

การค้นหาเพลงโดยการร้องทำนองอย่างแม่นยำ
โดยใช้การปรับขนาดเอกรูป ไดนามิกไทม์วอร์ปิง และฟังก์ชันขอบเขตล่าง

นายไวยณ์วุฒิ เอื้อจงประสิทธิ์

สถาบันวิทยบริการ

จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2550

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

ACCURATE MUSIC SEARCH BY HUMMING USING UNIFORM SCALING,
DYNAMIC TIME WARPING, AND LOWER BOUNDING FUNCTION

Mr. Waiyawuth Euachongprasit

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 2007

Copyright of Chulalongkorn University

ไวยณฺ์วุฒิ เอื้อจงประสิทธิ : การค้นหาเพลงโดยการร้องทำนองอย่างแม่นยำ โดยใช้การปรับขนาดเอกรูป ไดนามิกไทม์วอร์ปิง และฟังก์ชันขอบเขตล่าง. (ACCURATE MUSIC SEARCH BY HUMMING USING UNIFORM SCALING, DYNAMIC TIME WARPING, AND LOWER BOUNDING FUNCTION) อาจารย์ที่ปรึกษา : อ.ดร.โซติรัตน์ รัตนามหัทธนะ, 98 หน้า.

ระบบค้นหาเพลงโดยการร้องทำนองเป็นระบบที่อนุญาตให้ผู้ใช้สามารถค้นหาเพลงจากการร้องทำนองหรือการร้องเพลงในบางส่วนของเพลงที่ต้องการ อย่างไรก็ตามวิธีที่น่าจะสามารถนำมาประยุกต์ใช้กับระบบค้นหาเพลงโดยการร้องทำนองได้อย่างเหมาะสมมากที่สุดในปัจจุบัน ก็คือ วิธีปรับขนาดกับไทม์วอร์ปิง ซึ่งเป็นวิธีที่ผสมผสานการปรับขนาดเอกรูปและไดนามิกไทม์วอร์ปิงเข้าไว้ด้วยกัน อย่างไรก็ตามวิธีดังกล่าวก็ยังไม่สามารถที่จะค้นหาเพลงภายใต้ฐานข้อมูลขนาดใหญ่ได้อย่างแม่นยำ ทั้ง ๆ ที่วิธีดังกล่าวดูเหมือนว่าจะสามารถรองรับความแปรผันต่าง ๆ ของผู้ใช้ได้อย่างครบถ้วนไม่ว่าจะเป็นการร้องทำนองที่มีจังหวะช้าหรือเร็ว รวมทั้งการร้องทำนองที่มีจังหวะไม่คงที่ แต่การละเลยถึงความสำคัญในการแปลงข้อมูลให้เป็นบรรทัดฐาน ทำให้การเปรียบเทียบข้อมูลระหว่างเสียงร้องทำนองและเพลงจึงไม่สามารถทำได้ อย่างแม่นยำ ดังนั้น เพื่อที่จะสามารถค้นหาเพลงจากเสียงร้องทำนองได้อย่างแม่นยำ งานวิจัยนี้ จึงเสนอวิธีปรับขนาดเอกรูปและไดนามิกไทม์วอร์ปิงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน รวมไปถึงฟังก์ชันขอบเขตล่างของวิธีดังกล่าว เพื่อที่จะสามารถค้นหาเพลงได้อย่างมีประสิทธิภาพ ยิ่งไปกว่านั้น งานวิจัยนี้ยังได้ประยุกต์ใช้วิธีการค้นหาข้อมูลที่มีประสิทธิภาพกับวิธีที่นำเสนอ เพื่อที่จะเพิ่มความเร็วในการค้นหาข้อมูล สุดท้าย จากการทดลองในงานวิจัยนี้ ซึ่งใช้แฟ้มข้อมูลเสียงร้องทำนองจำนวน 100 แฟ้ม ที่ได้จากกลุ่มตัวอย่างจำนวน 15 คน ทั้งเพศชายและเพศหญิง ภายใต้ฐานข้อมูลขนาด 1,000 เพลง ความแม่นยำในการค้นหาเพลงจากเสียงร้องทำนองของวิธีที่นำเสนอเท่ากับ 77 เปอร์เซ็นต์ ด้วยระยะเวลาในการค้นหาเพลงเฉลี่ยต่ำกว่า 10 วินาที ซึ่งวิธีที่นำเสนอนั้นสามารถค้นหาข้อมูลได้อย่างถูกต้องแม่นยำกว่าวิธีอื่น ๆ ที่ได้มีการทดลองมาทั้งหมด รวมไปถึงวิธีปรับขนาดกับไทม์วอร์ปิง

ภาควิชา วิศวกรรมคอมพิวเตอร์
สาขาวิชา วิศวกรรมคอมพิวเตอร์
ปีการศึกษา 2550

ลายมือชื่อนิสิต
ลายมือชื่ออาจารย์ที่ปรึกษา
.....

5070459321 : MAJOR COMPUTER ENGINEERING

KEY WORD : TIME SERIES RETRIEVAL / QUERY BY HUMMING / DYNAMIC TIME WARPING / UNIFORM SCALING / LOWER BOUNDING FUNCTION

WAIYAWUTH EUACHONGPRASIT : ACCURATE MUSIC SEARCH BY HUMMING USING UNIFORM SCALING, DYNAMIC TIME WARPING, AND LOWER BOUNDING FUNCTION. THESIS ADVISOR : CHOTIRAT RATANAMAHATANA, Ph.D., 98 pp.

Query-by-humming system is a system that allows users to conveniently search for the music by just humming or singing some parts of the music. The most promising method for the query-by-humming system, Scaled and Warped Matching or SWM, which is a combination of Uniform Scaling and Dynamic Time Warping, still cannot accurately search for the music from very large databases, although SWM appears to deal with almost all users' variations properly such as singing or humming in faster, slower or inconsistency tempos. The rationale behind this inaccuracy is its failure to realize the importance of normalization in SWM. Thus, to achieve accurate music search by humming, this research proposes Uniform Scaling and Dynamic Time Warping under normalization transformation, including its lower bounding function in order to efficiently search for the music. Moreover, this research also applies an efficient search algorithm to the proposed method so as to speed up the calculation. Finally, according to the experiments in this research, which use 100 sung queries collected from 15 subjects of both genders under 1,000 songs, the accuracy of the proposed method is 77 percent with an average searching time below 10 seconds, giving significant improvement over all rival methods, including SWM.

Department Computer Engineering. Student's signature
 Field of study Computer Engineering. Advisor's signature
 Academic year 2007

กิตติกรรมประกาศ

ตลอดระยะเวลาในการจัดทำวิทยานิพนธ์ฉบับนี้ ได้มีอุปสรรคต่าง ๆ เกิดขึ้น นานัปการ อันเป็นบทเรียนที่ทรงคุณค่ายิ่งแก่ผู้จัดทำ เพื่อที่จะได้ฝึกฝน เรียนรู้ และแก้ไขปัญหา ตลอดจนได้เพิ่มพูนทักษะต่าง ๆ ที่จำเป็นสำหรับการวิจัย ซึ่งทั้งหมดนี้ล้วนเป็นปัจจัยที่ช่วย ส่งเสริมและผลักดันศักยภาพให้แก่ผู้จัดทำเป็นอย่างมาก อย่างไรก็ตาม วิทยานิพนธ์ฉบับนี้ จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี ถ้าขาดแรงสนับสนุนจากบุคคลหลายฝ่าย ซึ่งข้าพเจ้าซาบซึ้ง ในความกรุณาเหล่านี้เป็นอย่างล้นพ้น และใคร่ขอใช้เนื้อหาในกิตติกรรมประกาศของวิทยานิพนธ์ ฉบับนี้ เป็นสื่อกลางในการแสดงความขอบพระคุณอย่างสุดซึ้งจากผู้จัดทำ

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

ขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ฉบับนี้ ที่ให้ข้อคิดและข้อเสนอแนะ ต่าง ๆ อันเป็นประโยชน์อย่างยิ่งในการพัฒนาคุณภาพของวิทยานิพนธ์ฉบับนี้ ซึ่งคณะกรรมการ สอบวิทยานิพนธ์นั้น ประกอบไปด้วย ศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล ผู้ช่วยศาสตราจารย์ ดร.โปรดปราน บุญยพุกกณะ และผู้ช่วยศาสตราจารย์ ดร.ชาญยศ ปลื้มปิติวิริยะเวช

ขอบคุณเพื่อน ๆ ในห้องปฏิบัติการทุกคนที่ช่วยให้ชีวิตในการทำวิจัยมีสีสันและ มีความหมายมากยิ่งขึ้น รวมทั้งช่วยเสนอแนวคิดต่าง ๆ ในการแก้ไขปัญหา และให้ความร่วมมือ ในการเก็บตัวอย่างเสียงเพื่อใช้สำหรับการทดลองในงานวิทยานิพนธ์ฉบับนี้

สุดท้ายที่ขาดเสียมิได้ ขอบพระคุณครอบครัวที่น่ารักของผู้จัดทำทุก ๆ คน ที่ เป็นกำลังใจ และให้การสนับสนุนทุกสิ่งทุกอย่างอย่างดีเสมอมา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ	ฉ
สารบัญ	ช
สารบัญภาพ	ฌ
สารบัญตาราง	ฎ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 ขอบเขตของการวิจัย	2
1.4 ประโยชน์ที่ได้รับ	3
1.5 วิธีดำเนินการวิจัย	3
1.6 ผลงานตีพิมพ์จากงานวิจัย	4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	5
2.1 ทฤษฎีที่เกี่ยวข้อง	5
2.2 ความรู้ทางดนตรี	5
2.3 ขั้นตอนวิธีการตรวจจับระดับเสียง	6
2.4 รูปแบบแฟ้มข้อมูลมิติมาตรฐาน	6
2.5 ไดนามิกไทม์วอร์ปิง	7
2.5.1 เงื่อนไขบังคับโดยรวม	9
2.5.2 ฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปิง	10
2.6 การปรับขนาดเอกกรุป	12
2.6.1 ฟังก์ชันขอบเขตล่างของการปรับขนาดเอกกรุป	13
2.7 การปรับขนาดกับไทม์วอร์ปิง	15
2.7.1 ฟังก์ชันขอบเขตล่างของการปรับขนาดกับไทม์วอร์ปิง	17
2.8 การลดมิติข้อมูล	18
2.8.1 การลดมิติข้อมูลแบบพีเอเอ	18
2.8.2 ฟังก์ชันขอบเขตล่างของข้อมูลที่ถูกลดมิติแบบพีเอเอ	20
2.9 งานวิจัยที่เกี่ยวข้อง	21
2.9.1 ระบบค้นหาเพลงโดยการร้องทำนองที่ใช้การแทนข้อมูลด้วยสายอักขระ	21

2.9.2 ระบบค้นหาเพลงโดยการร้องทำนองที่ใช้การแทนข้อมูลด้วยอนุกรมเวลา ...	23
บทที่ 3 การค้นหาเพลงโดยการร้องทำนอง	26
3.1 การสกัดคุณลักษณะจากเสียงร้องทำนอง	26
3.2 การสกัดคุณลักษณะจากเพลงในฐานข้อมูล	28
3.3 การค้นหาเพลงจากเสียงร้องทำนองภายใต้ฐานข้อมูลอนุกรมเวลาขนาดใหญ่	29
3.3.1 การปรับขนาดเอกรูปและไดนามิกโทมวอร์ปปีงที่รองรับการแปลงข้อมูล ให้เป็นบรรทัดฐาน	29
3.3.2 ฟังก์ชันขอบเขตล่างของการปรับขนาดเอกรูปและไดนามิกโทมวอร์ปปีง ที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน	32
3.3.3 การค้นหาเพลงในฐานข้อมูลอนุกรมเวลาที่มีขนาดใหญ่	40
บทที่ 4 การทดลองและวิเคราะห์ผล	49
4.1 การทดลองเพื่อวิเคราะห์คุณภาพฟังก์ชันขอบเขตล่างที่นำเสนอ	49
4.2 การทดลองเกี่ยวกับการวิเคราะห์ประสิทธิภาพของวิธีการค้นหาเพลงโดยการ ร้องทำนองที่นำเสนอ	53
4.2.1 การทดลองเพื่อวิเคราะห์ประสิทธิภาพของวิธีที่นำเสนอ	55
4.2.2 การทดลองเพื่อวิเคราะห์ความแม่นยำของวิธีที่นำเสนอเมื่อเปรียบเทียบกับวิธีอื่น ๆ	63
4.2.3 การทดลองเพื่อวิเคราะห์ความเร็วในการค้นหาเพลงเมื่อเปรียบเทียบกับวิธีอื่น ๆ	67
บทที่ 5 สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ	69
5.1 สรุปผลการวิจัย	69
5.2 ข้อเสนอแนะ	70
รายการอ้างอิง	72
ภาคผนวก	75
ภาคผนวก ก	76
ภาคผนวก ข	84
ประวัติผู้เขียนวิทยานิพนธ์	98

สารบัญภาพ

หน้า

รูปที่ 2.1	การคำนวณค่าระยะทางโดยใช้วิธีไดนามิกไทม์วอร์ปปีง	7
รูปที่ 2.2	การปรับแนวในการหาระยะทางอย่างไม่เหมาะสมของวิธีไดนามิกไทม์วอร์ปปีง	8
รูปที่ 2.3	เงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะ	9
รูปที่ 2.4	การคำนวณค่าระยะทางขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปปีง	11
รูปที่ 2.5	การคำนวณค่าระยะทางด้วยวิธีการปรับขนาดเอกรูป	12
รูปที่ 2.6	การคำนวณค่าระยะทางขอบเขตล่างของการปรับขนาดเอกรูป	14
รูปที่ 2.7	การเปรียบเทียบวิธีคำนวณค่าระยะทางแบบต่าง ๆ	15
รูปที่ 2.8	การคำนวณค่าระยะทางขอบเขตล่างของการปรับขนาดกับไทม์วอร์ปปีง	17
รูปที่ 2.9	การลดมิติข้อมูลแบบพีเอเอ	19
รูปที่ 2.10	เส้นขอบเขตบนและเส้นขอบเขตล่างของข้อมูลที่ถูกลดมิติแบบพีเอเอ	20
รูปที่ 2.11	การแทนข้อมูลเพลงด้วยรหัสพาร์สัน	22
รูปที่ 2.12	การแทนข้อมูลเพลงและเสียงร้องทำนองด้วยอนุกรมเวลา	23
รูปที่ 3.1	คอนทัวร์ระดับเสียงที่ได้จากขั้นตอนต่าง ๆ ในการสกัดคุณลักษณะ	27
รูปที่ 3.2	การเปรียบเทียบแนวคิดในการแปลงข้อมูลให้เป็นบรรทัดฐานในงานวิจัยของ Fu และคนอื่น ๆ กับงานวิจัยนี้	30
รูปที่ 3.3	การคำนวณค่าระยะทางขอบเขตล่างในงานวิจัยของ Fu และคนอื่น ๆ สำหรับกรณีที่มีการแปลงข้อมูลให้เป็นบรรทัดฐานทุกครั้งก่อนการคำนวณค่าระยะทาง	33
รูปที่ 3.4	ฟังก์ชันขอบเขตล่างของวิธีที่นำเสนอ	34
รูปที่ 3.5	รหัสที่ยืมสำหรับวิธีการให้ค่าเริ่มต้นกับรายการเพลง 10 ลำดับแรก	42
รูปที่ 3.6	วิธีการหยุดคำนวณในตอนต้นของฟังก์ชันขอบเขตล่าง	44
รูปที่ 3.7	วิธีการหยุดคำนวณในตอนต้นของวิธีไดนามิกไทม์วอร์ปปีง	45
รูปที่ 3.8	รหัสที่ยืมสำหรับวิธีการค้นหาข้อมูลตามลำดับโดยใช้ฟังก์ชันขอบเขตล่าง	46
รูปที่ 4.1	ความสามารถในการลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริงของข้อมูลสอบถามที่ความยาวและอัตราส่วนการหดและยืดขนาดต่าง ๆ	50
รูปที่ 4.2	ความสามารถในการลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริงของข้อมูลสอบถามที่ความยาว และความกว้างของเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะที่ขนาดต่าง ๆ	51
รูปที่ 4.3	ความสามารถในการลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริง เมื่อฐานข้อมูลมีขนาดต่าง ๆ	52

รูปที่ 4.4	การกระจายตัวของเสียงร้องทำนองที่มีจังหวะซ้ำหรือเร็วกว่าจังหวะเพลงใน ฐานข้อมูล ที่อัตราส่วนการหดและยืดขนาดต่าง ๆ	56
รูปที่ 4.5	เปรียบเทียบความแม่นยำในการค้นหาเพลง 10 ลำดับแรก เมื่อความกว้างของ เงื่อนไขบังคับโดยรวมแบบซอกโก-ซิเบมีขนาดต่าง ๆ	57
รูปที่ 4.6	เปรียบเทียบค่าเฉลี่ยลำดับส่วนกลับในการค้นหาเพลง เมื่อความกว้างของ เงื่อนไขบังคับโดยรวมแบบซอกโก-ซิเบมีขนาดต่าง ๆ	58
รูปที่ 4.7	เปรียบเทียบความแม่นยำในการค้นหาเพลง 10 ลำดับแรกเมื่อฐานข้อมูลมีขนาด ต่าง ๆ	59
รูปที่ 4.8	เปรียบเทียบค่าเฉลี่ยลำดับส่วนกลับในการค้นหาเพลง เมื่อฐานข้อมูลมีขนาด ต่าง ๆ	59
รูปที่ 4.9	สัดส่วนของลำดับเพลงที่ต้องการในการค้นหาข้อมูลภายใต้ฐานข้อมูลเพลงขนาด 1,000 เพลง	60
รูปที่ 4.10	เปรียบเทียบความเร็วในการค้นหาเพลง เมื่อฐานข้อมูลมีขนาดต่าง ๆ	61
รูปที่ 4.11	การกระจายตัวของระยะเวลาในการค้นหาเพลงจากเสียงร้องทำนองภายใต้ ฐานข้อมูลขนาด 1,000 เพลง	62
รูปที่ 4.12	เปรียบเทียบความแม่นยำในการค้นหาเพลงของวิธีการค้นหาแบบต่าง ๆ	65
รูปที่ 4.13	เปรียบเทียบค่าเฉลี่ยลำดับส่วนกลับในการค้นหาเพลงของวิธีการค้นหาแบบ ต่าง ๆ	65
รูปที่ 4.14	เปรียบเทียบระยะเวลาในการค้นหาเพลงของวิธีการค้นหาแบบต่าง ๆ	68

สารบัญตาราง

หน้า

ตารางที่ 4.1 จำนวนของคอนทัวร์ระดับเสียงที่สกัดจากเพลงมิติที่ฐานข้อมูลขนาดต่าง ๆ54



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ระบบค้นหาเพลงโดยการร้องทำนองถือเป็นหนึ่งในนวัตกรรมสำหรับโลกในอนาคตอันใกล้ ซึ่งจะเป็นยุคที่การค้นหาข้อมูลข่าวสารในรูปแบบแฟ้มข้อมูลดิจิทัลสามารถทำได้ อย่างเป็นสะดวกสบายและเป็นมิตรกับผู้ใช้มากยิ่งขึ้น ระบบค้นหาเพลงโดยการร้องทำนองนี้เป็นระบบที่ช่วยให้ผู้ใช้สามารถค้นหาเพลงได้จากการร้องทำนองหรือการร้องเพลงของผู้ใช้ โดยที่ข้อมูลเสียงเหล่านี้จะถูกนำมาใช้เป็นข้อมูลสอบถามสำหรับระบบคอมพิวเตอร์ อันจะช่วยทำให้ผู้ใช้สามารถค้นหาแฟ้มข้อมูลเพลงจากฐานข้อมูลเพลงที่มีขนาดใหญ่ได้อย่างสะดวกสบายและมีประสิทธิภาพมากยิ่งขึ้น โดยเฉพาะอย่างยิ่ง ในกรณีที่ผู้ใช้ไม่สามารถจดจำข้อมูล หรือรายละเอียดต่าง ๆ ของเพลงที่ต้องการได้ เช่น ชื่อเพลง ชื่อศิลปิน หรือ เนื้อร้อง เป็นต้น อันจะเป็นอุปสรรคต่อการค้นหาเพลงจากระบบสืบค้นข้อมูลแบบทั่วไปในปัจจุบัน

อย่างไรก็ตาม นับตั้งแต่ปี ค.ศ. 1995 ที่ได้เริ่มมีการศึกษาวิจัยระบบค้นหาเพลง โดยการร้องทำนอง [1] จวบจนถึงปัจจุบัน ระบบค้นหาเพลงโดยการร้องทำนองนี้ก็ยังไม่สามารถนำมาใช้ในเชิงพาณิชย์ได้อย่างจริงจัง หรือแม้แต่ในห้องวิจัยต่าง ๆ ก็ยังไม่สามารถพัฒนาระบบค้นหาเพลงโดยการร้องทำนองที่สามารถตอบสนองความต้องการของผู้ใช้ได้อย่างมีประสิทธิภาพ เนื่องจากในการพัฒนาระบบค้นหาเพลงโดยการร้องทำนองนั้น ผู้พัฒนาจะต้องออกแบบระบบให้สามารถรองรับกับความแปรผันต่าง ๆ ที่เกิดขึ้นจากผู้ใช้ เช่น การร้องเพี้ยน การร้องผิดจังหวะ การร้องโน้ตขาดหรือเกิน เป็นต้น รวมทั้งความสามารถในการค้นหาเพลงภายใต้ฐานข้อมูลที่มีขนาดใหญ่ได้อย่างถูกต้องและรวดเร็ว

ในปัจจุบัน มีงานวิจัยเป็นจำนวนมากได้เลือกวิธีการค้นคืนข้อมูลอนุกรมเวลา (Time Series Retrieval) มาประยุกต์ใช้กับระบบค้นหาเพลงโดยการร้องทำนอง เพื่อที่จะให้ระบบสามารถค้นหาเพลงจากเสียงร้องทำนองได้อย่างถูกต้องแม่นยำมากยิ่งขึ้น ซึ่งวิธีไดนามิก-โทมวอร์ปปีง ก็เป็นหนึ่งในวิธีที่ได้รับการยอมรับมากที่สุดที่สามารถค้นคืนข้อมูลอนุกรมเวลาได้อย่างถูกต้องแม่นยำ พร้อมทั้งมีการนำไปประยุกต์ใช้ในงานวิจัยต่าง ๆ มากมาย [2-11] และเมื่อไม่นานมานี้ ได้มีงานวิจัยหนึ่งเสนอวิธีปรับขนาดกับโทมวอร์ปปีง [2] ซึ่งเป็นการผสานวิธีปรับขนาดเอกรูปและวิธีไดนามิกโทมวอร์ปปีงเข้าไว้ด้วยกัน เพื่อทำให้วิธีไดนามิกโทมวอร์ปปีงสามารถค้นคืนข้อมูลอนุกรมเวลาที่มีการหดหรือยืดขนาดได้อย่างเหมาะสมมากยิ่งขึ้น พร้อมทั้งเสนอฟังก์ชันขอบเขตล่าง เพื่อลดปริมาณการคำนวณของวิธีดังกล่าว นอกจากนั้น งานวิจัยนี้ที่เสนอวิธีปรับขนาดกับโทมวอร์ปปีง ยังได้กล่าวถึงความเป็นไปได้ที่จะนำวิธีปรับขนาดกับโทม-

วอร์ปิง มาใช้ในการค้นหาเพลงโดยการร้องทำนอง อย่างไรก็ตาม งานวิจัยดังกล่าวได้มองข้ามความจำเป็นในการแปลงข้อมูลให้เป็นบรรทัดฐานทุกครั้งที่มีการหดหรือยืดขนาดของข้อมูล ซึ่งถือเป็นขั้นตอนสำคัญที่จะทำให้ข้อมูลที่จะนำมาเปรียบเทียบนั้น อยู่ในระดับอ้างอิงเดียวกันกับระดับอ้างอิงของข้อมูลสอบถามเสมอ อันจะส่งผลให้การเปรียบเทียบความคล้ายคลึงกันระหว่างข้อมูลทั้งสองมีความแม่นยำมากยิ่งขึ้น

สำหรับงานวิจัยนี้ มีวัตถุประสงค์เพื่อนำเสนอวิธีการค้นหาเพลงโดยการร้องทำนอง ซึ่งสามารถให้ผลลัพธ์ที่มีความถูกต้องแม่นยำสูง โดยใช้การปรับขนาดเอกรูปและไดนามิกโถมวอร์ปิงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน และสามารถทำงานได้อย่างรวดเร็วภายใต้ฐานข้อมูลขนาดใหญ่

1.2 วัตถุประสงค์ของการวิจัย

งานวิจัยนี้มีวัตถุประสงค์เพื่อนำเสนอวิธีการค้นหาเพลงโดยการร้องทำนองอย่างแม่นยำโดยใช้การปรับขนาดเอกรูปและไดนามิกโถมวอร์ปิง ที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน พร้อมทั้งนำเสนอวิธีการค้นหาเพลงโดยการร้องทำนองด้วยวิธีดังกล่าวที่สามารถทำงานได้อย่างรวดเร็ว

1.3 ขอบเขตของการวิจัย

1. พัฒนาระบบการค้นหาเพลงโดยการร้องทำนองด้วยวิธีการปรับขนาดเอกรูปและวิธีไดนามิกโถมวอร์ปิง ที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน และสามารถทำงานได้อย่างถูกต้องรวดเร็ว ภายใต้ฐานข้อมูลเพลงขนาด 1,000 เพลง
2. พัฒนาฟังก์ชันขอบเขตล่างสำหรับการปรับขนาดเอกรูปและวิธีไดนามิกโถมวอร์ปิง ที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน พร้อมพิสูจน์ความเป็นฟังก์ชันขอบเขตล่างของวิธีที่นำเสนอ
3. ทดสอบความแม่นยำในการค้นหาเพลงที่ใกล้เคียงกับเสียงร้องทำนองมากที่สุด 10 ลำดับแรก จากแฟ้มข้อมูลเสียงร้องทำนองของผู้ใช้ที่มีพื้นฐานทางดนตรีที่แตกต่างกันจำนวน 100 แฟ้ม ภายใต้ฐานข้อมูลเพลงขนาด 1,000 เพลง โดยในการทดสอบความแม่นยำนี้ เพื่อที่จะแสดงให้เห็นว่าวิธีที่นำเสนอสามารถค้นหาเพลงจากเสียงร้องทำนองได้อย่างแม่นยำมากกว่าวิธีที่มีการนำไปประยุกต์ใช้กันอยู่ในปัจจุบัน งานวิจัยนี้จึงมีการเปรียบเทียบความแม่นยำในการค้นหาเพลงของวิธีที่นำเสนอกับวิธีไดนามิกโถมวอร์ปิง ซึ่งได้รับการยอมรับว่ามีความแม่นยำสูง และวิธีปรับขนาดกับโถมวอร์ปิง ที่เสนอในงานวิจัยของ Fu และคนอื่น ๆ [2] ซึ่งเสนอวิธีที่ทำให้วิธีไดนามิกโถมวอร์ปิงสามารถวัดระยะทางของข้อมูลที่มีการยืดหรือหดขนาดได้อย่างเหมาะสมมากยิ่งขึ้น

4. ทดสอบความเร็วในการค้นหาเพลงที่ใกล้เคียงกับเสียงร้องทำนองมากที่สุด 10 ลำดับแรก จากแฟ้มข้อมูลเสียงร้องทำนองของผู้ใช้ที่มีพื้นฐานทางดนตรีที่แตกต่างกันจำนวน 100 แฟ้ม ภายใต้ฐานข้อมูลเพลงขนาด 1,000 เพลง เพื่อที่จะแสดงให้เห็นว่าฟังก์ชันขอบเขตล่างของวิธีที่นำเสนอ นั้น เมื่อนำไปประยุกต์ใช้กับวิธีค้นหาข้อมูลแบบอื่น ๆ ที่มีประสิทธิภาพ แล้วจะสามารถค้นหาเพลงได้อย่างรวดเร็ว ภายใต้ฐานข้อมูลเพลงขนาดใหญ่ โดยจะมีการเปรียบเทียบเช่นเดียวกันกับการทดสอบในประเด็นของความแม่นยำในการค้นหาเพลง

1.4 ประโยชน์ที่ได้รับ

ได้วิธีการค้นหาเพลงโดยการร้องทำนองที่มีความแม่นยำสูง สามารถทำงานได้อย่างรวดเร็ว และรองรับฐานข้อมูลที่มีขนาดใหญ่ นอกจากนี้ยังสามารถนำวิธีที่นำเสนอไปประยุกต์ใช้กับการค้นหาข้อมูลอนุกรมเวลาแบบอื่น ๆ ที่มีการหดหรือยืดขนาดของข้อมูลได้ เช่น นำไปประยุกต์ใช้กับการค้นหาข้อมูลภาพเคลื่อนไหว การค้นหาข้อมูลยีน (Gene Expression Data) เป็นต้น

1.5 วิธีดำเนินการวิจัย

1. ศึกษากระบวนการค้นหาเพลงโดยการร้องทำนองที่มีอยู่ในปัจจุบัน
2. ศึกษาแนวทางในการพัฒนาระบบค้นหาเพลงโดยการร้องทำนองที่ใช้วิธีการค้นหาโดยการเปรียบเทียบข้อมูลอนุกรมเวลา
3. ออกแบบและพัฒนาระบบค้นหาเพลงโดยการร้องทำนองโดยใช้การปรับขนาดเอกรูปและไดนามิกไทม์วอร์ปปีง ที่สามารถรองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน และออกแบบฟังก์ชันขอบเขตล่างของวิธีดังกล่าว พร้อมทั้งพิสูจน์ความเป็นฟังก์ชันขอบเขตล่าง
4. ทดสอบความสามารถของฟังก์ชันขอบเขตล่างที่นำเสนอในการลดจำนวนข้อมูลเพลงในฐานข้อมูล ก่อนที่จะคำนวณหาค่าระยะทางโดยใช้วิธีปรับขนาดเอกรูปและวิธีไดนามิกไทม์วอร์ปปีง ที่สามารถรองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน
5. ออกแบบและพัฒนาวีธีค้นหาเพลงโดยการร้องทำนองโดยใช้วิธีปรับขนาดเอกรูปและวิธีไดนามิกไทม์วอร์ปปีง ที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน
6. ทดสอบความแม่นยำและความเร็วของวิธีที่นำเสนอเมื่อเทียบกับวิธีค้นหาเพลงโดยการร้องทำนองแบบต่าง ๆ
7. วิเคราะห์และสรุปผลการทดลอง
8. สรุป เรียบเรียง และจัดทำวิทยานิพนธ์

1.6 ผลงานตีพิมพ์จากงานวิจัย

ส่วนหนึ่งของงานวิทยานิพนธ์นี้ ได้รับการตีพิมพ์เป็นบทความทางวิชาการสองเรื่อง ดังนี้

- “Efficient Multimedia Time Series Data Retrieval Under Uniform Scaling and Normalisation” โดย ไวยณัฐ ใเอ้อจงประสิทธิ์ และโชติรัตน์ รัตนามัทธนะ ในงานประชุมวิชาการ “30th European Conference on Information Retrieval” ซึ่งจัดขึ้น ณ เมืองกลาสโกล ประเทศสก็อตแลนด์ ระหว่างวันที่ 30 มีนาคม ถึง 3 เมษายน 2551 ดังรายละเอียดในภาคผนวก ก
- “Accurate and Efficient Retrieval of Multimedia Time Series Data Under Uniform Scaling and Time Warping” โดย ไวยณัฐ ใเอ้อจงประสิทธิ์ และโชติรัตน์ รัตนามัทธนะ ในงานประชุมวิชาการ “The Pacific-Asia Conference on Knowledge Discovery and Data Mining” ซึ่งจัดขึ้น ณ เมืองโอซากา ประเทศญี่ปุ่น ระหว่างวันที่ 20 – 23 พฤษภาคม 2551 ดังรายละเอียดในภาคผนวก ข

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

สำหรับบทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง จะมีการนำเสนอทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับดนตรี วิธีการสกัดคุณลักษณะของเสียง รูปแบบของแฟ้มข้อมูลมิติมาตรฐาน ซึ่งทั้งหมดนี้เป็นพื้นฐานในการวิจัยและพัฒนาระบบค้นหาเพลงโดยการร้องทำนองโดยทั่วไป รวมไปถึงวิธีการค้นคืนข้อมูลอนุกรมเวลา ซึ่งจะถูกนำมาใช้ในการเปรียบเทียบความคล้ายคลึงกันระหว่างเสียงร้องทำนองของผู้ใช้กับเพลงในฐานข้อมูล สุดท้ายตามด้วยงานวิจัยที่เกี่ยวข้องกับระบบค้นหาเพลงโดยการร้องทำนอง

2.1 ทฤษฎีที่เกี่ยวข้อง

สำหรับหัวข้อทฤษฎีที่เกี่ยวข้องนี้ จะเริ่มต้นนำเสนอจาก ความรู้ทางดนตรีที่เกี่ยวข้องกับการวิจัยระบบค้นหาเพลงโดยการร้องทำนอง ตามด้วยขั้นตอนการตรวจจับระดับเสียง ซึ่งจะใช้สำหรับการสกัดคุณลักษณะของเสียงร้องทำนอง รูปแบบแฟ้มข้อมูลมิติมาตรฐาน จากนั้นจะเป็นส่วนของวิธีการเปรียบเทียบความคล้ายคลึงกันของข้อมูลอนุกรมเวลาที่เกี่ยวข้องกับงานวิจัยนี้ ซึ่งได้แก่ วิธีไดนามิกไทม์วอร์ปิง วิธีการปรับขนาดเอกรูป และวิธีปรับขนาดกับไทม์วอร์ปิง ตามลำดับ สุดท้ายจะเป็นส่วนของวิธีการลดขนาดของข้อมูลอนุกรมเวลา เพื่อให้การเปรียบเทียบความคล้ายคลึงกันของข้อมูลอนุกรมเวลาสามารถทำได้อย่างรวดเร็วมากยิ่งขึ้น

2.2 ความรู้ทางดนตรี

ความรู้ทางดนตรีที่เกี่ยวข้องกับการวิจัยระบบค้นหาเพลงโดยการร้องทำนอง มีดังต่อไปนี้

- ระดับเสียง (Pitch)

ในทางดนตรี ระดับเสียงของตัวโน้ต หมายถึง ความถี่ของตัวโน้ต ซึ่งตัวโน้ตแต่ละตัวจะมีความถี่มาตรฐานของตัวเอง เช่น โน้ต A₄ มีความถี่ 440 เฮิรตซ์ เป็นต้น

- คอนทัวร์ระดับเสียง (Pitch Contour)

คอนทัวร์ระดับเสียง คือ ระดับเสียงที่เรียงต่อกันตามลำดับ ในรูปแบบของข้อมูลอนุกรมเวลา

- ความเร็วจังหวะ (Tempo)

ความเร็วจังหวะเป็นตัวกำหนดความเร็วของเพลง โดยมีหน่วยเป็นบีตต่อนาที (Beats Per Minute หรือ BPM)

2.3 ขั้นตอนวิธีการตรวจจับระดับเสียง (Pitch Detection Algorithm)

ขั้นตอนวิธีการตรวจจับระดับเสียงเป็นวิธีการคำนวณหาระดับเสียง หรือความถี่มูลฐาน (Fundamental Frequency) ของสัญญาณเสียงต่าง ๆ เช่น เสียงพูดของมนุษย์ เสียงดนตรี หรือเสียงธรรมชาติ เป็นต้น โดยในระบบค้นหาเพลงโดยการร้องทำนองจะมีการสกัดคอนทัวร์ระดับเสียงออกจากเสียงร้องของผู้ใช้ เพื่อที่จะนำคุณลักษณะดังกล่าวไปใช้ในการเปรียบเทียบความคล้ายคลึงกันกับข้อมูลโน้ตดนตรีของเพลงที่จัดเก็บอยู่ในฐานข้อมูล ขั้นตอนวิธีการตรวจจับระดับเสียงสามารถทำได้ทั้งในโดเมนเวลา (Time Domain) เช่น วิธีสหสัมพันธ์ (Autocorrelation Method) [12] และในโดเมนความถี่ (Frequency Domain) เช่น ผลการแปลงฟูเรียร์ (Fourier Transform) [13] สำหรับการสกัดข้อมูลโน้ตดนตรีของเพลงในฐานข้อมูลนั้น จะมีการนำเสนอในหัวข้อถัดไป

2.4 รูปแบบแฟ้มข้อมูลมิติมาตรฐาน (Standard MIDI File Structure หรือ SMF)

มิติ หรือ มาตรฐานการประสานเครื่องดนตรีแบบดิจิทัล (Musical Instrument Digital Interface) คือ เทคโนโลยีการแทนเสียงเพลงในรูปแบบแฟ้มข้อมูลดิจิทัล ซึ่งข้อมูลในรูปแบบนี้ จะแยกเก็บข้อมูลโน้ตดนตรีของเครื่องดนตรีแต่ละชนิด ออกเป็นช่องสัญญาณ (Channel) ต่าง ๆ ที่แตกต่างกัน รวมทั้งข้อมูลโน้ตดนตรีสำหรับเครื่องดนตรีที่แทนเสียงร้องของมนุษย์ ซึ่งจากรูปแบบของแฟ้มข้อมูลมิตินี้เอง ทำให้การสกัดข้อมูลโน้ตดนตรีเพื่อเป็นตัวแทนเสียงร้องของมนุษย์นั้นสามารถทำได้ อันจะเป็นประโยชน์อย่างยิ่งในการพัฒนาระบบค้นหาเพลง โดยการร้องทำนอง เนื่องจากในการค้นหาเพลงโดยการร้องทำนองนั้น ข้อมูลโน้ตดนตรีของเสียงร้องทำนองที่สกัดได้จากแฟ้มข้อมูลมิติ จะถูกนำไปใช้ในการเปรียบเทียบความคล้ายคลึงกันกับคอนทัวร์ระดับเสียงของผู้ใช้ ซึ่งเป็นข้อมูลสอบถามสำหรับระบบค้นหาเพลงโดยการร้องทำนอง จากนั้นผลลัพธ์ที่ได้จากการค้นหาก็คือ รายการของเพลงซึ่งมีข้อมูลโน้ตดนตรีที่มีลักษณะใกล้เคียงกับคอนทัวร์ระดับเสียงของผู้ใช้มากที่สุด

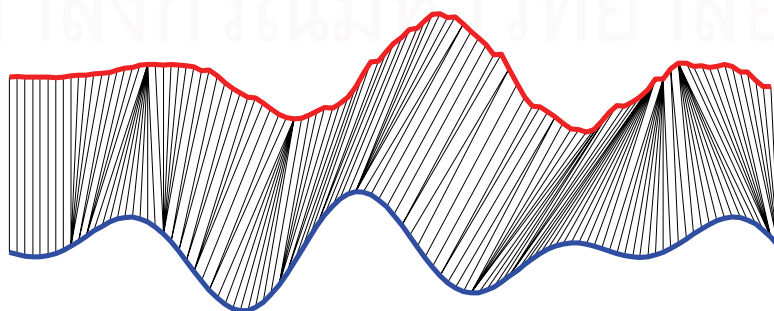
สำหรับโครงสร้างของแฟ้มข้อมูลมิตินั้น ประกอบไปด้วยกลุ่มข้อมูล (Chunks) ต่าง ๆ ที่เรียงต่อกัน ซึ่งในแต่ละกลุ่มข้อมูล จะมีหน้าที่ที่แตกต่างกันไป โดยสามารถแบ่งได้เป็นสองกลุ่มหลัก ๆ ด้วยกัน คือ กลุ่มข้อมูลส่วนต้น (Header Chunk) และกลุ่มข้อมูลแทรค (Track Chunks) โดยกลุ่มข้อมูลส่วนต้นนี้ จะเป็นส่วนที่แสดงรายละเอียดต่าง ๆ ของแฟ้มข้อมูลมิติ เช่น รูปแบบของแฟ้มข้อมูลมิติ จำนวนของกลุ่มข้อมูลส่วนแทรค เป็นต้น สำหรับกลุ่มข้อมูลแทรค

ในส่วนนี้จะเก็บข้อมูล หรือเหตุการณ์ต่าง ๆ ทางดนตรีไว้ เช่น จุดเริ่มต้นโน้ตดนตรี จุดสิ้นสุดโน้ตดนตรี ระยะเวลานับตั้งแต่เหตุการณ์สุดท้ายจนถึงปัจจุบัน (Delta Time) โน้ตดนตรีที่ใช้ช่องสัญญาณ เป็นต้น

อย่างไรก็ตาม ในการค้นหาเพลงโดยการร้องทำนองนั้น เมื่อมีข้อมูลสอบถามซึ่งได้จากการสกัดคุณลักษณะเกี่ยวกับคอนทิวรัระดับเสียงจากเสียงร้องทำนองของผู้ใช้แล้ว และมีข้อมูลที่จะนำมาใช้ในการเปรียบเทียบ ซึ่งก็คือ ข้อมูลโน้ตดนตรีที่อยู่ภายในแฟ้มข้อมูลมิติ จากนั้นจะเป็นการเปรียบเทียบความคล้ายคลึงกันระหว่างข้อมูลทั้งสอง ซึ่งจะมีการนำเสนอในลำดับถัดไป

2.5 ไดนามิกไทม์วอร์ปิง (Dynamic Time Warping หรือ DTW)

ไดนามิกไทม์วอร์ปิง เป็นวิธีกำหนดการพลวัต (Dynamic Programming) ที่ใช้สำหรับวัดความคล้ายคลึงกัน (Similarity Measure) ระหว่างข้อมูลอนุกรมเวลา 2 ชุด โดยผลลัพธ์ที่ได้จากการวัดความคล้ายคลึงดังกล่าว จะเป็นค่าระยะทางระหว่างข้อมูลอนุกรมเวลาทั้งสอง ซึ่งสามารถหาได้โดยการคำนวณค่าระยะทางสะสมระหว่างจุดข้อมูลในอนุกรมเวลาดังกล่าวที่มีการปรับแนว (Alignment) ระหว่างกัน ที่จะทำให้ได้ค่าระยะทางสะสมระหว่างจุดข้อมูลของข้อมูลอนุกรมเวลาทั้งสองนั้นมีค่าน้อยที่สุด ดังแสดงได้ในรูปที่ 2.1 ซึ่งแสดงวิธีการปรับแนวที่จะให้ค่าระยะทางสะสมต่ำสุดระหว่างจุดข้อมูลในอนุกรมเวลาทั้งสอง ทั้งนี้การปรับแนวระหว่างข้อมูลดังกล่าวในการคำนวณหาค่าระยะทางนั้น เพื่อที่จะรองรับความแปรผันเฉพาะที่เชิงเวลา (Local Variation) ที่เกิดขึ้นในข้อมูลอนุกรมเวลาต่าง ๆ เช่น ข้อมูลเสียงพูด หรือเสียงร้องทำนอง เนื่องจากในการพูดหรือการร้องทำนองนั้น ข้อมูลที่ได้มักจะมีจังหวะหรือความเร็วไม่คงที่ ซึ่งทำให้เมื่อสกัดคุณลักษณะจากข้อมูลดังกล่าวออกมาในรูปของข้อมูลอนุกรมเวลาแล้ว ข้อมูลอนุกรมเวลาที่ได้นั้น จะมีการเลื่อนของข้อมูลในแนวแกนเวลาอย่างไม่สม่ำเสมอ อันจะเป็นปัญหาสำหรับการวัดความคล้ายคลึงของข้อมูลดังกล่าวได้ ดังนั้นวิธีไดนามิกไทม์วอร์ปิงจึงได้รับความนิยมในการนำไปประยุกต์ใช้กับระบบรู้จำเสียงพูด (Speech Recognition) และระบบค้นหาเพลงโดยการร้องทำนอง (Query by Humming)



รูปที่ 2.1 การคำนวณค่าระยะทางโดยใช้วิธีไดนามิกไทม์วอร์ปิง

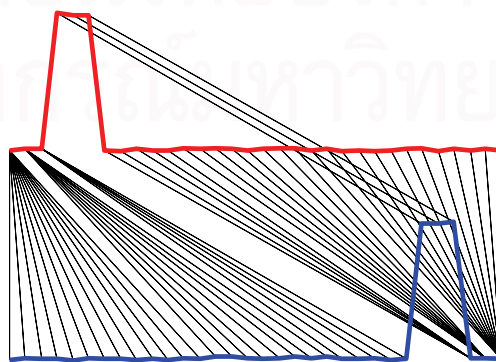
สำหรับนิยามของวิธีไดนามิกไทม์วอร์ปิ้งนั้น สามารถนิยามได้ดังนี้ ให้ข้อมูลอนุกรมเวลา $Q = q_1, q_2, q_3, \dots, q_m$ คือ ข้อมูลสอบถาม (Query Sequence) ที่มีความยาว m และ $C = c_1, c_2, c_3, \dots, c_n$ คือ ข้อมูลที่จะใช้ในการเปรียบเทียบ (Candidate Sequence) ความยาว n การหาระยะทางด้วยวิธีไดนามิกไทม์วอร์ปิ้ง จะสามารถแสดงเป็นสมการเวียนเกิดได้ดังสมการที่ (2.1)

$$DTW(Q, C) = D(q_i, c_j) + \min \begin{cases} DTW(q_{i-1}, c_{j-1}) \\ DTW(q_i, c_{j-1}) \\ DTW(q_{i-1}, c_j) \end{cases} \quad (2.1)$$

โดยที่ $1 \leq i \leq m$ และ $1 \leq j \leq n$ และฟังก์ชัน $D(q_i, c_j)$ เป็นฟังก์ชันยูคลิดที่ใช้สำหรับหาระยะทางระหว่างจุด q_i และจุด c_j อย่างไรก็ตาม ดังแสดงในสมการที่ (2.2) รากที่สองของฟังก์ชันยูคลิดนั้น สามารถละได้โดยที่ลำดับของความคล้ายคลึงกันของข้อมูลจะไม่มีเปลี่ยนแปลง ทั้งนี้เพื่อที่จะเพิ่มประสิทธิภาพในการคำนวณค่าระยะทางด้วยวิธีไดนามิกไทม์วอร์ปิ้งให้สามารถคำนวณได้อย่างรวดเร็วมากยิ่งขึ้น

$$D(q_i, c_j) = (q_i - c_j)^2 \quad (2.2)$$

แม้ว่าวิธีไดนามิกไทม์วอร์ปิ้ง จะสามารถคำนวณค่าระยะทางระหว่างข้อมูลอนุกรมเวลาที่มีความแปรผันเชิงเวลาได้เป็นอย่างดี แต่ในการคำนวณค่าระยะทางด้วยวิธีไดนามิกไทม์วอร์ปิ้งนั้น ในบางกรณี วิธีดังกล่าวอาจมีการปรับแนวเพื่อหาค่าระยะทางระหว่างคู่จุดอย่างไม่เหมาะสม ดังเช่นในรูปที่ 2.2 ที่แสดงให้เห็นว่า มีการปรับแนวให้มีการคำนวณค่าระยะทางระหว่างจุดยอดของข้อมูลอนุกรมเวลาทั้งสองที่อยู่ในส่วนต้นและส่วนปลายของข้อมูล โดยข้อมูลทั้งสองอาจเป็นข้อมูลต่างประเภทกัน ทำให้การคำนวณดังกล่าวจะค่าระยะทางที่ต่ำ ซึ่งเป็นค่าระยะทางที่ไม่เหมาะสม ดังนั้นจึงได้ผู้เสนอเงื่อนไขบังคับโดยรวมขึ้นเพื่อป้องกันกาปรับแนวอย่างในการหาค่าระยะทางที่ไม่เหมาะสมนี้

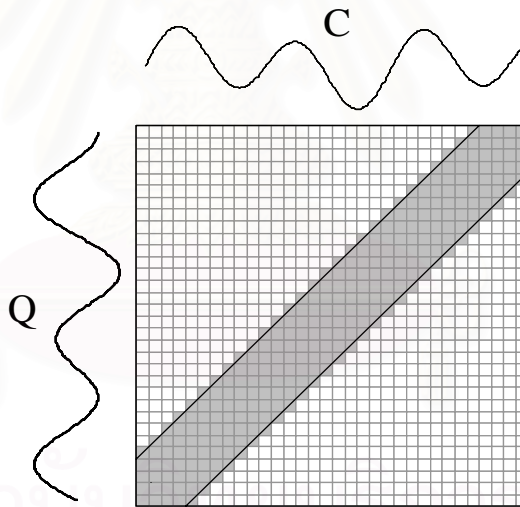


รูปที่ 2.2 การปรับแนวในการหาระยะทางอย่างไม่เหมาะสมของวิธีไดนามิกไทม์วอร์ปิ้ง [14]

2.5.1 เงื่อนไขบังคับโดยรวม (Global Constraint)

เงื่อนไขบังคับโดยรวมเป็นการบังคับให้การคำนวณหาค่าระยะทางด้วยวิธีไดนามิกโทมัสเวิร์ปป์ทำงานอยู่ในขอบเขตที่ถูกจำกัดไว้ หรือกล่าวอีกนัยหนึ่งคือ ทำให้วิธีไดนามิกโทมัสเวิร์ปป์ไม่สามารถคำนวณค่าระยะทางระหว่างคู่จุดที่อยู่นอกเหนือขอบเขตที่ถูกกำหนดไว้ล่วงหน้าได้ อันเป็นการป้องกันการคำนวณค่าระยะทางที่เกิดจากการปรับแนวอย่างไม่เหมาะสม และยังเพิ่มความแม่นยำโดยรวมของวิธีไดนามิกโทมัสเวิร์ปป์อีกด้วย [15]

เงื่อนไขบังคับโดยรวมนี้มีหลายประเภท แต่ในที่นี้จะกล่าวถึงเฉพาะเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะ (Sakoe-Chiba Band) ซึ่งเป็นการจำกัดขอบเขตการคำนวณของวิธีไดนามิกโทมัสเวิร์ปป์ไว้เป็นเส้นขนานในแนวเส้นทะแยงมุมบนตารางการคำนวณค่าระยะทาง ดังแสดงในรูปที่ 2.3 หรือสามารถกล่าวได้อีกนัยหนึ่งคือ ทุก ๆ คู่จุดที่นำมาคำนวณค่าระยะทางนั้น จะต้องมียะห่างระหว่างกันในแนวแกนเวลาหรือแนวแกน x ไม่เกินระยะทาง r ซึ่งเป็นค่าคงที่ที่ได้กำหนดไว้ล่วงหน้า โดยความยาวของข้อมูลสอบถามและความยาวของข้อมูลที่จะใช้ในการเปรียบเทียบจะต้องมีขนาดเท่ากัน นั่นคือ $m = n$



รูปที่ 2.3 เงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะ [15]

สำหรับเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะนั้น สามารถนิยามได้ดังต่อไปนี้

$$DTW_{\text{constraint}}(Q, C, r) \equiv D_{\text{constraint}}(q_i, c_i, r) + \min \begin{cases} DTW_{\text{constraint}}(q_{i-1}, c_{j-1}, r) \\ DTW_{\text{constraint}}(q_i, c_{j-1}, r) \\ DTW_{\text{constraint}}(q_{i-1}, c_j, r) \end{cases} \quad (2.3)$$

$$D_{\text{constraint}}(q_i, c_j, r) \equiv \begin{cases} D(q_i, c_j) & \text{where } |i-j| \leq r \\ \infty & \text{otherwise} \end{cases} \quad (2.4)$$

อย่างไรก็ตาม สำหรับระบบค้นหาเพลงโดยการร้องทำนอง ข้อมูลเพลงที่จะนำมาใช้ในการเปรียบเทียบกับเสียงร้องทำนองนั้นมีจำนวนมาก ถ้าหากเลือกใช้วิธีไดนามิกไทม์-วอร์ปปีงในการคำนวณค่าระยะทางเพื่อหาความคล้ายคลึงกันระหว่างเสียงร้องทำนองกับข้อมูลเพลง จะทำให้ต้องใช้ระยะเวลาในการประมวลผลนาน เพราะฉะนั้นจึงได้มีผู้เสนอวิธีที่จะประมาณค่าระยะทางที่ได้จากวิธีไดนามิกไทม์วอร์ปปีงก่อนที่จะคำนวณค่าระยะทางจริง เพื่อที่จะเลือกคำนวณค่าระยะทางด้วยวิธีไดนามิกไทม์วอร์ปปีงกับข้อมูลเพียงบางตัวเท่านั้น

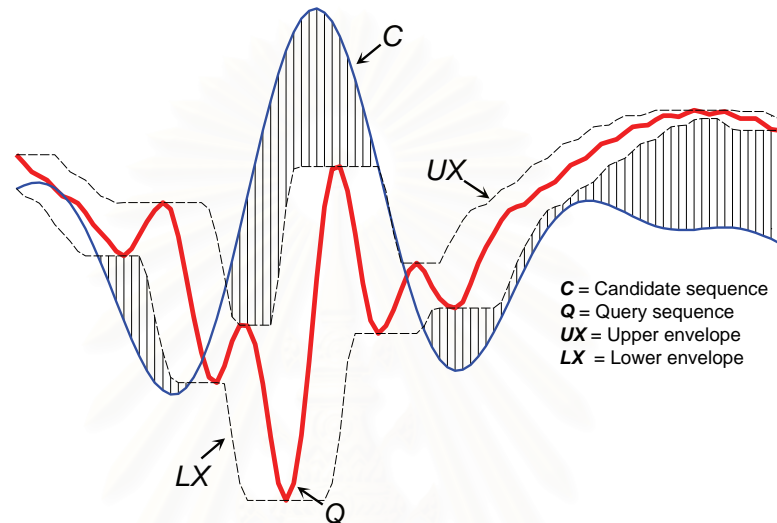
2.5.2 ฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปปีง (Lower Bounding Function of DTW)

การคำนวณระยะทางด้วยวิธีไดนามิกไทม์วอร์ปปีงนั้น จะมีขีดจำกัดเชิงสัญกรณ์ (Asymptotic Limit) เท่ากับ $O(n^2)$ นั่นคือ วิธีไดนามิกไทม์วอร์ปปีงจะใช้เวลาในการคำนวณเป็นฟังก์ชันพหุนาม (Polynomial Time) กับความยาวของข้อมูลขาเข้า ซึ่งมีประสิทธิภาพในเชิงสัญกรณ์ต่ำกว่าการหาระยะทางแบบยูคลิด (Euclidean Distance) ที่มีขีดจำกัดเชิงสัญกรณ์เป็นเชิงเส้น (Linear Time) หรือ $O(n)$ ดังนั้นจึงได้มีผู้เสนอฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปปีงที่ใช้การคำนวณแบบยูคลิดขึ้น [16] เพื่อใช้ในการประมาณค่าระยะทางระหว่างข้อมูลอนุกรมเวลาทั้งสองอย่างมีประสิทธิภาพ ก่อนที่จะคำนวณค่าระยะทางจริงด้วยวิธีไดนามิกไทม์วอร์ปปีง เพื่อที่จะลดปริมาณข้อมูลที่ต้องคำนวณค่าระยะทางด้วยวิธีไดนามิกไทม์วอร์ปปีงลง ค่าระยะทางที่ได้จากการคำนวณฟังก์ชันขอบเขตล่าง จะต้องไม่เกินค่าระยะทางจริงที่ได้จากการคำนวณด้วยวิธีไดนามิกไทม์วอร์ปปีง

สำหรับการใช้งานฟังก์ชันขอบเขตล่างในการค้นคืนข้อมูลอนุกรมเวลานั้น สามารถทำได้โดย คำนวณค่าระยะทางระหว่างข้อมูลสอบถามกับข้อมูลอนุกรมเวลาในฐานข้อมูลทุกตัว โดยใช้ฟังก์ชันขอบเขตล่างในการประมาณค่าระยะทาง และถ้าค่าระยะทางขอบเขตล่างที่ได้จากการคำนวณนั้น มีค่ามากกว่าค่าระยะทางจริงน้อยสุดตั้งแต่มีการคำนวณค่าระยะทาง นั้นแสดงว่าแม้ว่าจะคำนวณค่าระยะทางจริงด้วยวิธีไดนามิกไทม์วอร์ปปีงระหว่างข้อมูลสอบถามกับข้อมูลดังกล่าว ค่าระยะทางจริงที่ได้ก็จะมีค่ามากกว่าค่าระยะทางจริงน้อยสุดเสมอ ซึ่งแสดงว่าข้อมูลดังกล่าวไม่ได้มีความคล้ายคลึงกับข้อมูลสอบถามมากที่สุด ทำให้สามารถลดการคำนวณค่าระยะทางด้วยวิธีไดนามิกไทม์วอร์ปปีงลงได้ ในทางกลับกัน ถ้าค่าระยะทางขอบเขตล่างที่ได้มีค่าน้อยกว่าหรือเท่ากับค่าระยะทางจริงน้อยสุด นั้นแสดงว่า มีโอกาสที่ค่าระยะทางจริงระหว่างข้อมูลอนุกรมเวลาทั้งสอง จะมีค่าน้อยกว่าค่าระยะทางจริงน้อยสุด นั่นคือ ข้อมูลอนุกรมเวลาดังกล่าวมีโอกาสที่จะคล้ายคลึงกับข้อมูลสอบถามมากที่สุด

วิธีในการสร้างฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปปีงนั้น สามารถทำได้โดยสร้างเส้นขอบเขตบนและเส้นขอบเขตล่างจากเงื่อนไขบังคับโดยรวมของข้อมูลสอบถามเพื่อเป็นตัวแทนของข้อมูลสอบถามในการคำนวณค่าระยะทางขอบเขตล่างกับข้อมูลอนุกรม

เวลาที่จะใช้ในการเปรียบเทียบด้วยวิธียุคลิด ดังแสดงในรูปที่ 2.4 ซึ่งแสดงการคำนวณหาค่าระยะทางด้วยวิธียุคลิดระหว่างเส้นขอบเขตบน UX และเส้นขอบเขตล่าง LX กับข้อมูลอนุกรมเวลา C อย่างไรก็ตามการคำนวณค่าระยะทางด้วยวิธียุคลิดนั้นจะไม่มี การปรับแนวระหว่างข้อมูลในการหาค่าระยะทาง ทำให้ผลลัพธ์ที่ได้จึงเป็นค่าระยะทางสะสมระหว่างคู่จุดในข้อมูลอนุกรมเวลาทั้งสองที่อยู่ในลำดับเดียวกัน ซึ่งจะแสดงการคิดหาค่าระยะทางระหว่างคู่จุดของข้อมูลดังกล่าวได้เป็นเส้นในแนวตั้งดังรูป



รูปที่ 2.4 การคำนวณค่าระยะทางขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปิง โดยที่ค่า r เท่ากับ 5% ของความยาวข้อมูล

สำหรับฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปิงนั้น สามารถนิยามได้ดังนี้ ให้ $LBX(Q, C)$ คือ ฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปิงระหว่างข้อมูลอนุกรมเวลา Q และ C โดยที่ความยาวของข้อมูลอนุกรมเวลา Q และ C จะต้องมีความยาวเท่ากัน นั่นคือ $m = n$ และให้ UX และ LX เป็นเส้นขอบเขตบนและเส้นขอบเขตล่างตามลำดับ จะได้

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

$$UX_i = \max(q_{\min(1, i-r)}, \dots, q_{\max(i+r, n)}) \quad (2.6)$$

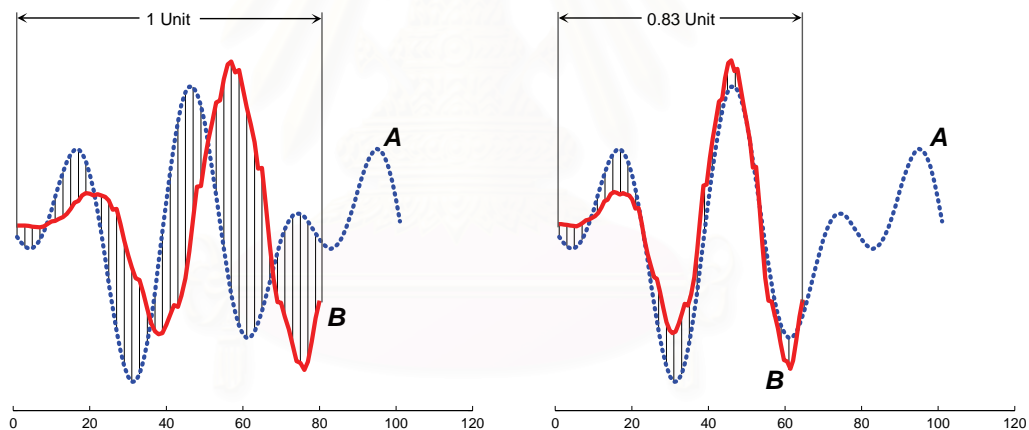
$$LX_i = \min(q_{\min(1, i-r)}, \dots, q_{\max(i+r, n)})$$

แม้ว่าการคำนวณค่าระยะทางด้วยวิธีไดนามิกไทม์วอร์ปิง จะสามารถรองรับความแปรผันภายในแกนเวลาได้เป็นอย่างดี อย่างไรก็ตามถ้าข้อมูลอนุกรมเวลา มีการหดหรือยืด

ขนาดอย่างสม่ำเสมอ อาจทำให้ความแม่นยำในการนำวิธีไดนามิกไทม์วอร์ปิงไปใช้ในการวัดความคล้ายคลึงกันของข้อมูลอนุกรมเวลาลดลง ดังนั้นเพื่อที่จะแก้ปัญหาดังกล่าว จึงผู้ที่เสนอวิธีในการวัดความคล้ายคลึงกันข้อมูลอนุกรมเวลาที่มีการหดหรือยืดขนาดอย่างสม่ำเสมอ ซึ่งจะกล่าวในหัวข้อถัดไป

2.6 การปรับขนาดเอกกรุป (Uniform Scaling)

การปรับขนาดเอกกรุปเป็นวิธีที่หดหรือยืดขนาดของข้อมูลอนุกรมเวลาอย่างสม่ำเสมอ ก่อนที่จะคำนวณหาระยะทางด้วยวิธียุคลิด ทั้งนี้เพื่อเพิ่มความแม่นยำในการคำนวณหาระยะทางของข้อมูลอนุกรมเวลาที่มีความแปรผันโดยรวมเชิงเวลา (Global Variation) ซึ่งในระบบค้นหาเพลงโดยการร้องทำนองนั้น ผู้ใช้มักจะร้องทำนองช้าหรือเร็วกว่าจังหวะเพลงที่เก็บไว้ในฐานข้อมูล ซึ่งวิธีการปรับขนาดเอกกรุปนี้สามารถรองรับความแปรผันของผู้ใช้ดังกล่าวได้เป็นอย่างดี ดังตัวอย่างในรูปที่ 2.5 ถ้ามีการหดขนาดของข้อมูล B แบบเอกกรุป (ในรูปด้านซ้ายมือ) ไปที่ความยาว 0.83 เท่าจากความยาวเดิม (ในรูปด้านขวามือ) จะทำให้การวัดความคล้ายคลึงกันระหว่างข้อมูล A และข้อมูล B เป็นไปอย่างถูกต้องมากยิ่งขึ้น



รูปที่ 2.5 การคำนวณหาระยะทางด้วยวิธีการปรับขนาดเอกกรุป

สำหรับการปรับขนาดเอกกรุปนั้น สามารถนิยามได้ดังนี้ ให้ $C = c_1, c_2, c_3, \dots, c_l$ เป็นข้อมูลในส่วนต้นของข้อมูลอนุกรมเวลา (Prefix of Candidate Sequence) ที่มีความยาว l และถ้าต้องการหดหรือยืดขนาดข้อมูลอนุกรมเวลานี้ให้มีความยาวเท่ากับ m จะสามารถทำได้ดังสมการต่อไปนี้ [17]

$$c_j = c_{\lfloor j * l / m \rfloor} \quad \text{where } 1 \leq j \leq m \quad (2.7)$$

อย่างที่ได้อธิบายไว้ในข้างต้น การปรับขนาดเอกรูปสามารถนำมาใช้ร่วมกับการวัดระยะทางแบบยุคลิดเพื่อหาค่าระยะทางระหว่างข้อมูลอนุกรมทั้งสองที่มีความแปรผันโดยรวมเชิงเวลาได้ ซึ่งสามารถนิยามได้ดังต่อไปนี้

กำหนดให้อัตราส่วนการหดและยืดขนาดของข้อมูลมากที่สุด คือ $sfmin$ และ $sfmax$ ตามลำดับ โดยที่ $n \geq sfmax * m$ และ m และ n คือความยาวของข้อมูลสอบถามและข้อมูลที่จะใช้ในการเปรียบเทียบ ตามลำดับ สำหรับแนวคิดในการนำวิธีปรับขนาดเอกรูปมาใช้ร่วมกับการวัดระยะทางแบบยุคลิดนั้น สามารถทำได้โดยการหาค่าระยะทางน้อยสุดระหว่างข้อมูลสอบถามและส่วนต้นของข้อมูลที่จะใช้ในการเปรียบเทียบ ที่มีความยาวระหว่าง $sfmin * m$ ถึง $sfmax * m$ ซึ่งถูกปรับขนาดให้เท่ากับความยาวของข้อมูลสอบถามที่มีความยาว m สำหรับสมการในการวัดระยะทางโดยการปรับขนาดเอกรูปนั้น สามารถแสดงได้ดังนี้

$$US(Q, C, sfmin, sfmax) = \min_{l=\lfloor sfmin * m \rfloor}^{\min(\lfloor sfmax * m \rfloor, n)} D(RP(C, m, l), Q) \quad (2.8)$$

$$RP(C, m, l)_i = c_{\lfloor i * l / m \rfloor} \quad \text{where } 1 \leq i \leq m \quad (2.9)$$

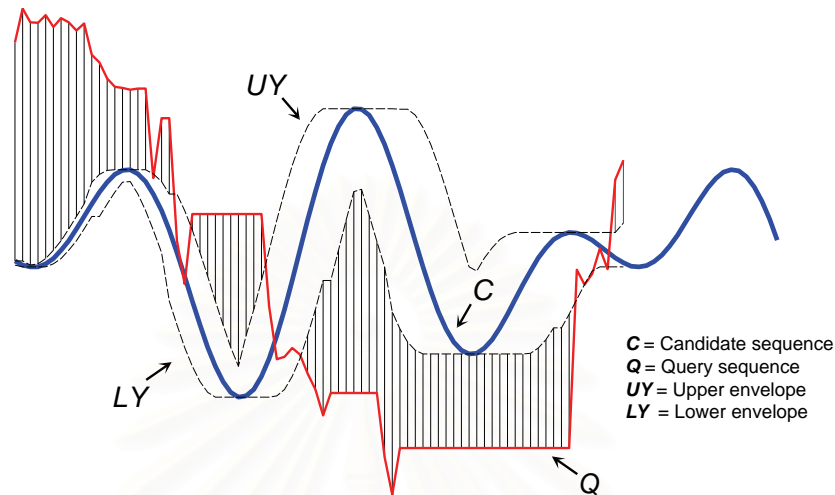
โดยที่ฟังก์ชัน $US(Q, C, sfmin, sfmax)$ คือ ฟังก์ชันที่ใช้สำหรับการคำนวณค่าระยะทางด้วยวิธีปรับขนาดเอกรูป และสำหรับฟังก์ชัน $RP(C, m, l)$ เป็นฟังก์ชันที่ใช้สำหรับการหดหรือยืดขนาดของข้อมูลส่วนต้นของอนุกรมเวลา C ที่ความยาว l ไปที่ความยาว m

อย่างไรก็ตาม แม้ว่าการคำนวณหาค่าระยะทางด้วยวิธีปรับขนาดเอกรูปนั้น จะทำให้การวัดความคล้ายคลึงกันระหว่างข้อมูลอนุกรมเวลาที่มีการหดหรือยืดขนาดสามารถทำได้เหมาะสมมากยิ่งขึ้น แต่ประสิทธิภาพในการคำนวณค่าระยะทางด้วยวิธีดังกล่าว จะเป็นฟังก์ชันพหุนามกับความยาวของข้อมูลขาเข้า ทำให้ไม่สามารถทำงานได้อย่างรวดเร็วและมีประสิทธิภาพ ดังนั้นจึงมีผู้เสนอฟังก์ชันขอบเขตล่างของการปรับขนาดเอกรูปขึ้น เพื่อแก้ไขปัญหาดังกล่าว

2.6.1 ฟังก์ชันขอบเขตล่างของการปรับขนาดเอกรูป (Lower Bounding Function of Uniform Scaling)

ฟังก์ชันขอบเขตล่างของการปรับขนาดเอกรูป มีวัตถุประสงค์เพื่อลดภาระการคำนวณค่าระยะทางด้วยวิธีปรับขนาดเอกรูป ซึ่งใช้เวลาในการคำนวณเป็นฟังก์ชันพหุนามกับความยาวของข้อมูลขาเข้า รวมทั้งใช้เส้นขอบเขตบนและเส้นขอบเขตล่างเป็นตัวแทนข้อมูลในการประมาณค่าระยะทางจริงที่ได้จากวิธีปรับขนาดเอกรูปดังแสดงในรูปที่ 2.6 เส้น UY และ LY เป็นเส้นขอบเขตบนและเส้นขอบเขตล่างตามลำดับ และ Q คือข้อมูลสอบถามและ C คือข้อมูลที่จะใช้ในการเปรียบเทียบ สำหรับเส้นในแนวตั้งจะแสดงถึงการหาค่าระยะทางด้วยวิธียุคลิด

ระหว่างข้อมูลสอบถามกับเส้นขอบเขตบนและเส้นขอบเขตล่าง สำหรับผลลัพธ์ที่ได้จากการคำนวณค่าฟังก์ชันขอบเขตล่างนี้ก็คือ ค่าระยะทางสะสมที่ได้จากการคำนวณค่าระยะทางระหว่างข้อมูลอนุกรมเวลาทั้งสอง



รูปที่ 2.6 การคำนวณค่าระยะทางขอบเขตล่างของการปรับขนาดเอกรูป ที่สามารถหดหรือยืดขนาดข้อมูลสอบถาม Q ได้ในช่วง $[0.9, 1.1]$

ฟังก์ชันขอบเขตล่างของวิธีการปรับขนาดเอกรูป [17] สามารถนิยามได้ดังนี้ ให้ $LB(Y, Q, C)$ คือ ฟังก์ชันขอบเขตล่างของการปรับขนาดเอกรูประหว่างข้อมูลอนุกรมเวลา Q และ C และให้ UY และ LY เป็นเส้นขอบเขตบนและเส้นขอบเขตล่างตามลำดับ ดังแสดงได้ในสมการที่ (2.10) และ (2.11)

$$LB(Y, Q, C) = \sum_{i=1}^m \begin{cases} (q_i - UY_i)^2 & \text{if } q_i > UY_i \\ (q_i - LY_i)^2 & \text{if } q_i < LY_i \\ 0 & \text{otherwise} \end{cases} \quad (2.10)$$

$$UY_i = \max(c_{[i*sfmin]}, \dots, c_{[i*sfmax]}) \quad (2.11)$$

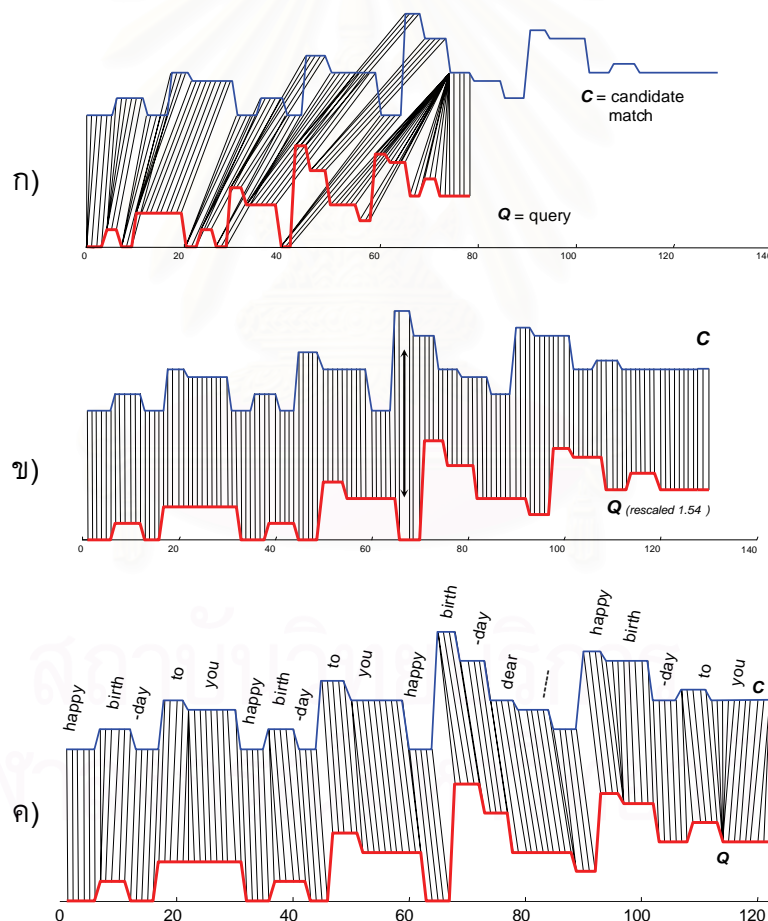
$$LY_i = \min(c_{[i*sfmin]}, \dots, c_{[i*sfmax]})$$

แม้ว่าการปรับขนาดเอกรูปจะช่วยให้การวัดความคล้ายคลึงกันของข้อมูลที่มีการหดหรือยืดขนาดสามารถทำได้เหมาะสม อย่างไรก็ตาม ในระบบค้นหาเพลงโดยการร้องทำนอง ผู้ใช้มักจะร้องทำนองในจังหวะที่ไม่คงที่ นั่นคือจะมีความแปรผันเชิงเวลาเกิดขึ้น ทำให้การวัดระยะทางโดยใช้การปรับขนาดเอกรูปเพียงอย่างเดียว อาจจะไม่เหมาะสมสำหรับการวัดความคล้ายคลึงกันของข้อมูลในรูปแบบดังกล่าว อย่างไรก็ตาม วิธีไดนามิกโทมอร์ฟิซมที่นำเสนอไปนั้น สามารถรองรับความแปรผันในเรื่องการร้องทำนองที่มีจังหวะไม่คงที่ได้ดี ดังนั้น

จึงได้มีผู้เสนอวิธีที่จะนำวิธีวัดระยะทางทั้งสองมาผสมผสานกันเพื่อให้การเปรียบเทียบความคล้ายคลึงกันของข้อมูลอนุกรมเวลาดังกล่าวสามารถทำได้ดียิ่งขึ้น

2.7 การปรับขนาดกับไทม์วอร์ปิง (Scaled and Warped Matching หรือ SWM)

การปรับขนาดกับไทม์วอร์ปิง คือการผสมผสานวิธีการปรับขนาดเอกรูปและวิธีไดนามิกไทม์วอร์ปิง [2] เข้าด้วยกัน เพื่อรองรับความแปรผันเฉพาะที่เชิงเวลาและความแปรผันโดยรวมเชิงเวลา ซึ่งเกิดขึ้นโดยทั่วไปในข้อมูลที่เก็บตัวอย่างได้จากพฤติกรรมของมนุษย์ เช่น ข้อมูลการเคลื่อนไหว ข้อมูลเสียงพูด และข้อมูลเสียงร้องทำนอง เป็นต้น ซึ่งการรวมวิธีทั้งสองนี้เข้าด้วยกัน จะช่วยให้การเปรียบเทียบความคล้ายคลึงกันระหว่างข้อมูลดังกล่าว มีความถูกต้องมากกว่าการวัดความคล้ายคลึงกันด้วยวิธีไดนามิกไทม์วอร์ปิงหรือการปรับขนาดเอกรูปเพียงอย่างเดียว ดังแสดงในรูปที่ 2.7



รูปที่ 2.7 การเปรียบเทียบวิธีคำนวณค่าระยะทางแบบต่าง ๆ ก) วิธีไดนามิกไทม์วอร์ปิง
 ข) วิธีปรับขนาดเอกรูป ค) วิธีการปรับขนาดกับไทม์วอร์ปิง [2]

สำหรับรูปที่ 2.7 จะเป็นการแสดงการเปรียบเทียบการคำนวณค่าระยะทางด้วยวิธีต่าง ๆ จากข้อมูลเสียงร้องทำนองและข้อมูลเพลงที่ถูกสกัดคุณลักษณะเกี่ยวกับโน้ตดนตรีจากเพลงแฮปปี้เบิร์ดเดย์ (Happy Birthday) โดยในรูปที่ 2.7 ก) จะแสดงการคำนวณค่าระยะทางด้วยวิธีไดนามิกไทม์วอร์ปิง ซึ่งจะเห็นว่าในช่วงต้นของอนุกรมเวลา การปรับแนวในการคำนวณค่าระยะทางระหว่างข้อมูลทั้งสองสามารถทำได้อย่างเหมาะสม อย่างไรก็ตามเนื่องจากข้อมูลทั้งสองมีจังหวะช้า เร็วไม่เท่ากัน ทำให้ข้อมูลในช่วงท้ายของเสียงร้องทำนอง จะไม่สามารถหาคู่จุดที่เหมาะสมในการคำนวณระยะทางได้ และในรูปที่ 2.7 ข) ซึ่งจะใช้วิธีปรับขนาดเอกรูปเพียงอย่างเดียว โดยวิธีนี้จะยืดขนาดของข้อมูลเสียงร้องทำนองไปที่ความยาวที่เหมาะสมก่อนการคำนวณค่าระยะทางด้วยวิธียุคลิด นั่นคือที่ความยาว 1.54 เท่าของความยาวเดิม อย่างไรก็ตาม วิธีปรับขนาดเอกรูปเพียงอย่างเดียวจะไม่สามารถรองรับความแปรผันเฉพาะที่เชิงเวลาได้ ทำให้ค่าระยะทางที่ได้จากการคำนวณดังกล่าวไม่เหมาะสมสำหรับการวัดความคล้ายคลึงกันของข้อมูลเสียงร้องทำนอง และสำหรับรูปที่ 2.7 ค) จะเป็นการผสานวิธีทั้งสองเข้าด้วยกัน ซึ่งก็คือวิธีการปรับขนาดกับไทม์วอร์ปิง การวัดระยะทางด้วยวิธีนี้จะให้ค่าระยะทางที่ถูกต้องและเหมาะสมมากที่สุด เนื่องจากสามารถรองรับทั้งความแปรผันที่เกิดจากการร้องทำนองที่มีจังหวะไม่คงที่ และการร้องทำนองที่ช้าหรือเร็วกว่าจังหวะเพลงในฐานข้อมูลได้อย่างเหมาะสม

สำหรับการคำนวณค่าระยะทางด้วยวิธีปรับขนาดกับไทม์วอร์ปิงสามารถนิยามได้ดังนี้ กำหนดให้อัตราส่วนการหดและยืดขนาดมากที่สุด คือ $sfmin$ และ $sfmax$ ตามลำดับ โดยที่ $n \geq sfmax * m$ และ m และ n คือความยาวของข้อมูลสอบถามและข้อมูลที่จะใช้ในการเปรียบเทียบ ตามลำดับ นั่นคือ ความยาวของข้อมูลที่จะใช้ในการเปรียบเทียบต้องไม่น้อยกว่าความยาวของข้อมูลสอบถามที่ถูกยืดขนาดมากที่สุด วิธีนี้ใช้สำหรับคำนวณหาค่าระยะทางน้อยสุดของข้อมูลอนุกรมเวลาทั้งสองด้วยวิธีไดนามิกไทม์วอร์ปิง ที่ข้อมูลสอบถามนั้นสามารถยืดหรือหดขนาดได้มากที่สุดที่ความยาว $sfmin * m$ ถึง $sfmax * m$ โดยสามารถแสดงเป็นสมการได้ดังนี้ [2]

$$SWM(Q, C, sfmin, sfmax, r) = \min_{l=[sfmin*m]}^{\min([sfmax*m], n)} DTW_{\text{constraint}}(RP(C, m, l), Q, r) \quad (2.12)$$

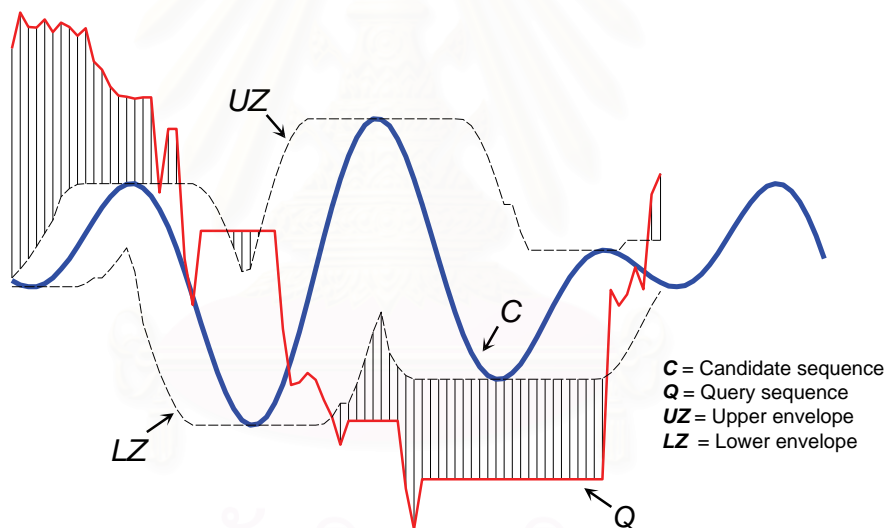
$$RP(C, m, l)_i = c_{[i*l/m]} \quad \text{where } 1 \leq i \leq m \quad (2.13)$$

โดยที่ฟังก์ชัน $SWM(Q, C, sfmin, sfmax, r)$ คือ ฟังก์ชันที่ใช้สำหรับการคำนวณค่าระยะทางด้วยวิธีปรับขนาดกับไทม์วอร์ปิง และสำหรับฟังก์ชัน $RP(C, m, l)$ เป็นฟังก์ชันที่ใช้สำหรับการหดหรือยืดขนาดของข้อมูลส่วนต้นของอนุกรมเวลา C ความยาว l ไปที่ความยาว m นั่นคือ ความยาวของข้อมูลสอบถามก่อนที่จะคิดค่าระยะทางด้วยวิธีไดนามิกไทม์วอร์ปิง

เช่นเดียวกับวิธีไดนามิกไทม์วอร์ปิง และวิธีปรับขนาดเอกรูป วิธีการปรับขนาดกับไทม์วอร์ปิงนี้ จะใช้ระยะเวลาในการประมวลผลนาน ซึ่งจะทำให้ไม่สามารถนำมาใช้กับการค้นหาข้อมูลในฐานข้อมูลที่มีขนาดใหญ่ได้ ดังนั้นจึงมีนักวิจัยได้เสนอฟังก์ชันขอบเขตล่างของการปรับขนาดกับไทม์วอร์ปิงขึ้น [2]

2.7.1 ฟังก์ชันขอบเขตล่างของการปรับขนาดกับไทม์วอร์ปิง (Lower Bound Function of SWM)

ฟังก์ชันขอบเขตล่างของการปรับขนาดกับไทม์วอร์ปิงนั้น มีไว้เพื่อลดภาระในการคำนวณค่าระยะทางด้วยวิธีปรับขนาดกับไทม์วอร์ปิง สำหรับแนวคิดของฟังก์ชันขอบเขตล่างของวิธีดังกล่าวนั้น คล้ายกันกับฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปิง และฟังก์ชันขอบเขตล่างของการปรับขนาดเอกรูป โดยสามารถแสดงได้ดังรูปที่ 2.8 ซึ่งเป็นการคำนวณระยะทางด้วยวิธียุคลิดระหว่างเส้นขอบเขตบนและเส้นขอบเขตล่าง UZ และ LZ ตามลำดับกับข้อมูลสอบถาม Q



รูปที่ 2.8 การคำนวณค่าระยะทางขอบเขตล่างของการปรับขนาดกับไทม์วอร์ปิง โดยที่กำหนดให้ $r = 5\%$ และให้สามารถหดหรือยืดขนาดข้อมูลสอบถามได้ในช่วง $[0.9, 1.1]$

ฟังก์ชันขอบเขตล่างของการปรับขนาดกับไทม์วอร์ปิง [2] สามารถนิยามได้ดังนี้ ให้ $LBZ(Q,C)$ คือ ฟังก์ชันขอบเขตล่างของการปรับขนาดกับไทม์วอร์ปิงระหว่างข้อมูลอนุกรมเวลา Q และ C และให้ UZ และ LZ เป็นเส้นขอบเขตบนและเส้นขอบเขตล่างตามลำดับดังแสดงได้ในสมการที่ (2.14) และ (2.15)

$$LBZ(Q, C) = \sum_{i=1}^m \begin{cases} (q_i - UZ_i)^2 & \text{if } q_i > UZ_i \\ (q_i - LZ_i)^2 & \text{if } q_i < LZ_i \\ 0 & \text{otherwise} \end{cases} \quad (2.14)$$

$$UZ_i = \max(c_{\max(1, \lfloor i * sfmin \rfloor - r)}, \dots, c_{\min(\lfloor i * sfmax \rfloor + r, n)}) \quad (2.15)$$

$$LZ_i = \min(c_{\max(1, \lfloor i * sfmin \rfloor - r)}, \dots, c_{\min(\lfloor i * sfmax \rfloor + r, n)})$$

สำหรับในลำดับถัดไปจะเป็นการนำเสนอวิธีลดมิติข้อมูลอนุกรมเวลา ซึ่งมีความสำคัญอย่างยิ่งในการเพิ่มความเร็วในการเปรียบเทียบความคล้ายคลึงกันของข้อมูลอนุกรมเวลาต่าง ๆ

2.8 การลดมิติข้อมูล (Dimensionality Reduction)

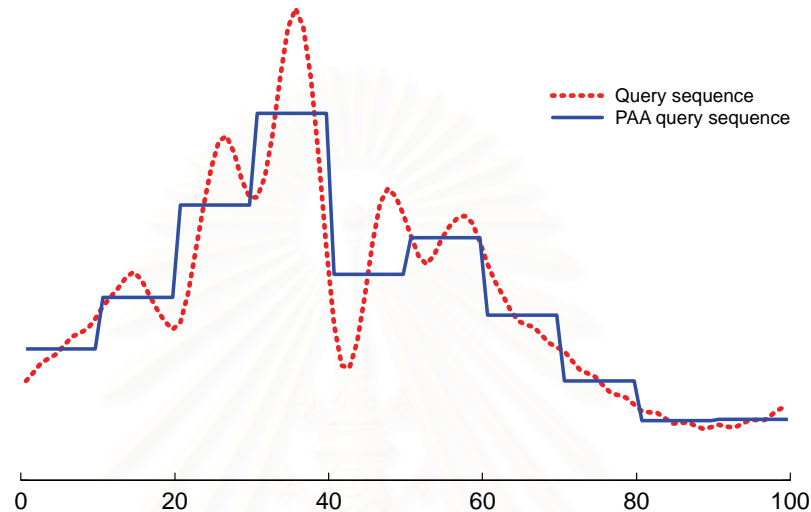
สำหรับการค้นคืนข้อมูลอนุกรมเวลาในฐานะข้อมูลขนาดใหญ่ ข้อมูลมักจะถูกจัดเก็บอยู่ในหน่วยเก็บข้อมูลแบบทุติยภูมิ (Secondary Storage) เช่น จานบันทึกแบบแข็ง (Hard disk) ซึ่งมีความจุมากกว่าหน่วยเก็บข้อมูลแบบปฐมภูมิ (Primary Storage) เช่น หน่วยความจำหลัก (Main Memory) หรือ แรม (RAM) อย่างไรก็ตาม การค้นคืนข้อมูลในหน่วยเก็บข้อมูลแบบทุติยภูมินั้น จะช้ากว่าการค้นคืนข้อมูลในหน่วยเก็บข้อมูลแบบปฐมภูมิมาก เนื่องจากความเร็วในการเข้าถึงข้อมูล (Access Time) ของจานบันทึกแบบแข็ง ช้ากว่าความเร็วในการเข้าถึงข้อมูลของแรม ดังนั้นการลดขนาดหรือการลดมิติของข้อมูลจึงมีความสำคัญเพื่อที่จะสร้างดัชนี (Index) ของฐานข้อมูลดังกล่าวที่มีขนาดเล็กเพียงพอที่จะจัดเก็บอยู่บนหน่วยเก็บข้อมูลแบบปฐมภูมิได้ เพื่อให้การค้นคืนข้อมูลมีประสิทธิภาพมากยิ่งขึ้น

ในช่วงกว่าสิบปีมานี้ ได้มีงานวิจัยที่ศึกษาเกี่ยวกับการลดมิติของข้อมูลอนุกรมเวลาออกมาเป็นจำนวนมาก เนื่องจากฐานข้อมูลอนุกรมเวลาโดยทั่วไป มักจะมีขนาดใหญ่ จึงทำให้มีความจำเป็นที่จะต้องสร้างดัชนีให้กับข้อมูลอนุกรมเวลาเหล่านั้น เพื่อให้การค้นคืนข้อมูลอนุกรมเวลาสามารถทำได้อย่างรวดเร็วและมีประสิทธิภาพ ในปัจจุบันการลดมิติข้อมูลแบบพีเอ-เอ [18] ซึ่งเป็นวิธีที่ใช้สำหรับลดมิติข้อมูลอนุกรมเวลา ได้มีการนำไปใช้กันอย่างแพร่หลายมากที่สุด รวมทั้งการนำไปใช้ลดมิติข้อมูลเพื่อสร้างดัชนีให้กับข้อมูลอนุกรมเวลา เนื่องจากเป็นวิธีที่มีประสิทธิภาพสูงและไม่ซับซ้อน [10, 14, 16, 19, 20]

2.8.1 การลดมิติข้อมูลแบบพีเอเอ (Piecewise Aggregate Approximation หรือ PAA)

การลดมิติข้อมูลแบบพีเอเอ เป็นวิธีการลดมิติของข้อมูลอนุกรมเวลา ซึ่งจะเริ่มจากการแบ่งข้อมูลอนุกรมเวลาออกเป็น N ส่วน โดยที่ N คือ ความยาวของข้อมูลที่ต้องการ

หลังจากการลดมิติข้อมูลแบบพีเอเอ โดยแต่ละส่วนที่แบ่งนั้นจะต้องมีความยาวเท่ากัน เช่น ข้อมูลอนุกรมเวลา X มีความยาวเท่ากับ 100 มิติ ต้องการที่จะลดขนาดของข้อมูลอนุกรมเวลา X นี้ให้เหลือความยาว 10 มิติ เพราะฉะนั้น เริ่มแรกจะต้องทำการแบ่งข้อมูลอนุกรมเวลา X ออกเป็น 10 ส่วนย่อย ๆ โดยที่ในแต่ละส่วนจะมีความยาวเท่ากัน คือ 10 มิติ จากนั้นจึงหาค่าเฉลี่ยของแต่ละส่วน เพื่อใช้เป็นตัวแทนข้อมูลในส่วนนั้น ซึ่งสามารถแสดงได้ดังรูปที่ 2.9



รูปที่ 2.9 การลดมิติข้อมูลแบบพีเอเอ

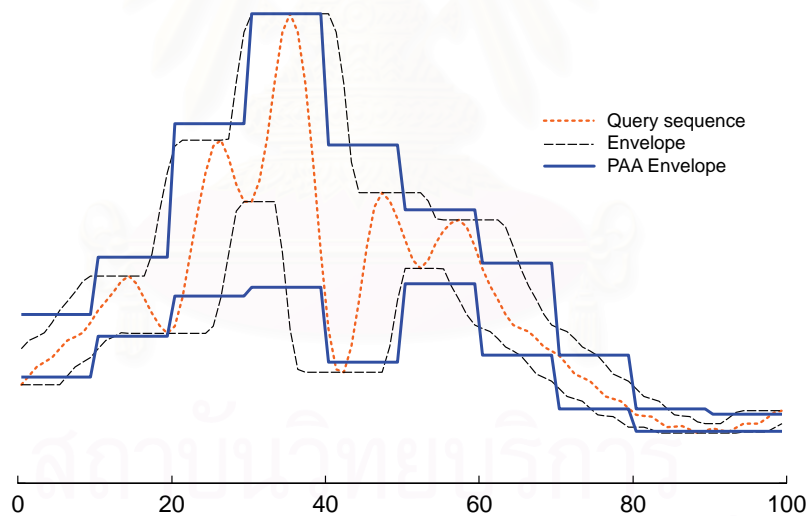
สำหรับฟังก์ชันของการลดมิติข้อมูลแบบพีเอเอ [18] สามารถนิยามได้ดังนี้ ให้ $X = x_1, x_2, x_3, \dots, x_n$ เป็นข้อมูลอนุกรมเวลาที่มีความยาว n และให้ N เป็นความยาวมิติที่ต้องการ ภายหลังการลดมิติข้อมูลแบบพีเอเอ โดยที่ $1 \leq N \leq n$ และ N เป็นตัวประกอบของ n ซึ่งผลลัพธ์ของการลดมิติข้อมูลแบบพีเอเอ จะได้ข้อมูลอนุกรมเวลา $\bar{X} = \bar{x}_1, \bar{x}_2, \bar{x}_3, \dots, \bar{x}_N$ ที่มีความยาว N โดยที่ค่า \bar{x}_i สามารถหาได้ดังสมการต่อไปนี้

$$\bar{x}_i = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n}{N}i} x_j \quad (2.16)$$

สำหรับการค้นคืนข้อมูลอนุกรมเวลานั้น จะต้องลดขนาดของข้อมูลสอบถามหรือลดขนาดของข้อมูลที่จะนำมาใช้ในการเปรียบเทียบ และเพื่อให้การค้นหามีประสิทธิภาพมากยิ่งขึ้น การพัฒนาฟังก์ชันขอบเขตล่างสำหรับข้อมูลที่ถูกลดมิติ ก็มีความสำคัญเช่นกัน

2.8.2 ฟังก์ชันขอบเขตล่างของข้อมูลที่ถูกลดมิติแบบพีเอเอ (PAA Envelope Transformation)

สำหรับฟังก์ชันขอบเขตล่างของข้อมูลที่ถูกลดมิตินั้น ได้มีผู้เสนอวิธีการต่าง ๆ ด้วยกันหลายวิธี เช่น Keogh และคนอื่น ๆ ได้เสนอฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ป-ปิง [16] และฟังก์ชันขอบเขตล่างของการปรับขนาดเอกรูป [19] สำหรับข้อมูลที่ถูกลดมิติแบบพีเอเอ และต่อมา Zhu และ Shasha ได้เสนอฟังก์ชันขอบเขตล่างของไดนามิกไทม์วอร์ป-ปิงสำหรับข้อมูลที่ถูกลดมิติ [10] ที่มีความกระชับ (Tightness) สูงกว่าวิธีที่เสนอโดย Keogh และคนอื่น ๆ [16] ดังนั้นในที่นี้ จะขอกล่าวถึงฟังก์ชันขอบเขตล่างของข้อมูลที่ถูกลดมิติแบบพีเอเอที่เสนอโดย Zhu และ Shasha โดยแนวคิดในการแปลงฟังก์ชันขอบเขตล่างนี้ให้สามารถรองรับการคำนวณค่าระยะทางขอบเขตล่างของข้อมูลที่ถูกลดมิติแบบพีเอเอนั้น เริ่มต้นจากการแปลงเส้นขอบเขตบนและเส้นขอบเขตล่างที่ได้จากฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ป-ปิงด้วยวิธีการลดมิติข้อมูลแบบพีเอเอ ดังแสดงในรูปที่ 2.10 จากนั้นจึงนำเส้นขอบเขตบนและเส้นขอบเขตล่างนี้ ไปคิดค่าระยะทางขอบเขตล่างกับข้อมูลที่จะนำมาเปรียบเทียบที่ถูกลดมิติด้วยวิธีพีเอเอต่อไป



รูปที่ 2.10 เส้นขอบเขตบนและเส้นขอบเขตล่างของข้อมูลที่ถูกลดมิติแบบพีเอเอ

สำหรับการแปลงฟังก์ชันขอบเขตล่างด้วยวิธีการลดมิติแบบพีเอเอ [10] สามารถนิยามได้ดังนี้

$$LB_{PAA}(\bar{Q}, \bar{C}) = \frac{n}{N} \sum_{i=1}^n \begin{cases} (\overline{UX}_i - \bar{c}_i)^2 & \text{if } \bar{c}_i > \overline{UX}_i \\ (\overline{LX}_i - \bar{c}_i)^2 & \text{if } \bar{c}_i < \overline{LX}_i \\ 0 & \text{otherwise} \end{cases} \quad (2.17)$$

$$\overline{UX}_i = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n}{N}i} UX_j, \quad \overline{LX}_i = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n}{N}i} LX_j \quad (2.18)$$

โดยที่ $LB_{PAA}(\overline{Q}, \overline{C})$ คือฟังก์ชันขอบเขตล่างของข้อมูลที่ถูกลดมิติแบบพีเอเอ \overline{C} คือข้อมูลที่จะใช้ในการเปรียบเทียบที่ผ่านการลดมิติแบบพีเอเอ n คือความยาวของข้อมูลสอบถาม ซึ่งมีความยาวเท่ากับความยาวของข้อมูลที่จะนำมาเปรียบเทียบ N คือ ความยาวของข้อมูลที่ต้องการภายหลังการลดมิติแบบพีเอเอ UX และ LX เป็นเส้นขอบเขตบนและเส้นขอบเขตล่างของไดนามิกไทม์วอร์ปิง และ \overline{UX} และ \overline{LX} เป็นเส้นขอบเขตบนและเส้นขอบเขตล่างที่ถูกลดมิติแบบพีเอเอ

สำหรับหัวข้อทฤษฎีที่เกี่ยวข้องดังที่กล่าวมาทั้งหมดนั้น บางส่วนจะเป็นทฤษฎีที่ได้นำมาประยุกต์ใช้ในระบบค้นหาเพลงโดยการร้องทำนองที่เคยมีการวิจัยมาในอดีต บางส่วนจะทฤษฎีที่งานวิจัยนี้จะนำมาปรับปรุงและประยุกต์ใช้กับระบบค้นหาเพลงโดยการร้องทำนองให้มีความแม่นยำและรวดเร็วมากยิ่งขึ้น โดยในหัวข้อถัดไปจะเป็นหัวข้องานวิจัยที่เกี่ยวข้อง ซึ่งจะกล่าวถึงงานวิจัยต่าง ๆ ที่เกี่ยวข้องกับระบบค้นหาเพลงโดยการร้องทำนองที่มีการวิจัยกันมาในอดีตจวบจนถึงปัจจุบัน รวมทั้งงานวิจัยที่เกี่ยวกับการค้นคืนข้อมูลอนุกรมเวลา

2.9 งานวิจัยที่เกี่ยวข้อง

ในช่วงกว่าทศวรรษที่ผ่านมา ตั้งแต่งานวิจัยของ Ghias และคนอื่น ๆ [1] ในปี ค.ศ.1995 ที่ได้เสนอระบบค้นหาเพลงโดยการร้องทำนองขึ้น ก็ได้มีงานวิจัยอื่น ๆ ที่ศึกษาและพัฒนาเกี่ยวกับเรื่องนี้ออกมาเป็นจำนวนมาก โดยในงานวิจัยต่าง ๆ ก็ได้เสนอวิธีการที่แตกต่างกันไป เพื่อที่จะปรับปรุงประสิทธิภาพของระบบค้นหาเพลงโดยการร้องทำนอง ทั้งในแง่ของความเร็ว ความแม่นยำ ความสามารถในการรองรับความแปรผันต่าง ๆ ของผู้ใช้ การลดข้อจำกัดในการใช้งานระบบค้นหาเพลงโดยการร้องทำนอง รวมทั้งการเพิ่มความสามารถในการรองรับกับฐานข้อมูลที่มีขนาดใหญ่ได้อย่างมีประสิทธิภาพ ซึ่งสามารถจำแนกประเภทของระบบค้นหาเพลงโดยการร้องทำนองที่มีอยู่ในปัจจุบัน ตามวิธีการแทนข้อมูลเพลงและเสียงร้องทำนองในรูปแบบต่าง ๆ ได้หลายประเภท แต่ที่ได้นำมาศึกษา วิจัย และพัฒนากันอย่างจริงจังมีเพียง 2 ประเภทหลัก ๆ ดังนี้

2.9.1 ระบบค้นหาเพลงโดยการร้องทำนองที่ใช้การแทนข้อมูลด้วยสายอักขระ

การแทนข้อมูลเพลงและเสียงร้องทำนองด้วยสายอักขระนี้ ได้นำมาใช้ในการศึกษาวิจัยระบบค้นหาเพลงโดยการร้องทำนองตั้งแต่ในยุคแรก ๆ ของการพัฒนา โดยแนวคิดของการค้นหาเพลงโดยการร้องทำนองที่ใช้การแทนข้อมูลในรูปแบบนี้ ก็คือ แปลงทั้ง

ข้อมูลเพลงและเสียงร้องทำนองให้อยู่ในรูปแบบของสายอักขระ ซึ่งอาจเป็นรหัสพาร์สัน (Parson's code) [21, 22] ดังแสดงในรูปที่ 2.11 ซึ่งเป็นการแทนข้อมูลเสียงร้องทำนองและข้อมูลเพลงด้วยอักขระ 3 ตัว ก็คือ U D และ R โดยที่อักขระ U หมายความว่า ระดับเสียงของโน้ตตัวปัจจุบันสูงกว่าระดับเสียงของโน้ตตัวก่อนหน้า อักขระ D หมายความว่า ระดับเสียงของโน้ตตัวปัจจุบันต่ำกว่าระดับเสียงของโน้ตตัวก่อนหน้า และอักขระ R คือ โน้ตตัวปัจจุบันและโน้ตตัวก่อนหน้ามีระดับเสียงเท่ากัน จากนั้นจึงนำสายอักขระที่ได้จากการแปลงเสียงร้องทำนองของผู้ใช้ ไปเปรียบเทียบกับคุณลักษณะของเพลงในฐานข้อมูลที่เป็นรหัสพาร์สันด้วยวิธีการเปรียบเทียบสายอักขระ เพื่อวัดความคล้ายคลึงกันระหว่างสายอักขระที่ได้จากเสียงร้องทำนองกับสายอักขระของเพลงต่าง ๆ ระบบค้นหาเพลงโดยการร้องทำนองที่ใช้การแทนข้อมูลเพลงและเสียงร้องทำนองด้วยสายอักขระนี้ ได้แก่ ระบบ Musipedia [22] ระบบ CubyHum [23] และในงานวิจัยอื่น ๆ อีกจำนวนมาก [1, 4, 7, 8]



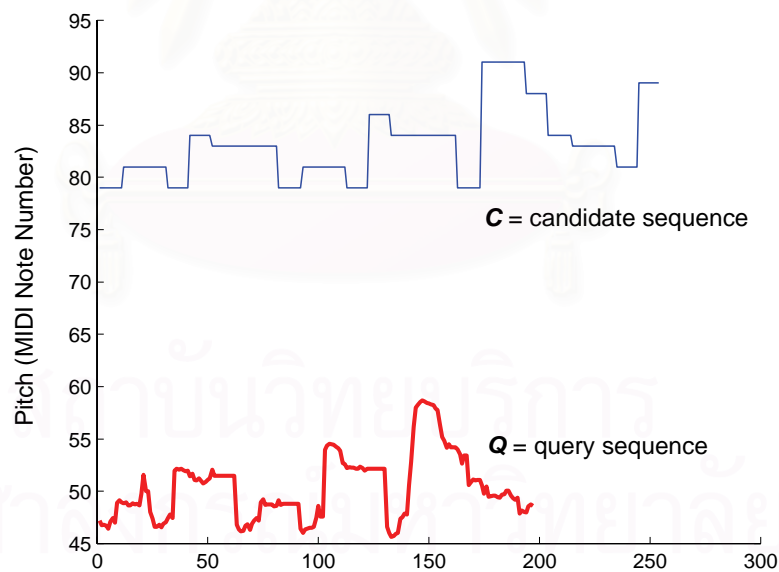
รูปที่ 2.11 การแทนข้อมูลเพลงด้วยรหัสพาร์สัน [21]

อย่างไรก็ตามระบบค้นหาเพลงโดยการร้องทำนองที่ใช้การแทนข้อมูลเพลงและเสียงร้องทำนองด้วยสายอักขระนั้น ยังไม่สามารถทำงานได้อย่างแม่นยำ เพราะว่าเสียงร้องทำนองของผู้ใช้มีความแปรผันอยู่มาก ทำให้การแปลงข้อมูลเสียงร้องทำนองให้เป็นสายอักขระไม่สามารถทำได้อย่างถูกต้อง ดังนั้นข้อมูลสอบถามสำหรับระบบค้นหาเพลงโดยการร้องทำนองที่ใช้การแทนข้อมูลด้วยสายอักขระจึงมีคุณภาพต่ำ สุดท้ายผลลัพธ์ที่ได้จากการค้นหาเพลงจึงไม่แม่นยำ ดังนั้นจึงได้มีผู้เสนอแนวคิดในการนำวิธีทางสถิติมาประยุกต์ใช้ในการค้นหาเพลงโดยการร้องทำนองเพื่อรองรับความแปรผันและข้อผิดพลาดที่เกิดจากการแปลงสายอักขระดังกล่าว เช่น งานวิจัยของ Hu และคนอื่น ๆ [24] เสนอให้นำตารางความน่าจะเป็นในการแปลงเสียงร้องทำนองเป็นโน้ตดนตรีมาใช้ร่วมกับวิธีระยะแก้ไข (Edit Distance) เพื่อแก้ปัญหการร้องเพี้ยนของผู้ใช้ งานวิจัยของ Liu และคนอื่น ๆ [25] ได้นำเสนอวิธีแบบจำลองฮิดเดนมาร์คอฟ (Hidden Markov Model) ในการค้นหาเพลงที่ใกล้เคียงกับเสียงร้องทำนองของผู้ใช้มากที่สุด อย่างไรก็ตามวิธีที่นำเสนอในงานวิจัยเหล่านี้ ยังมีข้อจำกัดในเรื่องความสามารถในการรองรับความแปรผันของผู้ใช้ เช่น ความสามารถในการรองรับข้อมูลเสียงร้องทำนองที่มีโน้ตขาดหรือเกินมากกว่า 1 ตัวโน้ต ความสามารถในการรองรับเสียงร้องของผู้ใช้ที่ร้องด้วยพยางค์อื่นนอกจากเสียง “ทา” “ตา” หรือ “ลา” เป็นต้น

แม้ว่าวิธีทางสถิติที่นำมาประยุกต์ใช้กับระบบค้นหาเพลงโดยการร้องทำนองนั้นสามารถเพิ่มความแม่นยำให้กับระบบได้ อย่างไรก็ตาม ในหลายระบบก็ไม่สามารถเพิ่มความแม่นยำได้อย่างมีนัยสำคัญ หรือแม้ว่าบางระบบสามารถเพิ่มความแม่นยำได้ดี เช่น ในงานวิจัยของ Liu และคนอื่น ๆ แต่ก็ทำให้ความเร็วในการค้นหาของระบบลดลงอย่างมาก

2.9.2 ระบบค้นหาเพลงโดยการร้องทำนองที่ใช้การแทนข้อมูลด้วยอนุกรมเวลา

เพื่อหลีกเลี่ยงความแปรผันของผู้ใช้ อันทำให้การแปลงข้อมูลเสียงร้องทำนองเป็นสายอักขระไม่สามารถทำงานได้อย่างถูกต้อง ดังนั้นจึงมีผู้เสนอวิธีค้นหาเพลงโดยการร้องทำนองที่ใช้การแทนข้อมูลเพลงและเสียงร้องทำนองด้วยข้อมูลอนุกรมเวลาขึ้น โดยข้อมูลอนุกรมเวลานี้ จะได้จากการสกัดคุณลักษณะเกี่ยวกับระดับเสียงของเสียงร้องทำนองออกมาในรูปแบบของคอนทัวร์ระดับเสียง ซึ่งมีรายละเอียดเกี่ยวกับเสียงร้องทำนองครบถ้วนและถูกต้องมากกว่าข้อมูลเสียงร้องทำนองที่ผ่านกระบวนการแปลงเป็นสายอักขระ จากนั้นจึงนำคอนทัวร์ระดับเสียงนี้ไปเปรียบเทียบกับคอนทัวร์ระดับเสียงของเพลงที่สกัดจากแฟ้มข้อมูลมิติในฐานข้อมูล สำหรับในรูปที่ 2.12 จะเป็นตัวอย่างของแฟ้มข้อมูลมิติของเพลงแฮปปี้เบิร์ดเดย์และเสียงร้องทำนองของเพลงเดียวกัน ที่ถูกสกัดคุณลักษณะเกี่ยวกับระดับเสียงออกมาในรูปแบบของคอนทัวร์ระดับเสียงซึ่งเป็นข้อมูลอนุกรมเวลา



รูปที่ 2.12 การแทนข้อมูลเพลง (บน) และเสียงร้องทำนอง (ล่าง) ด้วยอนุกรมเวลา

ระบบค้นหาเพลงโดยการร้องทำนองที่ใช้การแทนข้อมูลเพลงและเสียงร้องทำนองด้วยอนุกรมเวลานั้น ได้มีการศึกษาวิจัยกันอย่างกว้างขวาง เช่น งานวิจัยของ Jang และ Lee [9] Lee และคนอื่น ๆ [6] Wang และ Zhang [26] Zhu และ Shasha [10] เป็นต้น ซึ่ง

ทั้งหมดนี้ใช้วิธีไดนามิกไทม์วอร์ปปีงในการเปรียบเทียบความคล้ายคลึงกันระหว่างเสียงร้อง ทำนองและเพลงที่ถูกสกัดคุณลักษณะให้อยู่ในรูปแบบของอนุกรมเวลา แต่ทั้งนี้ส่วนที่แตกต่างกันของระบบทั้งหมดที่กล่าวมาก็คือ ส่วนของการเพิ่มประสิทธิภาพในเชิงความเร็ว และการลดข้อจำกัดต่าง ๆ ในการใช้งาน

วิธีไดนามิกไทม์วอร์ปปีงเป็นที่ยอมรับกันอย่างแพร่หลายถึงความแม่นยำในการเปรียบเทียบข้อมูลอนุกรมเวลาประเภทต่าง ๆ รวมทั้งในระบบการค้นหาเพลงโดยการร้อง ทำนอง [4, 8] ซึ่งระบบค้นหาเพลงโดยการร้องทำนองที่ใช้การแทนข้อมูลเพลงและเสียงร้อง ทำนองด้วยอนุกรมเวลานี้ สามารถรองรับความแปรผันต่าง ๆ ของผู้ใช้ได้เป็นอย่างดี อย่างไรก็ตาม แม้ว่าวิธีนี้จะให้ผลลัพธ์ที่มีความแม่นยำสูง แต่ก็ใช้ระยะเวลาในการประมวลผลนาน รวมทั้งในบางงานวิจัยยังมีข้อจำกัดในการใช้งานค่อนข้างมากอีกด้วย [2, 9, 10]

เมื่อไม่นานมานี้ มีงานวิจัยได้เปรียบเทียบถึงประสิทธิภาพของวิธีการค้นหาเพลงโดยการร้องทำนองแบบต่าง ๆ [4, 7, 8] ทั้งที่ใช้การค้นหาเพลงโดยการเปรียบเทียบสายอักขระแบบทั่วไป การค้นหาเพลงโดยใช้วิธีทางสถิติในการเปรียบเทียบสายอักขระ ซึ่งในงานวิจัยดังกล่าวได้เลือกใช้วิธีแบบจำลองฮิดเดนมาร์คอฟมาใช้ในการเปรียบเทียบสายอักขระ และวิธีที่สามที่ได้นำมาใช้เปรียบเทียบก็คือ การค้นหาเพลงโดยการร้องทำนองที่ใช้การแทนข้อมูลด้วยอนุกรมเวลา ซึ่งได้ใช้วิธีไดนามิกไทม์วอร์ปปีงในการเปรียบเทียบข้อมูล พร้อมทั้งได้กล่าวถึงข้อดี ข้อเสียของแต่ละวิธีไว้อย่างชัดเจน โดยวิธีที่ใช้การแทนข้อมูลอนุกรมเวลาจะมีความแม่นยำสูงที่สุด รองลงมาคือวิธีที่ใช้การแทนข้อมูลด้วยสายอักขระที่ใช้วิธีฮิดเดนมาร์คอฟในการเปรียบเทียบข้อมูล และวิธีที่ใช้การแทนข้อมูลด้วยสายอักขระที่ใช้วิธีการเปรียบเทียบแบบทั่วไป ตามลำดับ อย่างไรก็ตาม แม้ว่าวิธีที่ใช้การแทนข้อมูลด้วยอนุกรมเวลาจะให้ผลลัพธ์ที่มีความแม่นยำสูงที่สุด แต่วิธีนี้ก็ไม่สามารถทำงานได้อย่างรวดเร็วเช่นเดียวกับ วิธีการค้นหาเพลงโดยใช้การเปรียบเทียบสายอักขระแบบทั่วไป

อย่างไรก็ตาม ในปัจจุบันการศึกษาวิจัยกระบวนการค้นคืนข้อมูลอนุกรมเวลา (Time Series Retrieval) ได้มีวิธีต่าง ๆ ที่ทำให้การค้นหาข้อมูลอนุกรมเวลานั้น สามารถทำงานได้มีประสิทธิภาพมากยิ่งขึ้น ทั้งในด้านความเร็วและความแม่นยำในการประมวลผล เช่น ในงานวิจัยของ Ratanamahatana และ Keogh [15] ได้แสดงให้เห็นว่าเงื่อนไขบังคับโดยรวมที่มีขนาดเหมาะสมนั้น นอกจากจะช่วยให้สามารถคำนวณระยะทางด้วยวิธีไดนามิกไทม์วอร์ปปีงได้อย่างรวดเร็วแล้ว ยังสามารถเพิ่มความแม่นยำในการเปรียบเทียบข้อมูลอนุกรมเวลาได้ดีกว่าการใช้เงื่อนไขบังคับโดยรวมที่มีขนาดใหญ่เกินไปอีกด้วย และยังมีงานวิจัยอื่น ๆ อีก [10, 16, 19, 20] ที่แสดงให้เห็นว่าฟังก์ชันขอบเขตล่างที่ได้จากเงื่อนไขบังคับโดยรวมของวิธีไดนามิกไทม์วอร์ปปีงนั้น สามารถนำไปสร้างเป็นดัชนีเพื่อเพิ่มความเร็วในการค้นหาข้อมูล รวมทั้งในงานวิจัยของ Sakurai และคนอื่น ๆ [3] ที่แสดงให้เห็นถึงประสิทธิภาพของวิธีการค้นหาตามลำดับ (Sequential Search) ที่เสนอในงานวิจัยของพวกเขา นั้น สามารถค้นหาข้อมูลได้

รวดเร็วกว่าวิธีการค้นหาข้อมูลโดยใช้ดัชนี ที่เสนอในงานวิจัยของ Zhu และ Shasha [10] ถึง 222 เท่า

นอกจากจะมีการศึกษาวิจัยกระบวนการค้นคืนข้อมูลอนุกรมเวลาในแง่ของการเพิ่มประสิทธิภาพในการทำงานแล้ว เมื่อไม่นานมานี้ ได้มีงานวิจัยที่ได้เสนอวิธีที่จะเพิ่มความแม่นยำในการค้นคืนข้อมูลอนุกรมเวลาอีกด้วย นั่นคือ งานวิจัยของ Fu และคนอื่น ๆ [2] โดยในงานวิจัยนี้ได้เสนอวิธีปรับขนาดกับโทมวอร์ปิง ซึ่งเป็นการผสมวิธีปรับขนาดเอกรูปและไดนามิกโทมวอร์ปิง เพื่อรองรับข้อมูลที่มีการหดหรือยืดขนาดของข้อมูลอย่างสม่ำเสมอ และรองรับข้อมูลที่มีความแปรผันเฉพาะที่เชิงเวลา พร้อมทั้งเสนอฟังก์ชันขอบเขตล่างที่รองรับการสร้างดัชนี เพื่อเพิ่มความเร็วในการประมวลผลอีกด้วย

สำหรับงานวิจัยนี้ ผู้วิจัยมีความสนใจที่จะนำวิธีปรับขนาดกับโทมวอร์ปิง มาใช้ในงานวิจัยการค้นหาเพลงโดยการร้องทำนอง เนื่องจากวิธีปรับขนาดกับโทมวอร์ปิงนั้นสามารถรองรับผู้ใช้ที่ร้องทำนองช้าหรือเร็วกว่าจังหวะเพลงที่ผู้ใช้ต้องการได้อย่างเหมาะสม และน่าจะให้ความแม่นยำที่สูงกว่าการใช้วิธีไดนามิกโทมวอร์ปิงเพียงอย่างเดียว อย่างไรก็ตาม วิธีปรับขนาดกับโทมวอร์ปิงที่นำเสนอในงานวิจัยของ Fu และคนอื่น ๆ [2] นั้น ได้มองข้ามความจำเป็นในการแปลงข้อมูลให้เป็นบรรทัดฐานทุกครั้งที่มีการยืดหรือหดขนาดของข้อมูล ซึ่งถือเป็นขั้นตอนสำคัญที่ทำให้ข้อมูลที่จะนำมาเปรียบเทียบนั้น อยู่ในระดับอ้างอิงเดียวกันกับระดับอ้างอิงของข้อมูลสอบถามเสมอ อันจะส่งผลให้การเปรียบเทียบความคล้ายคลึงกันระหว่างข้อมูลทั้งสองมีความแม่นยำมากยิ่งขึ้น สำหรับวิธีการแก้ปัญหาดังกล่าวและการนำไปประยุกต์ใช้กับระบบค้นหาเพลงโดยการร้องทำนอง เพื่อให้สามารถค้นหาข้อมูลได้อย่างแม่นยำและรวดเร็วนั้น จะกล่าวไว้ในหัวข้อถัดไป

บทที่ 3

การค้นหาเพลงโดยการร้องทำนอง

การพัฒนาระบบค้นหาเพลงโดยการร้องทำนองโดยทั่วไป จะประกอบด้วย 3 ส่วนหลัก ๆ คือ 1. ส่วนของการสกัดคุณลักษณะจากเสียงร้องทำนองและแทนข้อมูลเสียงร้องดังกล่าวด้วยรูปแบบที่เหมาะสม 2. ส่วนของการสกัดคุณลักษณะจากเพลงในฐานข้อมูลพร้อมทั้งแทนข้อมูลคุณลักษณะดังกล่าวด้วยรูปแบบที่เหมาะสม 3. การค้นหาเพลงจากเสียงร้องทำนองที่อยู่ในรูปแบบที่เหมาะสม สำหรับงานวิจัยนี้ ให้ความสนใจเกี่ยวกับการค้นหาเพลงจากเสียงร้องทำนองที่อยู่ในรูปแบบอนุกรมเวลา ซึ่งถือว่ามีประสิทธิภาพในด้านความแม่นยำสูงสุด [4, 7, 8]

สำหรับบทที่ 3 นี้ จะเริ่มต้นนำเสนอจาก ส่วนของการสกัดคุณลักษณะจากเสียงร้องทำนอง ตามด้วยการสกัดคุณลักษณะจากเพลงในฐานข้อมูล และสุดท้ายคือ การค้นหาเพลงจากเสียงร้องทำนองที่อยู่ในรูปแบบอนุกรมเวลาภายใต้ฐานข้อมูลอนุกรมเวลาที่มีขนาดใหญ่

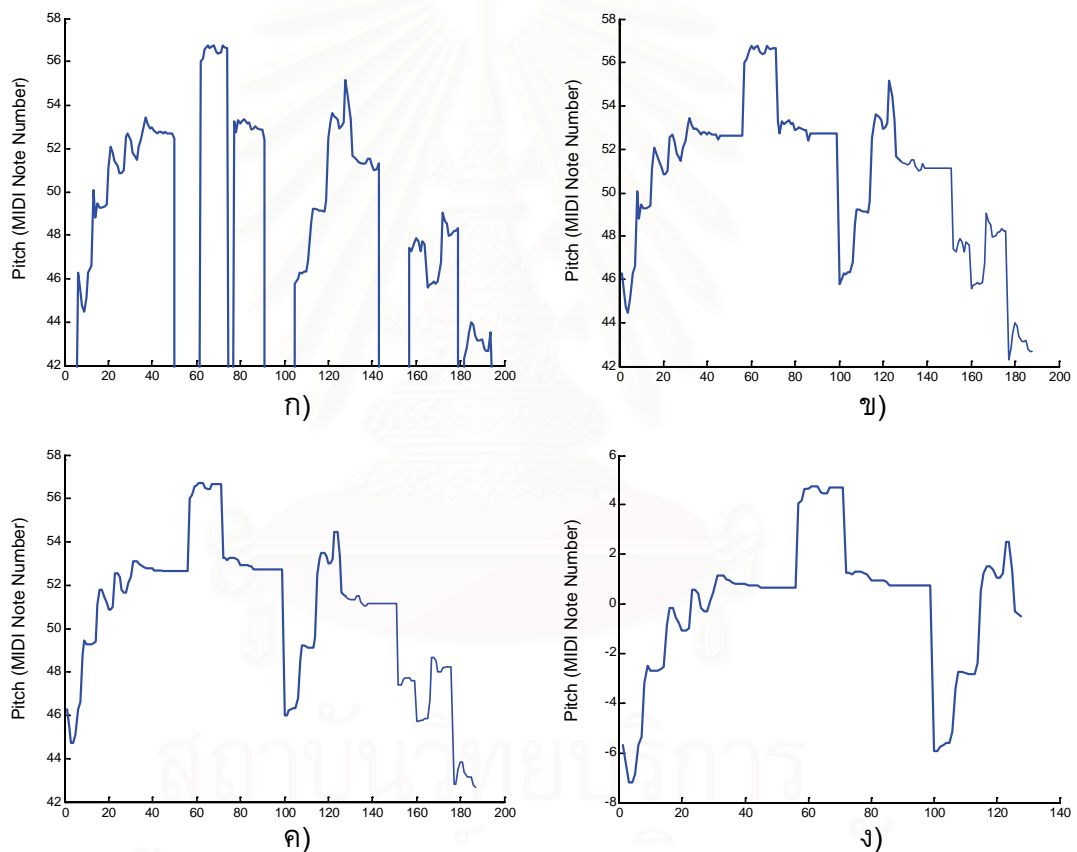
3.1 การสกัดคุณลักษณะจากเสียงร้องทำนอง

ในส่วนของการสกัดคุณลักษณะจากเสียงร้องทำนองจะเป็นส่วนที่รับเสียงร้องทำนองของผู้ใช้เข้ามาและทำการสกัดคุณลักษณะจากเสียงร้องทำนองดังกล่าว เพื่อที่จะนำคุณลักษณะที่ได้ไปใช้ในการเปรียบเทียบความคล้ายคลึงกันกับคุณลักษณะของเพลงที่เก็บไว้ในฐานข้อมูล เพื่อหาเพลงที่มีคุณลักษณะใกล้เคียงกับคุณลักษณะของเสียงร้องทำนองนั้นมากที่สุด สำหรับงานวิจัยนี้ ผู้วิจัยได้เลือกใช้โปรแกรม Praat [27] ในการสกัดคอนทัวร์ระดับเสียงออกจากเสียงร้องทำนอง ซึ่งเป็นคุณลักษณะที่สำคัญของเสียงร้องทำนอง โดยได้เลือกใช้วิธีสหสัมพันธ์ (Autocorrelation) ของโปรแกรมดังกล่าวในการสกัดคอนทัวร์ระดับเสียง และได้กำหนดให้ช่วงการซักรวบรวม (Sampling Interval) เท่ากับ 0.05 วินาที ผลลัพธ์ที่ได้จากการสกัดคุณลักษณะนี้ก็คือ คอนทัวร์ระดับเสียงดังแสดงในรูปที่ 3.1 ก)

อย่างไรก็ตาม โดยทั่วไปเสียงร้องทำนองของผู้ใช้มักจะมีช่วงที่ไม่มีเสียง หรือช่วงเงียบ อันเกิดจากการร้องทำนองหรือการร้องเพลงนั้น จะต้องมีการเว้นวรรคของแต่ละประโยคในเพลงหรือการเว้นวรรคระหว่างคำ ซึ่งจะทำให้คอนทัวร์ระดับเสียงที่ได้ จะมีลักษณะขาดจากกันเป็นช่วง ๆ ดังนั้นงานวิจัยนี้จึงได้เลือกที่จะเติมเต็มช่วงที่ไม่มีเสียงดังกล่าว ด้วยค่าเฉลี่ยจากข้อมูลคอนทัวร์ระดับเสียง 3 ลำดับแรกที่อยู่ก่อนหน้าช่วงที่ไม่มีเสียงนั้น ดังแสดงในรูปที่ 3.1 ข) จากนั้นเพื่อให้ข้อมูลคอนทัวร์ระดับเสียงที่ได้มีคุณภาพสูงขึ้น จึงได้ทำการปรับเรียบ (Smoothing) ข้อมูลคอนทัวร์ระดับเสียงนั้นด้วยตัวกรองค่ามัธยฐาน (Median Filter) ขนาด 3 มิติ ซึ่งจะได้ผลลัพธ์ดังรูปที่ 3.1 ค) การปรับเรียบคอนทัวร์ระดับเสียงนั้น จะช่วยให้การ

เปรียบเทียบความคล้ายคลึงกันระหว่างคุณลักษณะของเสียงร้องทำนองกับคุณลักษณะของเพลงในฐานข้อมูลสามารถทำได้อย่างแม่นยำมากยิ่งขึ้น

สุดท้ายจึงนำข้อมูลคอนทัวร์ระดับเสียงที่ได้ มาตัดแบ่งให้ได้ขนาดตามที่ต้องการ สำหรับงานวิจัยนี้ได้เลือกตัดข้อมูลคอนทัวร์ระดับเสียงที่ความยาว 128 มิติแรก มาเป็นตัวแทนของข้อมูลเสียงร้องทำนองที่ใช้ในการค้นหา นั่นก็คือ ประมาณ 6 วินาทีแรกหลังจากที่ผู้ใช้เริ่มเปล่งเสียง จากนั้นจึงนำคอนทัวร์ระดับเสียงที่ได้มาทำให้เป็นบรรทัดฐานโดยใช้ค่าเฉลี่ยของข้อมูลคอนทัวร์ระดับเสียงนั้น ๆ เพื่อให้ข้อมูลคอนทัวร์ระดับเสียงที่ได้ อยู่ในระดับอ้างอิงเดียวกับคอนทัวร์ระดับเสียงของเพลงในฐานข้อมูล จากนั้นจึงนำคอนทัวร์ระดับเสียงที่ได้นี้ ไปใช้ในการค้นหาเพลงต่อไป ดังแสดงในรูปที่ 3.1 ง)



รูปที่ 3.1 คอนทัวร์ระดับเสียงที่ได้จากขั้นตอนต่าง ๆ ในการสกัดคุณลักษณะ ก) คอนทัวร์ระดับเสียงที่ได้จากขั้นตอนการสกัดคุณลักษณะออกจากเสียงร้องทำนอง ข) คอนทัวร์ระดับเสียงหลังจากผ่านขั้นตอนการเติมเต็มช่วงที่ไม่มีเสียง ค) คอนทัวร์ระดับเสียงที่ได้หลังจากผ่านกระบวนการปรับเรียบ ง) คอนทัวร์ระดับเสียงที่ผ่านขั้นตอนการตัดขนาดและแปลงข้อมูลดังกล่าวให้เป็นบรรทัดฐาน

3.2 การสกัดคุณลักษณะจากเพลงในฐานข้อมูล

ฐานข้อมูลที่ใช้ในงานวิจัยนี้ เป็นฐานข้อมูลเพลงในรูปแบบแฟ้มข้อมูลมิตี ซึ่งสามารถดาวน์โหลดได้โดยทั่วไปจากอินเทอร์เน็ต แฟ้มข้อมูลมิตินั้นมีลักษณะที่เหมาะสมสำหรับการพัฒนาระบบค้นหาเพลงโดยการร้องทำนอง เนื่องจากแฟ้มข้อมูลมิตีมีการจัดเก็บข้อมูลเกี่ยวกับโน้ตดนตรีของเพลงนั้น ๆ เอาไว้ ทำให้สามารถเลือกสกัดข้อมูลเกี่ยวกับทำนองเพลงออกมาเพื่อใช้ในการเปรียบเทียบกับคุณลักษณะที่ได้จากเสียงร้องทำนองของผู้ใช้ได้

เนื่องจากงานวิจัยนี้ได้เลือกที่จะแทนข้อมูลเพลงและเสียงร้องทำนองด้วยข้อมูลอนุกรมเวลา ดังนั้นในขั้นตอนการสกัดคุณลักษณะจากเพลงในฐานข้อมูล จะต้องสกัดข้อมูลคอนทิวรัระดับเสียงของแฟ้มข้อมูลมิตี ซึ่งเป็นข้อมูลเกี่ยวกับทำนองเพลง และแปลงข้อมูลดังกล่าวให้เป็นข้อมูลอนุกรมเวลา เพื่อที่จะจัดเก็บข้อมูลคอนทิวรัระดับเสียงไว้ในฐานข้อมูล อย่างไรก็ตาม โดยทั่วไปการค้นคืนข้อมูลอนุกรมเวลาสามารถแบ่งได้เป็น 2 ประเภทหลัก ๆ ด้วยกันคือ การค้นคืนข้อมูลแบบการเปรียบเทียบข้อมูลทั้งลำดับ (Whole Sequence Matching) และการค้นคืนข้อมูลแบบการเปรียบเทียบข้อมูลลำดับย่อย (Subsequence Matching) ซึ่งการค้นคืนข้อมูลอนุกรมเวลาทั้งสองแบบจะส่งผลให้โครงสร้างในการจัดเก็บข้อมูลมีลักษณะแตกต่างกัน สำหรับงานวิจัยนี้ ได้เลือกใช้วิธีค้นคืนข้อมูลแบบการเปรียบเทียบทั้งลำดับ ซึ่งโดยทั่วไปการเปรียบเทียบข้อมูลแบบทั้งลำดับนั้น จะมีประสิทธิภาพในการค้นคืนข้อมูลสูงกว่าการเปรียบเทียบข้อมูลแบบลำดับย่อย เนื่องจากในฐานข้อมูลอนุกรมเวลาที่ใช้สำหรับการเปรียบเทียบข้อมูลทั้งลำดับนั้น สามารถเลือกที่จะจัดเก็บข้อมูลเฉพาะที่เป็นประโยชน์สำหรับการค้นหาเท่านั้นได้ ทำให้ปริมาณข้อมูลที่ใช้ในการเปรียบเทียบน้อยและมีคุณภาพสูง ซึ่งแตกต่างกับการเปรียบเทียบข้อมูลแบบลำดับย่อย ที่จะต้องจัดเก็บข้อมูลอนุกรมเวลาที่มีขนาดยาวทั้งลำดับ และขณะค้นหาข้อมูลจึงจะแบ่งข้อมูลย่อย ๆ นั้นออกมา เพื่อใช้เป็นข้อมูลสำหรับการเปรียบเทียบ

สำหรับการเลือกข้อมูลที่จะนำมาจัดเก็บในฐานข้อมูลนั้น ในงานวิจัยนี้ได้เลือกใช้วิธีการเลื่อนหน้าต่าง (Sliding Window) เพื่อที่จะตัดส่วนข้อมูลคอนทิวรัระดับเสียงที่ได้จากการสกัดคุณลักษณะจากแฟ้มข้อมูลมิตี และจัดเก็บข้อมูลดังกล่าวลงในฐานข้อมูลเพื่อใช้ในการเปรียบเทียบ โดยที่ความยาวของหน้าต่างจะมีความยาวเท่ากับความยาวของข้อมูลสอบถามคุณด้วยอัตราส่วนการยืดสูงสุด นั่นคือ $n = m * sfmax$ โดยที่ n คือความยาวของข้อมูลที่ถูกจัดเก็บในฐานข้อมูลและ m คือความยาวของข้อมูลสอบถาม $sfmax$ คือ อัตราส่วนการยืดขนาดสูงสุด โดยในการตัดส่วนนั้น จะทำการเลื่อนหน้าต่างครั้งละ 1 ตัวโน้ต นอกจากนี้ยังมีข้อกำหนดต่าง ๆ เพื่อให้การจัดเก็บข้อมูลคอนทิวรัระดับเสียงของเพลงมีประสิทธิภาพมากยิ่งขึ้น คือข้อมูลที่จัดเก็บต้องไม่ซ้ำซ้อนกับท่อนอื่นในเพลงเดียวกัน ต้องไม่มีโน้ตดนตรีตัวใดที่มีความยาว

มากกว่า 5 วินาที ซึ่งอาจจะเป็นช่วงเงียบในเพลง และข้อมูลที่จัดเก็บต้องมีจำนวนตัวโน้ตมากกว่า 3 ตัวขึ้นไป

3.3 การค้นหาเพลงจากเสียงร้องทำนองภายใต้ฐานข้อมูลอนุกรมเวลาขนาดใหญ่

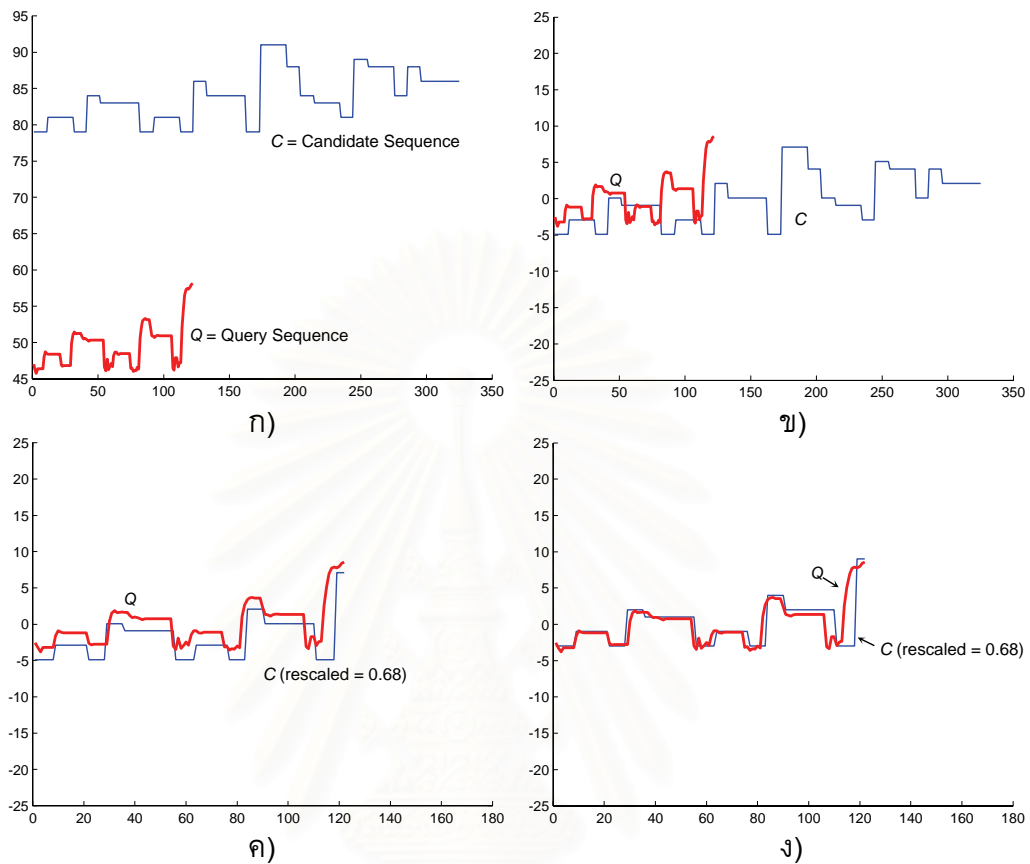
ในงานวิจัยนี้ ผู้วิจัยได้เลือกพัฒนาวิธีการค้นหาเพลงโดยการร้องทำนองที่ใช้วิธีการปรับขนาดเอกรูปและไดนามิกโทม์วอร์ปปีง ที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐานโดยวิธีดังกล่าวเกิดจากการผสมผสานวิธีปรับขนาดเอกรูปและวิธีไดนามิกโทม์วอร์ปปีงเข้าไว้ด้วยกัน พร้อมทั้งมีการแปลงข้อมูลให้เป็นบรรทัดฐานทุกครั้งหลังจากมีการปรับขนาดเอกรูปเพื่อให้ผลลัพธ์ในการค้นหามีความแม่นยำสูง พร้อมทั้งนำเสนอในส่วนของฟังก์ชันขอบเขตล่างของวิธีดังกล่าว และวิธีการค้นหาเพลงโดยใช้ฟังก์ชันขอบเขตล่าง เพื่อให้สามารถค้นหาเพลงในฐานข้อมูลอนุกรมเวลาที่มีขนาดใหญ่ได้อย่างมีประสิทธิภาพและรวดเร็ว ซึ่งจะนำเสนอตามลำดับ

3.3.1 การปรับขนาดเอกรูปและไดนามิกโทม์วอร์ปปีงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน

วิธีไดนามิกโทม์วอร์ปปีงได้มีการศึกษาวิจัยและค้นพบแล้วว่า เป็นวิธีที่เหมาะสมสำหรับระบบค้นหาเพลงโดยการร้องทำนอง [6, 8-10] เนื่องจากวิธีดังกล่าวสามารถรองรับความแปรผันทางเวลาที่เกิดขึ้นในเสียงร้องทำนองได้เป็นอย่างดี เช่น สามารถรองรับเสียงร้องทำนองที่มีจังหวะในการร้องไม่คงที่ หรือเสียงร้องทำนองที่มีจังหวะช้าหรือเร็วกว่าจังหวะเพลงที่จัดเก็บไว้ในฐานข้อมูล แต่ในการศึกษาวิจัยเมื่อเร็ว ๆ นี้ [2] Fu และคนอื่น ๆ ได้เสนอแนวคิดในการผสมผสานวิธีปรับขนาดเอกรูปเข้ากับวิธีไดนามิกโทม์วอร์ปปีง ซึ่งจะช่วยให้สามารถรองรับความแปรผันที่เกิดจากเสียงร้องทำนองที่มีจังหวะช้าหรือเร็วได้อย่างสมบูรณ์มากยิ่งขึ้น กล่าวคือ วิธีนี้สามารถรองรับความแปรผันได้ทั้งการร้องทำนองที่มีจังหวะในการร้องไม่คงที่ และการร้องทำนองที่ช้าหรือเร็วกว่าจังหวะของเพลงที่จัดเก็บอยู่ในฐานข้อมูล

แม้ว่าวิธีปรับขนาดกับโทม์วอร์ปปีงนั้น ดูเหมือนว่าจะสามารถรองรับความแปรผันของผู้ใช้ได้เป็นอย่างดี แต่วิธีที่เสนอในงานวิจัยของ Fu และคนอื่น ๆ นั้น [2] ได้มองข้ามความสำคัญของการแปลงข้อมูลให้เป็นบรรทัดฐานทุกครั้งหลังจากมีการปรับขนาดเอกรูป ซึ่งเป็นขั้นตอนก่อนการวัดระยะทางด้วยวิธีไดนามิกโทม์วอร์ปปีง โดยในการแปลงข้อมูลให้เป็นบรรทัดฐานทุกครั้งก่อนที่จะวัดระยะทางนั้น จะทำให้ทั้งข้อมูลสอบถามและข้อมูลที่จะนำมาใช้ในการเปรียบเทียบ อยู่ในระดับอ้างอิงเดียวกันเสมอ ซึ่งเป็นสาเหตุให้การวัดระยะทางด้วยวิธีปรับขนาดกับโทม์วอร์ปปีงเพียงอย่างเดียว นั้น อาจจะไม่เหมาะสมสำหรับการค้นหาเพลงโดยการร้องทำนอง และไม่สามารถทำงานได้อย่างถูกต้องแม่นยำ ดังนั้นในงานวิจัยนี้จึงได้เสนอวิธีการปรับ

ขนาดเอกรูปและไดนามิกไทมวอร์ปิง ที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน เพื่อที่จะเพิ่มความแม่นยำให้กับการคำนวณระยะทางระหว่างข้อมูลอนุกรมเวลาทั้งสอง



รูปที่ 3.2 การเปรียบเทียบแนวคิดในการแปลงข้อมูลให้เป็นบรรทัดฐานในงานวิจัยของ Fu และคนอื่น ๆ กับงานวิจัยนี้ ก) คอนทิวรัระดับเสียงที่สกัดจากเสียงร้องทำนอง และคอนทิวรัระดับเสียงบางส่วนของเพลงแฮปปี้เบิร์ดเดย์ ข) คอนทิวรัระดับเสียงของข้อมูลทั้งสองที่ถูกปรับระดับของข้อมูลให้อยู่ในระดับอ้างอิงเดียวกัน ด้วยการแปลงข้อมูลให้เป็นบรรทัดฐานด้วยค่าเฉลี่ยของข้อมูลแต่ละตัว ค) คอนทิวรัระดับเสียงของเพลงที่ถูกหดขนาดไปที่ความยาว 0.68 เท่าของความยาวเดิม พร้อมทั้งตัดแบ่งให้ขนาดเท่ากับความยาวของข้อมูลสอบถาม ง) คอนทิวรัระดับเสียงของเพลงที่ถูกหดขนาดไปที่ความยาว 0.68 เท่าของความยาวเดิม พร้อมทั้งปรับระดับของข้อมูลเพลงให้เป็นบรรทัดฐานและตัดแบ่งให้ขนาดข้อมูลอนุกรมเวลาของเพลงเท่ากับความยาวของข้อมูลสอบถาม

สำหรับแนวคิดของงานวิจัยนี้เมื่อเปรียบเทียบกับแนวคิดในงานวิจัยของ Fu และคนอื่น ๆ [2] สามารถแสดงได้ดังรูปที่ 3.2 โดยที่ในรูปที่ 3.2 ก) นั้น แสดงถึงคอนทิวรัระดับเสียงของเพลงและเสียงร้องทำนองที่ถูกสกัดคุณลักษณะออกมาในรูปแบบของอนุกรมเวลา และจากนั้นแปลงข้อมูลทั้งสองให้เป็นบรรทัดฐานโดยใช้ค่าเฉลี่ยของข้อมูลนั้น ๆ ดังแสดงในรูปที่ 3.2 ข) และในรูปที่ 3.2 ค) แสดงให้เห็นถึงแนวคิดของวิธีปรับขนาดกับไทมวอร์ปิงที่เสนอใน

งานวิจัยของ Fu และคนอื่น ๆ [2] ซึ่งจะมีการหัดขนาดของข้อมูลอนุกรมเวลาที่จะใช้ในการเปรียบเทียบที่ 0.68 เท่าของความยาวเดิม และตัดแบ่งความยาวให้เท่ากับความยาวของข้อมูลสอบถาม จากรูปจะเห็นได้ว่าแม้จะมีการแปลงข้อมูลทั้งสองให้เป็นบรรทัดฐานในครั้งแรก แต่ข้อมูลทั้งสองก็ยังไม่อยู่ในระดับอ้างอิงเดียวกัน เนื่องจากข้อมูลในส่วนที่ถูกตัดแบ่งและหัดขนาดมาให้เท่ากับความยาวของข้อมูลสอบถามนั้น ยังไม่ถูกแปลงให้เป็นบรรทัดฐาน ทำให้ค่าระยะทางที่ได้ มีค่ามากเกินไปจนความเป็นจริง สำหรับรูปที่ 3.2 ง) จะเป็นแนวคิดของงานวิจัยนี้ นั่นคือ ทุกครั้งที่มีการปรับขนาดเอกรูปของข้อมูลที่จะใช้ในการเปรียบเทียบ จะมีการแปลงข้อมูลดังกล่าวให้เป็นบรรทัดฐานด้วยค่าเฉลี่ย ซึ่งเห็นได้ชัดว่าการแปลงข้อมูลให้เป็นบรรทัดฐานทุกครั้งหลังจากมีการปรับขนาดเอกรูป จะทำให้ข้อมูลสอบถามและข้อมูลที่จะนำมาใช้ในการเปรียบเทียบ อยู่ในระดับอ้างอิงเดียวกันเสมอ อันจะช่วยใช้การวัดระยะทางระหว่างข้อมูลอนุกรมเวลาทั้งสองมีความถูกต้องแม่นยำมากยิ่งขึ้น

สำหรับนิยามของวิธีปรับขนาดเอกรูปและไดนามิกไทม์วอร์ปึงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน ในส่วนนี้ผู้วิจัยจะขอเปลี่ยนการนำเสนอวิธีการแปลงข้อมูลให้เป็นบรรทัดฐานจากการใช้ค่าเฉลี่ย มาเป็นการใช้คะแนน Z (Z-Score Normalization) แทน แม้ว่าการแปลงข้อมูลให้เป็นบรรทัดฐานโดยใช้ค่าเฉลี่ยนั้น จะเหมาะสมสำหรับการค้นหาเพลงโดยการร้องทำนองมากกว่า อย่างไรก็ตามการนำเสนอวิธีปรับขนาดเอกรูปและไดนามิกไทม์วอร์ปึงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐานโดยใช้คะแนน Z นั้น จะสามารถนำมาประยุกต์ใช้ได้กับวิธีปรับขนาดเอกรูปและไดนามิกไทม์วอร์ปึงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐานโดยใช้ค่าเฉลี่ยได้เช่นกัน รวมทั้งวิธีดังกล่าวที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐานด้วยคะแนน Z ยังอาจจะนำไปประยุกต์ใช้กับปัญหาอื่นได้อีกด้วย โดยวิธีปรับขนาดเอกรูปและไดนามิกไทม์วอร์ปึงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน สามารถนิยามได้ดังนี้

ให้ข้อมูลอนุกรมเวลา $Q = q_1, q_2, q_3, \dots, q_m$ คือ ข้อมูลสอบถาม ความยาว m และ $C = c_1, c_2, c_3, \dots, c_n$ คือ ข้อมูลที่เก็บอยู่ในฐานข้อมูล ความยาว n และ Q' คือ ข้อมูลสอบถามที่ถูกแปลงให้เป็นบรรทัดฐานด้วยคะแนน Z และอัตราส่วนการหัดและยืดขนาดมากที่สุดคือ $sfmin$ และ $sfmax$ ตามลำดับ โดยที่ $n \geq sfmax * m$ นั่นคือ ความยาวของข้อมูลที่จะใช้ในการเปรียบเทียบจะต้องยาวไม่น้อยกว่าความยาวของข้อมูลสอบถามที่ถูกยืดขนาดมากที่สุด r คือ ความกว้างของเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะ โดยฟังก์ชันปรับขนาดเอกรูปและไดนามิกไทม์วอร์ปึงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐานนั้น สามารถแสดงเป็นสมการได้ดังนี้

$$SWM_{norm}(Q', C, sfmin, sfmax, r) = \min_{l=\lfloor sfmin * m \rfloor}^{\min(\lfloor sfmax * m \rfloor, n)} DTW_{constraint}(RP(C, m, l), Q', r) \quad (3.1)$$

$$RP_{norm}(C, m, l)_i = \frac{c_{\lfloor i * l / m \rfloor} - c_{1..l}}{SD(c_{1..l})} \quad \text{where } 1 \leq i \leq m \quad (3.2)$$

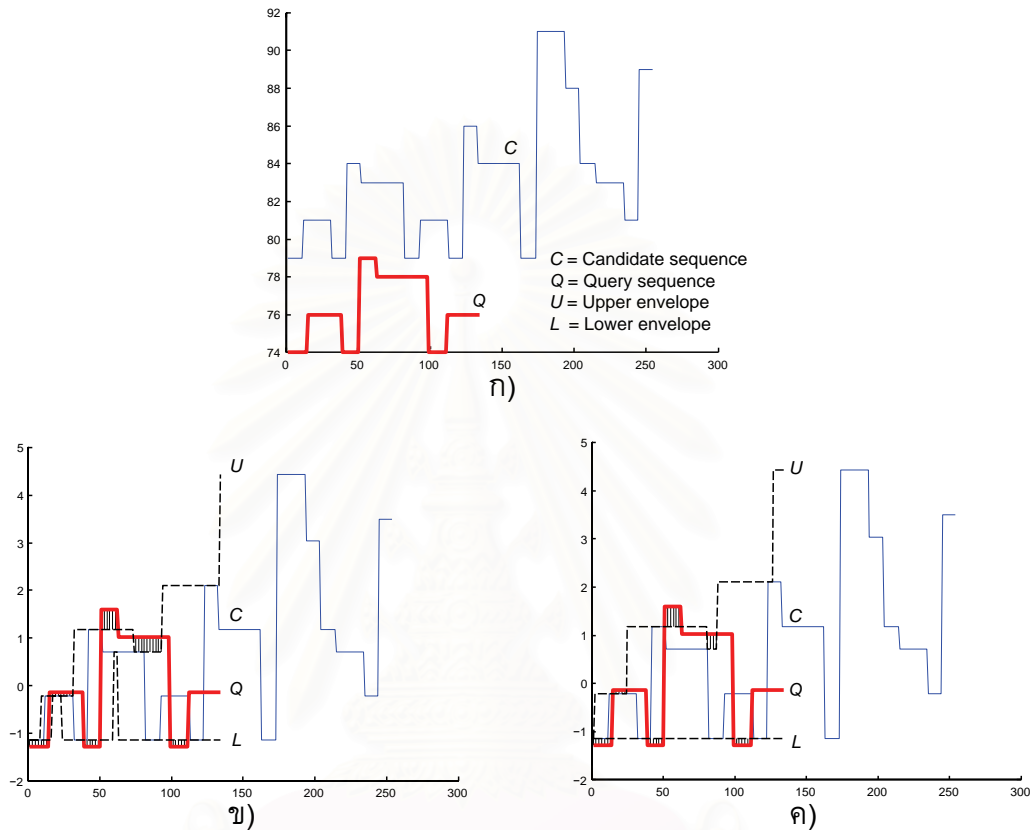
โดยที่ฟังก์ชัน $SWM_{norm}(Q', C, sfmin, sfmax, r)$ คือ ฟังก์ชันที่ใช้สำหรับการคำนวณค่าระยะทางด้วยวิธีปรับขนาดเอกรูปและไดนามิกไทม์วอร์ปิงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน และสำหรับฟังก์ชัน $RP_{norm}(C, m, l)$ เป็นฟังก์ชันที่ใช้สำหรับการหดหรือยืดขนาดของข้อมูลส่วนต้นของอนุกรมเวลา C ความยาว l ไปที่ความยาว m นั่นคือ ความยาวของข้อมูลสอบถาม พร้อมทั้งทำข้อมูลดังกล่าวให้เป็นบรรทัดฐานด้วยคะแนน Z ก่อนที่จะคำนวณระยะทางด้วยวิธี ไดนามิกไทม์วอร์ปิง $\overline{c_{1..l}}$ และ $SD(c_{1..l})$ คือ ค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของข้อมูลส่วนต้นของข้อมูลที่จะนำมาใช้ในการเปรียบเทียบที่ขนาด l

สำหรับการแปลงฟังก์ชันที่นำเสนอขึ้นให้รองรับกับการแปลงข้อมูลให้เป็นบรรทัดฐานด้วยค่าเฉลี่ย สามารถทำได้โดยการละในส่วนของการหารค่าเบี่ยงเบนมาตรฐานในสมการที่ (3.2) อย่างไรก็ตาม เพื่อให้การค้นหาข้อมูลสามารถทำได้มีประสิทธิภาพ ฟังก์ชันขอบเขตล่างของวิธีที่นำเสนอขึ้น ถือเป็นสิ่งจำเป็นเพื่อที่จะหลีกเลี่ยงการคำนวณค่าระยะทางจริงด้วยวิธีปรับขนาดเอกรูปและไดนามิกไทม์วอร์ปิงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐานกับข้อมูลในฐานะข้อมูลทุกตัว

3.3.2 ฟังก์ชันขอบเขตล่างของการปรับขนาดเอกรูปและไดนามิกไทม์วอร์ปิงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน

สำหรับวิธีปรับขนาดเอกรูปและไดนามิกไทม์วอร์ปิงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐานนั้น ฟังก์ชันขอบเขตล่างที่นำเสนอในงานวิจัยของ Fu และคนอื่น ๆ [2] ไม่สามารถทำงานได้อย่างถูกต้องในกรณีดังกล่าว ซึ่งสามารถแสดงได้ดังรูปที่ 3.3 นั่นคือ กรณีที่ตัดข้อมูลในส่วนต้นของข้อมูลอนุกรมเวลาและทำการยืดขนาดของข้อมูลนั้นไปที่ความยาว 1.2 เท่าของความยาวเดิม และกำหนดให้ข้อมูลอนุกรมเวลาในส่วนต้นนั้นเป็นข้อมูลสอบถาม ดังแสดงในรูปที่ 3.3 ก) จากนั้นจึงนำข้อมูลดังกล่าวและข้อมูลอนุกรมเวลาเดิม มาทำให้เป็นบรรทัดฐานโดยใช้คะแนน Z ที่ความยาวของข้อมูลสอบถาม (การที่นำข้อมูลอนุกรมเวลาเดิมมาทำให้เป็นบรรทัดฐานที่ความยาวของข้อมูลสอบถามนั้น ก็เพื่อให้ข้อมูลอนุกรมเวลาเดิมอยู่ในระดับอ้างอิงที่ใกล้เคียงกับข้อมูลสอบถามมากที่สุด) และเมื่อนำข้อมูลดังกล่าวมาคิดค่าฟังก์ชันขอบเขตล่างของวิธีในงานวิจัยเดิมที่เสนอวิธีปรับขนาดกับไทม์วอร์ปิง โดยกำหนดให้สามารถรองรับการหดและยืดขนาดของข้อมูลที่มีความยาว 0.7 ถึง 1.3 เท่าของความยาวข้อมูลสอบถาม ดังรูปที่ 3.3 ข) และ ค) โดยในรูปที่ 3.3 ข) นั้นได้กำหนดให้ ความกว้างของเงื่อนไขบังคับโดยรวมแบบซิกมา-ซิบะเท่ากับ 0% และในรูปที่ 3.3 ค) เท่ากับ 5% จากรูปจะเห็นได้ชัดว่า ค่าระยะทางที่ได้จากฟังก์ชันขอบเขตล่างดังกล่าว มีค่าไม่เท่ากับศูนย์ ทั้ง ๆ ที่ ถ้าหากนำข้อมูลสอบถามนี้ไปคิดค่าระยะทางกับฟังก์ชันปรับขนาดเอกรูปและไทม์วอร์ปิงโดยเพิ่มในส่วนของการแปลงข้อมูลให้เป็นบรรทัดฐานทุกครั้ง ก่อนการคำนวณระยะทางด้วยวิธีไดนามิกไทม์วอร์ปิง จะได้ค่าระยะทางเท่ากับศูนย์ ซึ่งหมายความว่าฟังก์ชันขอบเขตล่างดังกล่าวไม่มีคุณสมบัติ

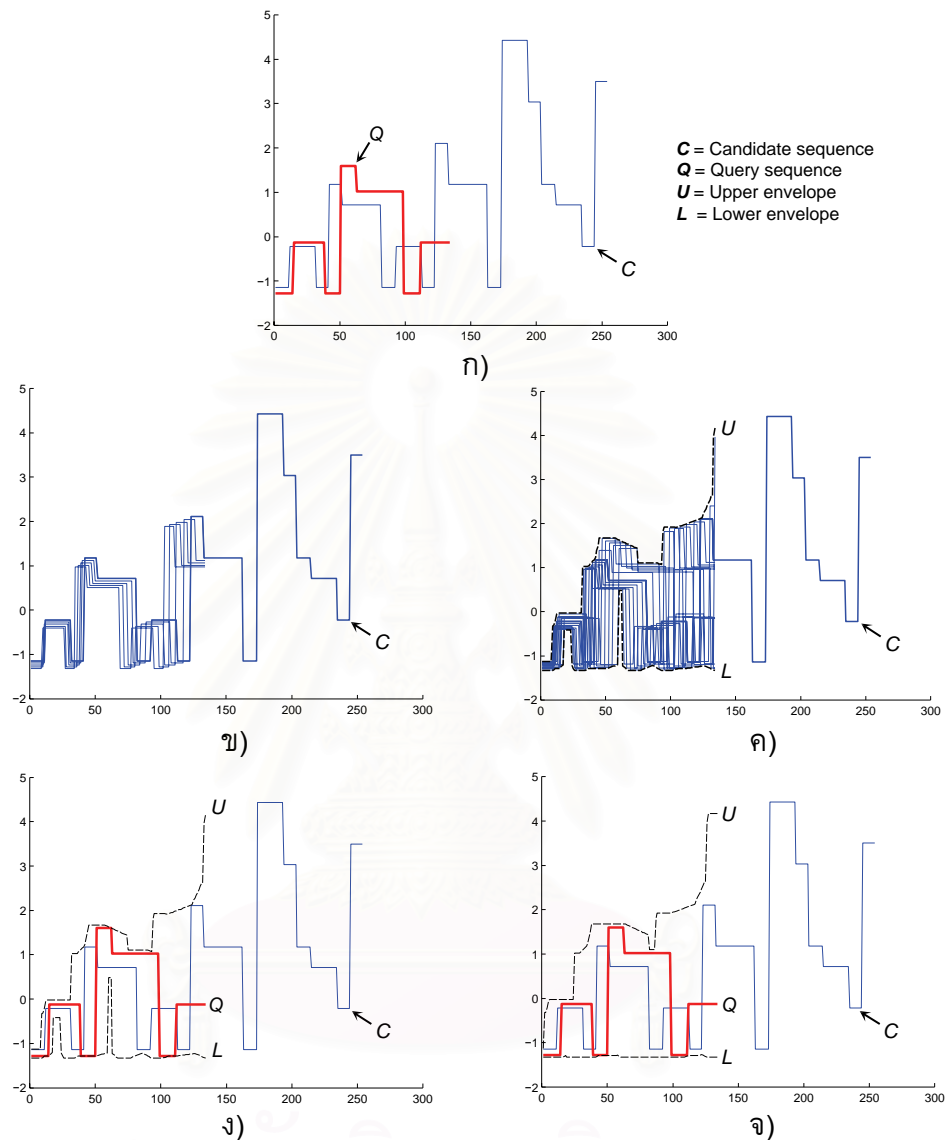
ของฟังก์ชันขอบเขตล่าง สำหรับกรณีที่ต้องมีการแปลงข้อมูลให้เป็นบรรทัดฐานทุกครั้งหลังจากมีการปรับขนาดเอกรูป เนื่องจากค่าระยะที่ได้จากฟังก์ชันขอบเขตล่างดังกล่าวมีค่ามากกว่าค่าระยะทางที่ได้จากการคำนวณจากฟังก์ชันปรับขนาดกับโทมวอร์ปิงที่มีการแปลงข้อมูลให้เป็นบรรทัดฐาน



รูปที่ 3.3 การคำนวณค่าระยะทางขอบเขตล่างในงานวิจัยของ Fu และคนอื่น ๆ สำหรับกรณีที่มีการแปลงข้อมูลให้เป็นบรรทัดฐานทุกครั้งก่อนการคำนวณค่าระยะทาง ก) คอนทัวร์ระดับเสียงของเพลงแฮปปีเบิร์ดเดย์ที่มีจังหวะไม่เท่ากัน โดยคอนทัวร์ระดับเสียงที่เป็นข้อมูลสอบถาม Q จะมีจังหวะที่ช้ากว่าจังหวะของคอนทัวร์ระดับเสียง C อยู่ 1.2 เท่า ข) และ ค) แสดงถึงคอนทัวร์ระดับเสียงของข้อมูลสอบถาม Q และข้อมูลเพลง C ที่ถูกทำให้เป็นบรรทัดฐานด้วยคะแนน Z ที่ความยาวของข้อมูลสอบถาม พร้อมกับฟังก์ชันขอบเขตล่างของวิธีปรับขนาดกับโทมวอร์ปิงที่นำเสนอในงานวิจัยของ Fu และคนอื่น ๆ [2] โดยที่กำหนดให้ $sfmin$ และ $sfmax$ เท่ากับ 0.7 และ 1.3 ตามลำดับ โดยที่ในรูป ข) มีค่า $r = 0\%$ และในรูป ค) มีค่า $r = 5\%$

สำหรับงานวิจัยนี้ จะนำเสนอฟังก์ชันขอบเขตล่างสำหรับวิธีการปรับขนาดเอกรูปและไดนามิกโทมวอร์ปิงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน โดยแนวคิดในการพัฒนาคือ จะมีการสร้างเส้นขอบเขตบนและเส้นขอบเขตล่างที่สามารถครอบคลุมทุกกรณีที่เป็นไปได้ในการหดและยืดขนาดของข้อมูลที่จะใช้ในการเปรียบเทียบ และมีการแปลงข้อมูลดังกล่าวให้เป็นบรรทัดฐานในทุกครั้งที่มีการหดหรือยืดขนาด จากนั้นเมื่อได้เส้นขอบเขตบนและ

เส้นขอบเขตล่างเป็นที่เรียบร้อยแล้ว จึงจะนำข้อมูลสอบถามมาคิดค่าระยะทางขอบเขตล่างกับเส้นขอบเขตบนและเส้นขอบเขตล่างดังกล่าว



รูปที่ 3.4 ฟังก์ชันขอบเขตล่างของวิธีที่นำเสนอ ก) คอนทิวรัระดับเสียงของข้อมูลสอบถาม Q และข้อมูลเพลง C ที่ถูกทำให้เป็นบรรทัดฐานด้วยคะแนน Z ที่ความยาวของข้อมูลสอบถาม Q ข) คอนทิวรัระดับเสียงของข้อมูลเพลงบางส่วน ที่มีการหดและยืดขนาดพร้อมทั้งทำข้อมูลให้เป็นบรรทัดฐาน ค) แสดงเส้นขอบเขตบนและเส้นขอบเขตล่างที่ครอบคลุมทุกกรณีที่เป็นไปได้ในการหดและยืดขนาดของข้อมูลเพลงในฐานข้อมูล ที่ขนาดตั้งแต่ 0.7 เท่า ถึง 1.3 เท่า พร้อมทั้งทำข้อมูลให้เป็นบรรทัดฐาน โดยที่ค่า $r = 0\%$ ง) และ จ) แสดงถึงฟังก์ชันขอบเขตล่างของวิธีที่นำเสนอที่สามารถหดและยืดขนาดได้ตั้งแต่ 0.7 เท่า ถึง 1.3 เท่า โดยที่ ง) มีค่า $r = 0\%$ และ จ) มีค่า $r = 5\%$

โดยแนวคิดของฟังก์ชันขอบเขตล่างสำหรับวิธีที่นำเสนอ นั้น สามารถแสดงได้ ดังรูปที่ 3.4 ซึ่งในรูปที่ 3.4 ก) แสดงถึงคอนทัวร์ระดับเสียงของข้อมูลสอบถาม Q และคอนทัวร์ระดับเสียงของข้อมูลเพลง C ที่ถูกทำให้เป็นบรรทัดฐานด้วยคะแนน Z และในรูปที่ 3.4 ข) แสดงให้เห็นว่า ถ้าพยายามที่จะยืดขนาดของข้อมูลสอบถาม Q ออกไป เพื่อใช้สำหรับวัตถุประสงค์ทาง จะ มีผลเช่นเดียวกับการตัดแบ่งข้อมูลในส่วนต้นของอนุกรมเวลา C ที่มีความยาวเท่ากับข้อมูล สอบถามที่ถูกยืดออกไป หดกลับมาที่ความยาวของข้อมูลสอบถามเดิม จากนั้นจึงแปลงข้อมูล เพลงดังกล่าวให้เป็นบรรทัดฐาน แล้วจึงนำมาใช้ในการวัดระยะทางต่อไป อย่างไรก็ตาม มี ข้อสังเกตว่า การที่นำข้อมูลส่วนต้นของอนุกรมเวลา C ที่ขนาดต่าง ๆ มาแปลงข้อมูลให้เป็น บรรทัดฐาน ข้อมูลอนุกรมเวลาที่ได้จะมีการเลื่อนในแนวแกน y ดังแสดงในรูปที่ 3.4 ข) ซึ่งเป็น สาเหตุให้ฟังก์ชันขอบเขตล่างของวิธีที่เสนอในงานวิจัยของ Fu และคนอื่น ๆ [2] ไม่สามารถ ทำงานได้อย่างถูกต้องในการวัดระยะทางสำหรับกรณีนี้ และในรูปที่ 3.4 ค) แสดงให้เห็นถึงการ สร้างเส้นขอบเขตบนและเส้นขอบเขตล่างที่สามารถครอบคลุมทุกกรณีที่เป็นไปได้ในการหดและ ยืดขนาดของคอนทัวร์ระดับเสียง Q ที่ 0.7 เท่า ถึง 1.3 เท่า โดยที่ค่าความกว้างของเงื่อนไข บังคับโดยรวมแบบซิกมา-ซิมบะเป็น 0% และในรูปที่ 3.4 ง) แสดงถึงการนำเส้นขอบเขตบนและ เส้นขอบเขตล่างที่ได้ไปคำนวณค่าฟังก์ชันขอบเขตล่างกับคอนทัวร์ระดับเสียง Q ซึ่งเป็นข้อมูล สอบถาม และสำหรับรูปที่ 3.4 จ) แสดงถึงการคำนวณฟังก์ชันขอบเขตล่างที่นำเสนอ สำหรับ กรณีที่ค่าความกว้างของเงื่อนไขบังคับโดยรวมแบบซิกมา-ซิมบะเท่ากับ 5% โดยเส้นขอบเขตบน และเส้นขอบเขตล่างในกรณีที่ค่า r เท่ากับ 5% นั้น สามารถหาได้โดยประยุกต์ใช้ฟังก์ชัน ขอบเขตล่างของวิธีไดนามิกโทมวอร์ปปีงเข้ากับเส้นขอบเขตบนและเส้นขอบที่นำเสนอในรูปที่ 3.4 ง) นอกจากนี้ ในรูปที่ 3.4 ง) และรูปที่ 3.4 จ) จะเห็นได้ชัดว่าค่าระยะทางที่ได้จากฟังก์ชัน ขอบเขตล่างที่นำเสนอ นั้น มีค่าเท่ากับศูนย์ ซึ่งถูกต้องตามข้อกำหนดของฟังก์ชันขอบเขตล่าง

สำหรับฟังก์ชันขอบเขตล่างของวิธีปรับขนาดเอกรูปและไดนามิกโทมวอร์ปปีง ที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน สามารถนิยามได้ดังสมการที่ (3.3) โดยให้ $LB_{\text{norm}}(Q', C)$ คือ ฟังก์ชันขอบเขตล่างของการปรับขนาดเอกรูปและไดนามิกโทมวอร์ปปีงที่ รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน ระหว่างข้อมูลอนุกรมเวลา Q' และ C โดยที่ Q' คือ ข้อมูลสอบถามที่ผ่านการแปลงข้อมูลให้เป็นบรรทัดฐานด้วยคะแนน Z และ C คือข้อมูลที่ใช้ใน การเปรียบเทียบ และให้ U' และ L' คือเส้นขอบเขตบนและเส้นขอบเขตล่างตามลำดับ ซึ่ง สามารถแสดงได้ดังสมการที่ (3.4) และ (3.5)

$$LB_{\text{norm}}(Q', C) = \sum_{i=1}^m \begin{cases} (q_i - U'_i)^2 & \text{if } q_i > U'_i \\ (q_i - L'_i)^2 & \text{if } q_i < L'_i \\ 0 & \text{otherwise} \end{cases} \quad (3.3)$$

$$\begin{aligned}
U'_i &= \max(UZ'_{\max(1,i-r)}, \dots, UZ'_{\min(i+r,m)}) \\
UZ'_j &= \max\left(\frac{C_{sfmin*j} - C_{1...sfmin*m}}{SD(c_{1...sfmin*m})}, \dots, \frac{C_{sfmax*j} - C_{1...sfmax*m}}{SD(c_{1...sfmax*m})}\right) \text{ where } 1 \leq j \leq m \\
&= \max_{k=0}^{\lceil m*(sfmax-sfmin) \rceil} \left(\frac{C_{\lfloor sfmin*j+k*\frac{j}{m} \rfloor} - C_{1...[\lfloor sfmin*m \rfloor+k]}}{SD(c_{1...[\lfloor sfmin*m \rfloor+k]})}, \frac{C_{\lfloor sfmin*j+k*\frac{j}{m} \rfloor} - C_{1...[\lfloor sfmin*m \rfloor+k]}}{SD(c_{1...[\lfloor sfmin*m \rfloor+k]})} \right), \\
&\quad \left(\frac{C_{\lfloor sfmin*j+k*\frac{j}{m} \rfloor} - C_{1...[\lfloor sfmin*m \rfloor+k]}}{SD(c_{1...[\lfloor sfmin*m \rfloor+k]})}, \frac{C_{\lfloor sfmin*j+k*\frac{j}{m} \rfloor} - C_{1...[\lfloor sfmin*m \rfloor+k]}}{SD(c_{1...[\lfloor sfmin*m \rfloor+k]})} \right)
\end{aligned} \tag{3.4}$$

$$\begin{aligned}
L'_i &= \min(LZ'_{\max(1,i-r)}, \dots, LZ'_{\min(i+r,m)}) \\
LZ'_j &= \min\left(\frac{C_{sfmin*j} - C_{1...sfmin*m}}{SD(c_{1...sfmin*m})}, \dots, \frac{C_{sfmax*j} - C_{1...sfmax*m}}{SD(c_{1...sfmax*m})}\right) \text{ where } 1 \leq j \leq m \\
&= \min_{k=0}^{\lceil m*(sfmax-sfmin) \rceil} \left(\frac{C_{\lfloor sfmin*j+k*\frac{j}{m} \rfloor} - C_{1...[\lfloor sfmin*m \rfloor+k]}}{SD(c_{1...[\lfloor sfmin*m \rfloor+k]})}, \frac{C_{\lfloor sfmin*j+k*\frac{j}{m} \rfloor} - C_{1...[\lfloor sfmin*m \rfloor+k]}}{SD(c_{1...[\lfloor sfmin*m \rfloor+k]})} \right), \\
&\quad \left(\frac{C_{\lfloor sfmin*j+k*\frac{j}{m} \rfloor} - C_{1...[\lfloor sfmin*m \rfloor+k]}}{SD(c_{1...[\lfloor sfmin*m \rfloor+k]})}, \frac{C_{\lfloor sfmin*j+k*\frac{j}{m} \rfloor} - C_{1...[\lfloor sfmin*m \rfloor+k]}}{SD(c_{1...[\lfloor sfmin*m \rfloor+k]})} \right)
\end{aligned} \tag{3.5}$$

โดยที่ UZ' และ LZ' คือเส้นขอบเขตบนและเส้นขอบเขตล่างสำหรับกรณีที่มีความกว้างเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะมีค่าเท่ากับ 0 และ $c_{1...sfmin*m}$ คือค่าเฉลี่ยของข้อมูลที่จะใช้ในการเปรียบเทียบตั้งแต่ตัวแรกถึงตัวที่ $sfmin*m$ และ $SD(c_{1...sfmin*m})$ คือค่าเบี่ยงเบนมาตรฐานของข้อมูลตั้งแต่ตัวแรกถึงตัวที่ $sfmin*m$ นั่นคือ $\frac{C_{sfmin*i} - C_{1...sfmin*m}}{SD(c_{1...sfmin*m})}$ คือค่าของข้อมูลที่จะนำมาใช้ในการเปรียบเทียบตัวที่ $sfmin*i$ ที่ผ่านการแปลงให้เป็นบรรทัดฐานด้วยคะแนน Z สำหรับในกรณีที่ต้องการนำฟังก์ชันขอบเขตล่างที่นำเสนอขึ้นไปใช้กับกรณีที่มีการแปลงข้อมูลให้เป็นบรรทัดฐานโดยใช้ค่าเฉลี่ย จะสามารถทำได้โดยการละในส่วนของการหารค่าเบี่ยงเบนมาตรฐานในสมการที่ (3.4) และ (3.5)

เพื่อยืนยันความถูกต้องของฟังก์ชันขอบเขตล่างที่ของการปรับขนาดเอกรูปและไดนามิกโทมัวร์ปิงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน จากนี้จะเป็นส่วนของบทพิสูจน์ความเป็นฟังก์ชันขอบเขตล่างของวิธีที่นำเสนอ ซึ่งจะเริ่มต้นนำเสนอจากกรณีที่มีความกว้างของเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะมีค่าเท่ากับ 0 จากนั้นจึงจะเริ่มพิสูจน์ในส่วนที่ความกว้างของเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะมีขนาดใด ๆ

ทฤษฎีบทที่ 1 ความเป็นฟังก์ชันขอบเขตล่างของฟังก์ชัน $LB_{norm}(Q', C)$ ในกรณีที่ความกว้างของเงื่อนไขบังคับโดยรวมแบบซิก-ซิมมีค่าเท่ากับ 0% สำหรับข้อมูลอนุกรมเวลา Q' และ C ใด ๆ ที่มีขนาด m และ n ตามลำดับ และ $n \geq sfmax * m$ โดยที่ Q' คือข้อมูลสอบถามที่ผ่านการแปลงข้อมูลให้เป็นบรรทัดฐานด้วยคะแนน Z และ C คือข้อมูลที่ใช้ในการเปรียบเทียบ รวมทั้งให้ข้อมูลสอบถาม Q' สามารถยืดหรือหดขนาดได้ในช่วง $sfmin * m$ ถึง $sfmax * m$ โดยที่ $0 < sfmin \leq 1$ และ $sfmax \geq 1$

พิสูจน์

การพิสูจน์ทฤษฎีบทที่ 1 จะเป็นการพิสูจน์เพื่อแสดงให้เห็นว่าค่าระยะทางที่ได้จากฟังก์ชันขอบเขตล่างที่นำเสนอจะมีค่าน้อยกว่าหรือเท่ากับค่าระยะทางที่ได้จากฟังก์ชันการปรับขนาดเอกรูปและไดนามิกไทม์วอร์ปิงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐานเสมอ นั่นคือ การพิสูจน์ว่า $LB_{norm}(Q', C) \leq SWM_{norm}(Q', C, sfmin, sfmax, 0)$ เป็นจริง

ถ้าข้อมูลสอบถามถูกหดหรือยืดขนาดไปที่ความยาว $sf * m$ โดยที่ sf คืออัตราส่วนการหดหรือการยืดขนาด ดังนั้นข้อมูลที่จะนำมาใช้ในการเปรียบเทียบในส่วนต้น จะถูกตัดแบ่งขนาดที่ความยาว $sf * m$ พร้อมทั้งทำการแปลงข้อมูลดังกล่าวให้เป็นบรรทัดฐานด้วยคะแนน Z ดังสมการที่ (3.6)

$$c'_j = \frac{c_j - c_{1...sf*m}}{SD(c_{1...sf*m})} \quad \text{where } 1 \leq j \leq m \quad (3.6)$$

สำหรับกรณีที่ความกว้างของเงื่อนไขบังคับโดยรวมแบบซิก-ซิมมีค่าเท่ากับ 0 เส้นขอบเขตบนและเส้นขอบเขตล่างของวิธีที่นำเสนอ ณ ตัวที่ i ในสมการที่ (3.4) และ (3.5) จะมีค่าเท่ากับ UZ'_i และ LZ'_i ตามลำดับ ซึ่งจะได้ว่า

$$UZ'_i \geq \frac{c_{sfmin*i} - c_{1...sfmin*m}}{SD(c_{1...sfmin*m})}, \dots, \frac{c_{sfmax*i} - c_{1...sfmax*m}}{SD(c_{1...sfmax*m})} \geq LZ'_i$$

นั่นคือ

$$UZ'_i \geq \frac{c_{sf*i} - c_{1...sf*m}}{SD(c_{1...sf*m})} \geq LZ'_i \quad \text{where } sfmin \leq sf \leq sfmax$$

ให้ $sf * i = k$ จะได้

$$UZ'_i \geq c'_k \geq LZ'_i \quad \text{where } sfmin * i \leq k \leq sfmax * i$$

ดังนั้น จะได้ว่า

$$\begin{aligned} (q_i - UZ'_i)^2 &< (q_i - c'_k)^2 && \text{if } q_i > UZ'_i \\ (q_i - LZ'_i)^2 &< (q_i - c'_k)^2 && \text{if } q_i < LZ'_i \\ 0 &&& \text{otherwise} \end{aligned}$$

พิจารณา $(q_i - c'_k)^2$ โดยที่ $sfmin * i \leq k \leq sfmax * i$ จะได้ว่า $(q_i - c'_k)^2$ ก็คือ การนำข้อมูล สอบถามตัวที่ i มาคิดค่าระยะทางกับข้อมูลที่จะใช้ในการเปรียบเทียบตัวที่ $sfmin * i$ ถึงตัวที่ $sfmax * i$ ซึ่งถูกแปลงให้เป็นบรรทัดฐานแล้วทั้งหมด ซึ่งก็คือ การคำนวณค่าระยะทางสำหรับ กรณีที่ข้อมูลสอบถามมีการหดและยืดขนาดตั้งแต่ $sfmin$ ถึง $sfmax$ นั้นเอง ดังนั้นจะได้ว่า

$$LB_{\text{norm}}(Q', C) \leq SWM_{\text{norm}}(Q', C, sfmin, sfmax, 0) \quad \square$$

จากการพิสูจน์ทฤษฎีบทที่ 1 จะได้ว่า ค่าระยะทางที่ได้จากฟังก์ชันขอบเขตล่าง ที่นำเสนอในกรณีที่มีความกว้างเงื่อนไขบังคับแบบซาโก-ชิบะมีค่าเป็น 0 จะมีค่าน้อยกว่าหรือ เท่ากับค่าระยะทางที่ได้จากฟังก์ชันการปรับขนาดเอกรูปและไดนามิกไทม์วอร์ปิงที่รองรับการ แปลงข้อมูลให้เป็นบรรทัดฐานเสมอ

ทฤษฎีบทที่ 2 ความเป็นฟังก์ชันขอบเขตล่างของฟังก์ชัน $LB_{\text{norm}}(Q', C)$ ในกรณีที่มีความ กว้างของเงื่อนไขบังคับแบบซาโก-ชิบะมีค่าใด ๆ

พิสูจน์

การพิสูจน์ทฤษฎีบทที่ 2 จะเป็นการนำฟังก์ชันขอบเขตล่างของไดนามิกไทม์- วอร์ปิงมาประยุกต์ใช้กับฟังก์ชันขอบเขตล่างที่นำเสนอไปสำหรับกรณีที่ความกว้างของเงื่อนไข บังคับโดยรวมแบบซาโก-ชิบะมีค่าเท่ากับ 0 แล้วค่าระยะทางที่ได้จากฟังก์ชันขอบเขตล่างที่ นำเสนอจะมีค่าน้อยกว่าหรือเท่ากับค่าระยะทางที่ได้จากฟังก์ชันการปรับขนาดเอกรูปและ ไดนามิกไทม์วอร์ปิงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐานเสมอ

จากฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปิงในสมการที่ (2.5) จะเห็น ว่ามีการสร้างเส้นขอบเขตบนและเส้นขอบเขตล่างจากข้อมูลสอบถาม Q ที่มีขนาด m เพื่อที่จะ นำมาคิดค่าระยะทางกับข้อมูลที่จะนำมาใช้ในการเปรียบเทียบ C อย่างไรก็ตาม ในกรณีที่มีการ สร้างเส้นขอบเขตบนและเส้นขอบเขตล่างจากข้อมูลที่จะนำมาใช้ในการเปรียบเทียบ C ที่มีขนาด เท่ากันคือ m แล้วนำมาคิดค่าระยะทางกับข้อมูลสอบถาม Q ก็จะได้ค่าระยะทางขอบเขตล่าง ของวิธีไดนามิกไทม์วอร์ปิงเช่นกัน ดังสมการต่อไปนี้

$$LBX(Q, C) = \sum_{i=1}^m \begin{cases} (q_i - UX_i)^2 & \text{if } q_i > UX_i \\ (q_i - LX_i)^2 & \text{if } q_i < LX_i \\ 0 & \text{otherwise} \end{cases} \quad (3.7)$$

$$UX_i = \max(c_{\min(1, i-r)}, \dots, c_{\max(i+r, n)}) \quad (3.8)$$

$$LX_i = \min(c_{\min(1, i-r)}, \dots, c_{\max(i+r, n)})$$

สำหรับกรณีของการปรับขนาดเอกรูปและไดนามิกไทม์วอร์ปิงที่นำเสนอ ข้อมูลที่นำมาใช้ในการเปรียบเทียบทุกตัวที่อัตราส่วนการหดหรือยืดขนาดต่าง ๆ จะถูกปรับขนาดเอกรูปมาที่ความยาว m ซึ่งก็คือ ความยาวของข้อมูลสอบถาม ดังนั้นค่าความกว้างของเงื่อนไขบังคับโดยแบบซาโก-ชิบะของข้อมูลที่จะนำมาใช้ในการเปรียบเทียบทุกตัวจะมีค่าเท่ากัน คือ r ดังนั้นถ้าให้เส้นขอบเขตบนและเส้นขอบเขตล่างของวิธีที่นำเสนอเป็นดังสมการที่ (3.9) โดยที่ UZ'_i และ LZ'_i เป็นเส้นขอบเขตบนและเส้นขอบเขตล่างของวิธีที่นำเสนอ ในกรณีที่ $r = 0$

$$U'_i = \max(UZ'_{\max(1, i-r)}, \dots, UZ'_{\min(i+r, m)}) \quad (3.9)$$

$$L'_i = \min(LZ'_{\max(1, i-r)}, \dots, LZ'_{\min(i+r, m)})$$

ก็จะได้ว่า

$$LB_{\text{norm}}(Q', C) \leq SWM_{\text{norm}}(Q', C, sfmin, sfmax, r) \quad \square$$

จากการพิสูจน์ทฤษฎีบทที่ 2 จะได้ว่า ค่าระยะทางที่ได้จากการคำนวณ ระยะทางด้วยฟังก์ชันขอบเขตล่างที่นำเสนอในกรณีที่ความกว้างเงื่อนไขบังคับแบบซาโก-ชิบะมีขนาดใด ๆ จะมีค่าระยะทางน้อยกว่าหรือเท่ากับค่าระยะทางที่ได้จากฟังก์ชันการปรับขนาดเอกรูปและไดนามิกไทม์วอร์ปิงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐานเสมอ

ในปัจจุบันได้มีวิธีการค้นหาข้อมูลอนุกรมเวลาที่มีประสิทธิภาพต่าง ๆ ที่นำฟังก์ชันขอบเขตล่างมาประยุกต์ใช้ ซึ่งได้แก่ วิธีการค้นหาข้อมูลแบบดัชนี [10, 16, 19, 20] ซึ่งจะช่วยให้อาจค้นหาข้อมูลอนุกรมเวลาจากฐานข้อมูลที่มีขนาดใหญ่ได้อย่างมีประสิทธิภาพ และวิธีการค้นหาข้อมูลแบบตามลำดับที่นำเสนอในงานวิจัยของ Sakurai และคนอื่น ๆ [3] ที่อ้างว่าสามารถค้นหาข้อมูลได้เร็วกว่าวิธีการค้นหาข้อมูลแบบดัชนีถึง 222 เท่า สำหรับในงานวิจัยนี้ จะมีการนำวิธีที่เสนอในงานวิจัยของ Sakurai และคนอื่น ๆ [3] มาประยุกต์ใช้กับการค้นหาเพลง โดยการร้องทำนอง ซึ่งจะมีการนำเสนอในลำดับถัดไป

3.3.3 การค้นหาเพลงในฐานข้อมูลอนุกรมเวลาที่มีขนาดใหญ่

การค้นหาเพลงในฐานข้อมูลอนุกรมเวลาที่มีขนาดใหญ่ นั้น งานวิจัยนี้ได้เลือกใช้วิธีการปรับขนาดเอกรูปและไดนามิกไทม์วอร์ปิงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน รวมทั้งใช้ฟังก์ชันขอบเขตล่างที่นำเสนอ เพื่อให้สามารถค้นหาเพลงได้อย่างรวดเร็ว และเพื่อที่จะแสดงให้เห็นว่าฟังก์ชันขอบเขตล่างที่เสนอนั้น สามารถนำไปประยุกต์ใช้กับเทคนิคในการค้นหาข้อมูลอนุกรมเวลาแบบอื่น ๆ ที่มีประสิทธิภาพได้ ดังนั้นในงานวิจัยนี้จึงมีการประยุกต์ใช้เทคนิคการค้นหาข้อมูลแบบตามลำดับที่เสนอในงานวิจัยของ Sakurai และคนอื่น ๆ [3] ในการค้นหาเพลงจากเสียงร้องทำนอง ซึ่งงานวิจัยดังกล่าวได้อ้างว่าสามารถค้นหาข้อมูลได้รวดเร็วกว่าการค้นหาข้อมูลแบบดัชนีที่นำเสนอในงานวิจัยของ Zhu และ Shasha [10] ซึ่งเป็นวิธีในการค้นหาข้อมูลอนุกรมเวลาที่เร็วที่สุด ณ ขณะนั้น ถึง 222 เท่า โดยผลลัพธ์ในการค้นหาเพลง ก็คือรายการเพลงที่มีคุณลักษณะใกล้เคียงกับเสียงร้องทำนองของผู้ใช้มากที่สุด 10 ลำดับแรก

สำหรับวิธีการค้นหาเพลงในฐานข้อมูลอนุกรมเวลา ที่จะนำเสนอในงานวิจัยนี้ ประกอบด้วยขั้นตอนหลัก ๆ 2 ขั้นตอนด้วยกัน ซึ่งสามารถอธิบายโดยสังเขปได้ดังนี้

1. การให้ค่าเริ่มต้นกับรายการเพลง 10 ลำดับแรกที่มีคุณลักษณะใกล้เคียงกับเสียงร้องทำนองมากที่สุด ในขั้นตอนนี้มีไว้เพื่อที่จะนำค่า *best_so_far* ซึ่งเป็นตัวแปรที่เก็บค่าระยะทางสูงสุดในรายการเพลง 10 ลำดับแรก มาใช้ในการลดภาระการคำนวณค่าระยะทางจริงด้วยวิธีที่นำเสนอในขั้นตอนที่ 2 สำหรับขั้นตอนนี้จะเริ่มจากการพิจารณาค่าระยะทางขอบเขตล่างระหว่างข้อมูลสอบถามที่ถูกลดมิติลงเหลือขนาด 16 มิติ กับเส้นขอบเขตบนและเส้นขอบเขตล่างของข้อมูลเพลงที่ถูกลดมิติลงเหลือขนาด 16 มิติทุกตัว เพื่อที่จะหารายการเพลงที่มีค่าระยะทางขอบเขตล่างต่ำสุด 10 เพลงแรก จากนั้นจึงนำรายการเพลงดังกล่าวไปหาค่าระยะทางจริง พร้อมทั้งนำข้อมูลที่ได้ไปให้ค่าเริ่มต้นกับรายการเพลง 10 ลำดับแรกที่มีคุณลักษณะใกล้เคียงกับเสียงร้องทำนองมากที่สุด
2. การค้นหาข้อมูลตามลำดับโดยใช้ฟังก์ชันขอบเขตล่าง ในขั้นตอนนี้จะเป็นการค้นหาข้อมูลแบบตามลำดับของข้อมูลเพลงที่เก็บอยู่ในฐานข้อมูลที่ละตัว โดยเริ่มจากการพิจารณาค่าระยะทางขอบเขตล่างของวิธีที่นำเสนอ ระหว่างข้อมูลสอบถามและข้อมูลเพลงในฐานข้อมูลที่มีขนาด 32 มิติ ถ้าค่าระยะทางขอบเขตล่างที่ได้มีค่าน้อยกว่าค่าพารามิเตอร์ *best_so_far* ก็จะมีการปรับขนาดเอกรูปกับข้อมูลเพลงที่จะใช้ในการเปรียบเทียบตามอัตราส่วนการหดและยืดขนาดที่กำหนด โดยข้อมูลนี้จะเป็นข้อมูลที่ยังไม่ถูกลดมิติ พร้อมทั้งแปลงข้อมูลดังกล่าวให้เป็นบรรทัดฐาน จากนั้นจึงคำนวณค่าระยะทางขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปิงกับข้อมูลเพลงที่ถูกปรับขนาดเอกรูปดังกล่าว ซึ่งถ้าค่าระยะทาง

ขอบเขตล่างที่ได้ยังคงมีค่าน้อยกว่าค่าพารามิเตอร์ *best_so_far* ก็จะมีการคำนวณค่าระยะทางจริงด้วยวิธีไดนามิกใหม่เวอร์ปึงต่อไป

สำหรับรายละเอียดของวิธีการค้นหาเพลงในฐานข้อมูลอนุกรมเวลาทั้งสองขั้นตอนนี้ จะนำเสนอในลำดับถัดไป รวมทั้งโครงสร้างของฐานข้อมูลที่รองรับวิธีการค้นหาดังกล่าว

3.3.3.1 การให้ค่าเริ่มต้นกับรายการเพลง 10 ลำดับแรกที่มีคุณลักษณะใกล้เคียงกับเสียงร้องทำนองมากที่สุด

อย่างที่กล่าวไว้ในข้างต้น ขั้นตอนนี้มีไว้เพื่อที่จะสามารถนำค่าพารามิเตอร์ *best_so_far* มาใช้ในการลดภาระการคำนวณค่าระยะทางจริงด้วยวิธีที่นำเสนอในขั้นตอนที่ 2 ซึ่งเป็นปัจจัยที่สำคัญอย่างยิ่งในการเพิ่มความเร็วในการประมวลผล เนื่องจากในขั้นตอนที่ 2 นั้น จะเป็นการค้นหาข้อมูลที่ใช้ฟังก์ชันขอบเขตล่างในการประมาณค่าระยะทางระหว่างข้อมูลเสียงร้องทำนองของผู้ใช้กับคุณลักษณะของเพลงที่จะใช้ในการเปรียบเทียบ โดยถ้าค่าระยะทางขอบเขตล่างดังกล่าวมีค่ามากกว่าค่า *best_so_far* นั้นแสดงว่าคุณลักษณะของเพลงที่ใช้ในการเปรียบเทียบนั้น ไม่ได้มีลักษณะคล้ายคลึงกับเสียงร้องทำนองของผู้ใช้มากที่สุด ใน 10 ลำดับเพลงแรก ทำให้สามารถที่จะลดภาระการคำนวณค่าระยะทางจริงระหว่างข้อมูลทั้งสองลงได้ ซึ่งจะช่วยให้การเปรียบเทียบความคล้ายคลึงกันระหว่างข้อมูลดังกล่าวสามารถทำงานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

อย่างไรก็ตาม ในการค้นหาเพลง k ลำดับแรกที่มีคุณลักษณะใกล้เคียงกับเสียงร้องทำนองของผู้ใช้มากที่สุดนั้น จะมีข้อแตกต่างจากการค้นหาข้อมูลที่ใกล้เคียงกับข้อมูลสอบถาม k ลำดับแรกโดยทั่วไป (K-Nearest Neighbor) ก็คือ ในการค้นหาข้อมูลที่ใกล้เคียงกับข้อมูลสอบถามโดยทั่วไปนั้น จะไม่สนใจว่าผลลัพธ์ที่ใกล้เคียง k ลำดับแรกนั้น จะเป็นเพลงที่ใกล้เคียง k ลำดับแรกด้วยหรือไม่ เนื่องจากในบางกรณีผลลัพธ์ที่ใกล้เคียงกับข้อมูลสอบถาม k ลำดับแรก อาจมีข้อมูลมากกว่า 1 ตัวที่เป็นเพลงเดียวกันปรากฏอยู่ก็ได้ ทำให้โดยสรุปผลลัพธ์ที่ได้จะมีจำนวนเพลงน้อยกว่า k เพลงดังที่ต้องการ ดังในงานวิจัยนี้จะมีการปรับเปลี่ยนแนวคิดการค้นหาข้อมูลใกล้เคียงกับข้อมูลสอบถาม k ลำดับแรก ให้สามารถนำมาใช้กับการค้นหาเพลง k ลำดับแรกที่มีคุณลักษณะใกล้เคียงกับเสียงร้องทำนองของผู้ใช้มากที่สุด ซึ่งจะทำให้การพิจารณาค่า *best_so_far* มีลักษณะที่แตกต่างจากการค้นหาข้อมูลที่ใกล้เคียงแบบทั่วไปด้วย

สำหรับหลักในการพิจารณาค่า *best_so_far* เพื่อใช้ในการเปรียบเทียบค่าระยะทางในงานวิจัยนี้ จะมีอยู่ด้วยกัน 4 กรณี คือ

1. ถ้าข้อมูลในรายการเพลงยังมีจำนวนไม่ครบตามต้องการ และในรายการเพลงไม่มีเพลงเดียวกันกับข้อมูลที่ใช้ในการพิจารณา ค่า *best_so_far* จะมีค่าเป็นอนันต์
2. ถ้าข้อมูลในรายการเพลงยังมีจำนวนไม่ครบตามต้องการ แต่ในรายการเพลงมีเพลงเดียวกันกับข้อมูลที่ใช้ในการพิจารณา ค่า *best_so_far* จะมีค่าเท่ากับค่าระยะทางของข้อมูลเพลงในรายการเพลงที่ตรงกันกับข้อมูลที่จะนำมาใช้ในการพิจารณา
3. ถ้าข้อมูลในรายการเพลงมีจำนวนครบตามต้องการ แต่ในรายการเพลงไม่มีเพลงเดียวกันกับข้อมูลที่ใช้ในการพิจารณา ค่า *best_so_far* จะมีค่าเท่ากับค่าระยะทางสูงสุดในรายการเพลงนั้น
4. ถ้าข้อมูลในรายการเพลงมีจำนวนครบตามต้องการ และในรายการเพลงมีเพลงเดียวกันกับข้อมูลที่ใช้ในการพิจารณา ค่า *best_so_far* จะมีค่าเท่ากับค่าระยะทางของข้อมูลเพลงในรายการเพลงที่ตรงกันกับข้อมูลที่จะนำมาใช้ในการพิจารณา

สำหรับวิธีการให้ค่าเริ่มต้นกับรายการเพลง 10 ลำดับแรกที่มีคุณลักษณะใกล้เคียงกับเสียงร้องทำนองมากที่สุดนั้น สามารถแสดงเป็นรหัสเทียม (Pseudo Code) ได้ดังนี้

Algorithm 1 : Initialize_ResultList(Q)

```

1:  Q_16 ← reduceDimensionality(Q,16)
2:
3:  Foreach C ∈ DATABASE do
4:    C_envelope_16 ← DATABASE
5:    dlb_16[C] ← LBPAA(Q_16,C_envelope_16)
6:    If dlb_16[C] ≤ best_so_far_tmp then
7:      TempList ← C,dlb_16[C]
8:    EndIf
9:  EndFor
10:
11: Foreach C ∈ TempList do
12:   C ← DATABASE
13:   dist ← SWMnorm(Q,C,sfmin,sfmax,r)
14:   ResultList ← C,dist
15: EndFor

```

รูปที่ 3.5 รหัสเทียมสำหรับวิธีการให้ค่าเริ่มต้นกับรายการเพลง 10 ลำดับแรก

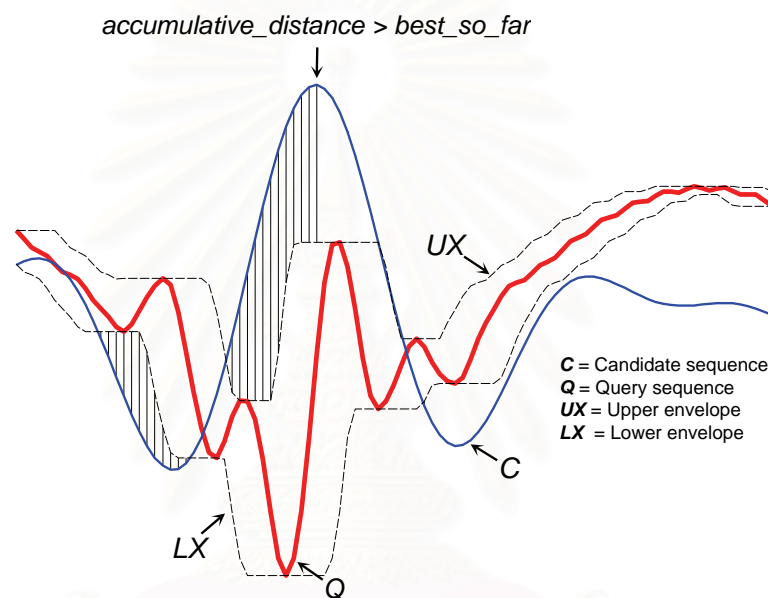
ในบรรทัดที่ 1 จะเป็นการลดมิติข้อมูลสอบถาม Q จากขนาด 128 มิติ ให้เหลือขนาด 16 มิติ ซึ่งจะได้ข้อมูลสอบถาม Q_{16} และในบรรทัดที่ 3 ถึง 9 จะเป็นส่วนของการค้นหาข้อมูลที่มีค่าระยะทางขอบเขตล่างต่ำสุด 10 เพลงแรก โดยในบรรทัดที่ 3 ถึง 4 จะเป็นการดึงข้อมูลเส้นขอบเขตบนและเส้นขอบเขตล่าง $C_{envelope_{16}}$ ที่ถูกลดมิติลงเหลือขนาด 16 มิติของข้อมูลที่จะใช้ในการเปรียบเทียบจากฐานข้อมูลที่ละตัว บรรทัดที่ 5 จะเป็นการคิดค่าระยะทางขอบเขตล่างระหว่างข้อมูลสอบถาม Q_{16} กับเส้นขอบเขตบนและเส้นขอบเขตล่าง $C_{envelope_{16}}$ ที่ได้จากบรรทัดที่ 4 ในบรรทัดที่ 6 7 และ 8 จะเป็นการพิจารณาว่า ข้อมูลที่จะใช้ในการเปรียบเทียบดังกล่าวสามารถที่จะเพิ่มเข้าไปในรายการเพลงชั่วคราว $TempList$ ได้หรือไม่ โดยพิจารณาจาก ถ้าค่าระยะทางขอบเขตล่างที่ได้มีค่าน้อยกว่าหรือเท่ากับค่า $best_so_far_tmp$ ของรายการเพลงชั่วคราวนั้น ก็จะสามารถเพิ่มข้อมูลดังกล่าวเข้าไปในรายการได้ อย่างไรก็ตามถ้าในรายการเพลงชั่วคราวมีเพลงเดียวกันกับข้อมูลที่จะเพิ่มเข้าไป ก็ให้ปรับปรุงข้อมูลในของเพลงที่ตรงกับกับข้อมูลที่จะเพิ่มในรายการเพลงชั่วคราวด้วยข้อมูลใหม่

จากนั้นในบรรทัดที่ 11 ถึง 15 จะเป็นการนำข้อมูลเพลงทั้งหมดในรายการเพลงชั่วคราวไปหาค่าระยะทางจริงด้วยวิธีการปรับขนาดเอกรูปและไดนามิกไทม์วอร์ปิงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน โดยงานวิจัยนี้จะมีการกำหนดอัตราการหดและยืดขนาดมากที่สุดไว้ที่ 0.6 เท่าและ 1.6 เท่า และความกว้างของเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะไว้ที่ 5% ของความยาวข้อมูลสอบถาม จากนั้นในบรรทัดที่ 14 จะเป็นการเพิ่มข้อมูลที่คำนวณค่าระยะทางจริงแล้วเข้าสู่รายการเพลงผลลัพธ์ $ResultList$ ซึ่งการเพิ่มข้อมูลที่มีแนวโน้มที่จะมีค่าระยะทางต่ำ ๆ ที่ได้จากการพิจารณาค่าระยะทางขอบเขตล่าง เข้าไปในรายการเพลง $ResultList$ จนเต็มรายการนี้เอง จะทำให้ค่า $best_so_far$ ที่ได้มีค่าน้อย ซึ่งจะทำให้การลดการคำนวณขั้นตอนที่ 2 นั้น สามารถทำได้อย่างมีประสิทธิภาพ

3.3.3.2 การค้นหาข้อมูลตามลำดับโดยใช้ฟังก์ชันขอบเขตล่าง

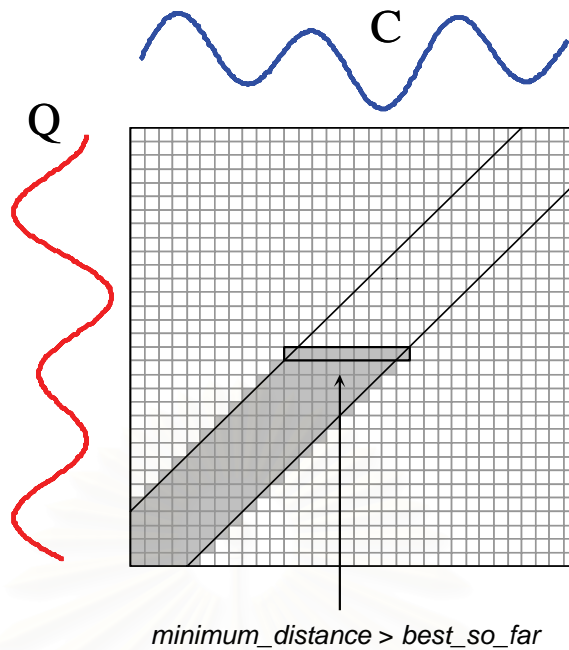
ขั้นตอนนี้จะเป็นการค้นหาข้อมูลแบบตามลำดับของข้อมูลเพลงที่เก็บอยู่ในฐานข้อมูลที่ละตัว โดยเริ่มจากการพิจารณาค่าระยะทางขอบเขตล่างของวิธีที่นำเสนอ ระหว่างข้อมูลสอบถามและข้อมูลเพลงในฐานข้อมูลที่ขนาด 32 มิติ โดยจะมีการแบ่งขนาดการหดหรือยืดขนาดออกเป็นสองช่วง คือ การหดและยืดขนาดตั้งแต่ 0.6 เท่าถึง 1 เท่า และ 1.2 เท่าถึง 1.6 เท่า ทั้งนี้เพื่อให้การพิจารณาค่าระยะทางขอบเขตล่างมีความแม่นยำมากยิ่งขึ้น และถ้าค่าระยะทางขอบเขตล่างที่ได้มีค่าน้อยกว่าค่าพารามิเตอร์ $best_so_far$ ก็จะมีการปรับขนาดเอกรูปกับคุณลักษณะของเพลงที่จะใช้ในการเปรียบเทียบตามอัตราส่วนการหดหรือยืดขนาดที่กำหนดไว้ โดยข้อมูลเหล่านี้จะเป็นข้อมูลที่ยังไม่ถูกลดมิติ พร้อมทั้งแปลงข้อมูลดังกล่าวให้เป็นบรรทัดฐาน จากนั้นจึงคำนวณค่าระยะทางขอบเขตล่างของไดนามิกไทม์วอร์ปิงกับข้อมูลเพลงที่ถูกปรับขนาดเอกรูปดังกล่าว ซึ่งถ้าค่าระยะทางขอบเขตล่างที่ได้ยังคงมีค่าน้อยกว่าค่าพารามิเตอร์ $best_so_far$ ก็จะมีการคำนวณค่าระยะทางจริงด้วยวิธีไดนามิกไทม์วอร์ปิงต่อไป

เพื่อให้การคำนวณฟังก์ชันขอบเขตล่างมีประสิทธิภาพมากยิ่งขึ้น ในงานวิจัยนี้ จะมีการนำวิธีการหยุดคำนวณในตอนต้น (Early Abandon) มาใช้ในการคำนวณหาค่าระยะทางขอบเขตล่าง โดยมีแนวคิดคือ ถ้าค่าระยะทางสะสมในการคำนวณค่าระยะทางขอบเขตล่างมีค่ามากกว่าค่า *best_so_far* แล้ว ให้หยุดการคำนวณระยะทางดังกล่าวทันที เพราะว่าแม้จะมีการคำนวณค่าระยะทางขอบเขตล่างดังกล่าวต่อไปจนจบการคำนวณ ค่าระยะทางที่ได้ก็จะมากกว่าค่า *best_so_far* เสมอ ซึ่งก็คือ ข้อมูลที่จะใช้ในการเปรียบเทียบดังกล่าว ไม่ได้มีความคล้ายคลึงกันกับข้อมูลเสียงร้องทำนองมากที่สุด 10 ลำดับแรก สำหรับแนวคิดนี้สามารถแสดงได้ดังรูปที่ 3.6



รูปที่ 3.6 วิธีการหยุดคำนวณในตอนต้นของฟังก์ชันขอบเขตล่าง

นอกจากที่จะสามารถนำวิธีการหยุดการคำนวณในตอนต้นมาใช้ในการคำนวณฟังก์ชันขอบเขตล่างแล้ว วิธีไดนามิกใหม่เวอร์ปิงนั้นก็สามารถนำวิธีดังกล่าวมาประยุกต์ใช้เพื่อเพิ่มประสิทธิภาพในการคำนวณได้เช่นกัน โดยการคำนวณระยะทางด้วยวิธีไดนามิกใหม่เวอร์ปิงจะสามารถหยุดการคำนวณได้ทันที ถ้าค่าระยะทางสะสมน้อยสุดในแถวมีค่ามากกว่าค่า *best_so_far* เนื่องจากแม้จะมีการคำนวณระยะทางต่อไป ค่าระยะทางที่ได้ก็จะมีความมากกว่าค่า *best_so_far* เสมอ ซึ่งแนวคิดดังกล่าวสามารถแสดงได้ดังรูปที่ 3.7



รูปที่ 3.7 วิธีการหยุดคำนวณในตอนต้นของวิธีไดนามิกโทมวอร์ปิง ถ้าค่าระยะทางน้อยสุดในแถวมีค่ามากกว่าค่า *best_so_far* ให้หยุดการคำนวณระยะทางดังกล่าวทันที

สำหรับวิธีการค้นหาข้อมูลตามลำดับโดยใช้ฟังก์ชันขอบเขตล่างสามารถแสดงเป็นรหัสเทียมได้ดังรูปที่ 3.8

ในบรรทัดที่ 1 จะเป็นการลดมิติของข้อมูลสอบถามจากขนาด 128 มิติ ลงเหลือขนาด 32 มิติ และในบรรทัดที่ 2 จะเป็นส่วนของการหาเส้นขอบเขตบนและเส้นขอบเขตล่างของข้อมูลสอบถาม เพื่อที่จะนำไปใช้ในการหาค่าระยะทางขอบเขตล่างต่อไป ในบรรทัดที่ 4 ถึง 39 นั้นจะเป็นวิธีการค้นหาข้อมูลแบบตามลำดับทุกตัวในฐานข้อมูลโดยใช้วิธีการปรับขนาดเอกรูปและไดนามิกโทมวอร์ปิงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน รวมทั้งใช้ฟังก์ชันขอบเขตล่างในการเพิ่มประสิทธิภาพในการทำงาน บรรทัดที่ 5 มีไว้เพื่อตรวจสอบว่าค่าระยะทางขอบเขตล่างของข้อมูลที่จะใช้ในการเปรียบเทียบที่ขนาด 16 มิติ ซึ่งคำนวณได้จากขั้นตอนที่ 1 มีค่าน้อยกว่าหรือเท่ากับค่า *best_so_far* หรือไม่ และข้อมูลที่จะใช้ในการเปรียบเทียบไม่เป็นสมาชิกของ *TempList* หรือไม่ ถ้าประพจน์ดังกล่าวเป็นจริง จึงจะคำนวณค่าระยะทางต่อไป ซึ่งในบรรทัดที่ 6 จะเป็นการดึงข้อมูลเส้นขอบเขตบนและเส้นขอบเขตล่างของข้อมูลที่จะใช้ในการเปรียบเทียบในกรณีที่ข้อมูลมีการยืดหดตั้งแต่ 0.6 เท่าถึง 1 เท่า โดยข้อมูลดังกล่าวจะถูกนำไปคำนวณหาค่าระยะทางขอบเขตล่างในบรรทัดที่ 7 สำหรับบรรทัดที่ 8 จะเป็นการพิจารณาว่าค่าระยะทางขอบเขตล่างที่ได้ มีค่าน้อยกว่าหรือเท่ากับค่า *best_so_far* หรือไม่ ถ้าใช่ ก็ให้ดึงข้อมูลที่จะใช้ในการเปรียบเทียบจากฐานข้อมูลออกมา

Algorithm 2 : Sequential_Search(Q)

```

1:   $Q_{32} \leftarrow \text{reduceDimension}(Q, 32)$ 
2:   $Q_{\text{envelope}_{128}} \leftarrow \text{get\_DTW\_Envelope}(Q, r)$ 
3:
4:  Foreach  $C \in \text{DATABASE}$  do
5:    If  $d_{lb_{16}}[C] \leq \text{best\_so\_far}$  AND  $C \notin \text{TempList}$  then
6:       $C_{\text{envelope}_{32\_L}} \leftarrow \text{DATABASE}$ 
7:       $\text{dist} \leftarrow \text{LB}_{\text{PAA}}\text{EARLY}(Q_{32}, C_{\text{envelope}_{32\_L}})$ 
8:      If  $\text{dist} \leq \text{best\_so\_far}$  then
9:         $C \leftarrow \text{DATABASE}$ 
10:       Foreach  $sf \in [0.6, 0.8, 1.0]$ 
11:          $C_{sf} \leftarrow \text{RP}_{\text{norm}}(C, sf)$ 
12:          $\text{dist} \leftarrow \text{LB}_{\text{DTW}}\text{EARLY}(Q_{\text{envelope}_{128}}, C_{sf})$ 
13:         If  $\text{dist} \leq \text{best\_so\_far}$  then
14:            $\text{dist} \leftarrow \text{DTW\_EARLY}(Q, C_{sf})$ 
15:           If  $\text{dist} \leq \text{best\_so\_far}$  then
16:              $\text{ResultList} \leftarrow C, \text{dist}$ 
17:           EndIf
18:         EndIf
19:       EndFor
20:     EndIf
21:      $C_{\text{envelope}_{32\_H}} \leftarrow \text{DATABASE}$ 
22:      $\text{dist} \leftarrow \text{LB}_{\text{PAA}}\text{EARLY}(Q_{32}, C_{\text{envelope}_{32\_H}})$ 
23:     If  $\text{dist} \leq \text{best\_so\_far}$  then
24:        $C \leftarrow \text{DATABASE}$ 
25:       Foreach  $sf \in [1.2, 1.4, 1.6]$ 
26:          $C_{sf} \leftarrow \text{RP}_{\text{norm}}(C, sf)$ 
27:          $\text{dist} \leftarrow \text{LB}_{\text{DTW}}\text{EARLY}(Q_{\text{envelope}_{128}}, C_{sf})$ 
28:         If  $\text{dist} \leq \text{best\_so\_far}$  then
29:            $\text{dist} \leftarrow \text{DTW\_EARLY}(Q, C_{sf})$ 
30:           If  $\text{dist} \leq \text{best\_so\_far}$  then
31:              $\text{ResultList} \leftarrow C, \text{dist}$ 
32:           EndIf
33:         EndIf
34:       EndFor
35:     EndIf
36:   EndIf
37: EndFor
39: Return  $\text{ResultList}$ 

```

รูปที่ 3.8 รหัสเทียมสำหรับวิธีการค้นหาข้อมูลตามลำดับโดยใช้ฟังก์ชันขอบเขตล่าง

จากนั้นในส่วนของบรรทัดที่ 10 ถึง 19 จะเป็นการคำนวณหาค่าระยะทางจริง ด้วยวิธีการปรับขนาดเอกรูปและไดนามิกโทมวอร์ปิงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน ซึ่งจะเริ่มจากอัตราส่วนการหดหรือยืดขนาดที่ 0.6 0.8 และ 1 เท่า สำหรับในบรรทัดที่ 11 จะเป็นนำข้อมูลในส่วนต้นของข้อมูลที่จะใช้ในการเปรียบเทียบที่ขนาดของข้อมูลสอบถามคุณด้วยอัตราส่วนการหดหรือยืดขนาดที่ใช้ จากนั้นจึงแปลงข้อมูลดังกล่าวให้เป็นบรรทัดฐานและหดขนาดของข้อมูลนั้นให้เท่ากับขนาดของข้อมูลสอบถาม ในบรรทัดที่ 12 จะเป็นการคำนวณหาระยะทางขอบเขตล่างของวิธีไดนามิกโทมวอร์ปิงระหว่างเส้นขอบเขตบนและเส้นขอบเขตล่างของข้อมูลสอบถามกับข้อมูลที่จะใช้ในการเปรียบเทียบที่ได้จากบรรทัดที่ 11 และในบรรทัดที่ 13 จะเป็นเงื่อนไขในการพิจารณาค่าระยะทางขอบเขตล่างนั้น ในบรรทัดที่ 14 จึงจะเป็นการคำนวณค่าระยะทางจริงด้วยไดนามิกโทมวอร์ปิง ซึ่งค่าระยะทางจริงที่ได้จะถูกพิจารณาต่อในบรรทัดที่ 15 และนำไปเพิ่มให้กับรายการเพลงที่จะเป็นคำตอบ *ResultList* ในบรรทัดที่ 16 ต่อไป สำหรับบรรทัดที่ 21 ถึง 35 ก็จะมีลักษณะเช่นเดียวกันกับรหัสเทียมในบรรทัดที่ 9 ถึง 20 สุดท้ายผลลัพธ์ของขั้นตอนวิธีนี้ก็คือ รายการเพลงที่ใกล้เคียงกับเสียงร้องทำนองมากที่สุด 10 ลำดับแรก

อย่างไรก็ตาม สำหรับวิธีการค้นหาข้อมูลที่ได้นำเสนอไปนั้น นอกจากจะมีขั้นตอนในการค้นหาข้อมูลอยู่ด้วยกัน 2 ขั้นตอนแล้ว สิ่งสำคัญอีกประการหนึ่งก็คือ การจัดการโครงสร้างของฐานข้อมูล เพื่อให้วิธีการค้นหาข้อมูลที่นำเสนอ สามารถทำงานได้อย่างมีประสิทธิภาพ

3.3.3.3 โครงสร้างของฐานข้อมูล

จากวิธีการค้นหาข้อมูลที่นำเสนอานั้น จะมีการค้นหาข้อมูลแบบตามลำดับจากหน่วยเก็บข้อมูลแบบทุดิยภูมิจำนวน 2 ครั้งด้วยกัน ก็คือในขั้นตอนที่ 1 และขั้นตอนที่ 2 โดยในขั้นตอนที่ 1 จะเป็นการดึงข้อมูลเส้นขอบเขตบนและเส้นขอบเขตล่างของข้อมูลเพลงที่ถูกลดมิติลงเหลือขนาด 16 มิติ และในขั้นตอนที่ 2 จะเป็นการดึงข้อมูลเส้นขอบเขตบนและเส้นขอบเขตล่างสำหรับการหดหรือยืดขนาดของข้อมูล 2 ช่วงที่ถูกลดมิติลงเหลือขนาด 32 มิติ พร้อมทั้งดึงข้อมูลอนุกรมเวลาที่เป็นคุณลักษณะของเพลงที่จะใช้ในการเปรียบเทียบ

ดังนั้นในการออกแบบโครงสร้างของฐานข้อมูลในงานวิจัยนี้ จะมีการสร้างแฟ้มฐานข้อมูลอยู่ด้วยกัน 2 แฟ้ม เพื่อรองรับการค้นหาข้อมูลทั้งสองขั้นตอน โดยในแฟ้มฐานข้อมูลแรก จะเก็บข้อมูลที่จะมีการเรียกใช้จากขั้นตอนที่ 1 ซึ่งก็คือ รหัสคุณลักษณะของเพลง และข้อมูลเส้นขอบเขตบนและเส้นขอบเขตล่างที่ถูกลดมิติลงเหลือขนาด 16 มิติ สำหรับแฟ้มฐานข้อมูลที่สอง จะเก็บข้อมูลที่จะมีการเรียกใช้จากขั้นตอนที่ 2 ซึ่งก็คือ รหัสคุณลักษณะของเพลง ข้อมูลเส้นขอบเขตบนและเส้นขอบเขตล่างที่ถูกลดมิติลงเหลือขนาด 32 มิติ จำนวน 2 ชุด และข้อมูลคุณลักษณะของเพลงที่จะใช้ในการเปรียบเทียบ

สำหรับในบทนี้ได้มีการนำเสนอวิธีการพัฒนาระบบค้นหาเพลงโดยการร้องทำนองซึ่งจะประกอบไปด้วยองค์ประกอบ 3 ส่วนหลัก ๆ คือ การสกัดคุณลักษณะจากเสียงร้องทำนอง การสกัดคุณลักษณะจากเพลงในฐานข้อมูล และวิธีการค้นหาเพลงจากเสียงร้องทำนองซึ่งในงานวิจัยนี้ได้ใช้วิธีการปรับขนาดเอกรูปและไดนามิกโทมวอร์ปปิงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน รวมทั้งยังได้นำเสนอฟังก์ชันขอบเขตล่างของวิธีดังกล่าวเพื่อให้สามารถค้นหาข้อมูลได้อย่างมีประสิทธิภาพมากยิ่งขึ้น พร้อมทั้งนำเสนอวิธีค้นหาข้อมูลเพลงภายใต้ฐานข้อมูลที่มีขนาดใหญ่โดยใช้ฟังก์ชันขอบเขตล่าง อย่างไรก็ตามการทดลองในเรื่องประสิทธิภาพทั้งในด้านความแม่นยำและความเร็วของวิธีที่นำเสนอ นั้น จะมีการนำเสนอในลำดับถัดไป



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

การทดลองและวิเคราะห์ผล

สำหรับงานวิจัยนี้ เพื่อที่จะสามารถประเมินคุณภาพและประสิทธิภาพของวิธีที่นำเสนอได้อย่างครบถ้วน ดังนั้นจะแบ่งการทดลองออกเป็นสองส่วนหลัก ๆ ด้วยกัน คือ

1. การทดลองเพื่อวิเคราะห์คุณภาพฟังก์ชันขอบเขตล่างของวิธีปรับขนาดเอกรูปและไดนามิก-โทมวอร์ปิงที่รองรับวิธีการแปลงข้อมูลให้เป็นบรรทัดฐาน และ
2. การทดลองเกี่ยวกับประสิทธิภาพ ทั้งในด้านความแม่นยำและความเร็วในการค้นหาเพลงโดยการร้องทำนองจากวิธีที่นำเสนอ

4.1 การทดลองเพื่อวิเคราะห์คุณภาพฟังก์ชันขอบเขตล่างที่นำเสนอ

ในการคำนวณหาค่าระยะทางจริงด้วยวิธีปรับขนาดเอกรูปและไดนามิก-โทมวอร์ปิงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐานนั้น จะใช้ระยะเวลาในการประมวลผลนาน ทำให้ไม่สามารถค้นหาข้อมูลภายใต้ฐานข้อมูลที่มีขนาดใหญ่ได้อย่างรวดเร็วและมีประสิทธิภาพ ดังนั้นจึงต้องมีฟังก์ชันขอบเขตล่างที่สามารถประมวลผลได้อย่างรวดเร็ว รวมทั้งสามารถลดปริมาณข้อมูลที่จะต้องคำนวณหาค่าระยะทางจริงได้เป็นจำนวนมาก มาใช้ในการค้นหาข้อมูล อย่างไรก็ตามการวัดคุณภาพของฟังก์ชันขอบเขตล่างโดยทั่วไปนั้นจะสามารถวัดได้หลากหลายวิธี แต่หนึ่งในวิธีที่ได้รับความนิยมมากที่สุดก็คือ การวัดความสามารถในการลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริง (Pruning Power) [2] ซึ่งก็คือ อัตราส่วนของจำนวนข้อมูลที่ไม่ต้องคำนวณค่าระยะทางจริง หารด้วยจำนวนข้อมูลทั้งหมด ดังแสดงในสมการที่ (4.1) โดยในงานวิจัยนี้ จะมีการนำวิธีดังกล่าวมาใช้วัดคุณภาพของฟังก์ชันขอบเขตล่างที่นำเสนอ

$$\text{Pruning Power} = \frac{\text{Number of pruned candidates}}{\text{Total number of candidate sequences}} \quad (4.1)$$

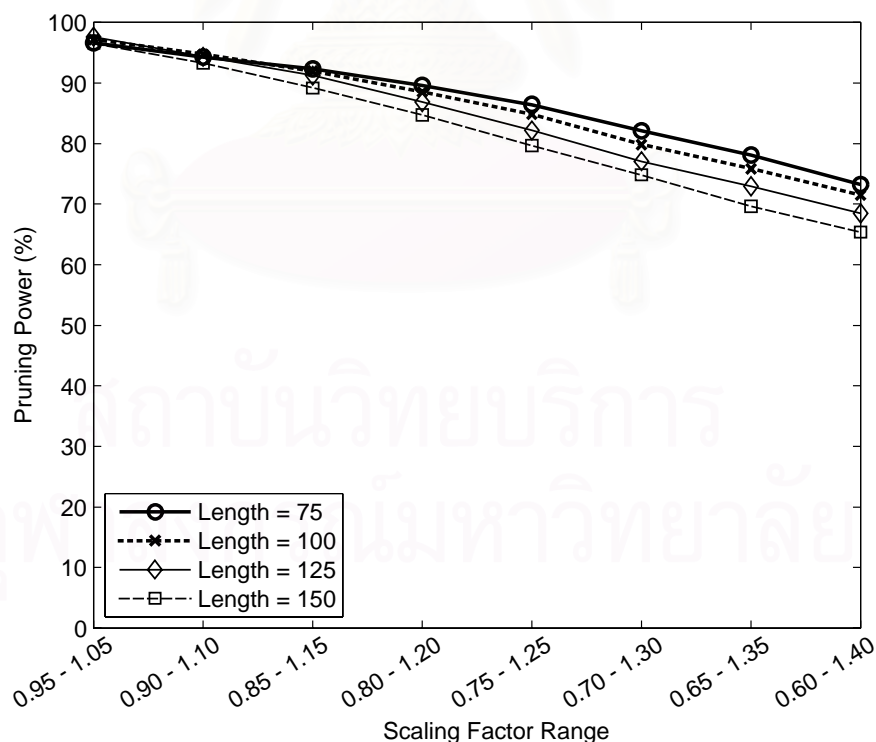
ดังนั้นเพื่อที่จะวัดคุณภาพฟังก์ชันขอบเขตล่างของวิธีที่นำเสนอ ในการทดลองนี้จะพัฒนาระบบค้นหาเพลงโดยการร้องทำนองอย่างง่าย มาใช้ในการวัดคุณภาพของฟังก์ชันขอบเขตล่างดังกล่าว โดยระบบค้นหาเพลงโดยการร้องทำนองอย่างง่ายนั้น จะเป็นการค้นหาเพลงที่มีคุณลักษณะใกล้เคียงกับเสียงร้องทำนองมากที่สุดเพียงเพลงเดียว

ฐานข้อมูลเพลงที่ใช้ในการทดลองนี้จะใช้แฟ้มข้อมูลมิตี โดยจะมีขนาดตั้งแต่ 100 เพลง ไปจนถึง 2,000 เพลง และสำหรับการสกัดคุณลักษณะของเพลงนั้น สามารถทำได้โดยใช้วิธีการเลื่อนหน้าต่าง เพื่อที่จะตัดแบ่งข้อมูลคอนทิวรัระดับเสียงที่ความยาว $sfmax * 150$ มิตี โดยที่ $sfmax$ คือ อัตราส่วนการยืดขนาดสูงสุด ซึ่งในการทดลองนี้ กำหนดให้ $sfmax$ มีค่า

เท่ากับ 1.4 และให้ข้อมูลเสียงร้องทำนองมีความยาวสูงสุดที่ 150 มิต สำหรับข้อมูลเสียงร้องทำนองในการทดลองนี้ ได้สุ่มเก็บตัวอย่างเสียงร้องทำนองจำนวน 55 ตัวอย่าง จากผู้ร้องจำนวน 12 คน ซึ่งไม่มีพื้นฐานในการร้องเพลง

เพื่อที่จะประเมินคุณภาพของฟังก์ชันขอบเขตล่างที่นำเสนอไปได้อย่างครบถ้วน งานวิจัยนี้จะแบ่งการทดลองออกเป็น 3 การทดลองย่อย ๆ เพื่อพิจารณาถึงปัจจัยต่าง ๆ ที่มีผลกระทบต่อคุณภาพของฟังก์ชันขอบเขตล่าง ซึ่งก็คือ ความยาวของข้อมูลสอบถาม อัตราส่วนการหดและยืดขนาดสูงสุด ความกว้างของเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะ และขนาดของฐานข้อมูล

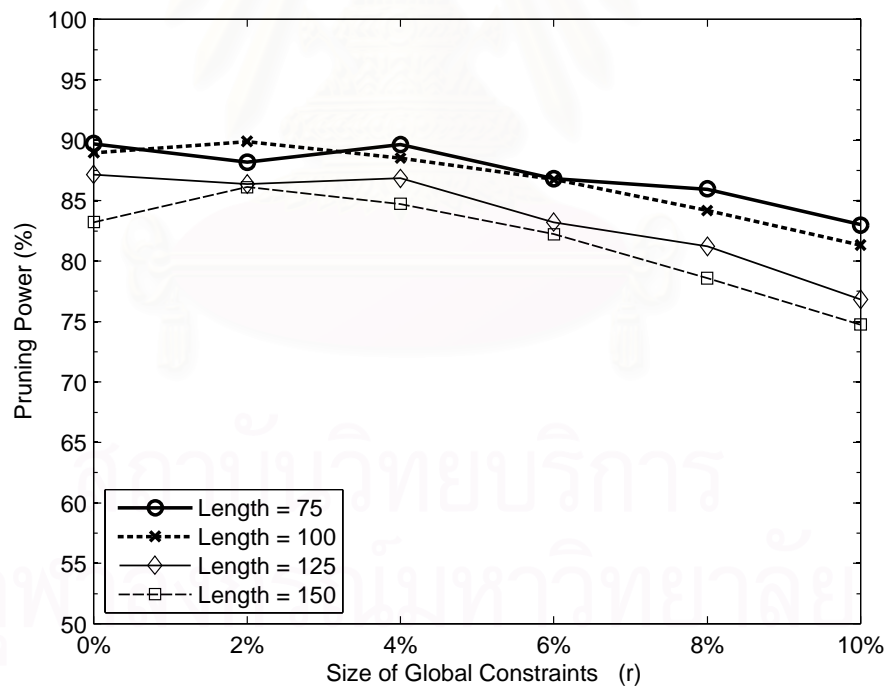
การทดลองแรกจะพิจารณาปัจจัยเกี่ยวกับความยาวของข้อมูลสอบถาม และอัตราส่วนการหดและยืดขนาดสูงสุด ที่จะมีผลต่อความสามารถในการลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริง โดยจะมีการทดลองกับข้อมูลสอบถามที่มีความยาว 75 100 125 และ 150 มิต รวมทั้งมีการปรับอัตราส่วนการหดและยืดขนาดสูงสุดที่อัตราส่วนต่าง ๆ อย่างไรก็ตามในการทดลองนี้ จะทดลองภายใต้ฐานข้อมูลอนุกรมเวลาขนาด 22,441 อนุกรม ซึ่งสกัดจากเพลงจำนวน 100 เพลง และกำหนดให้ความกว้างของเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะไว้ที่ 4% โดยผลการทดลองนี้ สามารถแสดงเป็นกราฟดังรูปที่ 4.1



รูปที่ 4.1 ความสามารถในการลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริงของข้อมูลสอบถามที่มีความยาวและอัตราส่วนการหดและยืดขนาดต่าง ๆ

จากผลการทดลองในรูปที่ 4.1 จะเห็นได้ว่าฟังก์ชันขอบเขตล่างที่นำเสนอ นั้นสามารถลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริงได้เป็นจำนวนมาก ทำให้มีข้อมูลเพียงส่วนน้อยเท่านั้นที่จะต้องคำนวณค่าระยะทางจริง อย่างไรก็ตามปัจจัยเรื่องความยาวของข้อมูลสอบถามนั้น เมื่อข้อมูลสอบถามมีความยาวมากขึ้น แนวโน้มความสามารถในการลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริงก็จะลดลง ซึ่งสอดคล้องกับอัตราส่วนการหัดและยึดขนาดสูงสุด ที่ความสามารถในการลดปริมาณข้อมูลนั้นจะลดลง เมื่ออัตราส่วนการหัดและยึดขนาดสูงสุดมีขนาดกว้างขึ้น

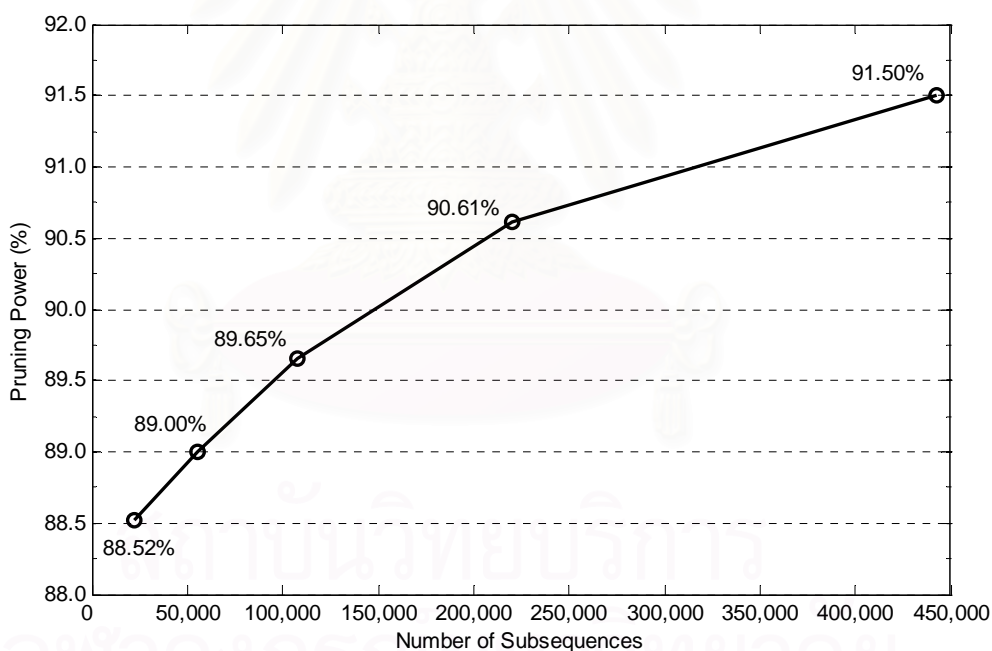
สำหรับการทดลองที่สอง จะพิจารณาปัจจัยเกี่ยวกับความยาวของข้อมูลสอบถาม และความกว้างของเงื่อนไขบังคับโดยรวมแบบซาทโก-ชิบะ โดยจะมีการทดลองกับข้อมูลสอบถามที่มีความยาว 75 100 125 และ 150 มิติ รวมทั้งมีการปรับความกว้างของเงื่อนไขบังคับโดยรวมแบบซาทโก-ชิบะที่ขนาด 0% 2% 4% 6% 8% และ 10% ของความยาวข้อมูลสอบถาม อย่างไรก็ตาม ในการทดลองนี้ จะทดลองภายใต้ฐานข้อมูลอนุกรมเวลาขนาด 22,441 อนุกรม ซึ่งสกัดจากเพลงจำนวน 100 เพลง และกำหนดให้อัตราส่วนการหัดและยึดขนาดสูงสุดไว้ที่ 0.8 ถึง 1.2 เท่า โดยผลการทดลองสามารถแสดงเป็นกราฟดังรูปที่ 4.2



รูปที่ 4.2 ความสามารถในการลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริงของข้อมูลสอบถามที่มีความยาว และความกว้างของเงื่อนไขบังคับโดยรวมแบบซาทโก-ชิบะที่ขนาดต่าง ๆ

ผลการทดลองในรูปที่ 4.2 จะสอดคล้องกับผลการทดลองในรูปที่ 4.1 ที่ว่าเมื่อข้อมูลสอบถามมีความยาวมากขึ้น ความสามารถในการลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริงก็จะมีแนวโน้มที่ลดลง อย่างไรก็ตามปัจจัยในเรื่องของความกว้างของเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะนั้น ดูเหมือนจะส่งผลกระทบต่อความสามารถในการลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริงน้อย แต่ก็มีแนวโน้มที่จะลดลงเมื่อความกว้างของเงื่อนไขบังคับโดยรวมมีขนาดใหญ่ขึ้น

สำหรับการทดลองสุดท้าย จะทดสอบในเรื่องขนาดของฐานข้อมูลที่จะส่งผลกระทบต่อความสามารถในการลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริง โดยจะทดลองกับฐานข้อมูลขนาด 22,441 55,595 107,993 220,378 และ 442,366 อนุกรม ซึ่งสกัดจากเพลงจำนวน 100 250 500 1,000 และ 2,000 เพลงตามลำดับ โดยที่กำหนดให้ความยาวของข้อมูลสอบถามเท่ากับ 100 มิติ อัตราส่วนการหดหรือยืดขนาดสูงสุดที่ 0.8 ถึง 1.2 เท่า และความกว้างของเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะเท่ากับ 4% โดยผลการทดลองสามารถแสดงเป็นกราฟดังรูปที่ 4.3



รูปที่ 4.3 ความสามารถในการลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริง เมื่อฐานข้อมูลมีขนาดต่าง ๆ

จากผลการทดลองในรูปที่ 4.3 จะเห็นได้ชัดว่าความสามารถในการลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริงนั้น มีค่าสูงขึ้นตามขนาดของฐานข้อมูลที่มีขนาดใหญ่ขึ้น ซึ่งคุณลักษณะนี้จะเป็นคุณลักษณะที่ต้องการ เนื่องจากเมื่อฐานข้อมูลมีขนาดใหญ่ขึ้น สัดส่วน

ของปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริงจะมีปริมาณลดลง ดังนั้นความเร็วในการค้นหาจะไม่เพิ่มขึ้นเป็นฟังก์ชันเชิงเส้นเมื่อเปรียบเทียบกับปริมาณข้อมูลที่เพิ่มขึ้น แต่จะมีอัตราในการเพิ่มขึ้นน้อยกว่าฟังก์ชันเชิงเส้น

โดยสรุป จากผลการทดลองทั้งหมดนั้น จะเห็นได้ว่าฟังก์ชันขอบเขตล่างที่นำเสนอสามารถลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริงได้เป็นจำนวนมาก ซึ่งจะช่วยให้การค้นหาข้อมูลสามารถทำงานได้อย่างรวดเร็วและมีประสิทธิภาพมากยิ่งขึ้น อย่างไรก็ตามปัจจัยต่าง ๆ ที่จะทำให้ระยะห่างระหว่างเส้นขอบเขตบนและเส้นขอบเขตล่างมีขนาดใหญ่ขึ้น ก็จะส่งผลให้ความสามารถในการลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริงลดลง เนื่องจากเมื่อระยะห่างระหว่างเส้นขอบเขตบนและเส้นขอบเขตล่างมีขนาดใหญ่ขึ้น ค่าระยะทางที่ได้จากการคำนวณฟังก์ชันขอบเขตล่างก็จะมีค่าน้อยลง นั่นคือฟังก์ชันขอบเขตล่างที่ใช้ในการคำนวณจะมีความกระชับลดลง ซึ่งจะส่งผลโดยตรงให้ความสามารถในการลดปริมาณข้อมูลนั้นลดลงไปด้วย อย่างไรก็ตามความสามารถในการลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริงจะสูงขึ้น ถ้าฐานข้อมูลมีขนาดใหญ่ขึ้น ซึ่งเป็นคุณลักษณะที่ดีและเป็นที่ต้องการอย่างยิ่งในการนำไปประยุกต์ใช้กับการค้นหาข้อมูลภายใต้ฐานข้อมูลที่มีขนาดใหญ่

นอกจากนั้น ฟังก์ชันขอบเขตล่างที่นำเสนอไป ยังสามารถที่จะนำไปใช้ร่วมกับวิธีการค้นหาข้อมูลแบบอื่น ๆ ที่มีประสิทธิภาพสูงได้ เช่น วิธีการค้นหาข้อมูลแบบดัชนี หรือวิธีการค้นหาข้อมูลแบบตามลำดับที่นำเสนอในงานวิจัยของ Sakurai และคนอื่น ๆ [3] ซึ่งในงานวิจัยนี้จะมีการนำวิธีดังกล่าวมาประยุกต์ใช้ โดยจะมีการนำเสนอผลการทดลองถึงประสิทธิภาพของวิธีดังกล่าวในลำดับถัดไป

4.2 การทดลองเกี่ยวกับการวิเคราะห์ประสิทธิภาพของวิธีการค้นหาเพลงโดยการร้องทำนองที่นำเสนอ

สำหรับการทดลองเกี่ยวกับประสิทธิภาพของวิธีการค้นหาเพลงโดยการร้องทำนองที่นำเสนอ นั้น จะแบ่งการทดลองออกเป็น 3 การทดลองหลัก ๆ คือ 1. การทดลองเพื่อวิเคราะห์ประสิทธิภาพของวิธีที่นำเสนอ 2. การทดลองเพื่อวิเคราะห์ความแม่นยำของวิธีที่นำเสนอเมื่อเปรียบเทียบกับวิธีอื่น ๆ และ 3. การทดลองเพื่อวิเคราะห์ความเร็วในการค้นหาเพลงเมื่อเปรียบเทียบกับวิธีอื่น ๆ โดยข้อมูลเสียงร้องทำนองที่จะมาใช้เป็นข้อมูลสอบถามสำหรับระบบค้นหาเพลงโดยการร้องทำนองในการทดลองต่าง ๆ นั้น ได้สุ่มเก็บตัวอย่างเสียงร้องทำนองจำนวน 100 ตัวอย่างจากผู้ร้องทั้งสิ้น 15 คน เป็นชายจำนวน 11 คน และเป็นหญิงจำนวน 4 คน โดยผู้ร้องทุกคนไม่เคยได้รับการฝึกฝนทักษะการร้องเพลงมาก่อน แต่อาจมีพื้นฐานทางดนตรีที่แตกต่างกันไป เช่น บางคนสามารถเล่นเปียโนได้ บางคนสามารถเล่นกีตาร์ได้ หรือบางคนไม่เล่นเครื่องดนตรีชนิดใดเลย สำหรับขั้นตอนการเก็บตัวอย่างเสียงร้องทำนอง ผู้ใช้จะได้รับคำแนะนำให้ร้องทำนองอย่างเป็นธรรมชาติในท่อนใดของเพลงในรายการที่กำหนด

ก็ได้ เป็นระยะเวลาประมาณ 7-8 วินาที ซึ่งสภาพแวดล้อมในการทดสอบนั้น จะเป็นสภาพแวดล้อมแบบเปิด ทำให้อาจมีเสียงอื่นที่ไม่ใช่เสียงร้องทำนองเล็ดลอดเข้าสู่การทดลอง และในบางครั้งผู้ร้องจะร้องเนื้อเพลงแทนการร้องทำนอง รวมไปถึงการร้องทั้งเนื้อเพลงและทำนองเพลงปะปนกัน และอาจจะมีเสียงสนทนาของผู้ร้องเล็ดลอดเข้ามาในแฟ้มข้อมูลที่ใช้ในการทดลองอีกด้วย

สำหรับฐานข้อมูลเพลงมิตินี้ที่ใช้ในการทดสอบนั้น จะเป็นเพลงสากลทั้งในประเทศและต่างประเทศ โดยในแต่ละการทดลองจะทดลองกับฐานข้อมูลที่มีขนาดแตกต่างกัน ตั้งแต่ 100 เพลง ไปจนถึง 2,500 เพลง อย่างไรก็ตามฐานข้อมูลจริงที่จะถูกนำมาใช้ในการเปรียบเทียบความคล้ายคลึงกันกับเสียงร้องทำนองก็คือ คอนทิวรัลระดับเสียงของแต่ละเพลง ซึ่งเป็นคุณลักษณะที่สกัดได้จากแฟ้มข้อมูลมิตินี้ โดยจะสกัดออกมาในรูปแบบของข้อมูลอนุกรมเวลา สำหรับจำนวนของคอนทิวรัลระดับเสียงในฐานข้อมูลที่มีขนาดต่าง ๆ นั้น สามารถแสดงได้ดังตารางที่ 4.1

ตารางที่ 4.1 จำนวนของคอนทิวรัลระดับเสียงที่สกัดจากเพลงมิตินี้ฐานข้อมูลขนาดต่าง ๆ

จำนวนเพลง (เพลง)	จำนวนข้อมูลอนุกรมเวลา (ข้อมูล)
100	20,567
250	53,397
500	99,468
750	153,206
1,000	203,461
1,500	303,432
2,000	391,823
2,500	472,349

นอกจากเสียงร้องทำนองของผู้ใช้และฐานข้อมูลเพลงแล้ว สิ่งสำคัญอีกประการหนึ่งในการทำการทดลองนี้ก็คือ ระบบที่ใช้ในการทดลอง สำหรับระบบที่ใช้ในการทดลองนั้นใช้เครื่องโน้ตบุ๊กคอมพิวเตอร์ Fujitsu รุ่น A6030 โดยมี Intel® Core™2 Duo Processor T7300 ความเร็ว 2.0 GHz เป็นซีพียูในการประมวลผล แรมขนาด 2 GB DDR2 ความเร็ว 667 MHz และจานบันทึกแบบแข็งขนาด 120 GB เชื่อมต่อแบบ SATA ความเร็ว 5,400 รอบต่อนาที

ในส่วนถัดไปจะเป็นส่วนของการทดลอง ผลการทดลอง และการวิเคราะห์ผลของการทดลองต่าง ๆ

4.2.1 การทดลองเพื่อวิเคราะห์ประสิทธิภาพของวิธีที่นำเสนอ

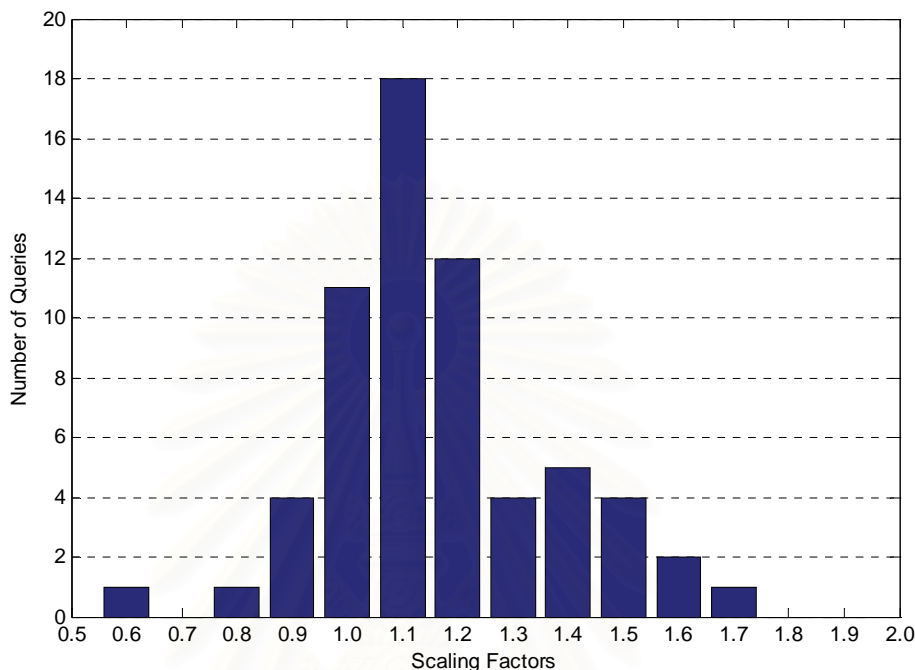
สำหรับการทดลองนี้ มีจุดประสงค์เพื่อที่จะวิเคราะห์หาค่าพารามิเตอร์ต่าง ๆ ที่สามารถรองรับความแปรผันที่เกิดจากการร้องทำนองของผู้ใช้ได้อย่างเหมาะสม รวมไปถึงวิเคราะห์ความแม่นยำและความเร็วของวิธีที่นำเสนอเมื่อฐานข้อมูลมีขนาดต่าง ๆ ดังนั้นในการทดลองนี้จะแบ่งเป็นการทดลองย่อย ๆ ด้วยกัน 4 การทดลอง คือ 1. การวิเคราะห์หาอัตราส่วนการหดและยืดขนาดสูงสุดที่เหมาะสมสำหรับระบบค้นหาเพลงโดยการร้องทำนอง 2. การวิเคราะห์หาค่าความกว้างของเงื่อนไขบังคับโดยรวมแบบซอกโก-ชิบะที่เหมาะสม 3. การวิเคราะห์ความแม่นยำของวิธีที่นำเสนอเมื่อฐานข้อมูลมีขนาดต่าง ๆ 4. การวิเคราะห์ความเร็วของวิธีที่นำเสนอเมื่อฐานข้อมูลมีขนาดต่าง ๆ

4.2.1.1 การวิเคราะห์หาอัตราส่วนการหดและยืดขนาดสูงสุดที่เหมาะสมสำหรับระบบค้นหาเพลงโดยการร้องทำนอง

การวิเคราะห์หาอัตราส่วนการหดและยืดขนาดสูงสุดที่เหมาะสมนั้น เพื่อที่จะหาอัตราส่วนการหดและยืดขนาดที่เหมาะสมที่จะสามารถรองรับความแปรผันในเรื่องการร้องซ้ำหรือการร้องเร็วของผู้ใช้ได้ ซึ่งการวิเคราะห์ดังกล่าวสามารถทำได้โดยการนำระบบค้นหาเพลงโดยการร้องทำนองที่นำเสนอ มาทดลองภายใต้ฐานข้อมูลเพลงขนาด 1,000 เพลง โดยมีการปรับแก้ให้มีอัตราส่วนการหดและยืดขนาดสูงสุดที่ 0.5 ถึง 2 เท่า ซึ่งมาจากแนวคิดในงานวิจัยต่าง ๆ [4, 10] ที่ว่า ผู้ใช้โดยทั่วไปมีโอกาสที่จะร้องทำนองซ้ำหรือเร็วกว่าจังหวะของเพลงในฐานข้อมูลถึง 2 เท่า และกำหนดให้มีการปรับอัตราส่วนในการหดและยืดขนาดเพิ่มครั้งละ 0.1 เท่า นั่นคือ จะมีการหดและยืดขนาดที่อัตราส่วน 0.5 0.6 0.7 ไปเรื่อย ๆ จนถึง 2.0 เท่า ตามลำดับและกำหนดให้ความกว้างของเงื่อนไขบังคับโดยรวมแบบซอกโก-ชิบะมีขนาดเท่ากับ 5% โดยผลการทดลองนั้นสามารถแสดงได้ดังรูปที่ 4.4

จากผลการทดลองจะเห็นได้ชัดว่าผู้ใช้โดยส่วนใหญ่ มีแนวโน้มที่จะร้องทำนองเร็วกว่าจังหวะของเพลงในฐานข้อมูล ทำให้จะต้องมีการยืดขนาดของข้อมูลสอบถามก่อนที่จะวัดค่าระยะทาง อย่างไรก็ตามอัตราส่วนของการหดและยืดขนาดสูงสุดที่ได้จากการทดลองนี้จะมีขนาดเพียง 0.6 ถึง 1.7 เท่า เท่านั้น ซึ่งเป็นเหตุผลให้ในงานวิจัยนี้จะมีการกำหนดให้ค่าอัตราส่วนของการหดและยืดขนาดสูงสุดเท่ากับ 0.6 ถึง 1.6 เท่า แทนที่จะกำหนดให้อัตราส่วนการหดและยืดขนาดสูงสุดที่ 0.5 ถึง 2.0 เท่าเช่นเดียวกับงานวิจัยอื่น ๆ โดยที่จะมีการปรับอัตราส่วนการหดและยืดขนาดเพิ่มครั้งละ 0.2 เท่า เพื่อให้สามารถทำงานได้อย่างถูกต้องและรวดเร็ว อย่างไรก็ตามแม้ว่าการปรับอัตราส่วนการหดและยืดขนาดให้เพิ่มขึ้นครั้งละ 0.2 เท่านั้น

จะทำให้ไม่สามารถครอบคลุมช่วง 0.6 ถึง 1.7 ได้อย่างสมบูรณ์ แต่จากการทดลองพบว่า การปรับอัตราส่วนการหดและยืดขนาดที่ความละเอียดสูงกว่านี้และกว้างกว่านี้ จะไม่ช่วยให้ความแม่นยำเพิ่มขึ้นมากนัก



รูปที่ 4.4 การกระจายตัวของเสียงร้องทำนองที่มีจังหวะช้าหรือเร็วกว่าจังหวะเพลงในฐานข้อมูล ที่อัตราส่วนการหดและยืดขนาดต่าง ๆ

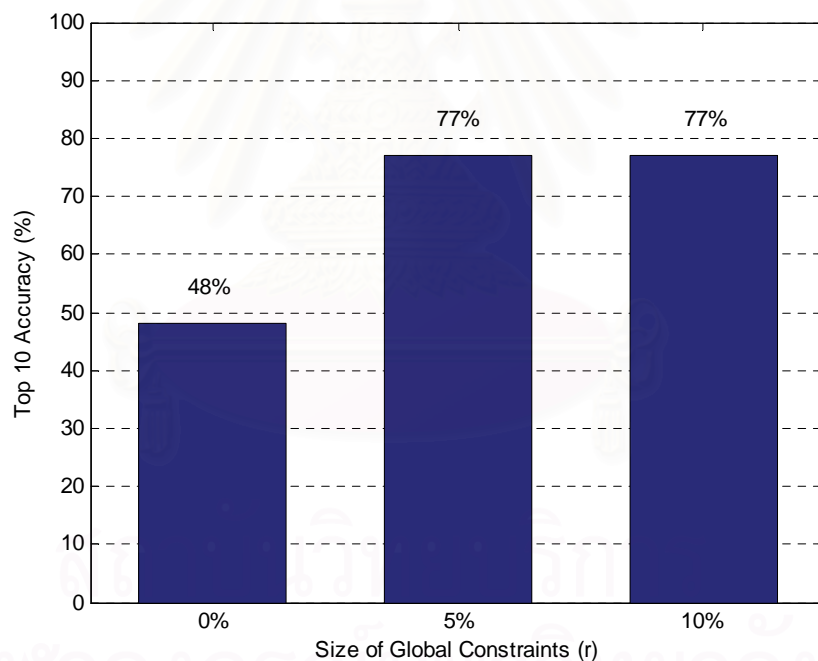
4.2.1.2 การวิเคราะห์หาความกว้างของเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะที่เหมาะสม

สำหรับการวิเคราะห์หาความกว้างของเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะที่เหมาะสม จะมีการเปรียบเทียบความแม่นยำในการค้นหาข้อมูล เมื่อความกว้างของเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะมีขนาดต่าง ๆ คือ มีขนาดเท่ากับ 0% 5% และ 10% ของขนาดข้อมูลสอบถาม โดยจะมีการทดลองกับฐานข้อมูลขนาด 1,000 เพลง และอัตราส่วนการหดและยืดขนาดสูงสุดที่ 0.6 ถึง 1.6 เท่า อย่างไรก็ตาม ในการเปรียบเทียบความแม่นยำของการค้นหาเพลงโดยการร้องทำนองนั้น จะสามารถวัดได้โดยวัดความแม่นยำในการค้นหาเพลง 10 ลำดับแรก ซึ่งผลการทดลองดังกล่าวจะสามารถแสดงได้ดังรูปที่ 4.5 อย่างไรก็ตามการวัดความแม่นยำในลักษณะนี้ อาจจะไม่สะท้อนถึงคุณภาพของคำตอบได้อย่างชัดเจน เนื่องจากไม่ได้พิจารณาถึงลำดับของคำตอบในรายการเพลงผลลัพธ์ ดังนั้นเพื่อที่จะวิเคราะห์คุณภาพของคำตอบได้อย่างเหมาะสม ในการทดลองนี้จะมีการพิจารณาค่าเฉลี่ยลำดับส่วนกลับ (Mean Reciprocal Rank หรือ MRR) ในการวัดค่าความแม่นยำ ซึ่งสามารถคำนวณได้ดังนี้สมการที่ (4.2) โดยที่ $rank_i$

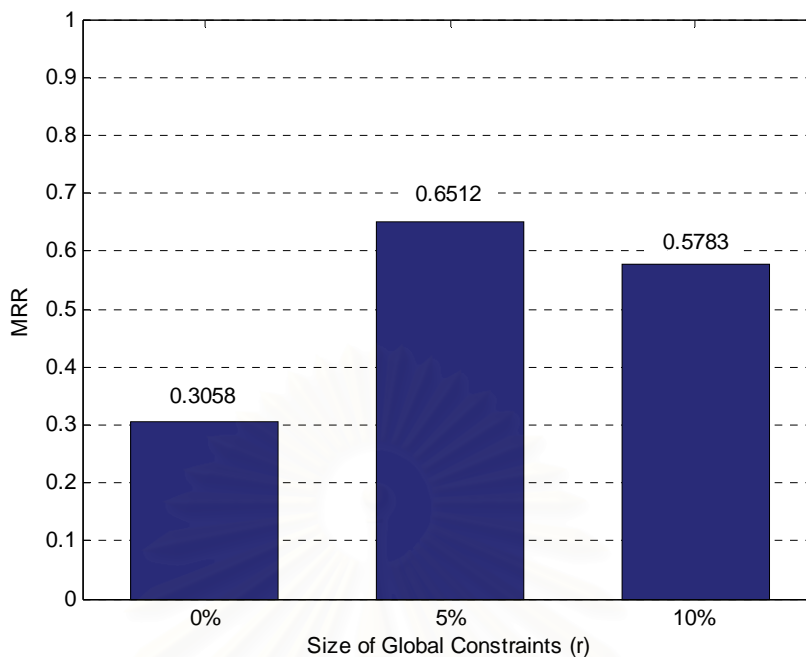
คือ ลำดับของคำตอบในรายการเพลงผลลัพธ์ของข้อมูลสอบถามตัวที่ i และ n คือ จำนวนของข้อมูลสอบถาม

$$MRR = \frac{\sum_{i=1}^n \frac{1}{rank_i}}{n} \quad (4.2)$$

อย่างไรก็ตาม ถ้าเพลงที่ต้องการนั้นไม่ปรากฏอยู่ในรายการเพลงผลลัพธ์ จะกำหนดให้ค่าลำดับส่วนกลับของข้อมูลดังกล่าวมีค่าเท่ากับ 0 นั่นคือ ถ้าเพลงที่ต้องการไม่เคยปรากฏอยู่ในรายการเพลงผลลัพธ์ ค่าเฉลี่ยลำดับส่วนกลับที่ได้ก็จะมีค่าเท่ากับ 0 ในทางกลับกัน ถ้าเพลงที่ต้องการปรากฏอยู่ในรายการเพลงผลลัพธ์ในลำดับที่ 1 เสมอ ค่าเฉลี่ยลำดับส่วนกลับก็จะมีค่าเท่ากับ 1 แสดงว่าถ้าค่าเฉลี่ยลำดับส่วนกลับมีค่าเข้าใกล้ 1 มากเท่าไร คุณภาพของคำตอบที่ได้ก็จะมีค่าสูงมากขึ้นเท่านั้น ซึ่งผลการทดลองเรื่องความแม่นยำที่ใช้ค่าเฉลี่ยลำดับส่วนกลับในการพิจารณานั้น จะสามารถแสดงได้ดังรูปที่ 4.6



รูปที่ 4.5 เปรียบเทียบความแม่นยำในการค้นหาเพลง 10 ลำดับแรก เมื่อความกว้างของเงื่อนไขบังคับโดยรวมแบบซอกโก-ชิบะมีขนาดต่าง ๆ

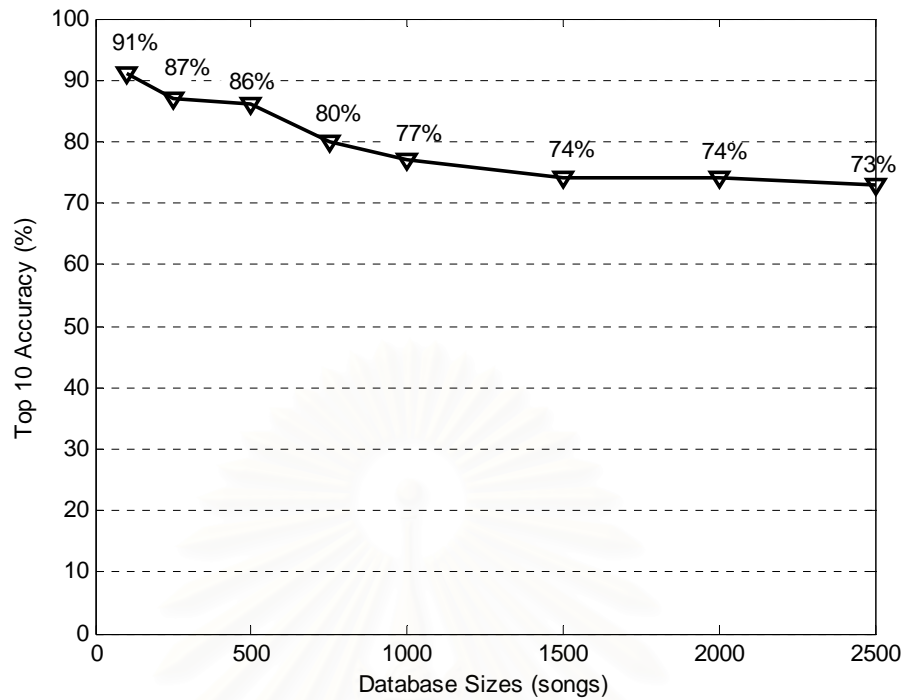


รูปที่ 4.6 เปรียบเทียบค่าเฉลี่ยลำดับส่วนกลับในการค้นหาเพลง เมื่อความกว้างของเงื่อนไข บังคับโดยรวมแบบซาโก-ชิบะมีขนาดต่าง ๆ

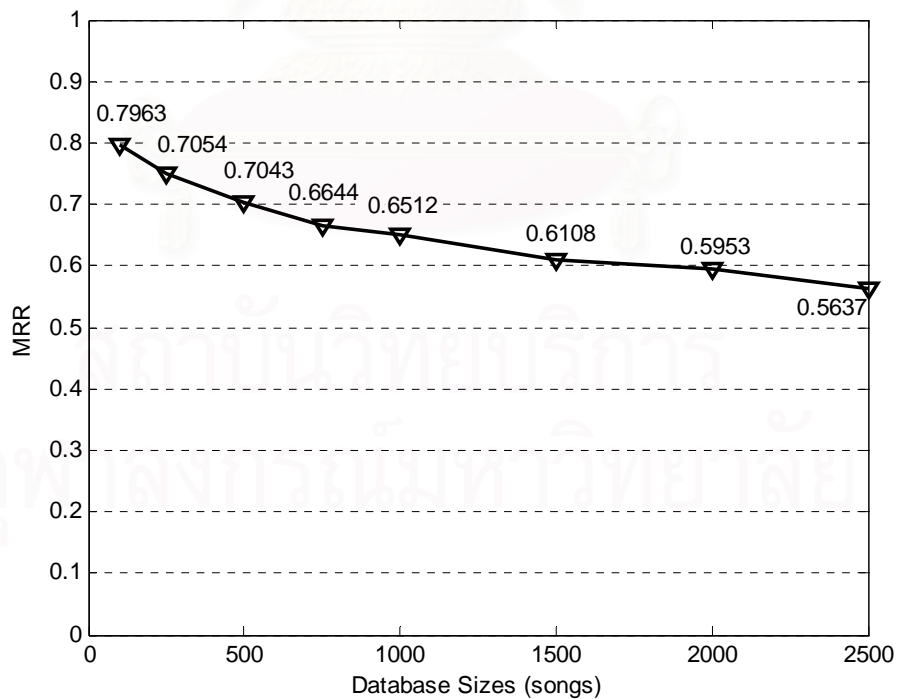
จากผลการทดลองในรูปที่ 4.5 จะได้ว่า เมื่อค่าความกว้างของเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะมีขนาดเท่ากับ 5% และ 10% ความแม่นยำในการค้นหาข้อมูลจะมีค่าสูงกว่าเมื่อความกว้างของเงื่อนไขบังคับดังกล่าวมีขนาดเท่ากับ 0% อย่างไรก็ตาม เมื่อค่าความกว้างของเงื่อนไขบังคับโดยรวมมีขนาดเท่ากับ 5% และ 10% นั้น ความแม่นยำในการค้นหาข้อมูลจะมีค่าเท่ากัน แต่เมื่อพิจารณาผลการทดลองในรูปที่ 4.6 ประกอบ ซึ่งจะเป็นการวัดคุณภาพของคำตอบที่ได้ จะเห็นได้ชัดว่าความกว้างของเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะที่มีขนาดเท่ากับ 5% คุณภาพของผลลัพธ์ในการค้นหาข้อมูลจะสูงกว่าเมื่อความกว้างของเงื่อนไขบังคับดังกล่าวมีขนาดเท่ากับ 10% ดังนั้นในงานวิจัยนี้จึงกำหนดให้ความกว้างของเงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะในการทดลองอื่น ๆ มีค่าเท่ากับ 5% ของความยาวข้อมูลสอบถาม

4.2.1.3 การวิเคราะห์ความแม่นยำของวิธีที่นำเสนอเมื่อฐานข้อมูลมีขนาดต่าง ๆ

ส่วนนี้จะเป็นการวิเคราะห์ความแม่นยำของวิธีการค้นหาเพลงโดยการร้องทำนองที่นำเสนอ เมื่อฐานข้อมูลมีขนาดต่าง ๆ ตั้งแต่ขนาด 100 เพลง ไปจนถึงขนาด 2,500 เพลง โดยจะมีการวัดความแม่นยำในการค้นหาเพลงโดยการร้องทำนองใน 10 ลำดับแรก ซึ่งผลการทดลองจะแสดงในรูปที่ 4.7 รวมทั้งจะมีการวัดคุณภาพของคำตอบที่ใช้ค่าเฉลี่ยลำดับส่วนกลับในการวัดผล ดังแสดงในรูปที่ 4.8



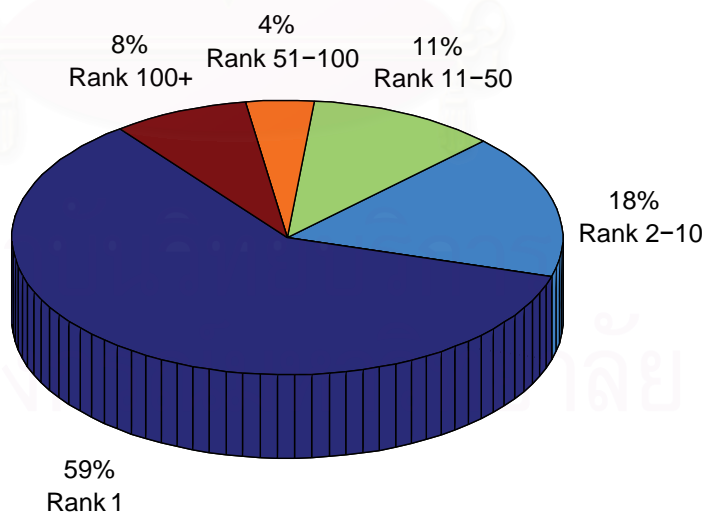
รูปที่ 4.7 เปรียบเทียบความแม่นยำในการค้นหาเพลง 10 ลำดับแรกเมื่อฐานข้อมูลมีขนาดต่าง ๆ



รูปที่ 4.8 เปรียบเทียบค่าเฉลี่ยลำดับส่วนกลับในการค้นหาเพลง เมื่อฐานข้อมูลมีขนาดต่าง ๆ

จากผลการทดลองในรูปที่ 4.7 และรูปที่ 4.8 จะเห็นได้ว่า เมื่อฐานข้อมูลมีขนาดใหญ่ขึ้น แนวโน้มทั้งความแม่นยำและคุณภาพของคำตอบจะมีทิศทางที่ลดลง อย่างไรก็ตาม อัตราในการลดลงของทั้งความแม่นยำและคุณภาพของคำตอบนั้นจะลดลงอย่างต่อเนื่อง จนกระทั่งในช่วงที่ฐานข้อมูลมีขนาด 1,500 เพลง ถึง 2,500 เพลง ความแม่นยำในการค้นหาข้อมูลแทบจะคงที่ นั่นคือ อยู่ในช่วง 74% ถึง 73% ซึ่งถ้าพิจารณาคุณภาพของคำตอบที่แสดงในรูปที่ 4.8 ประกอบ ก็จะได้เห็นว่าค่าเฉลี่ยลำดับส่วนกลับในการค้นหาข้อมูลนั้นก็ลดลงอยู่ในช่วงแคบ ๆ คือ 0.6108 ถึง 0.5637 นั่นแสดงว่า ถ้าฐานข้อมูลมีขนาดใหญ่ขึ้น ความแม่นยำของวิธีที่นำเสนอขึ้นมาก็จะมีแนวโน้มที่ลดลงในอัตราที่ลดลง ซึ่งสอดคล้องกับคุณภาพของคำตอบที่ได้ โดยคุณสมบัตินี้ถือเป็นคุณสมบัติที่ต้องการเป็นอย่างยิ่งในการพัฒนาวิธีการค้นหาเพลงโดยการร้องทำนอง รวมไปถึงระบบค้นหาข้อมูลแบบอื่น ๆ

ถ้าวิเคราะห์ลึกลงไปถึงสัดส่วนของลำดับเพลงที่ต้องการ ในการค้นหาข้อมูลภายใต้ฐานข้อมูลขนาด 1,000 เพลง ดังแสดงในรูปที่ 4.9 จะเห็นว่าวิธีค้นหาเพลงโดยการร้องทำนองที่นำเสนอขึ้นสามารถค้นหาเพลงได้อย่างถูกต้องแม่นยำและได้คำตอบที่มีคุณภาพสูงมาก โดยที่ 59% ของเสียงร้องทำนองที่ใช้ในการทดสอบทั้งหมด จะสามารถค้นหาเพลงที่ถูกต้องได้ในลำดับที่ 1 และ 18% ต่อมาจะสามารถค้นหาเพลงที่ถูกต้องในลำดับที่ 2 ถึง 10 นั่นคือวิธีการค้นหาเพลงโดยการร้องทำนองที่นำเสนอ สามารถค้นหาเพลงได้อย่างถูกต้องถึง 77% ภายใต้ฐานข้อมูลขนาด 1,000 เพลง และโดยส่วนใหญ่เพลงในอันดับแรกของรายการเพลงผลลัพธ์จะเป็นเพลงที่ต้องการ

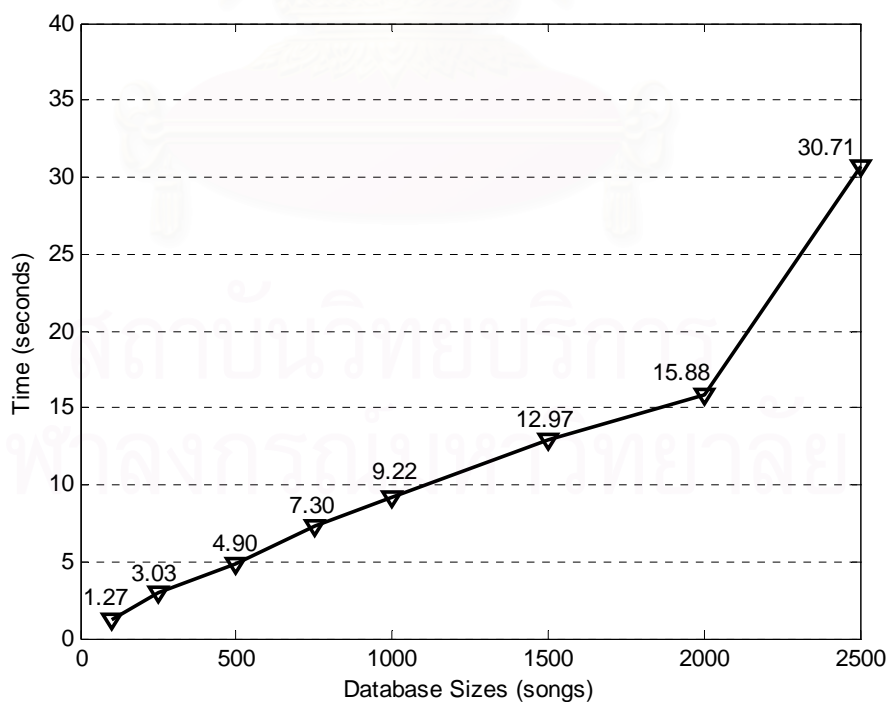


รูปที่ 4.9 สัดส่วนของลำดับเพลงที่ต้องการในการค้นหาข้อมูลภายใต้ฐานข้อมูลเพลงขนาด 1,000 เพลง

อย่างไรก็ตาม ถ้าวิธีที่นำเสนอ นั้นสามารถค้นหาเพลงจากเสียงร้องทำนองได้ อย่างแม่นยำ แต่ไม่สามารถทำงานได้อย่างรวดเร็ว วิธีที่นำเสนอ ก็อาจจะไม่สามารถนำไป ประยุกต์ใช้ได้จริงกับระบบค้นหาเพลงโดยการร้องทำนอง ดังนั้นความเร็วในการค้นหาข้อมูลจึง เป็นสิ่งสำคัญอีกประการหนึ่งของระบบค้นหาเพลงโดยการร้องทำนอง ซึ่งในงานวิจัยนี้ จะมีการ วิเคราะห์ความเร็วของวิธีที่นำเสนอ เมื่อฐานข้อมูลมีขนาดต่าง ๆ เพื่อแสดงให้เห็นว่าวิธีที่นำเสนอ นั้น สามารถค้นหาข้อมูลได้อย่างรวดเร็วภายใต้ฐานข้อมูลขนาดใหญ่

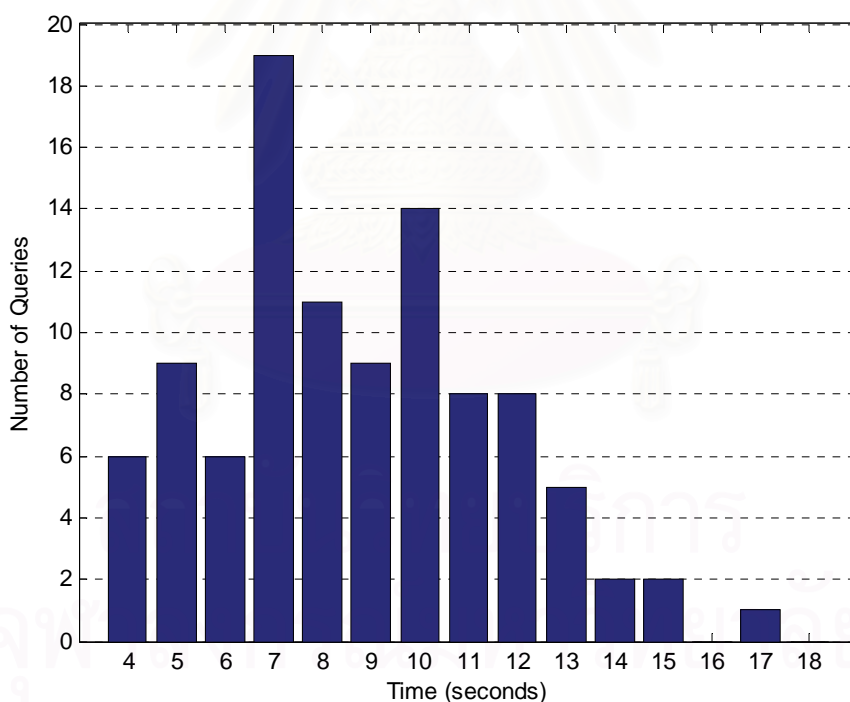
4.2.1.4 การวิเคราะห์ความเร็วของวิธีที่นำเสนอเมื่อฐานข้อมูลมีขนาดต่าง ๆ

ในการวิเคราะห์ความเร็วของวิธีที่นำเสนอ นั้น จะมีการนำแฟ้มข้อมูลเสียงร้อง ทำนองประมาณ 5 ถึง 10 แฟ้ม มาใช้ในการค้นหาเพลงก่อนทำการทดลอง เพื่อที่จะปรับ สภาพแวดล้อมที่ใช้ในการทดสอบให้คงที่ จากนั้นจึงจะเริ่มจับเวลาที่ใช้ในการค้นหาเพลงจาก แฟ้มข้อมูลเสียงร้องทำนองแต่ละแฟ้มและหาค่าเฉลี่ยของระยะเวลาดังกล่าว สุดท้ายจึงนำ ผลลัพธ์ที่ได้ไปวิเคราะห์ต่อไป อย่างไรก็ตาม แม้ว่าในระบบค้นหาเพลงโดยการร้องทำนองนั้น การค้นหาข้อมูลแต่ละครั้งจะต้องมีการสกัดคุณลักษณะเกี่ยวกับคอนทิวรัระดับเสียงออกจาก เสียงร้องทำนอง แต่ในงานวิจัยนี้ สนใจเฉพาะส่วนของการค้นหาเพลงเท่านั้น ดังนั้นระยะเวลาที่ ได้ จะวัดเฉพาะขั้นตอนการค้นหาข้อมูลเท่านั้น ผลการทดลองเกี่ยวกับการวิเคราะห์ความเร็ว ของวิธีที่นำเสนอเมื่อฐานข้อมูลมีขนาดต่าง ๆ นั้น สามารถแสดงได้ดังรูปที่ 4.10



รูปที่ 4.10 เปรียบเทียบความเร็วในการค้นหาเพลง เมื่อฐานข้อมูลมีขนาดต่าง ๆ

จากผลการทดลองจะเห็นได้ว่าสำหรับฐานข้อมูลขนาด 1,000 เพลง ซึ่งประกอบไปด้วยข้อมูลคอนทราต์ระดับเสียงของเพลงจำนวน 203,461 ข้อมูล วิธีการค้นหาเพลงโดยการร้องทำนองที่นำเสนอ นั้น สามารถค้นหาข้อมูลโดยใช้ระยะเวลาในการค้นหาข้อมูลเฉลี่ยเพียง 9.22 วินาที อย่างไรก็ตามแนวโน้มระยะเวลาที่ใช้ในการค้นหาเพลงนั้นจะมีค่าสูงขึ้นตามขนาดของฐานข้อมูลที่ใช้ โดยในช่วง 100 เพลงไปจนถึง 2,000 เพลง จะเห็นว่าแนวโน้มระยะเวลาที่ใช้ในการค้นหานั้น จะมีค่าต่ำกว่าฟังก์ชันเชิงเส้นกับขนาดของฐานข้อมูล ซึ่งอาจเป็นผลมาจากฟังก์ชันขอบเขตล่างที่นำเสนอ นั้น จะมีคุณสมบัติที่เมื่อฐานข้อมูลมีขนาดใหญ่ขึ้นความสามารถในการลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริงก็จะมีค่าสูงขึ้น อย่างไรก็ตาม เมื่อขนาดของฐานข้อมูลเพิ่มขึ้นไปจนถึง 2,500 เพลง ซึ่งประกอบไปด้วยข้อมูลอนุกรมเวลาจำนวน 472,349 ข้อมูล ระยะเวลาที่ใช้ในการค้นหากลับเพิ่มสูงขึ้นอย่างรวดเร็ว สำหรับประเด็นปัญหานี้ อาจเกิดมาจากฐานข้อมูลขนาด 2,500 เพลงนั้น ไม่สามารถที่จะจัดเก็บอยู่บนแรมได้อย่างครบถ้วนตั้งแต่ในขั้นตอนการปรับสภาพแวดล้อมของระบบ ทำให้จะต้องมีการเข้าถึงข้อมูลบนจานบันทึกแบบแข็งในกรณีที่ต้องการยังไม่ถูกจัดเก็บอยู่บนแรม ซึ่งจะส่งผลให้การค้นหาข้อมูลโดยรวมทั้งหมดช้าลง



รูปที่ 4.11 การกระจายตัวของระยะเวลาในการค้นหาเพลงจากเสียงร้องทำนองภายใต้ฐานข้อมูลขนาด 1,000 เพลง

อย่างไรก็ตาม ถ้ามีการวิเคราะห์ลึกลงไปถึงการกระจายตัวของระยะเวลาที่ใช้ในการค้นหาเพลงจากเสียงร้องทำนองภายใต้ฐานข้อมูลระดับ 1,000 เพลงดังแสดงในรูปที่ 4.11

จะเห็นได้ว่าเกินกึ่งหนึ่งของข้อมูลสอบถามจะใช้ระยะเวลาในการค้นหาข้อมูลไม่ถึง 10 วินาที และระยะเวลาในการค้นหาข้อมูลที่เร็วที่สุดจะอยู่ที่ 4.63 วินาที ส่วนระยะเวลาในการค้นหาข้อมูลที่นานที่สุดในการทดลองนี้จะอยู่ที่ 17.24 วินาที

จากการทดลองเพื่อวิเคราะห์ประสิทธิภาพของวิธีที่นำเสนอไปนั้น จะเห็นได้ว่าวิธีที่นำเสนอสามารถค้นหาข้อมูลเพลงได้อย่างถูกต้องแม่นยำและรวดเร็ว ภายใต้อาณัติข้อมูลขนาดใหญ่ อย่างไรก็ตาม เพื่อที่จะแสดงให้เห็นว่าวิธีที่นำเสนอสามารถค้นหาเพลงได้อย่างมีประสิทธิภาพเมื่อเทียบกับวิธีที่ใช้ในระบบค้นหาเพลงโดยการร้องทำนองโดยทั่วไป ดังนั้นในงานวิจัยนี้จึงจะมีการทำการทดลองเพื่อเปรียบเทียบความแม่นยำและความเร็วของวิธีที่นำเสนอกับวิธีการค้นหาแบบต่าง ๆ

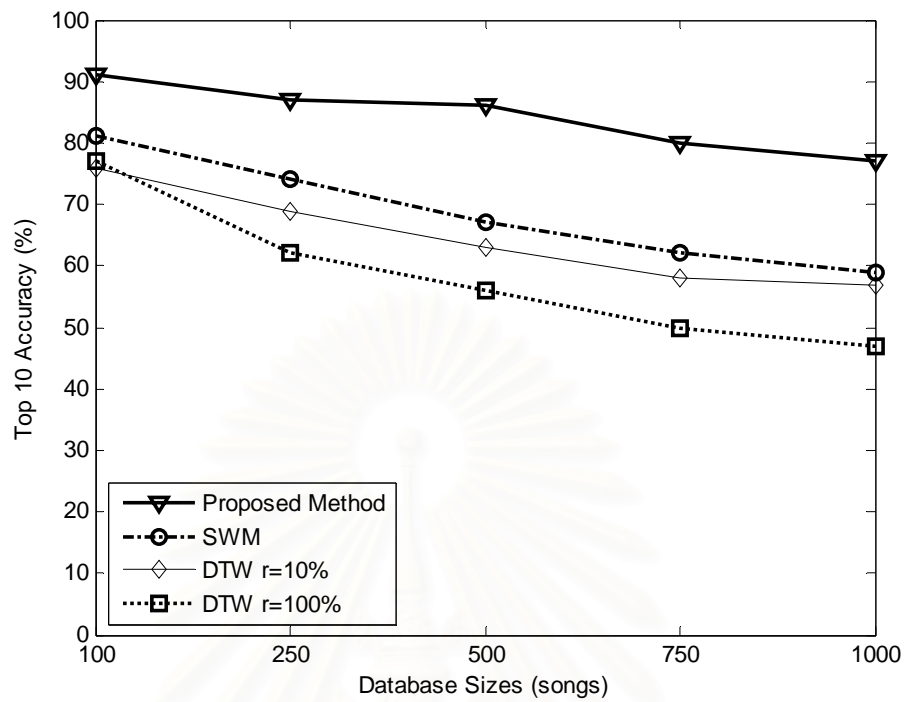
4.2.2 การทดลองเพื่อวิเคราะห์ความแม่นยำของวิธีที่นำเสนอเมื่อเปรียบเทียบกับวิธีอื่น ๆ

สำหรับการทดลองในเรื่องความแม่นยำของวิธีที่นำเสนอเมื่อเปรียบเทียบกับวิธีอื่น ๆ นั้น งานวิจัยนี้จะนำวิธีที่นำมาเปรียบเทียบกับวิธีไดนามิกโทมวอร์ปิง ซึ่งเป็นวิธีที่นิยมนำมาประยุกต์ใช้กับระบบค้นหาเพลงในการร้องทำนอง เนื่องจากมีความแม่นยำสูง [4, 6, 11, 28, 29] โดยวิธีไดนามิกโทมวอร์ปิงที่นำมาใช้ในการทดลองนั้น จะมีอยู่ด้วยกัน 2 แบบ คือวิธีไดนามิกโทมวอร์ปิงที่ไม่มีการนำเงื่อนไขบังคับโดยรวมมาใช้ในการคำนวณ และวิธีไดนามิกโทมวอร์ปิงที่ใช้เงื่อนไขบังคับโดยรวมแบบซาโก-ชิชะที่ขนาด 10% รวมทั้งนำวิธีปรับขนาดกับโทมวอร์ปิง [2] ซึ่งเป็นวิธีที่ช่วยให้วิธีไดนามิกโทมวอร์ปิงสามารถวัดระยะทางของข้อมูลที่มีการหดหรือยืดขนาดได้อย่างเหมาะสมมากยิ่งขึ้น สำหรับรายละเอียดของวิธีที่จะนำมาใช้ในการเปรียบเทียบนั้น จะสามารถแสดงได้ดังต่อไปนี้

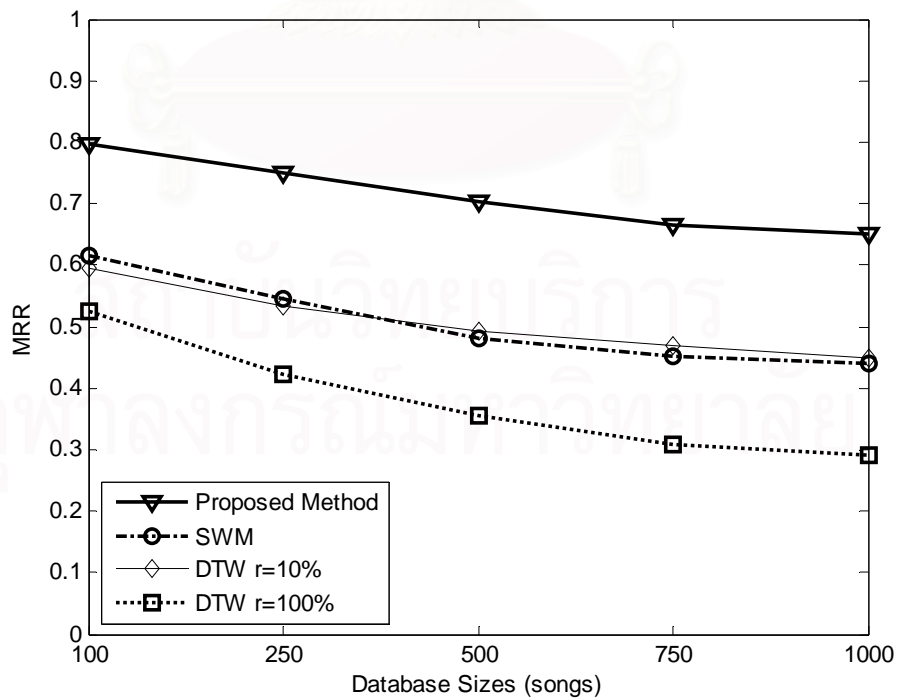
1. วิธีไดนามิกโทมวอร์ปิงที่ไม่มีการนำเงื่อนไขบังคับโดยรวมมาใช้ในการคำนวณ โดยในการนำวิธีดังกล่าวมาใช้นั้น จะใช้แนวคิดการค้นหาแบบการเปรียบเทียบข้อมูลทั้งลำดับ ซึ่งเป็นวิธีเดียวกับวิธีที่นำมาใช้ในงานวิจัยนี้ รวมทั้งมีการตัดแบ่งข้อมูลคอนทราสต์ระดับเสียงออกจากแฟ้มข้อมูลมิติ แปลงข้อมูลอนุกรมเวลาที่ได้ให้เป็นบรรทัดฐานด้วยค่าเฉลี่ย และการสกัดคุณลักษณะออกจากเสียงร้องทำนอง ซึ่งทั้งหมดนั้นจะเป็นวิธีเดียวกันกับวิธีที่นำเสนอ ทั้งนี้เพื่อที่จะสามารถเปรียบเทียบความแม่นยำของวิธีการต่าง ๆ ได้อย่างเหมาะสม รวมทั้งจะมีการเปรียบเทียบข้อมูลที่มีการหดและยืดขนาดที่ความยาว 0.5 เท่า 1 เท่า และ 2 เท่าตามลำดับ ซึ่งเป็นขั้นตอนที่มักจะนำมาประยุกต์ใช้ในงานวิจัยต่าง ๆ เพื่อให้วิธีไดนามิกโทมวอร์ปิงสามารถรองรับความแปรผันของผู้ใช้ได้ อย่างสมบูรณ์มากยิ่งขึ้น

2. วิธีไดนามิกไทม์วอร์ปิงที่ใช้เงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะ โดยกำหนดให้ความกว้างของเงื่อนไขข้อบังคับโดยรวมมีขนาดเท่ากับ 10% สำหรับการนำวิธีดังกล่าวมาใช้ในการทดลองนั้น จะมีการจัดเตรียมฐานข้อมูล การสกัดคุณลักษณะจากเสียงร้องทำนอง และการหดหรือยืดขนาดเช่นเดียวกันกับการทดลองที่ใช้วิธีไดนามิกไทม์วอร์ปิงที่ไม่มีการนำเงื่อนไขบังคับมาใช้ในการคำนวณ อย่างไรก็ตามในงานวิจัยของ Ratanamahatana และ Keogh [15] ได้แสดงให้เห็นว่าความกว้างของเงื่อนไขข้อบังคับโดยรวมแบบซาโก-ชิบะนั้น จะส่งผลกระทบต่อความแม่นยำในการค้นหา ดังนั้นการทดลองในงานวิจัยนี้ ได้มีการเปรียบเทียบความแม่นยำในการค้นหาข้อมูลเมื่อความกว้างของเงื่อนไขข้อบังคับดังกล่าวมีขนาดต่าง ๆ ทั้งนี้ เพื่อที่จะหาความกว้างของเงื่อนไขข้อบังคับโดยรวมที่จะให้ความแม่นยำในการค้นหาข้อมูลสูงสุด ซึ่งผลลัพธ์ที่ได้ก็คือความกว้างของเงื่อนไขข้อบังคับโดยรวมแบบซาโก-ชิบะที่ขนาด 10% จะให้ความแม่นยำสูงสุด ซึ่งสอดคล้องกับผลการทดลองในงานวิจัยก่อนหน้านี้ [10] รวมทั้งมีการนำฟังก์ชันขอบเขตล่างของวิธีไดนามิกไทม์วอร์ปิงมาใช้ในการค้นหาด้วย เพื่อที่จะสามารถค้นหาข้อมูลได้อย่างรวดเร็วมากยิ่งขึ้น
3. วิธีปรับขนาดกับไทม์วอร์ปิง วิธีดังกล่าวเป็นวิธีที่เสนอในงานวิจัยของ Fu และคนอื่น ๆ [2] ซึ่งเป็นวิธีที่ช่วยให้วิธีไดนามิกไทม์วอร์ปิงสามารถวัดระยะทางของข้อมูลที่มีการหดหรือยืดขนาดได้อย่างเหมาะสมมากยิ่งขึ้น อย่างไรก็ตามเนื่องจากงานวิจัยดังกล่าวไม่ได้มีวัตถุประสงค์เพื่อที่จะวิจัยระบบค้นหาเพลงโดยการร้องทำนองเป็นหลัก รวมทั้งแนวคิดในงานวิจัยนี้ เริ่มต้นมาจากงานวิจัยดังกล่าว ดังนั้นการกำหนดค่าพารามิเตอร์ต่าง ๆ สำหรับวิธีดังกล่าว จะใช้ค่าเดียวกันกับวิธีที่นำเสนอ นั่นคือ กำหนดให้ให้อัตราส่วนการหดและยืดขนาดสูงสุดเท่ากับ 0.6 ถึง 1.6 เท่า โดยจะมีการปรับอัตราส่วนการหดและยืดขนาดครั้งละ 0.2 เท่า ความกว้างของเงื่อนไขข้อบังคับโดยรวมแบบซาโก-ชิบะเท่ากับ 5% นอกจากนี้จะมีการแปลงข้อมูลอนุกรมเวลาในฐานข้อมูลดังกล่าวให้เป็นบรรทัดฐานด้วยค่าเฉลี่ย เพื่อที่จะให้วิธีปรับขนาดกับไทม์วอร์ปิงสามารถวัดค่าระยะทางได้อย่างถูกต้องแม่นยำมากยิ่งขึ้น รวมทั้งการนำฟังก์ชันขอบเขตล่างของวิธีดังกล่าวมาประยุกต์ใช้เพื่อให้สามารถค้นหาข้อมูลได้อย่างรวดเร็ว

เพื่อให้การเปรียบเทียบความแม่นยำของวิธีที่ต่าง ๆ สามารถทำได้อย่างครบถ้วน ดังนั้นในงานวิจัยนี้ จะมีการเปรียบเทียบความแม่นยำในการค้นหาเพลงที่ใกล้เคียงกับเสียงร้องทำนองมากที่สุด 10 อันดับแรก และเปรียบเทียบคุณภาพของคำตอบที่ได้จากการค้นหาด้วยค่าเฉลี่ยลำดับส่วนกลับ โดยผลการทดลองดังกล่าวสามารถแสดงได้ดังรูปที่ 4.12 และรูปที่ 4.13 ตามลำดับ



รูปที่ 4.12 เปรียบเทียบความแม่นยำในการค้นหาเพลงของวิธีการค้นหาแบบต่าง ๆ



รูปที่ 4.13 เปรียบเทียบค่าเฉลี่ยลำดับส่วนกลับในการค้นหาเพลงของวิธีการค้นหาแบบต่าง ๆ

จากผลการทดลองในรูปที่ 4.12 และรูปที่ 4.13 จะเห็นได้ชัดว่าวิธีที่นำเสนอ นั้นสามารถค้นหาข้อมูลได้อย่างถูกต้องแม่นยำมากกว่าวิธีต่าง ๆ ที่เคยนำมาประยุกต์ใช้กับระบบค้นหาเพลงโดยการร้องทำนองทั้งหมด ตัวอย่างเช่น ภายใต้ฐานข้อมูลเพลงขนาด 1,000 เพลง วิธีที่นำเสนอ นั้นสามารถค้นหาเพลงได้ถูกต้องถึง 77% เมื่อเปรียบเทียบกับวิธีปรับขนาดกับไทม์วอร์ปิงที่สามารถค้นหาเพลงได้ถูกต้องเพียง 59% ซึ่งเป็นวิธีที่มีความแม่นยำรองลงมา ถัดมาจะเป็นวิธีไดนามิกไทม์วอร์ปิงที่ใช้เงื่อนไขบังคับโดยรวมแบบซาโก-ซิเบที่ขนาด 10% ซึ่งมีความถูกต้อง 57% และวิธีที่มีความแม่นยำน้อยที่สุดในการทดลองนี้ก็คือ วิธีไดนามิกไทม์วอร์ปิงที่ไม่ใช้เงื่อนไขบังคับโดยรวมในการคำนวณ ซึ่งจะมีความถูกต้องอยู่ที่ 47% และเมื่อพิจารณาในส่วนของคุณภาพของคำตอบที่ได้นั้น วิธีที่นำเสนอ ก็ยังคงให้คำตอบที่มีคุณภาพสูงสุด คือมีค่าเฉลี่ยลำดับส่วนกลับอยู่ที่ 0.6512 และวิธีที่มีคุณภาพรองลงมาก็คือวิธีไดนามิกไทม์วอร์ปิงที่ใช้เงื่อนไขบังคับที่ขนาด 10% ซึ่งจะมีค่าเฉลี่ยลำดับส่วนกลับอยู่ที่ 0.4494 ซึ่งจะมีคุณภาพใกล้เคียงกับวิธีปรับขนาดกับไทม์วอร์ปิงที่มีค่าเฉลี่ยลำดับส่วนกลับอยู่ที่ 0.4390 และวิธีที่ให้คำตอบที่มีคุณภาพต่ำที่สุดก็คือ วิธีไดนามิกไทม์วอร์ปิงที่ไม่ใช้เงื่อนไขบังคับในการคำนวณหาค่าระยะทาง ซึ่งจะมีค่าเฉลี่ยลำดับส่วนกลับอยู่ที่ 0.2923

แม้ว่าวิธีการที่นำเสนอ นั้นจะสามารถค้นหาข้อมูลได้อย่างถูกต้องแม่นยำมากที่สุด แต่ถ้าพิจารณาแนวโน้มความแม่นยำของวิธีต่าง ๆ นั้น จะเห็นได้ว่าทุกวิธี มีแนวโน้มที่จะค้นหาข้อมูลได้ถูกต้องแม่นยำลดลง เมื่อฐานข้อมูลมีขนาดใหญ่ขึ้น ซึ่งอัตราในการลดลงดังกล่าวจะมีค่าใกล้เคียงกันหมดสำหรับทุกวิธีที่นำมาใช้ในการเปรียบเทียบ นั้นแสดงว่า ถ้ามีการทดลองกับฐานข้อมูลที่มีขนาดใหญ่ขึ้น วิธีที่นำเสนอ นั้นจะยังคงมีความแม่นยำในการค้นหาข้อมูลสูงที่สุด

สำหรับเหตุผลหลัก ๆ ที่ทำให้วิธีที่นำเสนอ นั้นสามารถค้นหาข้อมูลได้ถูกต้องแม่นยำกว่าวิธีอื่น ๆ นั้นก็คือ การหดหรือยืดขนาดก่อนการคำนวณระยะทาง และการแปลงข้อมูลให้เป็นบรรทัดฐานทุกครั้งที่มีการปรับขนาด ซึ่งจากการทดลองจะเห็นว่าวิธีที่นำเสนอ ซึ่งเป็นวิธีที่มีการหดและยืดขนาดก่อนการคำนวณนั้นจะให้ค่าความแม่นยำสูงกว่าวิธีไดนามิกไทม์วอร์ปิงที่มีการหดและยืดขนาดเพียง 3 ระดับ ซึ่งการหดและยืดขนาดนี้จะช่วยให้ข้อมูลถูกหดหรือยืดขนาดไปในอัตราส่วนที่เหมาะสมก่อนที่จะคำนวณค่าระยะทาง ทำให้ค่าระยะทางระหว่างข้อมูลสอบถามกับข้อมูลที่มีลักษณะใกล้เคียงกับข้อมูลสอบถามนั้นมากที่สุดมีค่าใกล้เคียงกันมากขึ้น อย่างไรก็ตาม แม้ว่าวิธีปรับขนาดกับไทม์วอร์ปิงจะมีการปรับขนาดเอกรูปที่ระดับต่าง ๆ เช่นกัน แต่การที่ไม่ได้ตระหนักถึงผลกระทบของการปรับขนาดว่า เมื่อมีการหดหรือยืดขนาดของข้อมูลออกไป ข้อมูลที่จะใช้ในการเปรียบเทียบนั้นอาจจะไม่อยู่ในระดับเดียวกันกับข้อมูลสอบถาม ทำให้เมื่อวัดค่าระยะทางระหว่างข้อมูลทั้งสอง ค่าระยะทางที่ได้จะมีค่ามาก ซึ่งหมายถึงข้อมูลทั้งสองมีความแตกต่างกันมากนั่นเอง ดังนั้นการแปลงข้อมูลให้เป็นบรรทัดฐานจึงมีความสำคัญอย่างยิ่ง ดังแสดงในผลการทดลองจะเห็นได้ว่า ทั้ง ๆ ที่วิธีที่นำเสนอ กับวิธีปรับ

ขนาดกับไทม์วอร์ปิง จริง ๆ แล้วจะแตกต่างกันส่วนของการแปลงข้อมูลให้เป็นบรรทัดฐาน แต่วิธีที่นำเสนอ นั้น กลับสามารถค้นหาข้อมูลได้แม่นยำมากกว่าถึง 18% ภายใต้ฐานข้อมูลขนาด 1,000 เพลง

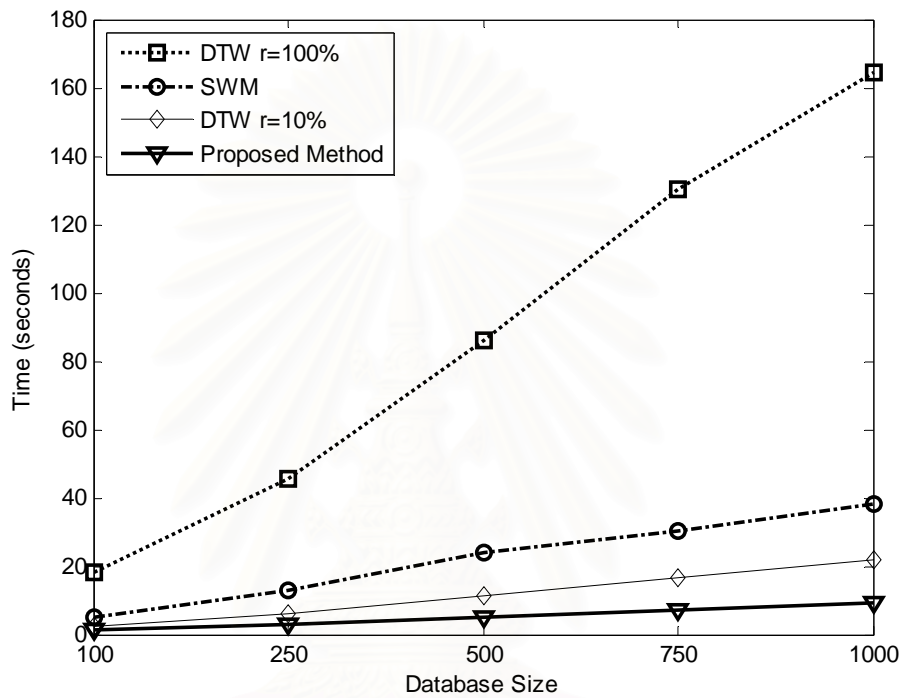
อย่างไรก็ตาม มีข้อสังเกตว่าวิธีไดนามิกไทม์วอร์ปิงที่ไม่มีการนำเงื่อนไขข้อบังคับโดยรวมมาใช้ในการคำนวณ จะเป็นวิธีที่ใกล้เคียงกับวิธีที่ใช้เปรียบเทียบในงานวิจัยของ Dannenberg และคนอื่น ๆ [4] ที่แสดงให้เห็นว่าวิธีไดนามิกไทม์วอร์ปิงนั้น มีความแม่นยำสูงกว่าวิธีการเปรียบเทียบสายอักขระแบบทั่วไป รวมทั้งวิธีเปรียบเทียบสายอักขระที่ใช้แบบจำลองฮิดเดนมาร์คอฟ ซึ่งจากผลการทดลองจะเห็นว่าวิธีที่นำเสนอ นั้นสามารถค้นหาข้อมูลได้แม่นยำมากกว่าวิธีดังกล่าวถึง 30% ภายใต้ฐานข้อมูลขนาด 1,000 เพลง ดังนั้น จึงสามารถกล่าวเป็นนัยได้ว่า วิธีที่นำเสนอ นั้น สามารถค้นหาข้อมูลได้อย่างแม่นยำมากกว่าวิธีการเปรียบเทียบสายอักขระทั้งสองด้วย

4.2.3 การทดลองเพื่อวิเคราะห์ความเร็วในการค้นหาเพลงเมื่อเปรียบเทียบกับวิธีอื่น ๆ

แม้ว่าวิธีที่นำเสนอ นั้นจะสามารถค้นหาข้อมูลได้อย่างแม่นยำ แต่ถ้าความแม่นยำที่เพิ่มขึ้น กลับสร้างภาระในการคำนวณอย่างมหาศาล วิธีที่นำเสนอ ก็อาจจะไม่สามารถนำไปประยุกต์ใช้กับระบบค้นหาเพลงโดยการร้องทำนองได้จริง สำหรับการทดลองเพื่อวิเคราะห์ความเร็วในการค้นหาเพลงเมื่อเปรียบเทียบกับวิธีอื่น ๆ นี้ มีจุดประสงค์เพื่อที่จะแสดงให้เห็นว่าวิธีที่นำเสนอ ซึ่งจะเป็นการนำฟังก์ชันขอบเขตล่างที่เสนอในงานวิจัยนี้ ไปประยุกต์ใช้กับวิธีการค้นหาข้อมูลที่มีประสิทธิภาพ แล้วจะสามารถค้นหาข้อมูลได้อย่างรวดเร็ว เมื่อเปรียบเทียบกับวิธีค้นหาข้อมูลโดยทั่วไป สำหรับวิธีที่จะนำมาใช้ในการเปรียบเทียบนั้นจะเป็นวิธีเดียวกันกับวิธีที่ใช้เปรียบเทียบในการทดลองเรื่องความแม่นยำที่ผ่านมา โดยในขั้นตอนการทดลองนั้น จะมีการนำข้อมูลสอบถามจำนวนประมาณ 5 ถึง 10 ข้อมูล มาใช้ในการค้นหาข้อมูลก่อนการจับเวลาในการทดลอง เพื่อที่จะปรับสภาพแวดล้อมในการทดลองให้คงที่ โดยผลการทดลองนั้น สามารถแสดงได้ดังรูปที่ 4.14

จากผลการทดลองในรูปที่ 4.14 จะเห็นว่าวิธีที่นำเสนอ นั้นสามารถค้นหาข้อมูลได้อย่างรวดเร็วภายใต้ฐานข้อมูลขนาดใหญ่ แม้ว่าภายใต้ฐานข้อมูลขนาดถึง 1,000 เพลง ที่ประกอบไปด้วยข้อมูลอนุกรมเวลาจำนวนทั้งสิ้น 203,461 ข้อมูล ซึ่งถ้าคำนวณเป็นจำนวนครั้งที่จะต้องนำมาใช้หาค่าระยะทางด้วยวิธีไดนามิกไทม์วอร์ปิงก็จะเท่ากับ 1,220,766 ครั้ง แต่เมื่อใช้ฟังก์ชันขอบเขตล่างที่นำเสนอร่วมกับวิธีการค้นหาที่มีประสิทธิภาพที่ได้รับการปรับปรุงให้เหมาะสมกับฟังก์ชันขอบเขตล่างที่นำเสนอแล้ว วิธีที่นำเสนอ ก็จะสามารถค้นหาเพลงโดยใช้เวลาเพียง 9.22 วินาที ซึ่งเมื่อเปรียบเทียบกับวิธีไดนามิกไทม์วอร์ปิงที่ไม่มีการนำเงื่อนไขข้อบังคับมาใช้ในการค้นหาที่ใช้เวลาถึง 164.4 วินาที ซึ่งวิธีดังกล่าวไม่มีการนำฟังก์ชันขอบเขตล่าง

มาประยุกต์ใช้ในการค้นหา หรือแม้แต่วิธีปรับขนาดกับโทมวอร์ปิงที่ใช้เวลาค้นหาเท่ากับ 38.03 วินาที ซึ่งวิธีดังกล่าวเป็นวิธีที่มีลักษณะใกล้เคียงกับงานวิจัยที่เสนอมากที่สุด แต่ไม่ได้มีการนำวิธีการค้นหาที่มีประสิทธิภาพมาประยุกต์ใช้ ทำให้ความเร็วในการค้นหาข้อมูลนั้นจะช้ากว่าวิธีที่นำเสนอ และวิธีไดนามิกโทมวอร์ปิงที่ใช้เงื่อนไขบังคับโดยรวมแบบซาโก-ชิบะที่ขนาด 10% จะใช้เวลาในการค้นหาข้อมูลเท่ากับ 21.74 วินาที



รูปที่ 4.14 เปรียบเทียบระยะเวลาในการค้นหาเพลงของวิธีการค้นหาแบบต่าง ๆ

ซึ่งจากผลการทดลองนี้ พอจะสรุปได้ว่าฟังก์ชันขอบเขตล่างที่นำเสนอ นั้นสามารถที่จะนำมาประยุกต์ใช้กับวิธีค้นหาข้อมูลที่มีประสิทธิภาพสูง แล้วจะช่วยให้สามารถค้นหาข้อมูลได้อย่างรวดเร็วภายใต้ฐานข้อมูลขนาดใหญ่ แม้ว่าจะมีการหดและยืดขนาดถึง 6 ระดับด้วยกัน

จากผลการทดลองต่าง ๆ ที่ได้นำเสนอไปนั้น จะเห็นได้ว่าวิธีค้นหาเพลงโดยการร้องทำนองที่นำเสนอ สามารถค้นหาเพลงจากเสียงร้องทำนองได้อย่างแม่นยำและรวดเร็ว ซึ่งในเรื่องความแม่นยำนั้น วิธีที่นำเสนอจะมีความแม่นยำสูงกว่าวิธีอื่น ๆ ที่นำมาใช้ในการทดลองทั้งหมด ซึ่งแต่ละวิธีล้วนเป็นวิธีที่ได้รับการยอมรับจากนักวิจัยเป็นจำนวนมากว่ามีความแม่นยำสูง รวมทั้งในเรื่องความเร็วในการค้นหา จากผลการทดลองจะเห็นได้ว่า วิธีนำเสนอ นั้นสามารถค้นหาข้อมูลได้อย่างรวดเร็วภายใต้ฐานข้อมูลที่มีขนาดใหญ่

บทที่ 5

สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

งานวิจัยนี้ได้นำเสนอวิธีการค้นหาเพลงโดยการร้องทำนองที่มีการประยุกต์ใช้เทคนิค การปรับขนาดเอกรูป ไดนามิกโหม่วอร์ปปีง และฟังก์ชันขอบเขตล่างเข้าไว้ด้วยกัน เพื่อที่จะให้วิธีการที่นำเสนอ สามารถค้นหาเพลงได้อย่างถูกต้องแม่นยำและรวดเร็ว โดยได้มีการทำการทดลองและวิเคราะห์ผลไว้อย่างละเอียดดังที่นำเสนอในบทที่ 4 ซึ่งจะเห็นได้ว่าวิธีที่นำเสนอ นั้นสามารถค้นหาข้อมูลได้อย่างแม่นยำและรวดเร็ว โดยผลจากการวิจัยทั้งหมดที่ได้ นำเสนอไปนั้น สามารถสรุปได้ดังนี้

5.1 สรุปผลการวิจัย

1. วิธีปรับขนาดเอกรูปและไดนามิกโหม่วอร์ปปีงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน สามารถที่จะนำมาประยุกต์ใช้ในระบบค้นหาเพลงโดยการร้องทำนอง และช่วยให้การค้นหาเพลงจากเสียงร้องทำนองนั้นมีความแม่นยำสูงถึง 77% ภายใต้ฐานข้อมูลขนาด 1,000 เพลง รวมทั้งคุณภาพของรายการเพลงผลลัพธ์ที่ได้ก็จะมีคุณภาพสูง ซึ่งส่วนใหญ่เพลงที่อยู่ในลำดับที่ 1 ของรายการเพลงผลลัพธ์ จะเป็นเพลงที่ต้องการ โดยทั้งความแม่นยำในการค้นหาเพลงและคุณภาพของรายการเพลงผลลัพธ์ที่ได้จะมีค่าสูงกว่าวิธีอื่น ๆ ที่นำมาใช้ในการเปรียบเทียบทั้งหมด

2. ฟังก์ชันขอบเขตล่างที่นำเสนอ สามารถลดปริมาณข้อมูลที่จะต้องคำนวณค่าระยะทางจริงได้เป็นจำนวนมาก ทำให้มีข้อมูลเพียงจำนวนน้อยเท่านั้นที่จะต้องคำนวณค่าระยะทางจริงด้วยวิธีปรับขนาดเอกรูปและไดนามิกโหม่วอร์ปปีงที่รองรับการแปลงข้อมูลให้เป็นบรรทัดฐาน โดยฟังก์ชันขอบเขตล่างที่นำเสนอ นั้น จะเป็นปัจจัยหลักที่จะช่วยให้กระบวนการเปรียบเทียบความคล้ายคลึงกันของข้อมูลสามารถทำงานได้อย่างรวดเร็ว

3. ฟังก์ชันขอบเขตล่างที่นำเสนอสามารถที่จะนำไปประยุกต์ใช้กับวิธีค้นหาข้อมูลที่มีประสิทธิภาพได้ ซึ่งจะช่วยให้สามารถค้นหาข้อมูลได้อย่างรวดเร็วภายใต้ฐานข้อมูลขนาดใหญ่ แม้ว่าจะมีการหดและยืดขนาดของข้อมูลที่อัตราส่วนต่าง ๆ โดยจากผลการทดลองจะเห็นได้ว่า ภายใต้ฐานข้อมูลขนาด 1,000 เพลง วิธีที่นำเสนอ นั้นสามารถค้นหาข้อมูลได้อย่างรวดเร็ว โดยใช้ระยะเวลาในการค้นหาข้อมูลเฉลี่ยเพียง 9.22 วินาที

5.2 ข้อเสนอแนะ

จากวิธีที่นำเสนอ นั้นจะเห็นว่า แม้ว่าวิธีที่นำเสนอจะสามารถค้นหาข้อมูลในระดับ 1,000 เพลงได้อย่างรวดเร็ว แต่เมื่อฐานข้อมูลมีขนาดใหญ่ขึ้นจนถึงในระดับ 2,500 เพลง ซึ่งเป็นขนาดของฐานข้อมูลที่ใหญ่ที่สุดที่ได้มีการทดลอง จะใช้ระยะเวลาในการค้นหาข้อมูลเฉลี่ยถึง 30.71 วินาที และมีแนวโน้มที่จะใช้ระยะเวลาในการค้นหาข้อมูลมากขึ้นเมื่อฐานข้อมูลมีขนาดใหญ่ขึ้น อย่างไรก็ตามถ้าต้องการที่จะค้นหาข้อมูลในระดับ 10,000 เพลงขึ้นไป วิธีที่นำเสนอทั้งหมดนั้น อาจจะไม่สามารถค้นหาเพลงได้อย่างรวดเร็ว อย่างไรก็ตาม มีข้อสังเกตว่าฐานข้อมูลที่นำมาใช้ในการทดลองนั้น แม้ว่าจะมีขนาด 1,000 เพลง แต่ปริมาณข้อมูลที่จะใช้ในการเปรียบเทียบจริงกลับสูงถึง 203,461 ข้อมูล นั้นแสดงว่าใน 1 เพลง จะมีการสกัดคุณลักษณะต่าง ๆ ของเพลงนั้นออกมาถึงประมาณ 200 คุณลักษณะ ซึ่งถือว่าปริมาณข้อมูลที่ใช้ในงานวิจัยนี้มากกว่าปริมาณข้อมูลที่ใช้งานวิจัยอื่น ๆ มาก เนื่องจากในงานวิจัยอื่น ๆ มักจะสกัดคุณลักษณะเฉพาะของท่อนสร้อยในเพลงออกมาเท่านั้น ดังตัวอย่างเช่น งานวิจัยของ Dannenberg และคนอื่น ๆ [4, 7] จะมีการทดสอบกับข้อมูลสองชุด โดยข้อมูลในชุดแรกฐานข้อมูลเพลงที่ใช้จะมีขนาด 258 เพลง ซึ่งจะมีการสกัดคุณลักษณะของเพลงเฉพาะในท่อนสร้อยออกมาได้ 2,844 คุณลักษณะ และในข้อมูลชุดที่สอง ฐานข้อมูลเพลงจะมีขนาด 868 เพลง ซึ่งจะมีการสกัดคุณลักษณะออกมาได้ 8,926 คุณลักษณะ ซึ่งจะเห็นได้ชัดว่าข้อมูลเพลง 1 เพลงที่ใช้ใน การทดลองของงานวิจัยนี้ถึง 20 เท่า และในงานวิจัยของ Wang [28] ฐานข้อมูลเพลงจะมีขนาด 1,400 เพลง และสกัดคุณลักษณะออกมาจำนวน 40,891 คุณลักษณะ ซึ่งจะเห็นได้ชัดว่าขนาดของฐานข้อมูลที่ใช้ในงานวิจัยดังกล่าวแม้ว่าจะมีจำนวนเพลงมากกว่า 1,000 เพลง แต่จำนวนคุณลักษณะที่ใช้ในการเปรียบเทียบนั้น กลับน้อยกว่าคุณลักษณะในฐานข้อมูลขนาด 1,000 เพลงของงานวิจัยนี้ถึง 5 เท่า ดังนั้นถ้าในงานวิจัยนี้มีการทดลองโดยใช้การสกัดคุณลักษณะของเพลงเฉพาะท่อนสร้อยดังเช่นในงานวิจัยอื่น ๆ วิธีที่ค้นหาเพลงโดยการร้องทำนองที่นำเสนอไปนั้น ก็น่าจะสามารถค้นหาข้อมูลเพลงจากเสียงร้องทำนองภายใต้ฐานข้อมูลในระดับสูงกว่า 10,000 เพลงได้อย่างแม่นยำและรวดเร็ว อย่างไรก็ตามการสกัดคุณลักษณะเฉพาะท่อนสร้อยของเพลงออกมาเพื่อใช้ในการเปรียบเทียบนั้น อาจทำให้ความแม่นยำในการค้นหาเพลงลดลง เนื่องจากผู้ใช้ไม่จำเป็นที่จะต้องร้องทำนองในท่อนสร้อยเท่านั้น ในบางเพลงผู้ใช้อาจจะร้องทำนองในช่วงต้นหรือช่วงท้ายของเพลงก็เป็นได้

นอกจากนี้งานวิจัยนี้ จะกำหนดให้ผู้ใช้ต้องร้องทำนองเป็นระยะเวลา 7-8 วินาที แล้วจึงนำข้อมูลดังกล่าวมาตัดแบ่งให้ได้ขนาดที่ต้องการคือประมาณ 6 วินาที แต่ถ้ามีการกำหนดให้ผู้ใช้ร้องต้องร้องทำนองนานขึ้น ความแม่นยำในการค้นหาข้อมูลก็จะมีแนวโน้มที่สูงขึ้น เนื่องจากเมื่อผู้ใช้ร้องทำนองนานขึ้น ข้อมูลที่ได้ก็จะมีคุณลักษณะต่าง ๆ มากขึ้น ทำให้ในการ

ค้นหาข้อมูลนั้นก็จะสามารถค้นหาข้อมูลได้อย่างถูกต้องมากยิ่งขึ้น ซึ่งในงานวิจัย [4, 7] จะกำหนดให้ผู้ใช้ร้องทำนองเพลงเฉลี่ย 9 ถึง 10 วินาที อย่างไรก็ตามวิธีที่นำเสนอโน้มนั้นยังมีข้อจำกัดในเรื่องระยะเวลาในการร้องทำนอง นั่นคือ ผู้ใช้จะต้องร้องทำนองในระยะเวลาที่กำหนดเท่านั้น ทั้งนี้ก็เพื่อที่จะสามารถค้นหาข้อมูลได้อย่างรวดเร็ว

สำหรับการนำวิธีที่เสนอในงานวิจัยอื่นมาประยุกต์ใช้กับวิธีที่นำเสนอในงานวิจัยนี้สามารถทำได้ เช่น การนำวิธีจับคู่ข้อมูลอนุกรมเวลาโดยใช้ตัวกรองหลายระดับ (Time Series Matching : a Multi-filter Approach) ที่เสนอในงานวิจัยของ Wang [28] ซึ่งเป็นการนำค่าทางสถิติต่าง ๆ มาใช้เป็นตัวกรองก่อนที่จะวัดค่าระยะทางด้วยวิธีไดนามิกไทม์วอร์ปิง อันจะช่วยให้การค้นหาข้อมูลสามารถทำงานได้อย่างรวดเร็วมากยิ่งขึ้น รวมทั้งการนำวิธีค้นหาข้อมูลแบบมีประสิทธิภาพอื่น ๆ เช่น วิธีค้นหาข้อมูลแบบดัชนี ที่รองรับข้อมูลที่มีจำนวนมิติสูง ๆ ได้

นอกจากนี้วิธีที่นำเสนอโน้มนั้น มีความเป็นไปได้ที่จะนำไปประยุกต์ใช้กับงานวิจัยในแขนงอื่น ๆ ได้ [17] เช่น การค้นหาข้อมูลภาพเคลื่อนไหว การค้นหาข้อมูลยีน เป็นต้น

รายการอ้างอิง

- [1] Ghias, A., Logan, J., Chamberlin, D., and Smith, B.C. (1995). Query by humming: musical information retrieval in an audio database. Proceedings of 3rd ACM international conference on Multimedia, pp. 231–236. San Francisco, CA, USA: ACM Press.
- [2] Fu, A.W.-c., Keogh, E., Lau, L.Y.H., and Ratanamahatana, C.A. (2005). Scaling and time warping in time series querying. Proceedings of 31st international conference on Very large data bases, pp. 649–660. Trondheim, Norway: VLDB Endowment.
- [3] Sakurai, Y., Yoshikawa, M., and Faloutsos, C. (2005). FTW: Fast Similarity Search under the Time Warping Distance. Proceedings of 24th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 326–337. Baltimore, MA, USA: ACM Press.
- [4] Dannenberg, R.B., Birmingham, W.P., Pardo, B., Hu, N., Meek, C., and Tzanetakis, G. (2007). A comparative evaluation of search techniques for query-by-humming using the MUSART testbed. Journal of the American Society for Information Science and Technology 58: pp. 687–701.
- [5] Shu, S., Narayanan, S., and Kuo, C.-C.J. (2006). Efficient Rotation Invariant Retrieval of Shapes using Dynamic Time Warping with Applications in Medical Databases. Proceedings of 19th IEEE International Symposium on Computer-Based Medical Systems (CBMS), pp. 673–678.
- [6] Lee, H.-R., Chen, C., and Jang, J.-S.R. (2005). Approximate lower-bounding functions for the speedup of DTW for melody recognition. International Workshop on Cellular Neural Networks and Their Applications, pp. 178–181.
- [7] Dannenberg, R.B., Birmingham, W.P., Tzanetakis, G.P., Meek, C.P., Hu, N.P., and Pardo, B.P. (2004). The MUSART Testbed for Query-by-Humming Evaluation. Computer Music Journal 28: pp. 34–48.
- [8] Hu, N., and Dannenberg, R.B. (2002). A comparison of melodic database retrieval techniques using sung queries. Proceedings of 2nd ACM/IEEE-CS joint conference on Digital libraries, pp. 301–307. Portland, OR, USA: ACM Press.

- [9] Jang, J.-S.R., and Lee, H.-R. (2001). Hierarchical filtering method for content-based music retrieval via acoustic input. Proceedings of 9th ACM international conference on Multimedia, pp. 401–410. Ottawa, Canada: ACM Press.
- [10] Zhu, Y., and Shasha, D. (2003). Warping indexes with envelope transforms for query by humming. Proceedings of the 2003 ACM SIGMOD international conference on Management of data, pp. 181–192. San Diego, CA, USA: ACM Press.
- [11] Adams, N., Marquez, D., and Wakefield, G. (2005). Iterative Deepening for Melody Alignment and Retrieval. Proceedings of 6th International Conference on Music Information Retrieval, pp. 199–206. London, UK.
- [12] Weisstein, E. (2005). Autocorrelation [Online]. MathWorld. Available from: <http://mathworld.wolfram.com/Autocorrelation.html> [3 August 2007]
- [13] Weisstein, E. (2004). Fourier transform [Online]. MathWorld. Available from: <http://mathworld.wolfram.com/FourierTransform.html> [3 August 2007]
- [14] 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.
- [15] Ratanamahatana, C.A., and Keogh, E.J. (2004). Everything you know about dynamic time warping is wrong. 3rd Workshop on Mining Temporal and Sequential Data, in conjunction with 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004), Seattle, WA, USA.
- [16] Keogh, E., and Ratanamahatana, C.A. (2005). Exact indexing of dynamic time warping. Knowledge and Information Systems (KAIS) 7: pp. 358–386.
- [17] Keogh, E. (2003). Efficiently Finding Arbitrarily Scaled Patterns in Massive Time Series Databases. European conference on principles and practice of knowledge discovery in databases (PKDD 2003 : knowledge discovery in databases), pp. 253–265. Cavtat-Dubrovnik , Croatia: Springer Berlin / Heidelberg.
- [18] Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S. (2001). Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. Knowledge and Information Systems 3: pp. 263–286.
- [19] Keogh, E., Palpanas, T., Zordan, V.B., Gunopulos, D., and Cardle, M. (2004). Indexing Large Human-Motion Databases. Proceedings of 30th VLDB Conference, Toronto, Canada.

- [20] Keogh, E., Celly, B., Ratanamahatana, C.A., and Zordan, V.B. (2003). A novel technique for indexing video surveillance data. 1st ACM SIGMM international workshop on Video surveillance pp. 98–106. Berkeley, CA, USA: ACM Press.
- [21] Kim, H.-G., Moreau, N., and Sikora, T. (2005). Introduction to MPEG-7 audio. England: John Wiley & Sons.
- [22] Typke, R. (2006). Musipedia [Online]. Available from: <http://www.musipedia.org> [20 July 2007]
- [23] Pauws, S. (2002). CubyHum: A Fully Operational Query by Humming System. Proceedings of ISMIR, pp. 187–196.
- [24] Hu, N., Dannenberg, R.B., and Lewis, A.L. (2002). A Probabilistic Model of Melodic Similarity. Proceedings of the International Computer Music Conference (ICMC), San Francisco, USA: International Computer Music Association.
- [25] Liu, B., Wu, Y., and Li, Y. (2003). Linear hidden Markov model for music information retrieval based on humming. Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03).
- [26] Wang, Z., and Zhang, B. (2005). Quotient space model of hierarchical query-by-humming system. Proceedings of IEEE International Conference on Granular Computing, pp. 671–674.
- [27] Boersma, P., and Weenink, D. (2005). Praat: doing phonetics by computer (version 4.4.13) [Computer program]. Institute of Phonetic Sciences (Producer). Available from: <http://www.fon.hum.uva.nl/praat/> [August 8 2006]
- [28] Wang, Z. (2006). Time series Matching: a Multi-filter Approach. Doctoral dissertation. Department of Computer Science, New York University.
- [29] Adams, N.H., Bartsch, M.A., Shifrin, J.B., and Wakefield, G.H. (2004). Time series alignment for music information retrieval. Proceedings of 5th International Conference on Music Information Retrieval, Barcelona, Spain.



ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

บทความทางวิชาการเรื่อง “Efficient Multimedia Time Series Data Retrieval Under Uniform Scaling and Normalisation” โดยไวยณ์วุฒิ เอื้อจงประสิทธิ์ และโชติรัตน์ รัตน์นัทธนะ ในงานประชุมวิชาการ “30th European Conference on Information Retrieval” ซึ่งจัดขึ้น ณ เมืองกลาสโกว ประเทศสกอตแลนด์ ระหว่างวันที่ 30 มีนาคม ถึง 3 เมษายน 2551



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Efficient Multimedia Time Series Data Retrieval Under Uniform Scaling and Normalisation

Wanyawuth Boachongprasit and Chutirat Ann Ratanamahatana

Chulalongkorn University
Department of Computer Engineering
Phayathai Rd., Pathumwan, Bangkok 10330 Thailand
{g50weh, ann}@ep.eng.chula.ac.th

Abstract. As the world has shifted towards manipulation of information and its technology, we have been increasingly overwhelmed by the amount of available multimedia data while having higher expectations to fully exploit these data at hand. One of the attempts is to develop content-based multimedia information retrieval systems, which greatly facilitate us to intuitively search by its contents; a classic example is a Query-by-Humming system. Nevertheless, typical content-based search for multimedia data usually requires a large amount of storages and is computationally intensive. Recently, time series representation has been successfully applied to a wide variety of research, including multimedia retrieval due to the great reduction in time and space complexity. Besides, an enhancement, Uniform Scaling, has been proposed and applied prior to distance calculation, as well as it has been demonstrated that Uniform Scaling can outperform Euclidean distance. These previous work on Uniform Scaling, nonetheless, overlook the importance and effects of normalisation, which make their frameworks impractical for real world data. Therefore, in this paper, we justify this importance of normalisation in multimedia data and propose an efficient solution for searching multimedia time series data under Uniform Scaling and normalisation.

Keywords: Content-Based Multimedia Retrieval, Time Series, Uniform Scaling.

1 Introduction

As the world has shifted towards manipulation of information and its technology, multimedia data have played a crucial role in our daily lives. We feel much more comfortable in using multimedia data as a medium to communicate with each other, to present new ideas, or even to entertain ourselves. While the amount of multimedia data has been increasing dramatically and continually, our expectations of manipulating these data have as well been escalating. However, multimedia data manipulation typically requires a large amount of storages and is computationally intensive. Recently, time series representation has been proposed to help alleviate this burden, and it has been successfully applied in multi disciplines such as science, bioinformatics, economics, and multimedia [1].

Transformation from multimedia data into time series data is simple and straightforward, e.g., audio data in query by humming. We can just extract a sequence of pitch from a sung query and then use this time series as a query to retrieve an intended song from the database. For other types of multimedia data such as videos and images, several techniques have been proposed to convert them into time series data [2-5].

The main objective of the time series transformation of multimedia data is to achieve an efficient representation for data manipulation, including information retrieval. The heart of information retrieval, especially in time series retrieval, is similarity measurement, where Euclidean distance is prevalent. However, Euclidean distance seems to be impractical in several applications, particularly for multimedia applications, where shrinking and stretching of the data are very typical. For example, in query-by-humming system, users tend to sing queries slower or faster than the songs in the database. Hence, scaling of the data before distance calculation is very important. Uniform Scaling (US) was introduced to solve this problem [3]. Unfortunately, US comes at a cost and cannot scale well with large databases. Therefore, a lower bounding of US [3] was introduced to achieve significant speedup over the calculation by efficiently pruning a large number of unwanted sequences before costly distance calculation. Although this technique appears to be a practical solution for multimedia retrieval, unawareness of normalisation causes serious flaws in the previous framework.

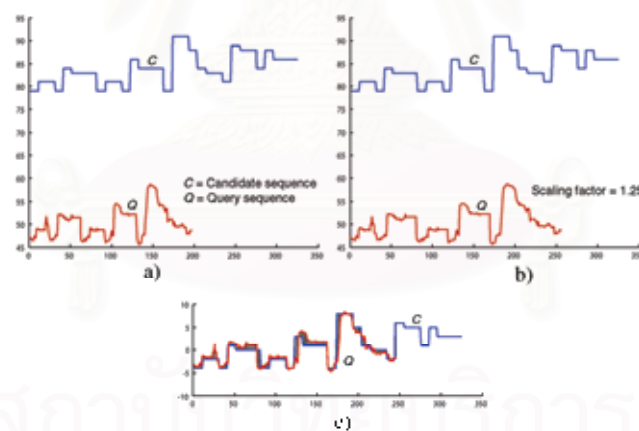


Fig. 1. a) A sequence of pitch extracted from a sung query sequence of a “Happy Birthday” song represented by a query sequence Q , and a MIDI pitch contour of the same song represented by a candidate sequence C . b) A rescaled query sequence with a scaling factor = 1.25 c) Both sequences after normalisation at the query’s length. The shaded region shows their Euclidean distance.

Generally, normalisation is crucial for similarity measurement in time series data since it enables us to measure the underlying shapes of time series. For example, in query by humming, we want to search a song database for the one with a segment that

is most similar to the sung query, regardless of the higher or lower key sung. However, to achieve this music key invariance, we must remove any existing offsets of both a query and a candidate sequence. If we do not normalise both sequences before distance calculation, this measurement will not be sensible. In Fig. 1 a), the shape of a query sequence extracted from a “Happy Birthday” sung query and the shape of a candidate sequence from the same part of the song are quite similar. Nevertheless, if we measure the similarity of these sequences by using only Euclidean distance without any pre-processing to the data, the distance will be excessive. Thus, rescaling and then normalising both sequences before distance calculation are crucial steps to achieve accurate and meaningful retrieval (see Fig. 1 b) and c)).

Though normalisation is one of the most important parts of similarity measurement, the current technique [3] has been developed while overlooking the importance of normalisation, which causes their framework to appear impractical for multimedia retrieval. Besides, their proposed lower-bounding function, which claims large pruning power, is invalid under normalisation condition. Hence, we propose a lower-bounding function that specifically deals with this normalisation problem efficiently and calculates a distance under the US, where no false dismissals are guaranteed.

2 Background

We begin with a formal problem definition and reviews of necessary background.

Problem definition. Suppose we have a query sequence Q of length m , where $Q = q_1, q_2, q_3, \dots, q_m$. It is scalable between lengths $sfmin*m$ and $sfmax*m$, where $sfmin$ and $sfmax$ are minimum and maximum scaling factors respectively, i.e., we can shrink or stretch a query sequence from any length $sfmin*m$ to $sfmax*m$, where $sfmax \geq 1$ and $0 < sfmin \leq 1$. In addition, each candidate sequence C of length n , $C = c_1, c_2, c_3, \dots, c_n$, is stored in a database D . For simplicity, here, we define $n \geq sfmax*m$. Finally, we want to find the most similar-shaped candidate sequence C in the database D to the query sequence Q , which is also scalable in arbitrary lengths between $sfmin*m$ and $sfmax*m$.

Definition 1. Squared Euclidean distance: We define a squared Euclidean distance measure in eq.(1), which calculates distance between two sequences of equal length m (query’s length). Note that the square root from the original Euclidean distance has been removed for an optimization purpose [3].

$$D(Q, C) \equiv \sum_{i=1}^m (q_i - c_i)^2 \quad (1)$$

Definition 2. Uniform Scaling: Uniform Scaling is a technique that uniformly stretches or shrinks a time series. In this approach, if we want to stretch a prefix of a candidate sequence C of length l to length m , we can use a scaling function in eq.(2); shrinking of a candidate sequence is done similarly to a stretching process.

We can formally define the US function as follows.

$$c_j = c_{\lfloor j \cdot l/m \rfloor} \quad \text{where } 1 \leq j \leq m \quad (2)$$

For the US distance calculation, prefixes of a candidate sequence C of length l , where $\lfloor sfmin * m \rfloor \leq l \leq \min(\lfloor sfmax * m \rfloor, n)$, are rescaled to length m (query's length). Then we use a squared Euclidean distance function to calculate distance between a query sequence and all rescaled prefix sequences in order to find a minimum distance value ranging from $sfmin$ to $sfmax$.

The formal definition of a Uniform Scaling distance function (US) is defined in eq.(3), where $RP(C, m, l)$ is a Rescaled Prefix function that returns a prefix of a candidate sequence of length l rescaled to length m .

$$US(Q, C, sfmin, sfmax) = \min_{l=\lfloor sfmin * m \rfloor}^{\min(\lfloor sfmax * m \rfloor, n)} D(RP(C, m, l), Q) \quad (3)$$

$$\text{where } RP(C, m, l)_i = c_{\lfloor i \cdot l/m \rfloor} ; 1 \leq i \leq m \text{ and } 1 \leq j \leq m$$

Definition 3. Lower bounding of Uniform Scaling [3, 6]: Lower bounding of Uniform Scaling is a distance approximation function, which can quickly compute a lower-bounding distance between a query and a candidate sequences; however, this lower bound value must not exceed the true distance value in order to be a valid lower-bounding function. To illustrate the idea, two new sequences are created, an upper envelope sequence UY and a lower envelope sequence LY enclosing a candidate sequence. This envelope represents all scaled candidate sequences for a lower-bounding distance calculation.

UY and LY are formally defined in eq.(4). Note that a lower bounding distance can simply be a squared Euclidean distance between a query sequence and the candidate's envelope, as defined in eq.(5).

$$UY_i = \max(c_{\lfloor i * sfmin \rfloor}, \dots, c_{\lfloor i * sfmax \rfloor}) \quad (4)$$

$$LY_i = \min(c_{\lfloor i * sfmin \rfloor}, \dots, c_{\lfloor i * sfmax \rfloor})$$

$$LBY(Q, C) = \sum_{i=1}^m \begin{cases} (q_i - UY_i)^2 & \text{if } q_i > UY_i \\ (q_i - LY_i)^2 & \text{if } q_i < LY_i \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

3 Our Proposed Method

As mentioned earlier, without realising an importance of normalisation, the distance measurement under US scheme becomes almost meaningless. Besides, the existing lower-bounding function cannot correctly calculate the distance for normalised sequences without false dismissals. For example, in Fig. 2, a query Q is a rescaled version of the candidate's prefix C ; hence, the distance between these two sequences should be zero according to their shape similarity. However, in Fig. 2 b), it is apparent

that the lower-bounding distance between a normalised query and the lower-bounding envelope is not zero as it should be. This phenomenon definitely violates the lower-bounding criteria, where the lower-bounding distance must guarantee not to exceed the true distance. Therefore, the existing lower-bounding function may cause some false dismissals. In an attempt to correct this flaw, we propose the US under *normalisation* condition, together with a corresponding lower-bounding function, which guarantees not to cause false dismissals, as shown in Fig. 2 c).

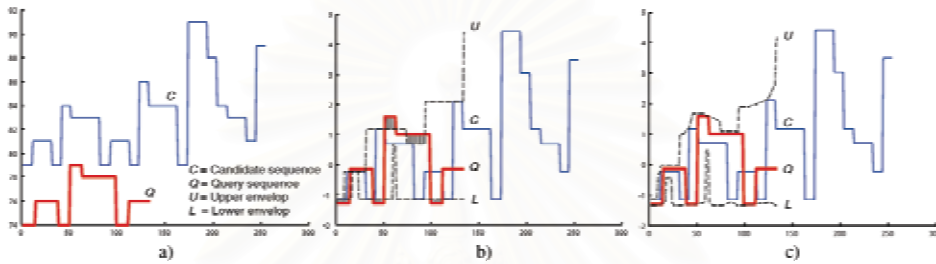


Fig. 2. a) Raw pitch contours are extracted from a “Happy Birthday” song sequence. C represents the candidate song, and Q is the query sung in a slower tempo (scaling factor = 1.2). b) and c) the query and the candidate sequences are z-normalised within the query’s length enclosed by different lower-bounding envelopes with the scaling factors ranging from 0.7 to 1.3. b) The previous lower bounding of US function [3]. c) Our proposed lower bounding of US that is guaranteed not to cause any false dismissals.

Definition 4. Uniform Scaling with Normalisation: The formal definition of US with normalisation is shown in eq.(6), where Q' is a normalised query, and $\overline{c_{1..l}}$ and $SD(c_{1..l})$ are mean and standard deviation values of a candidate’s prefix of length l respectively.

$$US_{\text{norm}}(Q', C, sfmin, sfmax) = \min_{l=[sfmin \cdot m]}^{\min(\lfloor sfmax \cdot m \rfloor, n)} D(RP_{\text{norm}}(C, m, l), Q') \quad (6)$$

where $RP_{\text{norm}}(C, m, l)_i = \frac{\overline{c_{\lfloor j \cdot l/m \rfloor}} - \overline{c_{1..l}}}{SD(c_{1..l})}$; $1 \leq i \leq m$ and $1 \leq j \leq m$

Definition 5. Lower bounding of Uniform Scaling with Normalisation: We develop a bounding envelope as shown in eq.(7), where UZ'_i and LZ'_i are an upper envelope and a lower envelope respectively. The distance calculation function is shown in eq.(8).

$$UZ'_i = \max\left(\frac{\overline{c_{sfmin \cdot i}} - \overline{c_{1..sfmin \cdot m}}}{SD(c_{1..sfmin \cdot m})}, \dots, \frac{\overline{c_{sfmax \cdot i}} - \overline{c_{1..sfmax \cdot m}}}{SD(c_{1..sfmax \cdot m})}\right) \quad (7)$$

$$= \max_{j=0}^{\lfloor m \cdot (sfmax - sfmin) \rfloor} \left(\frac{\overline{c_{[a]}} - \overline{c_{1..[b]+j}}}{SD(c_{1..[b]+j})}, \frac{\overline{c_{[a]}} - \overline{c_{1..[b]+j}}}{SD(c_{1..[b]+j})}, \frac{\overline{c_{[a]}} - \overline{c_{1..[b]+j}}}{SD(c_{1..[b]+j})}, \frac{\overline{c_{[a]}} - \overline{c_{1..[b]+j}}}{SD(c_{1..[b]+j})} \right)$$

$$\begin{aligned}
LZ'_i &= \min\left(\frac{c_{sfmin+i} - \overline{c_{L...sfmin+m}}}{SD(c_{L...sfmin+m})}, \dots, \frac{c_{sfmax+i} - \overline{c_{L...sfmax+m}}}{SD(c_{L...sfmax+m})}\right) \\
&= \min_{j=0}^{\lfloor m+(sfmax-sfmin) \rfloor} \left(\frac{c_{\lceil a \rceil} - \overline{c_{L...[b]+j}}}{SD(c_{L...[b]+j})}, \frac{c_{\lfloor a \rfloor} - \overline{c_{L...[b]+j}}}{SD(c_{L...[b]+j})}, \frac{c_{\lceil a \rceil} - \overline{c_{L...[b]+j}}}{SD(c_{L...[b]+j})}, \frac{c_{\lfloor a \rfloor} - \overline{c_{L...[b]+j}}}{SD(c_{L...[b]+j})} \right) \\
&\quad \text{where } a = sfmin * i + j * \frac{i}{m}, b = sfmin * m \\
LBZ(Q', C) &= \sum_{i=1}^m \begin{cases} (q_i - UZ'_i)^2 & \text{if } q_i > UZ'_i \\ (q_i - LZ'_i)^2 & \text{if } q_i < LZ'_i \\ 0 & \text{otherwise} \end{cases} \quad (8)
\end{aligned}$$

4 Experiment

In this section, we conduct an experiment to reconfirm that our approach is able to efficiently search through large multimedia databases by evaluating pruning power of the proposed lower-bounding function. The pruning power directly reflects the quality of the lower-bounding function. It is defined as the ratio of the candidate objects that can be disregarded from further calculations to the total number of candidates [6, 7], as shown in eq.(9).

$$Pruning\ Power = \frac{\text{The number of pruned candidates}}{\text{The total number of candidate sequences}} \quad (9)$$

In our experiments, we use 55 sung queries collected from 12 subjects of both genders. Then we extract sequences of pitch from the sung queries by using autocorrelation algorithm [8]. To observe the effect of different sequence lengths, we select sung queries that are sufficiently long, and crop their prefixes to lengths 75, 100, and 125 data points. The candidate sequences in databases are generated from the subsequences extracted from the MIDI files using a sliding window size of $125 * sfmax$ data points, where maximum scaling factor is 1.4.

To demonstrate the quality and utilities of our lower-bounding function, we test our proposed lower-bounding function on a simple query-by-humming system in two aspects using 1-nearest-neighbour algorithm. First, we inspect the pruning power using different lengths and different ranges of scaling as shown in Fig. 3, where the database contains 22441 subsequences. Second, we conduct an experiment to observe the effect of database sizes on pruning power by varying the number of subsequences; we use 22441, 55595, 107993 and 220378 subsequences from 100, 250, 500 and 1000 songs respectively to construct the databases. Note that queries' length is 100 data points, and the scaling factor ranges from 0.7 to 1.3. The result is shown in Fig. 4.

According to the experiment result in Fig. 3, the pruning power of our proposed method is quite impressive since the proposed lower-bounding function can prune a

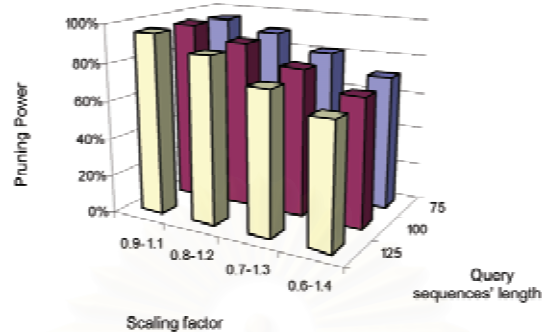


Fig. 3. The pruning power of different length and different ranges of scaling factors

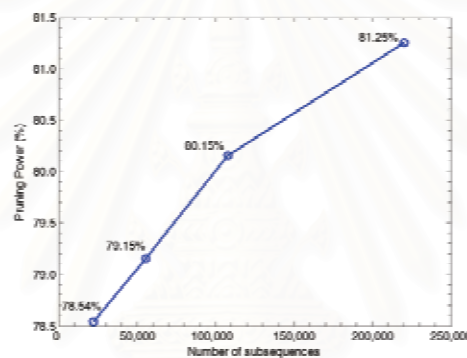


Fig. 4. The pruning powers in different-sized databases

large number of candidate sequences in database in every scaling. However, the wider the range of scaling factors are, the smaller the pruning power will become because the increasing range of scaling factors will also increase the size of the lower-bounding envelope. In addition, longer sequences appear to have smaller pruning power than that of the shorter sequences.

Fig. 4 demonstrates that pruning power will increase as the size of database increases. This is one of the most desirable properties for lower-bounding functions. However, the normalisation also affects the pruning power because the distance between the normalised query and the normalised candidate sequences is greatly reduced comparing with the distance between unnormalised sequences. Nonetheless, we would like to reemphasize the importance and necessity of normalisation, especially in multimedia applications.

5 Conclusions

We have shown that a lower-bounding function of US for *normalised* time series can efficiently prune a large number of candidate sequences in the database and

significantly reduce the time complexity in the retrieval, especially for multimedia data. Although this lower-bounding function can achieve dramatic speedups by pruning almost all the candidate sequences before finer calculations, scalability to a truly massive database is still a challenge for future research directions.

Acknowledgments. We would like to thank Dr. Boonserm Kijssirikul for valuable comments and enlightening discussions. This work is partially supported by the Thailand Research Fund (Grant No. MRG5080246).

References

1. Sakurai, Y., Yoshikawa, M., Faloutsos, C.: FTW: Fast Similarity Search under the Time Warping Distance. In: Proceedings of 24th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 326–337. ACM Press, Baltimore, Maryland (2005)
2. Keogh, E., Palpanas, T., Zordan, V.B., Gunopulos, D., Curdle, M.: Indexing Large Human Motion Databases. In: Proceedings of 30th VLDB Conference, Toronto, Canada (2004)
3. Keogh, E.: Efficiently Finding Arbitrarily Scaled Patterns in Massive Time Series Databases. In: Lavraç, N., Gamberger, D., Todorovski, L., Blockeel, H., eds. PKDD 2003, LNCS (LNAI), vol. 2838, pp. 253–265. Springer, Heidelberg (2003)
4. Keogh, E., Celly, B., Ratanamahatana, C.A., Zordan, V.B.: A novel technique for indexing video surveillance data. In: 1st ACM SIGMM international workshop on Video surveillance, pp. 98–106. ACM Press, Berkeley, California, USA (2003)
5. Shu, S., Narayanan, S., Kuo, C.-C.J.: Efficient Rotation Invariant Retrieval of Shapes using Dynamic Time Warping with Applications in Medical Databases. In: Proceedings of 19th IEEE International Symposium on Computer-Based Medical Systems (CBMS), pp. 673–678 (2006)
6. Fu, A., Wu, K.E., Lam, L.Y.H., Ratanamahatana, C.A.: Scaling and time warping in time series querying. In: Proceedings of 31st international conference on Very large data bases, VLDB Endowment, Trondheim, Norway, pp. 649–660 (2005)
7. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. *Knowledge and Information Systems: An International Journal (KAIS)* 7, 358–386 (2004)
8. Boersma, P., Weenink, D.: Praat: doing phonetics by computer (version 4.4.13) (2005). <http://www.praat.org/>

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข

บทความทางวิชาการเรื่อง “Accurate and Efficient Retrieval of Multimedia Time Series Data Under Uniform Scaling and Time Warping” โดยไวยณัฐติ เอื้อจง-ประสิทธิ์ และโชติรัตน์ รัตนามัทธนะ ในงานประชุมวิชาการ “The Pacific-Asia Conference on Knowledge Discovery and Data Mining” ซึ่งจัดขึ้น ณ เมืองโอซากา ประเทศญี่ปุ่น ระหว่างวันที่ 20 – 23 พฤษภาคม 2551



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Accurate and Efficient Retrieval of Multimedia Time Series Data Under Uniform Scaling and Time Warping

Waiyawuth Boachongprasit and Chutirat Ann Ratananahatana

Chulalongkorn University
 Department of Computer Engineering
 Phayathai Rd., Pathumwan, Bangkok 10330 Thailand
 (g55wch, ann)@cp.eng.chula.ac.th

Abstract. In this digital age, great interest has been shifted toward multimedia data manipulations. This includes videos, images, and audios, where typical manipulations require fairly large storage and are computationally intensive. Recent research has demonstrated the utilities of time series representation in various data mining tasks, allowing considerable reduction in time and space complexity. Specifically, the utilities of Uniform Scaling (US) and Dynamic Time Warping (DTW) have been shown to be necessary in several human-related domains, where uniform stretching or shrinking, as well as some local variation are typical. Classic examples include a query-by-humming system and motion capture data. However, all the past work has neglected the importance of data normalization before distance calculations, and therefore does not guarantee accurate retrievals. In this work, we discuss this concern and present a technique that accurately and efficiently searches under the US with DTW for normalized time series data, where no-false-dismissals are guaranteed.

Keywords: Data Mining, Content-based Multimedia Retrieval, Time Series, Uniform Scaling, Dynamic Time Warping.

1 Introduction

At present, multimedia data have evolved into our lives, where we increasingly have higher expectations in exploiting these data at hand. Typical manipulation usually requires fairly large amount of storage and is computationally intensive. Recently, it has been demonstrated that time series representation could be more efficient and effective in several domains, including science, bioinformatics, economics, and especially in multimedia [1]. For example, in a query-by-humming system, we can just extract a sequence of pitch from a sung query [2-6] to retrieve an intended song from the database. In motion retrieval, we can extract a sequence of motion in each video frame from a centroid of the object of interest in X, Y, and Z axes [7-9]. Similarly, in a content-based image search, image's shape can also be transformed into time series data for an efficient retrieval [10].

For the past decade, the most widely used distance measure in time series data has been Euclidean distance. It has been used for data retrieval, classification, clustering,

etc. However, Euclidean distance appears to be unsuitable for many applications because of its high sensitivity to variability in the time axis, and the superiority in accuracy of Dynamic Time Warping (DTW) over the Euclidean distance, which have been noted in various domains [7-9, 11, 12]. DTW, nonetheless, can handle only *local* variations in the data. Thus, it appears to be inappropriate for many applications, especially for multimedia or human-related domains where uniform stretching and shrinking are very typical [8, 12]; for example, in a query-by-humming system, most users tend to sing slower or faster than the original song. Similar problems also arise in other applications, such as motion capture and image retrievals [7, 9]. Recently, the use of Uniform Scaling (US) together with DTW has been introduced to mitigate this problem since US allows global scaling of time series before DTW distance calculation. However, this combination comes at a cost and cannot scale well with large databases. Fortunately, we have a lower bounding function of this distance measure [12], which can efficiently prune most of the dissimilar candidate sequences to achieve significant speedup over these calculations.

The Importance of Normalization

Even though we now have a relatively efficient technique to speed up US with DTW calculations, in almost all applications, data pre-processing is still mandatory for accurate and meaningful similarity measurements. In addition, *z*-normalization and mean normalization are typically used when our primary interest is the time series' shapes.

In query by humming, for instance, we want to search the song database for the one whose segment is most similar to the sung query. However, most users may not sing queries in the same key or the same pitch level as the songs stored in the database. Thus, normalization of both the query and candidate sequences is needed to remove any existing offsets [3, 4, 13] before any distance calculations. An example of this problem is illustrated in Fig. 1, where Q is the extracted sung query segment of the popular "Happy Birthday" song, and C is one of the candidate sequences stored in the database. We can quickly see that the shape of Q and a prefix of C are quite similar. Nevertheless, if we measure the similarity of these sequences directly using either Euclidean distance or DTW distance without any normalization nor rescaling, the distance will become excessive since both sequences may not be in the same pitch level nor in the right scale. Therefore, normalization and rescaling of both sequences before distance calculation are crucial steps to achieve an accurate retrieval (see Fig. 1 b) and c)).

In addition, the current lower-bounding method to prune off a large number of candidate sequences is developed without realization of the importance nor effects of data normalization. Hence, we propose a lower-bounding function to deal with this normalization problem efficiently and to calculate a distance under the US with DTW, where no false dismissals are guaranteed.

The rest of this paper is organized as follows. In section 2, we describe related research work and our motivation behind solving this normalization problem. Section 3 covers necessary background. In section 4, we describe our proposed method with a proof of no false dismissals. Section 5 verifies correctness of our method with a set of experiments to confirm the large pruning power in massive databases. Finally, section 6 gives some conclusions and offers possible future work.

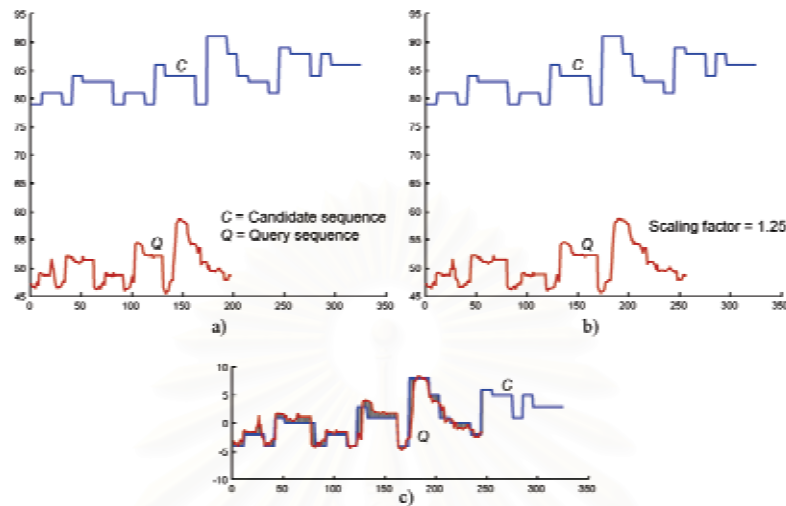


Fig. 1. a) A raw pitch contour extracted from a sung query represents a query sequence Q , and a MIDI pitch contour of “Happy Birthday” song represents a candidate sequence C . b) A rescaled query sequence Q with scaling factor = 1.25. c) Both sequences after mean normalization at the query’s length. The shaded region shows their Euclidean distance.

2 Related Work

For the past decade, DTW has attracted many researchers because of its superiority in accuracy over the ubiquitous Euclidean distance, which has been widely known in a variety of domains and applications [1-3, 5, 7-12, 14, 15]. However, lack of ability to globally stretch or shrink a time series of DTW in dealing with tempo variations has been known in music retrieval community [6, 16]. A straightforward solution is to generate every possible scaled version of the query or the candidates to be used in the measurement; it is, however, impractical for large databases. Thus, some researchers have proposed the methods to address and resolve this concern efficiently. Keogh has proposed a lower-bounding function for the US that can speed up the calculation by two to three orders of magnitude [8]. In 2005, Fu et al. have extended Keogh’s method providing a solution for a lower-bounding distance calculation under US with DTW [12]. At this point, although there have been relatively efficient solutions to deal with both US and DTW, practically none of the researchers has realized the importance and effects of normalization under US, and this is a primary cause of flaws in their methods due to the inapplicable distance calculation. Regardless of the dire need in normalization as mentioned earlier, it has been neglected and in turn has blocked up both US and DTW to achieve high accuracy and efficient retrieval.

Our contribution is to propose an efficient lower-bounding function for US with DTW distance calculation under normalization requirement, which can prune a significant number of unqualified candidate sequences. Nonetheless, we would like to reemphasize that normalization is a crucial step to achieve a meaningful distance calculation, especially in multimedia applications, as well as for efficient retrieval of the time series data.

3 Background

We begin with a formal problem definition as well as reviews of necessary background.

Problem definition. Suppose we have a query sequence Q of length m , where $Q = q_1, q_2, q_3, \dots, q_m$. It is scalable between lengths $sfmin * m$ and $sfmax * m$, where $sfmin$ and $sfmax$ are minimum and maximum scaling factors respectively, i.e., we can shrink or stretch a query sequence from lengths $sfmin * m$ to $sfmax * m$, where $sfmax \geq 1$ and $0 < sfmin \leq 1$. In addition, each candidate sequence C of length n , $C = c_1, c_2, c_3, \dots, c_n$ is stored in a database D . For simplicity, here, we define $n > sfmax * m$. Finally, we want to find the most similar-shaped candidate sequence C in the database D to the query sequence Q , which is also scalable in arbitrary lengths between $sfmin * m$ and $sfmax * m$.

Definition 1. Squared Euclidean distance: We define a squared Euclidean distance measure in eq.(1), which calculates distance between two sequences of equal length m (query's length). Note that the square root from the original Euclidean distance has been removed for an optimization purpose since the rankings of the results from both of these approaches are identical [8].

$$D(Q, C) = \sum_{i=1}^m (q_i - c_i)^2 \quad (1)$$

Definition 2. Uniform Scaling: Uniform Scaling is a technique that uniformly stretches or shrinks a time series. In this approach, if we want to stretch a prefix of a candidate sequence C of length l to length m , we can use the Uniform Scaling function in eq.(2); shrinking of a candidate is done similarly to a stretching process.

We can formally define the Uniform Scaling function as follows.

$$c_j = c_{\lfloor j * m / l \rfloor} \quad \text{where } 1 \leq j \leq m \quad (2)$$

For US distance calculation, prefixes of a candidate sequence C of length l , where $\lfloor sfmin * m \rfloor \leq l \leq \min(\lfloor sfmax * m \rfloor, n)$, are rescaled to length m (query's length). Then we use a squared Euclidean distance function to calculate distance between a query sequence and all rescaled prefix sequences in order to find a minimum distance value ranging from $sfmin$ to $sfmax$.

The formal definition of a Uniform Scaling distance function (US) is provided in eq.(3) and eq.(4), where $RP(C, m, l)$ is a Rescaled Prefix function that returns a prefix of a candidate sequence of length l rescaled to length m .

$$RP(C, m, l)_i = c_{\lfloor i * m / l \rfloor} \quad \text{where } 1 \leq i \leq m \text{ and } 1 \leq j \leq m \quad (3)$$

$$US(Q, C, sfmin, sfmax) = \min_{l \in [sfmin * m, sfmax * m]} D(RP(C, m, l), Q) \quad (4)$$

Definition 3. Lower bounding of Uniform Scaling [8, 12]: Lower bounding of Uniform Scaling is a distance approximation function, which can quickly compute a lower-bounding distance between a query and a candidate sequences; however, this lower bound value must not exceed the true distance value in order to be a valid lower-bounding function. To illustrate the idea, two new sequences are created, an upper envelope sequence UY and a lower envelope sequence LY , which enclose a candidate sequence. This envelope represents all scaled candidate sequences for a lower-bounding distance calculation.

UY and LY are formally defined in eq.(5), which was proposed in [8]. Note that a lower-bounding distance can simply be a squared Euclidean distance between a query sequence and the candidate's envelope, as defined in eq.(6).

$$UY_i = \max(c_{i, \min}, \dots, c_{i, \max}) \quad (5)$$

$$LY_i = \min(c_{i, \min}, \dots, c_{i, \max})$$

$$\text{LBY}(Q, C) = \sum_i \begin{cases} (q_i - UY_i)^2 & \text{if } q_i > UY_i \\ (LY_i - q_i)^2 & \text{if } q_i < LY_i \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Interested readers may consult [12] for further details about US and DTW.

4 Our Proposed Method

Because the existing lower-bounding functions are not designed for distance calculation under normalization requirement, they are flawed and do not give correct and meaningful calculation. Consequently, this paper is highly motivated to fix this flaw by proposing US with DTW function under *normalization* condition as well as their efficient lower-bounding functions. Furthermore, our proposed lower-bounding function is able to prune a large number of candidate sequences without undergoing such costly distance calculation, primarily to speed up the computation with no false dismissals.

The failure in lower-bounding distance calculation under z -normalization condition of the previous work is illustrated in Fig. 2 b) and c). The shown query in Fig. 2 a) is a rescaled version of the candidate's prefix (scaling factor = 1.2). Then we normalize both sequences by using z -normalization as presented in Fig. 2 b) to e). However, in Fig. 2 b) and c) being the previously proposed lower-bounding functions [8, 12], it is apparent that their results (lower-bounding distance) are not zero, i.e., the normalized query is not fully contained within the lower-bounding envelopes, as illustrated by the shaded regions. This phenomenon definitely violates the lower-bounding rule because the lower-bounding distance must not exceed the true distance; in this case, the true distance should in fact be zero. Therefore, these existing lower-bounding functions could cause some false dismissals in normalization scheme. Actually, this phenomenon is not surprising since both of the previously proposed lower-bounding functions are not developed for the normalization problem. Figs. 2 d) and e) are our proposed lower-bounding function that satisfy all the lower bounding conditions.

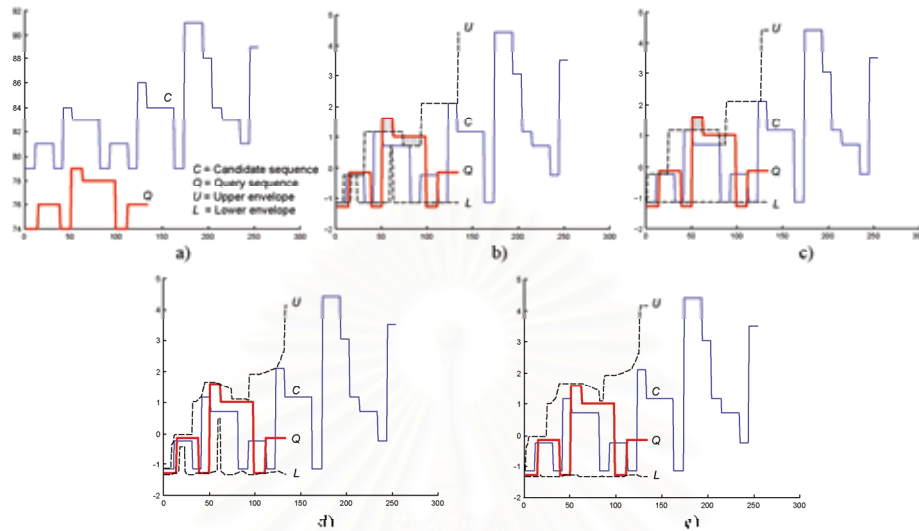


Fig. 2. a) Raw pitch contours extracted from a “Happy Birthday” song, where C represents the candidate, and Q is the query sung in a slower tempo (scaling factor = 1.2). Assume that a query sequence is no longer than the candidate sequence. b)–e) the query and the candidate sequences are z -normalized within the query’s length, enclosed by different lower-bounding envelopes with the scaling factor of range $[0.7, 1.3]$ b) A lower-bounding function of US that do not satisfy the lower-bounding condition [8]. c) A lower-bounding function of US with DTW, where a size of global constraint is 5%, which also does not satisfy the lower-bounding condition [12]. d) Our proposed lower-bounding function of US. e) Our proposed lower-bounding function of US with DTW, where a size of global constraint is 5%.

In this section, we begin with solutions for US and US with DTW distance measure, which satisfy normalization requirement, followed by the proof of no false dismissals.

Definition 4. Uniform Scaling with Normalization: The formal definition of a US with z -normalization is shown in eq.(7) and eq.(8), where Q' is a z -normalized query, and $c_{i,j}$ and $SD(c_{i,j})$ are mean and standard deviation of a candidate’s prefix of length l , respectively. Although different scalings of the same sequence through interpolation may yield different mean and standard deviation values, our observation discovers no statistically significant difference of mean and standard deviation between normalization before rescaling the sequences and rescaling the sequences before normalization.

$$RP_{norm}(C, m, l) = \frac{c_{i,m} - c_{i,l}}{SD(c_{i,l})} \quad \text{where } 1 \leq i \leq m \text{ and } 1 \leq j \leq m \quad (7)$$

$$US_{norm}(Q', C, s, \min, s, \max) = \min_{i, \min \leq i \leq \max}^{s, \min \leq i \leq \max} D(RP_{norm}(C, m, l), Q') \quad (8)$$

Definition 5. Lower bound of Uniform Scaling with Normalization: We develop a bounding envelope as expressed in eq.(9) and eq.(10), where UZ'_i and LZ'_i are upper and lower envelope sequences respectively. The corresponding distance calculation function is shown in eq.(11).

$$\begin{aligned}
 UZ'_i &= \max \left(\frac{C_{(1,1)} - C_{(1,2)}}{SD(c_{(1,1,2)})}, \dots, \frac{C_{(m,1)} - C_{(m,2)}}{SD(c_{(1,2,m)})} \right) \\
 &= \max_{1 \leq i \leq m} \left(\frac{C_{(i,1)} - C_{(i,2)}}{SD(c_{(1,2,i)})}, \dots, \frac{C_{(i,m)} - C_{(i,1)}}{SD(c_{(1,m,i)})} \right), \quad (9) \\
 &= \max_{1 \leq i \leq m} \left(\frac{C_{(i,1)} - C_{(i,2)}}{SD(c_{(1,2,i)})}, \dots, \frac{C_{(i,m)} - C_{(i,1)}}{SD(c_{(1,m,i)})} \right)
 \end{aligned}$$

$$\begin{aligned}
 LZ'_i &= \min \left(\frac{C_{(m,1)} - C_{(1,2)}}{SD(c_{(1,2,m)})}, \dots, \frac{C_{(m,1)} - C_{(1,2)}}{SD(c_{(1,2,m)})} \right) \\
 &= \min_{1 \leq i \leq m} \left(\frac{C_{(i,1)} - C_{(i,2)}}{SD(c_{(1,2,i)})}, \dots, \frac{C_{(i,m)} - C_{(i,1)}}{SD(c_{(1,m,i)})} \right), \quad (10) \\
 &= \min_{1 \leq i \leq m} \left(\frac{C_{(i,1)} - C_{(i,2)}}{SD(c_{(1,2,i)})}, \dots, \frac{C_{(i,m)} - C_{(i,1)}}{SD(c_{(1,m,i)})} \right)
 \end{aligned}$$

$$\text{LBZ}(Q', C) = \sum_{i=1}^m \begin{cases} (q_i - UZ'_i)^2 & \text{if } q_i > UZ'_i \\ (LZ'_i - q_i)^2 & \text{if } q_i < LZ'_i \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

In case of US with DTW distance measure with normalization, we can simply change distance calculation function D in eq.(8) from the squared Euclidean function to DTW function. Additionally, its lower-bounding function is also quite straightforward that we can apply lower bounding of DTW over our envelope from eq.(9) and eq.(10), as shown in eq.(12): ϵ is the size of a global constraint when using DTW distance calculation. The calculation of this lower-bounding function is similar to other functions as stated earlier. Due to space limitations, we omit full details of this distance function and its proof for brevity.

$$\begin{aligned}
 U'_i &= \max(UZ'_{(1,1)}, \dots, UZ'_{(m,1)}) \\
 L'_i &= \min(LZ'_{(1,1)}, \dots, LZ'_{(m,1)}) \quad (12)
 \end{aligned}$$

Lastly, to validate the correctness of the proposed method, we complete this section with a proof of our lower bounding properties.

Proposition 1. Let Q' be a normalized query sequence of length m , and C be a candidate sequence of length n . In addition, a minimum scaling factor and a maximum scaling factor are $sfmin$ and $sfmax$ respectively, where $0 < sfmin \leq 1$ and $sfmax \geq 1$, i.e., the query can be scalable between $sfmin * m$ and $sfmax * m$. Then the value of $LBZ(Q', C)$ is a lower-bounding distance of $US_{norm}(Q', C, sfmin, sfmax)$.

Proof. Suppose $UZ'_i \geq c'_j \geq LZ'_i$, where UZ'_i is an upper envelope, LZ'_i is a lower envelope, and c'_j is a normalized data point of a candidate sequence at an arbitrary scaling between $sfmin$ and $sfmax$, i.e., $sfmin * i \leq j \leq sfmax * i$. Then $LBZ(Q', C) \leq US_{norm}(Q', C, sfmin, sfmax)$.

If a query is stretched to length $sf * m$, where sf is a scaling factor, a prefix of a candidate sequence with length $sf * m$ will be rescaled into length m and z-normalized by its mean and standard deviation, as shown in eq.(13).

$$c'_j = \frac{c_j - \overline{c_{1..sf * m}}}{SD(c_{1..sf * m})} \quad \text{where } 1 \leq j \leq m \quad (13)$$

From eq.(9) and eq.(10), the upper and lower envelopes are defined as follows.

$$UZ'_i \geq \frac{c_{sfmin * i} - \overline{c_{1..sfmin * m}}}{SD(c_{1..sfmin * m})}, \dots, \frac{c_{sfmax * i} - \overline{c_{1..sfmax * m}}}{SD(c_{1..sfmax * m})} \geq LZ'_i \quad (14)$$

From eq.(13) and (14), it follows that

$$UZ'_i \geq c'_k \geq LZ'_i \quad \text{where } sfmin * i \leq k \leq sfmax * i$$

$$\begin{aligned} (q_i - UZ'_i)^2 &< (q_i - c'_k)^2 & \text{if } q_i > UZ'_i \\ (LZ'_i - q_i)^2 &< (q_i - c'_k)^2 & \text{if } q_i < LZ'_i \\ 0 & & \text{otherwise} \end{aligned}$$

$$LBZ(Q', C) \leq US_{norm}(Q', C, sfmin, sfmax) \quad \text{Q.E.D.}$$

5 Experiment

In the previous section, we introduce our proposed lower-bounding function as well as justification of its correctness in preserving the lower-bounding properties. In this section, we carefully evaluate the efficiency of our proposed method by conducting sets of experiments to observe the pruning power [11, 12]. Note that the pruning power is the fraction of the total candidate objects that can be discarded from further calculation, as defined in eq.(15).

$$Pruning\ Power = \frac{\text{Number of pruned candidates}}{\text{Total number of candidate sequences}} \quad (15)$$

In these experiments, we develop a simple Query-by-Humming system based on one-nearest-neighbor time series matching technique [3, 4, 6, 12, 17] in order to demonstrate the quality and utilities of our proposed method in multimedia database. We use 100 to 2,000 different international songs in MIDI file format and generate candidate sequences from this MIDI songs by using sliding windows of length $150 \cdot sf_{max}$ data points, where $sf_{max} = 1.4$, and then store it in a database. For query sequences, we collect 55 sung queries from 12 subjects of both genders with various singing abilities and then extract sequences of pitch from these sung queries by using autocorrelation algorithm [18].

To carefully evaluate each factor that affects quality of the lower-bounding function, we conduct three experiments. In the first experiment, we examine an effect of different lengths of sequences with different ranges of scaling factors under 22441 sequences (generated from 100 songs), where a size of the global constraint is set to 4 percent of the sequences' length (see Fig. 3 a)). In the second experiment, we investigate an effect of different lengths of the sequences with different sizes of global constraint under 22441 sequences, where range of scaling factor is between 0.8 and 1.2 (see Fig. 3 b)). In the last experiment, we use 22441, 55595, 107993, 220378, and 442366 subsequences from 100, 250, 500, 1000, and 2000 songs to construct different-sized databases in order to observe their pruning powers, as shown in Fig. 4.

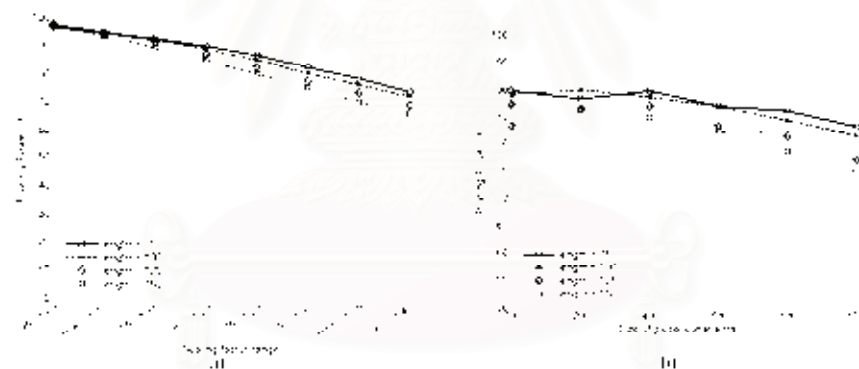


Fig. 3. a) The pruning powers of different time series lengths with various ranges of scaling factors. b) The pruning powers of different lengths with various sizes of global constraint.

According to these experiment results, our proposed lower-bounding function can prune a great number of candidate sequences in every parameter setting, as shown in Figs. 3 and 4. However, from these experiments, we found that there are several factors influencing the pruning power, and range of scaling factors is one of them. In Fig. 3 a), increases in range of scaling factors are likely to decrease the pruning power directly. Besides, the pruning power slightly decreases as the sizes of global constraint increase (see Fig. 3 b)). The rationale behind these results is that both increases in range of scaling factors and in size of global constraint definitely enlarge the size of

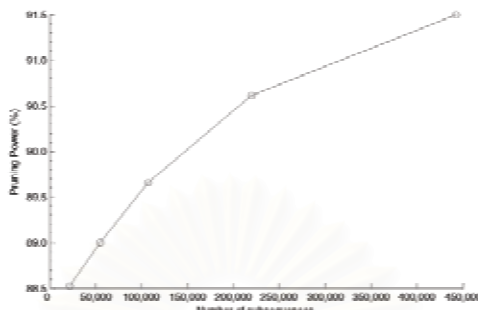


Fig. 4. The pruning powers of different database's sizes

the lower-bounding envelopes, causing a reduction in the lower bounding distance, and hence its pruning power. In addition, longer sequences appear to give smaller pruning power than that of the shorter sequences as illustrated in Figs. 3 and 4. In contrast, the pruning power is found to increase as the database size increases (see Fig. 4), which is a highly desirable property for lower-bounding functions.

Regardless of a few factors that decrease the pruning power, we discover no significant improvement in accuracy when we increase scaling factor range over 0.6-1.4 in our experiment. In addition, from recent research [19], wider sizes of the global constraint do not imply higher accuracy. In fact, in most cases, the size of the global constraint of less than 10 percent often yields optimal accuracy.

Notice that the normalization does affect the pruning power because the distances between the normalized query and the normalized candidate sequences are greatly reduced, comparing with the distances among unnormalized sequences. However, we would like to reemphasize that *normalization* is essential in many applications.

6 Discussion and Conclusions

We have shown that this proposed lower-bounding function of US with DTW under *normalization* requirement can efficiently prune a large number of candidates in the database, significantly reducing the time complexity in the data retrieval, especially for multimedia retrieval, while no false dismissals are also guaranteed. Furthermore, our approach can work well with other types of normalization. Nonetheless, we would like to reemphasize the importance and necessity of normalization, especially in multimedia applications.

Besides dramatically speeding up the calculations by pruning almost all candidates, this lower-bounding function is possible to utilize dimensionality reduction and indexing techniques [9] in order to be scalable to truly massive databases.

Acknowledgments. We would like to thank Dr. Boonserm Kijirikul for valuable comments and enlightening discussions. This work is partially supported by the Thailand Research Fund (Grant No. MRG5080246).

References

1. Sakurai, Y., Yoshikawa, M., Faloutsos, C.: FTW: Fast Similarity Search under the Time Warping Distance. In: Proceedings of 24th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 326–337. ACM Press, Baltimore, Maryland (2005)
2. Adams, N.H., Bartsch, M.A., Shifrin, J.B., Wakefield, G.H.: Time series alignment for music information retrieval. In: Proceedings of 5th International Conference on Music Information Retrieval, Barcelona, Spain (2004)
3. Zhu, Y., Shasha, D.: Warping indexes with envelope transforms for query by humming. In: Proceedings of the 2003 ACM SIGMOD international conference on Management of data, pp. 181–192. ACM Press, San Diego, California (2003)
4. Jang, J.-S.R., Lee, H.-R.: Hierarchical filtering method for content-based music retrieval via acoustic input. In: Proceedings of 9th ACM international conference on Multimedia, pp. 401–410. ACM Press, Ottawa, Canada (2001)
5. Lee, H. R., Chen, C., Jang, J. S.R.: Approximate lower bounding functions for the speedup of DTW for melody recognition. In: International Workshop on Cellular Neural Networks and Their Applications, pp. 178–181 (2005)
6. Hu, N., Dannenberg, R.B.: A comparison of melodic database retrieval techniques using sung queries. In: Proceedings of 2nd ACM/IEEE-CS joint conference on Digital Libraries, pp. 301–307. ACM Press, Portland, Oregon, USA (2002)
7. Keogh, E., Palpanas, T., Zordan, V.B., Gunopulos, D., Cardle, M.: Indexing Large Human-Motion Databases. In: Proceedings of 30th VLDB Conference, Toronto, Canada (2004)
8. Keogh, E.: Efficiently Finding Arbitrarily Scaled Patterns in Massive Time Series Databases. In: Lavrac, N.,amberger, D., Tudorovski, L., Blockeel, H. (eds.) PKDD 2003, LNCS (LNAI), vol. 2838, pp. 253–265. Springer, Heidelberg (2003)
9. Keogh, E., Celly, B., Ratanamahatana, C.A., Zordan, V.B.: A novel technique for indexing video surveillance data. In: 1st ACM SIGMM international workshop on Video surveillance, pp. 98–106. ACM Press, Berkeley, California, USA (2003)
10. Shu, S., Narayanan, S., Kuo, C.-C.J.: Efficient Rotation Invariant Retrieval of Shapes using Dynamic Time Warping with Applications in Medical Databases. In: Proceedings of 19th IEEE International Symposium on Computer-Based Medical Systems (CBMS), pp. 673–678 (2006)
11. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. Knowledge and Information Systems: An International Journal (KAIS) 7, 358–386 (2004)
12. Fu, A.W.-c., Keogh, E., Lau, J.Y.H., Ratanamahatana, C.A.: Scaling and time warping in time series querying. In: Proceedings of 31st international conference on Very large data bases. VLDB Endowment, Trondheim, Norway, pp. 649–660 (2005)
13. Wang, Z., Zhang, B.: Quotient space model of hierarchical query-by-humming system. In: Proceedings of IEEE International Conference on Granular Computing, vol. 2, pp. 671–674 (2005)
14. Ratanamahatana, C.A., Keogh, E.: Multimedia Retrieval Using Time Series Representation and Relevance Feedback. In: Fox, E.A., Neuhoff, E.J., Premsmit, P., Wuwongse, V. (eds.) ICADL 2005, LNCS, vol. 3815, pp. 400–405. Springer, Heidelberg (2005)
15. Ratanamahatana, C.A., Keogh, E.: Making Time-Series Classification More Accurate Using Learned Constraints. In: Proceedings of SIAM International Conference on Data Mining (SDM), Lake Buena Vista, Florida, USA, pp. 11–22 (2004)

16. Meek, C., Birmingham, W.P.: The dangers of parsimony in query-by-humming applications. In: Proceedings of 4th International Symposium on Music Information Retrieval (ISMIR) (2003)
17. Dannenberg, R.B., Birmingham, W.P., Pardo, B., Hu, N., Meek, C., Tzanetakis, G.: A comparative evaluation of search techniques for query-by-humming using the MUSART testbed. *Journal of the American Society for Information Science and Technology* 58, 687–701 (2007)
18. Boersma, P., Weenink, D.: Praat: doing phonetics by computer (version 4.4.13) (2005), <http://www.praat.org/>
19. Ratanamahatana, C.A., Keogh, E.J.: Everything you know about dynamic time warping is wrong. In: 3rd Workshop on Mining Temporal and Sequential Data, in conjunction with 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004) (2004)



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายไวยวัฒน์ ธี้อัจประสิทธิ์ เกิดวันที่ 1 กันยายน พ.ศ. 2527 สำเร็จ การศึกษาระดับมัธยมศึกษาที่โรงเรียนสวนกุหลาบวิทยาลัย จากนั้นจึงเข้าศึกษาต่อที่คณะ วิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2546 และในปีการศึกษา 2549 จึง สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ เกียรตินิยม อันดับสอง และเข้าศึกษาในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรม คอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2550



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย