal of Applied Mathematics*, vol. 14, no. 1, 1974, pp. 89-103.

[16] E. Keogh, X. Xi, L. Wei, and C.A. Ratanamahatana, "UCR time series classification/ clustering page. http://www.cs.ucr.edu/~eamonn/time_series_data," 2008; www.cs.ucr.edu/~eamonn/time_series_data/.

ภาคผนวก ค

บทความทางวิชาการเรื่อง “Efficient Time Series Classification under Template Matching using Time Warping Alignment” โดยดารารัตน์ ศรีใส และโชติรัตน์ รัตนามหัทธนะ ในงานประชุมวิชาการนานาชาติครั้งที่ 4 “The 2009 International Conference on Computer Sciences and Convergence Information Technology” ซึ่งจัดขึ้น ณ เมืองโซล ประเทศเกาหลีใต้ ระหว่างวันที่ 24 พฤศจิกายน ถึง 26 พฤศจิการยน 2552

Efficient Time Series Classification under Template Matching using Time Warping Alignment

Dararat Srisai
Dept. of Computer Engineering
Chulalongkorn University
Bangkok, Thailand
51dsr@cp.eng.chula.ac.th

Chotirat Ann Ratanamahatana
Dept. of Computer Engineering
Chulalongkorn University
Bangkok, Thailand
ann@cp.eng.chula.ac.th

***Abstract*—One of the most widely used time series classification is the 1-Nerest Neighbor (1-NN) classification algorithm which utilizes Dynamic Time Warping (DTW) as a similarity measure. On large training data, though DTW is demonstrated to be highly accurate, its 1-NN classification typically takes significant amount of time to classify a given test sequence. The hotspot for this type of computation lies in the repeated DTW computations. In limited storage applications such as some real-time embedded systems, there might not be sufficient amount of resources for such computation. In this paper, we propose a novel template construction algorithm based on the Accurate Shape Averaging (ASA) technique. Each training class is represented simply by only one sequence. Our experiments show that the 1-NN classification with our proposed template construction algorithm can gain significant performance improvement while maintaining its high accuracy.**

Keywords-component; Time Series; 1-NN classification; Template Matching; Shape Averaging Data; Dynamic Time Warping

I. INTRODUCTION

Time series classification [1, 2] has become an interesting research because it can be applied to a variety of applications in many fields such as medical, financial, entertainment, and other industries. Therefore, researchers around the world have used many algorithms to tackle the time series classification problems such as Decision Tree [3], Artificial Neural Networks [4], Support Vector Machine (SVM) [5], etc. However, the most popular and highly accurate method is the DTW-based 1-NN classification technique.

The DTW-based 1-NN classification matches the query sequence (Q) to the candidate sequences (C_i). The matching involves computing similarity computation between the Q sequence and each C_i sequences using DTW distance. The classification outcome is determined by the class that belongs to the nearest candidate. It is obvious that each classification would involve multiple DTW computations. For example, if a given training data contains three classes and each class has 100,000 sequences. There will be 300,000 DTW distance computations for each 1-NN classification. There is also the storage issue for such a huge training data. The tiny real-time embedded system platforms may not have sufficient storage to accommodate all the data. Therefore, numerosity reduction [6-8] is the key to relieve the large training data problem.

As another alternative, template construction is an approach to reduce the number of training sequences by selecting some sequences as class representatives which can be implemented in various ways such as random sampling [9], ranking [10], etc. Recently, DTW based 1-NN classification is frequently used in template construction for time series data.

In 2006, Keogh et al. [11] have proposed a sequence reduction algorithm called Adaptive WARping winDow (AWARD) algorithm which relies on numerosity reduction technique. Its main concept is to find optimal candidate sequences among the training sequences within the same class. Though this algorithm can reduce the classification time, the accuracy becomes deteriorated. However, the authors state that their algorithm mainly focuses on determining the correct sequences instead of discovering appropriate norms for each class.

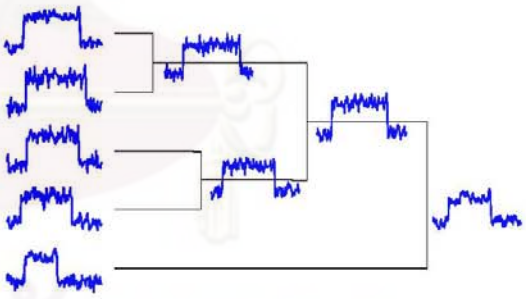


Figure 1. The PSA construction algorithm using hierachical clustering.

In 2009, Niennattrakul et al. [12] introduced a time series template construction algorithm called Prioritized Shape Averaging (PSA). This work constructs class templates based on a shape averaging technique, which uses hierarchical clustering to create an order of averaging. The purpose of the clustering is to prioritize the averaging pairs which can possibly perform on two sequences at a time

(shown in Fig. 1). The averaged time series sequences may suffer from additional data points and non-uniformly distributed data points. Therefore, some of the data points are discarded which leads to the loss of shape information. As a result, the template accuracy is suffered.

In this work, we proposed a novel template construction algorithm called Accurate Shape Averaging (ASA) which tries to minimize the number of class representative sequences through the use of Re-sampling Shape Averaging (RSA) [13] with Hybrid Dynamic Time Warping (HDTW) distance measure. The proposed algorithm is validated against AWARD, PSA, and Traditional One Nearest Neighbor (T-1NN) algorithms on 1-NN classification with DTW problems.

II. Background

A. Dynamic Time Warping

Dynamic Time Warping (DTW) [14, 15] is a shaped-based similarity measure algorithm that provides a distance measurement between two time series sequences. If there exist two times series Q and C of length m and n, respectively, where:

$$Q = q_1, q_2, \ldots, q_i, \ldots, q_m \tag{1}$$

$$C = c_1, c_2, \ldots, c_j, \ldots, c_n , \tag{2}$$

an $m \times n$ distance matrix, $D = \{d\}_{m \times n}$, can be constructed using a following equation,

$$d_{i,j} = (q_i - c_j)^2 , \tag{3}$$

where q_i and c_j are i^{th} and j^{th} elements in Q and C, respectively. A warping path, W, through the matrix that minimizes the cumulative distance provides an optimal alignment between Q and C (shown in Fig. 2). The warping path's element, w_k, equals to $(i, j)_k$ where $1 \le i \le m$, $1 \le j \le n$, $1 \le k \le K$, and $\max(m, n) \le K \le m+n-1$. The optimal path, W_o, is the path that minimizes the warping cost,

$$DTW(Q,C) = \sqrt{\sum_{k=1}^{K} D(w_k)} . \tag{4}$$

This path is discovered via dynamic programming to evaluate the minimum cumulative distance, $\gamma_{i,j}$, from three adjacent elements, $\gamma_{i,j} = d_{i,j} + \min\{\gamma_{i-1,j-1}, \gamma_{i-1,j}, \gamma_{i,j-1}\}$ with additional conditions as follows.

Boundary conditions: The path must start on $w_1 = (1, 1)$ and end on $w_K = (m, n)$.

Continuity condition: The i and j indices can only increase by 0 or 1 on each step along the path.

Monotonic condition: The warping path can only move forward in time (i and j monotonically increase). Given $w_k = (q, c)$ and $w_{k-1} = (q', c')$. The values $q - q'$ and $c - c'$ must be greater than or equal to zero.

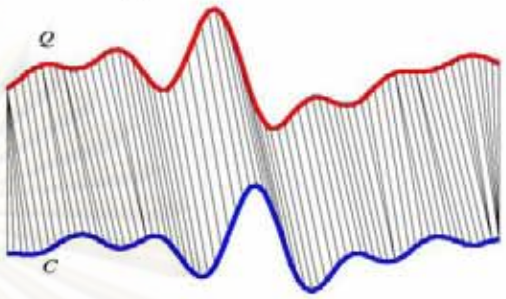


Figure 2. An optimal alignment between two time series, Q and C.

B. Derivative Dynamic Time Warping

The main concept of the Derivative Dynamic Time Warping (DDTW) [16] is to base the similarity measure on the rate of change or the first order derivative instead of the value of the data points. In some applications, the data is likely to be susceptible to offset in the signal or some low frequency noises. Therefore, the first order derivative will provide a better representation. In the implementation, the $m \times n$ distance matrix, $D = \{d\}_{m \times n}$, can be constructed using a following equation,

$$d_{i,j} = (q_i' - c_j')^2 . \tag{5}$$

where q_i' and c_j' are the i^{th} and j^{th} elements in the first order derivative of Q and C, respectively. In the original DDTW algorithm [16], a unique numerical differentiation method is used with a $O(h^2)$ accuracy where h is step size or the sampling interval. The estimated first order derivative can be computed as follows,

$$q_i' = \frac{(q_i - q_{i-1}) + ((q_{i+1} - q_{i-1})/2)}{2} , \tag{6}$$

where q_{i+1} and q_{i-1} are the neighboring data points of q_i. The rest of the computations are identical to the original DTW implementation.

The drawback of this method is that it is only attentive to the slope of the data. Therefore, it can only capture some similarities, and it could be susceptible to high frequency noises.

III. Hybrid Dynamic Time Warping

One problem arisen from the use of either the original DTW or the DDTW is that both are only attentive to one type of shape information. The original DTW only uses the data point value as the only source of information while the DDTW only uses the slope information of the data points. In this paper, we propose a Hybrid Dynamic Time Warping (HDTW) which combines the similarity captured by both the

value and the slope of the data points. Our implementation of the first order derivative uses the five-point stencil method which provides a better $O(h^4)$ accuracy instead of $O(h^2)$ provided by the original DDTW where h is the step size or the sampling interval. The estimated first order derivative using five-point stencil can be computed as follows,

$$f'(x) \approx \frac{-f(x+2)+8f(x+1)-8f(x-1)+f(x-2)}{12}, \quad (7)$$

where $f(x+2)$, $f(x+1)$, $f(x-1)$, and $f(x-2)$ are the neighboring data points of $f(x)$. Since the value and the slope of the data points are of different norms and scales, the distance matrix calculation will require normalization to balance the fusion of both parameters. The normalized distance equation is proposed as,

$$d_{i,j} = \left(\frac{d_{i,j}^{(0)} - \mu_0}{\sigma_0}\right) + \left(\frac{d_{i,j}^{(1)} - \mu_1}{\sigma_1}\right), \quad (8)$$

where $d^{(0)}$ and $d^{(1)}$ are the distances computed from the value of the data points and first order derivative of the data points, respectively. The μ_0, μ_1, σ_0, and σ_1 are the means and standard deviations of the distance $d^{(0)}$ and $d^{(1)}$, respectively.

The HDTW uses both the data point values and the slopes of the data points in the construction of the DTW distance matrix. Therefore, more similarity can be captured. As a result, it provides greater accuracy in similarity measure and shape averaging.

IV. Template Construction Algorithm

A. Accurate Shape Averaging (ASA)

1) Locate a Minimum Distance Pair

Shape averaging algorithms can typically be used for template construction. Most shape averaging algorithms can only compute the average between two sequences. If there are more than two time series sequences, the first averaging pair must be determined before subsequent averaging operations can be performed. In practice, the averaging step is performed on the most similar pair until there is only one sequence left. The most similar pair implies that the distance between the two sequences is at minimum. One way to determine the distance between the two sequences is to use DTW as the similarity measure. Lower bounding [17] technique can be integrated into the DTW implementation to reduce the amount of computation required from all possible pairs.

In each averaging operation, the most similar pair of the sequences is replaced by their average. The operation continues until there is only one sequence left. In the subsequent operations, there is no need to compute every distance pairs possible. Only the pairs involved in the previously averaged sequence are required to update the distances. To keep it simple, the data structure maintains the updated record of the nearest neighbor and the distance between them.

2) Time Warping Alignment

The DTW technique is used to find the most similar time series sequences and also the alignment between the two sequences. In traditional DTW operation, the distance is determined by the value of the data points within the sequences. This can capture some similarity, but completely ignores the rate of change of all the data points. A DDTW uses only the first order derivative of the data points in the computation and can capture different types of similarity. Our proposed work, a Hybrid DTW (HDTW) which combines both the data points value and the first order derivative of the data points in the distance calculation, can capture better similarity. There are two outputs from any DTW operation: cost and alignment. The cost represents the distance between the two input sequences while the alignment matches the corresponding data points between the two input sequences. This alignment provides the key to an accurate shape average.

For example, a sequence Q = [1 2 2 3 3 2 2 1 1] and a sequence C = [1 1 1 2 2 3 3 2 1], the cumulative distance, D, and the warping path, W = {(1,1), (1,2), (1,3), (2,4), (3,5), (4,6), (5,7), (6,8), (7,8), (8,9), (9,9)}, are shown in Fig. 3.

D_{ij}									Q_i	i
23.8	22.9	23.5	22.8	22.8	28.0	20.7	12.6	5.2	1	9
23.8	22.8	25.8	20.4	20.4	26.2	12.8	6.8	4.4	1	8
22.6	21.9	24.6	15.0	15.0	17.7	6.0	4.3	4.6	2	7
20.6	20.1	22.8	12.1	12.1	13.1	4.3	3.4	4.6	2	6
18.6	18.3	20.0	9.2	9.2	7.5	2.5	4.3	10.0	3	5
11.6	12.0	9.9	3.9	3.9	2.4	6.7	16.9	25.6	3	4
4.8	5.0	4.3	2.4	2.4	4.2	9.1	16.3	20.2	2	3
2.9	3.2	3.5	2.4	2.4	4.2	9.1	16.3	20.2	2	2
1.0	2.4	2.4	4.4	6.4	13.2	23.7	33.6	36.8	1	1
C_j	1	1	1	2	2	3	3	2	1	
j	1	2	3	4	5	6	7	8	9	

Figure 3. The cumulative distance matrix, D, and the warping path (shown in shaded regions) between two time series Q and C.

From the warping path, the sequence alignment can be constructed as shown in Fig. 4.

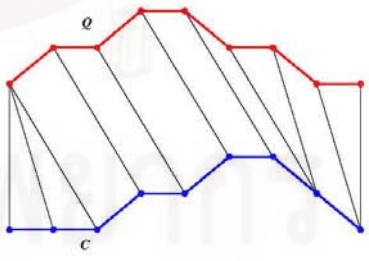


Figure 4. The corresponding HDTW alignment between the two time series, Q and C.

3) *Shape Averaging*

The traditional shape averaging using arithmetic mean cannot provide accurate result because the two sequences are not properly aligned (as illustrated in Fig. 5).

To resolve the problem, HDTW-based shape averaging must first determine the optimal sequence alignment before proceeding with the averaging calculation. The shape average, S, is computed by the following formula,

$$S_k = \left(\frac{w_{k,1} + w_{k,2}}{2}, \frac{Q(w_{k,1}) + C(w_{k,2})}{2} \right), \qquad (8)$$

where $w_{k,1}$ and $w_{k,2}$ are k^{th} index pair of the warping path, W. The average result, S, is a sequence {(1,1), (1.5,1), (2,1), (3,2), (4,2), (5,3), (6,3), (7,2), (7.5,2), (8.5,1), (9,1)}.

The HDTW computation provides an alignment where the corresponding samples between the two sequences are identified. Therefore, shape averaging can be performed with better accuracy.

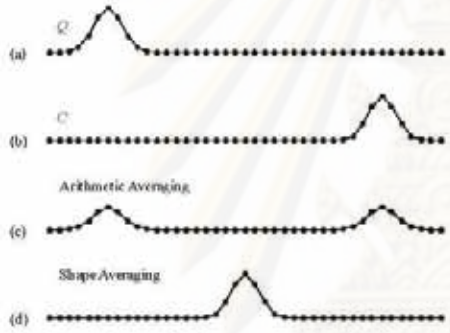


Figure 5. Time series averaging between sequences Q (a) and C (b) using arithmetic averaging (c) and the shape averaging (d).

4) *Re-sampling*

The averaging results tend to have more data points than the original sequences due to singularity existed in the alignment. The singularity condition happens when multiple data points on a sequence are paired with one data point on the other sequence. This will lead to additional data points on the averaging result as shown in Fig. 6(a). Initially, the time series Q and C both contain 9 data points. Their average shown in dotted line contains 11 data points. In fact, some of these data points (2nd, 9th and 10th points) may not aligned with the original data grid as shown in Fig. 6(a). Therefore, repeated averaging would result in a longer averaging result with many more data points unaligned with the initial data grid. This problem can be overcome by using a cubic spline approximation to re-sample the sequence back to the original grid location as shown in Fig. 6(b). Therefore, the size of the averaging result remains constant while all data points are grid aligned.

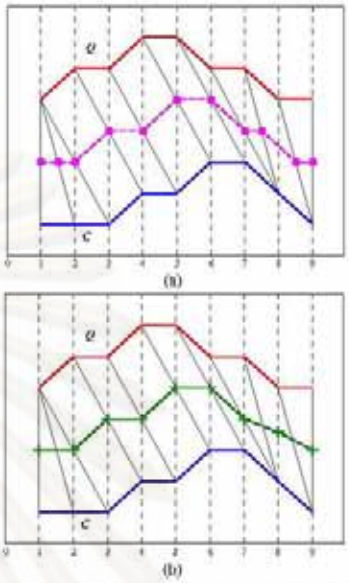


Figure 6. Illustrations of (a) the shape average (dotted line with "•" markers) between two time series, Q and C. (b) The cubic spline approximation (solid line with "+" markers).

V. EXPERIMENTS

We conduct two experiments to examine the validity and the accuracy/effectiveness of the templates computed from the proposed algorithm on various datasets from the UCR datasets classification/clustering archive [18] (shown in Table I).

TABLE I. CHARACTERISTICS OF THE DATASETS USED IN OUR EXPERIMENTS

Datasets	Number of classes	Size of training set	Size of test set	Time series length
50Words	50	450	455	270
Adiac	37	390	391	176
CBF	3	30	900	128
Coffee	2	28	28	286
ECG	2	100	100	96
Face (all)	14	560	1690	131
Face(four)	4	24	88	350
Gun-Point	2	50	150	150
OSU Leaf	6	200	242	427
Swedish Leaf	15	500	625	128
Synthetic Control	6	300	300	60
Trace	4	100	100	275
Two Patterns	4	1000	4000	128

A. *Experiment I*

The validity experiment uses Intra-class Distance, d, to measure the distance from the shape average to all the sequences of the same class by using the following formula,

$$d = \frac{1}{N} \sum_{c \in C} \left(\frac{1}{M_c} \sum_{q \in Q_c} DTW(c, q) \right), \quad (9)$$

where N is the number of classes in the dataset and C is the set of all classes. M_c refers to the number of sequences belong to class c and Q_c is the set of all sequences belong to class c. We compare these distances between the templates of our proposed ASA and the existing PSA algorithms.

B. Experiment II

We conduct an experiment to evaluate the accuracy of each algorithm. For the PSA algorithm, the templates are computed using hierarchical shape averaging on the training set. Each test sequence will be measured using DTW distance. The classification results are taken from the class of the nearest neighbor. The result will be compared with the original class of each test sequence. The accuracy η is computed as follows,

$$\eta = \frac{n_p}{n_p + n_n}, \quad (10)$$

where n_p and n_n are the number of positive and negative classifications. For the AWARD algorithm, the templates are sorted by using the Naïve Rank Reduction algorithm before being reduced by taking the higher ranked portion of the templates that contains every class. The reduced templates are used to classify the test sequences.

VI. Results and Discussion

A. Results of Experiment I

The results of the experiments are listed in Table II. The intra-class distances represent the average distance from each class' shape average to all its members. The smaller distance implies that this template is closer to the ideal template. The example templates are shown in Fig. 7.

TABLE II. The Intra-class Distances Comparison between ASA and PSA Algorithms.

Datasets	ASA	PSA
50Words	**2.67**	2.75
Adiac	**0.36**	0.59
CBF	**3.64**	3.87
Coffee	**0.57**	0.89
ECG	**2.60**	2.90
Face (all)	**3.89**	6.01
Face(four)	**5.58**	5.35
Gun-Point	**2.80**	2.92
OSU Leaf	**6.47**	6.82
Swedish Leaf	**1.29**	2.08
Synthetic Control	**3.46**	3.59
Trace	**1.35**	1.53
Two Patterns	**3.48**	3.85

The results in Table II show that the ASA Intra-class Distance is smaller than that of the PSA method. The smaller distances implies that our proposed ASA method produce a more compact group than the PSA method. This desirable feature is due to the use of cubic spline resampling to approximate the shape averaging results back to align with the original grid. In PSA method, the averaging results are resampled by choosing some data points with a specific rule. Therefore, its template is a rougher shape average than our proposed ASA.

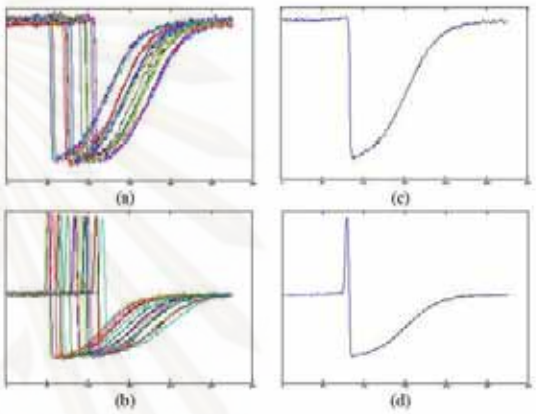


Figure 7. Illustrations of (a), (b) example sequences from Trace dataset; (c), (d) the templates from our proposed ASA.

B. Results of Experiment II

The accuracy and storage requirement results listed in Table III and Table IV show that our proposed ASA algorithm is superior to other algorithms in most cases. The winning results are shown in bold. The results are also compared with the T-1NN algorithm which is considered to be the upper bound of achievable accuracy. In this case, it is obvious that a proper shape averaging algorithm could yield a better result.

TABLE III. The Classification Accuracy Comparison among ASA, PSA, AWARD, and T-1NN.

Datasets	Accuracy (%)			
	ASA	*PSA*	*AWARD*	*T-1NN*
50Words	**71.21**	40.66	72.00	69.01
Adiac	**70.84**	33.50	58.00	60.36
CBF	**96.33**	95.11	61.00	99.67
Coffee	**100.00**	**100**	46.00	82.14
ECG	**80.00**	61.00	76.00	77.00
Face (all)	**87.22**	62.19	77.00	80.77
Face(four)	**76.14**	67.50	46.00	82.96
Gun-Point	**77.33**	70.00	58.00	90.67
OSU Leaf	**58.26**	25.51	38.00	59.09
Swedish Leaf	**88.96**	87.50	71.00	79.20
Synthetic Control	**97.33**	95.67	92.00	99.33
Trace	**100.00**	**100**	58.00	100.00
Two Patterns	**99.40**	93	45.00	100.00

TABLE IV. THE STORAGE REQUIRMENT COMPARISON AMONG ASA, PSA, AWARD, AND T-1NN.

Datasets	Storage Requirement (Number of Sequences)			
	ASA	*PSA*	*AWARD*	*T-1NN*
50Words	**50**	**50**	387	450
Adiac	**37**	**37**	330	390
CBF	**3**	**3**	4	30
Coffee	**2**	**2**	**2**	28
ECG	**2**	**2**	**2**	100
Face (all)	**14**	**14**	43	560
Face(four)	**4**	**4**	5	24
Gun-Point	**2**	**2**	3	50
OSU Leaf	**6**	**6**	15	200
Swedish Leaf	**15**	**15**	61	500
Synthetic Control	**6**	**6**	8	300
Trace	**4**	**4**	13	100
Two Patterns	**4**	**4**	6	1000

This experiment shows the accuracy and memory requirement for 1-NN DTW classification problem among our proposed ASA, PSA, AWARD, and T-1NN algorithms. Table III shows the accuracy results of the four algorithms on each dataset, it is obvious that our proposed ASA algorithm can outperform both PSA and AWARD algorithms for the 1-NN DTW classification. Both ASA and PSA algorithms are shape average based computation, but ASA uses the Hybrid DTW instead of the conventional DTW used by PSA. In addition, the ASA is more accurate in the shape averaging operation through the use of cubic spline approximation. To compare with AWARD which is a template reduction method, there exists only one case (50Words dataset) that its accuracy is about one percent better than that of ASA, This could come from the fact that the ASA algorithm has only one template per class.

The storage requirement in Table IV shows the amount of storage required for each algorithm to store the templates for classification. The results shows that our proposed ASA and the PSA algorithms use the least amount of storage (shown in bold) because both methods use only one template per class while the AWARD and T-1NN algorithms require much more storage. Because AWARD algorithm uses parts of the training set as its templates, multiple templates are often required to accurately represent a class. For example, it requires 7.74 times larger storage comparing to ASA while the T-1NN which uses all the training set as its templates requires as large as 24.88 times the storage of ASA.

VII. CONCLUSION

The proposed ASA algorithm has been shown to achieve better 1-NN DTW classification accuracy than the PSA and AWARD algorithms. The template based method can be more accurate than the T-1NN which is believed to be the upper bound for its classification accuracy. Furthermore, the computational complexity for the template based methods is much lower.

REFERENCES

[1] E. Keogh and M. J. Pazzani, "An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback," in Proceedings of 4th International Conference of Knowledge Discovery and Data Mining New York, NY,USA, 1998, pp. 239–243.

[2] C. A. Ratanamahatana and E. Keogh, "Making Time-Series Classification More Accurate Using Learned Constraints," in Proceedings of SIAM International Conference on Data Mining (SDM), Lake Buena Vista, FL, USA, 2004, pp. 11–22.

[3] J. J. Rodríguez and C. J. Alonso, "Interval and dynamic time warping-based decision trees," in Proceedings of the 2004 ACM symposium on Applied computing (SAC), ACM New York, NY, USA, 2004, pp. 548–552.

[4] A. Nanopoulos, R. Alcock, and Y. Manolopoulos, "Feature-based Classification of Time-series Data," International Journal of Computer Research, vol. 10, pp. 49–61, 2001.

[5] Y. Wu and E. Y. Chang, "Distance-function design and fusion for sequence data," in Proceedings of the thirteenth ACM international conference on Information and knowledge management, ACM New York, NY, USA, 2004, pp. 324–333.

[6] H. Brighton and C. Mellish, "Advances in instance selection for instance-based learning algorithms," Data mining and knowledge discovery, vol. 6, pp. 153–172, 2002.

[7] D. R. Wilson and T. R. Martinez, "Reduction techniques for instance-based learning algorithms," Machine Learning, vol. 38, pp. 257–286, 2000.

[8] E. Pekalska, R. P. W. Duin, and P. Paclik, "Prototype selection for dissimilarity-based classifiers," Pattern Recognition, vol. 39, pp. 189–208, 2006.

[9] D. B. Skalak, "Prototype and feature selection by sampling and random mutation hill climbing algorithms.," in Proceedings of the Eleventh International Conference on Machine Learning (ML94), 1994, pp. 293–301.

[10] D. R. Wilson and T. R. Martinez, "Instance pruning techniques," in Proceedings of the Fourteenth International Conference (ICML'97), San Francisco, CA, 1997, pp. 403–411.

[11] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana, "Fast Time Series Classification Using Numerosity Reduction," in Proceedings of 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 2006, pp. 1033–1040.

[12] V. Niennattrakul and C. Ratanamahatana, "Shape Averaging under Time Warping," in 6th International Conference on Electrical Engineering/Electronics, Computer,Telecommunications, and Information Technology (ECTI-CON 2009) Pattaya, Thailand, 2009.

[13] D. Srisai and C. A. Ratanamahatana, "Time Series Shape Averaging Using Time-Warping Alignment with Re-sampling," in 6th International Joint Conference on Computer Science and Software Engineering (JCSSE2009), Phuket, Thailand, 2009, pp. 286-291.

[14] D. Berndt and J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series," in Proceedings of AAAI Workshop on Knowledge Discovery in Databases, 1994, pp. 359-370.

[15] 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.

[16] E. Keogh and M. J. Pazzani, "Derivative Dynamic Time Warping," in Proceeding of the First SIAM International Conference on Data Mining (SDM'2001), Chicago, USA, 2001, pp. 5–7.

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

[18] E. Keogh, X. Xi, L. Wei, and C. A. Ratanamahatana, "The UCR Time Series Classification/Clustering Homepage," 2008; www.cs.ucr.edu/~eamonn/time_series_data/.

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวดารารัตน์ ศรีใส เกิดวันที่ 30 มีนาคม พ.ศ. 2528 สำเร็จการศึกษาระดับมัธยมศึกษาที่โรงเรียนเบ็ญจะมะมหาราช จากนั้นจึงเข้าศึกษาต่อที่คณะวิศวกรรมศาสตร์ มหาวิทยาลัยมหิดล ในปีการศึกษา 2547 และในปีการศึกษา 2550 จึงสำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ และเข้าศึกษาในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2551