

การรู้จำตัวอักษรภาษาไทยแบบออนไลน์โดยใช้แบบจำลองฮิดเดนมาร์คอฟและซัพพอร์ตเวกเตอร์แมชชีน



นายปริญญา สวงนิตย์

สถาบันวิทยบริการ

จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า

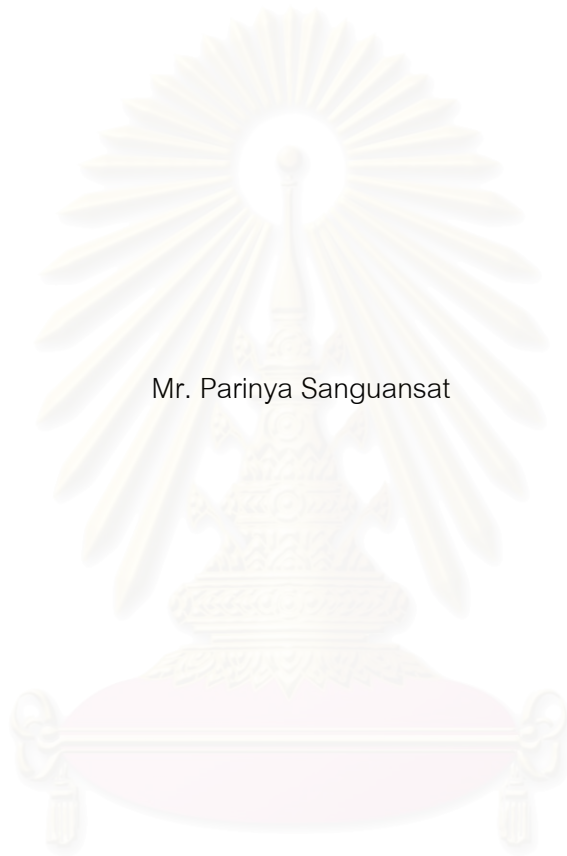
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2547

ISBN 974-17-6044-2

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

ONLINE THAI CHARACTER RECOGNITION USING HIDDEN MARKOV MODELS
AND SUPPORT VECTOR MACHINES



Mr. Parinya Sanguansat

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย
A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Electrical Engineering

Department of Electrical Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2004

ISBN 974-17-6044-2

หัวข้อวิทยานิพนธ์ การรู้จำตัวอักษรภาษาไทยแบบออนไลน์โดยใช้แบบจำลองฮิดเดน
มาร์คอฟและซัพพอร์ตเวกเตอร์แมชชีน
โดย นาย ปริญญา สงวนสัตย์
สาขาวิชา วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษา รongศาสตราจารย์ ดร.สมชาย จิตะพันธ์กุล
อาจารย์ที่ปรึกษาร่วม อาจารย์วิทยากร อัครวิเศษ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยาลัย
หนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร. ดิเรก ลาวัญศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(อาจารย์สุวิทย์ นาคพิระยุทธ)

..... อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ ดร. สมชาย จิตะพันธ์กุล)

..... อาจารย์ที่ปรึกษาร่วม
(อาจารย์วิทยากร อัครวิเศษ)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. บุญเสริม กิจศิริกุล)

นาย ปริญญา สงวนสัตย์ : การรู้จำตัวอักษรภาษาไทยแบบออนไลน์โดยใช้แบบจำลองฮิดเดนมาร์คอฟและซัพพอร์ตเวกเตอร์แมชชีน. (ONLINE THAI CHARACTER RECOGNITION USING HIDDEN MARKOV MODELS AND SUPPORT VECTOR MACHINES) อ. ที่ปรึกษา : รศ. ดร.สมชาย จิตะพันธ์กุล อ.ที่ปรึกษาร่วม : อ.วิทยากร อัครวิเศษ จำนวน 94 หน้า. ISBN 974-17-6044-2.

วิทยานิพนธ์ฉบับนี้นำเสนอวิธีรู้จำแบบใหม่ด้วยแบบจำลองฮิดเดนมาร์คอฟทำงานร่วมกับซัพพอร์ตเวกเตอร์แมชชีน สำหรับการรู้จำตัวอักษรเดี่ยวหลายมือเขียนภาษาไทยจากคำแบบออนไลน์ ในปัจจุบันงานวิจัยทางการรู้จำแบบรูปมีแนวโน้มเข้าสู่ 2 วิธีการหลักคือ วิธีการสร้างแบบจำลองแบบเจเนอเรทีฟเพื่อใช้จำลองรูปแบบของชนิดต่างๆ ซึ่งเทคนิคที่นิยมใช้ในการรู้จำตัวอักษรได้แก่ แบบจำลองฮิดเดนมาร์คอฟ และ วิธีการวิเคราะห์หัตถสคริมิแนนต์เชิงเส้นทั้งโดยตรงและทางอ้อม ซึ่งเทคนิคที่กำลังอยู่ในความสนใจอย่างมากในปัจจุบัน คือ ซัพพอร์ตเวกเตอร์แมชชีน ขณะที่การรู้จำด้วยแบบจำลองฮิดเดนมาร์คอฟประสบปัญหาในการรู้จำกลุ่มตัวอักษรภาษาไทยที่คล้ายกันส่งผลให้อัตราการรู้จำต่ำเนื่องจากการสร้างแบบจำลองของชนิดที่คล้ายกัน แบบจำลองที่ได้ยอมคล้ายกันทำให้ตัดสินใจผิดพลาด ซึ่งซัพพอร์ตเวกเตอร์แมชชีนสามารถวิเคราะห์หัตถสคริมิแนนต์เพื่อหากไฮเปอร์เพลนในการจำแนกชนิดที่คล้ายกันเหล่านี้ได้ แต่ซัพพอร์ตเวกเตอร์แมชชีนเป็นตัวจำแนกแบบไบนารีจึงไม่เหมาะสมกับจำนวนชนิดที่มาก เช่น จำนวนตัวอักษรภาษาไทย กรรมวิธีในงานวิจัยนี้จึงนำเอาข้อดีของทั้งสองแนวทางในการรู้จำมาทำงานร่วมกันเพื่อลดเซตอัตราการรู้จำต่ำในแบบจำลองฮิดเดนมาร์คอฟในกลุ่มชนิดที่คล้ายกัน และลดจำนวนชนิดที่ต้องจำแนกในซัพพอร์ตเวกเตอร์แมชชีน

ผลการทดสอบกระทำบนไมโครคอมพิวเตอร์ที่ใช้หน่วยประมวลผลกลางเป็นเพนเทียมโฟร์ ความเร็ว 2.4 กิกะเฮิรตซ์ และมีหน่วยความจำหลัก 512 เมกะไบต์ การทดสอบระบบกับตัวอักษรตัวเดียวระดับบน กลาง และล่างของผู้เขียนอิสระได้อัตราการรู้จำเฉลี่ยร้อยละ 90.57 92.20 และ 97.88 ตามลำดับ

ภาควิชา วิศวกรรมไฟฟ้า ลายมือชื่อนิสิต

สาขาวิชา วิศวกรรมไฟฟ้า ลายมือชื่ออาจารย์ที่ปรึกษา

ปีการศึกษา 2547 ลายมือชื่ออาจารย์ที่ปรึกษาร่วม

4570411321 : MAJOR ELECTRICAL ENGINEERING

KEY WORD: ON-LINE CHARACTER RECOGNITION / HIDDEN MARKOV MODEL /
SUPPORT VECTOR MACHINES / HANDWRITTEN THAI CHARACTER RECOGNITION

PARINYA SANGUANSAT : ONLINE THAI CHARACTER RECOGNITION USING
HIDDEN MARKOV MODEL AND SUPPORT VECTOR MACHINES. THESIS
ADVISOR : ASSOC. PROF. SOMCHAI JITAPUNKUL, Dr.Ing., THESIS
COADVISOR : WIDHYAKORN ASDORNWISED, 94 pp. ISBN 974-17-6044-2.

This thesis presents a novel on-line recognition method using Hidden Markov Models and Support Vector Machines in combination for recognizing on-line isolated Thai handwritten characters. Nowadays, active researches in pattern recognition are converged into two distinct methods: model estimation by means of generative method, of which the popular technique is Hidden Markov Models (HMM), and direct/indirect linear discriminant analysis, Support Vector Machines (SVM) is presently being interested. HMM shows inferior performance for recognizing some Thai characters that have similar patterns but it is appropriate technique for multi-classification. On the other hand, SVM provides the discriminant analysis for classifying these characters but it is a binary classifier. Therefore, our method is proposed to exploit the best of two worlds by combining the advantages of each other in order to compensate the HMM's poor recognition rate of similar characters and reduce the number of class candidates in SVM.

The system was tested on a 2.4 GHz Pentium IV processor and 512 MB of RAM. The experimental results of upper, middle, and lower level character case for independent writers showed an average recognition rate are 90.57, 92.20, and 97.88, respectively

Department Electrical Engineering Student's signature

Field of study Electrical Engineering Advisor's signature

Academic year 2004 Coadvisor's signature

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลือของ รองศาสตราจารย์ ดร.สมชาย จิตะพันธ์กุล อาจารย์ที่ปรึกษาวิทยานิพนธ์ อ.วิทย์กร อัครวิเศษ อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม คุณ Nathan Smith นักศึกษาปริญญาเอกมหาวิทยาลัย Cambridge และ นักวิจัยแห่ง Naval Undersea Warfare Center (NUWC) ประเทศอเมริกา Dr. Paul M. Baggenstoss ซึ่งได้ให้คำแนะนำและข้อคิดเห็น รวมถึงซอฟต์แวร์ต่าง ๆ ตลอดช่วงเวลาในการทำวิจัย ผู้วิจัยจึงขอกราบขอบคุณมา ณ ที่นี้

ขอขอบคุณโครงการเสริมสร้างความเชื่อมโยงระหว่างภาควิชาวิศวกรรมไฟฟ้า และภาคเอกชนทางด้านการวิจัยและพัฒนา ปี พ.ศ. 2545 ถึง พ.ศ. 2547 ที่มอบทุนสนับสนุนแก่ผู้วิจัย

ขอขอบคุณห้องปฏิบัติการวิจัยกรรมวิธีสัญญาณดิจิทัล สถานที่ทำการวิจัย รวมถึงเพื่อนพี่น้องนิสิตห้องปฏิบัติการ ที่มีส่วนช่วยเหลือและให้ข้อคิดเห็นและคำแนะนำ และ กำลังใจตลอดระยะเวลาการทำวิจัยอย่างยิ่ง

สุดท้ายนี้ขอขอบคุณบิดา มารดา รวมถึงเพื่อนพี่น้อง ซึ่งได้ให้การสนับสนุน และเป็นกำลังใจให้แก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ง
สารบัญ	ช
สารบัญตาราง	ญ
สารบัญภาพ.....	ฎ
1. บทนำ.....	1
1.1 ประวัติและความเป็นมา	1
1.2 วัตถุประสงค์	2
1.3 เป้าหมายและขอบเขตของการวิจัย	2
1.4 ขั้นตอนและวิธีการดำเนินงาน	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
2. ทฤษฎีและแนวคิด	4
2.1 กระบวนการก่อนหน้า.....	4
2.2 แบบจำลองฮิดเดนมาร์คอฟ	5
2.2.1 องค์ประกอบของแบบจำลองฮิดเดนมาร์คอฟแบบไม่ต่อเนื่อง	5
2.2.2 การใช้แบบจำลองฮิดเดนมาร์คอฟในการรู้จำ	6
2.2.3 แบบจำลองฮิดเดนมาร์คอฟแบบต่อเนื่อง	13
2.3 ซัพพอร์ตเวกเตอร์แมชชีน.....	14
2.3.1 มิติ Vapnik/Chervonenkis (VC-dimension)	15
2.3.2 ซัพพอร์ตเวกเตอร์แมชชีนแบบเชิงเส้น.....	16
2.3.3 ซัพพอร์ตเวกเตอร์แมชชีนแบบไม่เชิงเส้น.....	21
2.4 ปริภูมิคะแนน (Score-spaces).....	23
2.4.1 อาร์กิวเมนต์ของคะแนน (Score-argument).....	25
2.4.2 ตัวดำเนินการคะแนน (Score-operator)	26
2.4.3 การประยุกต์ใช้ปริภูมิคะแนนกับซัพพอร์ตเวกเตอร์แมชชีน.....	26
2.4.4 ปริภูมิคะแนนที่ถูกนิยามด้วยตัวดำเนินการคะแนนแบบอนุพันธ์อันดับหนึ่ง.....	27

2.4.5	นอร์มัลไลซ์เคอร์เนล (Normalized Kernel)	28
2.4.6	ปริภูมิคะแนนสำหรับเวกเตอร์มิติคงที่	29
3.	การรู้จำตัวอักษรลายมือเขียนภาษาไทยแบบออนไลน์ด้วยแบบจำลองฮิดเดนมาร์คคอฟ และซัพพอร์ตเวกเตอร์แมชชีน	33
3.1	การทำงานของแบบจำลองฮิดเดนมาร์คคอฟในระบบรู้จำตัวอักษร	34
3.1.1	การแบ่งส่วนของเส้น (Stroke Segmentation)	34
3.1.2	การสกัดคุณลักษณะสำคัญ (Feature Extraction)	36
3.1.3	การรู้จำตัวอักษรด้วยแบบจำลองฮิดเดนมาร์คคอฟ	37
3.2	การทำงานของซัพพอร์ตเวกเตอร์แมชชีนในระบบรู้จำตัวอักษร	38
3.2.1	การสกัดคุณลักษณะสำคัญ (Feature Extraction)	38
3.2.2	การนอร์มัลไลซ์ปริภูมิคะแนน (Score space Normalization)	40
3.2.3	กลุ่มของชนิดที่มีการตัดสินสับสน	43
3.2.4	การใช้ซัพพอร์ตเวกเตอร์แมชชีนในการรู้จำตัวอักษร	44
3.2.4	ระเบียบวิธี Max Win	45
3.3	การรู้จำสระและวรรณยุกต์ระดับบน	45
3.3.1	กระบวนการก่อนหน้า	46
3.3.2	การสกัดคุณลักษณะสำคัญ	46
4.	ขั้นตอนการทดลอง ผลการทดลองและวิเคราะห์ผลการทดลอง	48
4.1	เครื่องมือที่ใช้ในการทดสอบ	48
4.2	การจัดเก็บข้อมูล	49
4.3	การทดสอบประสิทธิภาพการรู้จำด้วยแบบจำลองฮิดเดนมาร์คคอฟ	49
4.4	วิเคราะห์การทดสอบประสิทธิภาพของการรู้จำด้วยแบบจำลองฮิดเดนมาร์คคอฟ	55
4.5	กลุ่มของชนิดที่มีการตัดสินสับสนที่ได้จากผลการทดสอบ	55
4.5.1	กลุ่มของชนิดที่มีการตัดสินสับสนที่ได้จากกลุ่มผู้เขียนที่ใช้ในการฝึกฝน ...	56
4.5.2	กลุ่มของชนิดที่มีการตัดสินสับสนที่ได้จากกลุ่มผู้เขียนอิสระ	57
4.6	การทดสอบประสิทธิภาพการรู้จำด้วยระบบรู้จำที่นำเสนอ	58
4.6.1	การทดสอบประสิทธิภาพการรู้จำตัวอักษรระดับกลางและล่างด้วยระบบ รู้จำที่นำเสนอ	58
4.6.2	การทดสอบประสิทธิภาพการรู้จำตัวอักษรระดับบนด้วยระบบรู้จำที่ นำเสนอ	65

4.7 วิเคราะห์การทดสอบประสิทธิภาพของการรู้จำด้วยระบบรู้จำที่นำเสนอเมื่อ
เปรียบเทียบกับระบบรู้จำที่ใช้แบบจำลองฮิดเดนมาร์คอฟพื้นฐาน..... 68

5. สรุปผลการวิจัยและข้อเสนอแนะ..... 81

 5.1 บทสรุป..... 81

 5.2 ข้อเสนอแนะสำหรับงานในอนาคต 83

รายการอ้างอิง..... 84

ภาคผนวก..... 87

ประวัติผู้เขียนวิทยานิพนธ์ 94



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

	หน้า
ตารางที่ 1.1 ตัวอักษรภาษาไทยแบ่งตามระดับ.....	3
ตารางที่ 3.1 ตัวอย่างเมตริกซ์ความสับสน.....	44
ตารางที่ 4.1 อัตราจำเฉลี่ยของระบบรู้จำตัวอักษรภาษาไทยด้วยแบบจำลองฮิดเดนมาร์คอฟ 50	
ตารางที่ 4.2 ผลการทดสอบระบบด้วยแบบจำลองฮิดเดนมาร์คอฟโดยใช้ข้อมูลจากสระและวรรณยุกต์ระดับบนของกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ.....	50
ตารางที่ 4.3 ผลการทดสอบระบบด้วยแบบจำลองฮิดเดนมาร์คอฟโดยใช้ข้อมูลจากสระและวรรณยุกต์ระดับบนของกลุ่มผู้เขียนอิสระ	51
ตารางที่ 4.4 ผลการทดสอบระบบด้วยแบบจำลองฮิดเดนมาร์คอฟโดยใช้ข้อมูลจากสระระดับล่างของกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ	51
ตารางที่ 4.5 ผลการทดสอบระบบด้วยแบบจำลองฮิดเดนมาร์คอฟโดยใช้ข้อมูลจากสระระดับล่างของกลุ่มผู้เขียนอิสระ.....	52
ตารางที่ 4.6 ผลการทดสอบระบบรู้จำด้วยแบบจำลองฮิดเดนมาร์คอฟด้วยข้อมูลจากตัวอักษรและสระระดับกลางของกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ	53
ตารางที่ 4.7 ผลการทดสอบระบบรู้จำด้วยแบบจำลองฮิดเดนมาร์คอฟด้วยข้อมูลจากตัวอักษรและสระระดับกลางของกลุ่มผู้เขียนอิสระ	54
ตารางที่ 4.8 กลุ่มของชนิดที่มีการตัดสินใจสับสนที่ได้จากกลุ่มผู้เขียนที่ใช้ในการฝึกฝน.....	56
ตารางที่ 4.9 กลุ่มของชนิดที่มีการตัดสินใจสับสนที่ได้จากกลุ่มผู้เขียนอิสระ	57
ตารางที่ 4.10 อัตราจำเฉลี่ยของระบบรู้จำตัวอักษรภาษาไทยระดับกลางด้วยระบบรู้จำที่นำเสนอโดยใช้กลุ่มชนิดที่มีการตัดสินใจสับสนที่ได้จากกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ.....	59
ตารางที่ 4.11 ผลการทดสอบระบบรู้จำที่นำเสนอด้วยข้อมูลจากตัวอักษรและสระระดับกลางของกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ.....	60
ตารางที่ 4.12 ผลการทดสอบระบบรู้จำที่นำเสนอด้วยข้อมูลจากตัวอักษรและสระระดับกลางของกลุ่มผู้เขียนอิสระ.....	61
ตารางที่ 4.13 อัตราจำเฉลี่ยของระบบรู้จำตัวอักษรภาษาไทยระดับกลางด้วยระบบรู้จำที่นำเสนอโดยใช้กลุ่มชนิดที่มีการตัดสินใจสับสนที่ได้จากกลุ่มผู้เขียนอิสระ.....	62

ตารางที่ 4.14	ผลการทดสอบระบบรู้งำนานำเสนอด้วยข้อมูลจากตัวอักษรและสระระดับกลาง ของกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ.....	63
ตารางที่ 4.15	ผลการทดสอบระบบรู้งำนานำเสนอด้วยข้อมูลจากตัวอักษรและสระระดับกลาง ของกลุ่มผู้เขียนอิสระ.....	64
ตารางที่ 4.16	ผลการทดสอบระบบรู้งำนานำเสนอโดยใช้ข้อมูลจากสระระดับกลางของกลุ่มผู้เขียนที่ ใช้ในการฝึกฝนระบบ.....	65
ตารางที่ 4.17	ผลการทดสอบระบบรู้งำนานำเสนอโดยใช้ข้อมูลจากสระระดับกลางของกลุ่มผู้เขียน อิสระ.....	65
ตารางที่ 4.18	ผลการทดสอบระบบรู้งำนานำเสนอโดยใช้ข้อมูลจากสระและวรรณยุกต์ระดับบน ของกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ.....	67
ตารางที่ 4.19	ผลการทดสอบระบบรู้งำนานำเสนอโดยใช้ข้อมูลจากสระและวรรณยุกต์ระดับบน ของกลุ่มผู้เขียนอิสระ.....	68
ตารางที่ 4.20	ผลการเปรียบเทียบอัตรารู้งำนตัวอักษรระดับต่าง ๆ ระหว่างระบบรู้งำนานำเสนอกับ ระบบรู้งำนด้วยแบบจำลองฮิดเดนมาร์คอฟพื้นฐาน.....	69
ตารางที่ 4.21	ร้อยละของอัตราการรู้งำนเฉลี่ยของระบบรู้งำนานำเสนอที่เพิ่มขึ้นจากระบบรู้งำนด้วย แบบจำลองฮิดเดนมาร์คอฟพื้นฐาน.....	69
ตารางที่ 4.22	ร้อยละของอัตราการรู้งำนของกลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่ทิศทาง เขียนของหัว.....	70
ตารางที่ 4.23	ร้อยละของอัตราการรู้งำนของกลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่ความกว้าง.....	71
ตารางที่ 4.24	ร้อยละของอัตราการรู้งำนของกลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่รอยหยัก.....	72
ตารางที่ 4.25	ร้อยละของอัตราการรู้งำนของกลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่วิธีการเขียน หาง.....	73
ตารางที่ 4.26	ร้อยละของอัตราการรู้งำนของกลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันทั้งรอยหยักและ วิธีการเขียนหาง.....	74
ตารางที่ 4.27	ร้อยละของอัตราการรู้งำนของกลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่สัดส่วนของหัว	75
ตารางที่ 4.28	ร้อยละของอัตราการรู้งำนของกลุ่มตัวอักษรที่ตำแหน่งจุดต้น จุดปลาย และทิศ ทางการวนคล้ายกัน.....	76

สารบัญภาพ

	หน้า
รูปที่ 2.1 ตัวอย่างตัวอักษรที่มีหาง และกรอบตัวอักษรที่ไม่รวมหาง.....	5
รูปที่ 2.2 (ก) แผนภูมิลำดับการคำนวณตัวแปรไปข้างหน้า (ข) แผนภูมิในการสร้างการคำนวณ..	8
รูปที่ 2.3 แผนภูมิลำดับการคำนวณตัวแปรย้อนกลับ	9
รูปที่ 2.4 (ก) มิติ VC ใน \mathbb{R}^2 กรณี 3 ตัวอย่างใน \mathbb{R}^2 (ข) กรณี 4 ตัวอย่างใน \mathbb{R}^2	15
รูปที่ 2.5 ไฮเปอร์เพลนแบบเชิงเส้นสำหรับกรณีแยกกันได้.....	16
รูปที่ 2.6 ไฮเปอร์เพลนที่เหมาะสมที่สุด	18
รูปที่ 2.7 ไฮเปอร์เพลนแบบเชิงเส้นสำหรับกรณีแยกกันไม่ได้	20
รูปที่ 2.8 โครงสร้างข้อมูลแบบไม่เป็นเชิงเส้น	21
รูปที่ 2.9 การส่งปริภูมิขาเข้า (ซ้าย) ไปสู่ ปริภูมิลักษณะ (ขวา)	22
รูปที่ 2.10 ฟังก์ชันเคอร์เนลแบบ Polynomial (ซ้าย) และ แบบ RBF (ขวา)	23
รูปที่ 3.1 แผนภาพการทำงานระบบรู้จำตัวอักษรลายมือเขียนภาษาไทยแบบออนไลน์ด้วย แบบจำลองฮิดเดนมาร์คอฟและซัพพอร์ตเวกเตอร์แมชชีน	33
รูปที่ 3.2 มุมที่พบว่าเป็นมุมแหลม	34
รูปที่ 3.3 มุมระหว่างเส้นสัมผัสของจุดที่พิจารณากับเส้นสัมผัสของจุดแรกของเส้นย่อย	35
รูปที่ 3.4 มุมระหว่างเส้นที่ลากจากจุดแรกของเส้นย่อยถึงจุดที่พิจารณากับเส้นสัมผัสที่จุดแรก ของเส้นย่อย	35
รูปที่ 3.5 ตัวอย่างลายมือเขียนที่ถูกแบ่งส่วนของเส้นตามข้อกำหนดต่าง ๆ	35
รูปที่ 3.6 คุณลักษณะสำคัญของเส้นย่อย.....	36
รูปที่ 3.7 ตัวอย่างเส้นตรงที่ลากจากจุดศูนย์กลางของเส้นย่อยถึงจุดศูนย์กลางของเส้นย่อย สุดท้าย	37
รูปที่ 3.8 การนอร์มัลไลซ์เชิงทรงกลม.....	42
รูปที่ 3.9 แผนภาพการทำงานระบบรู้จำตัวอักษรลายมือเขียนภาษาไทยระดับบนแบบออนไลน์ ด้วยแบบจำลองฮิดเดนมาร์คอฟแบบ 2 ชั้นตอน	46
รูปที่ 4.1 แบบจำลองฮิดเดนมาร์คอฟที่ใช้ในวิทยานิพนธ์	49
รูปที่ 4.2 อัตรการรู้จำเฉลี่ยของกลุ่มสระ $\bar{}$, $\bar{}$, $\bar{}$ และ $\bar{}$ ของกลุ่มผู้เขียนอิสระเมื่อใช้จำนวน สถานะต่าง ๆ.....	66

รูปที่ 4.3	แผนภูมิเปรียบเทียบอัตราการเรียนรู้จำตัวอักษรระดับกลางของข้อมูลตัวอักษรจากกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ.....	77
รูปที่ 4.4	แผนภูมิเปรียบเทียบอัตราการเรียนรู้จำตัวอักษรระดับกลางของข้อมูลตัวอักษรจากกลุ่มผู้เขียนอิสระ.....	78
รูปที่ 4.5	แผนภูมิเปรียบเทียบอัตราการเรียนรู้จำตัวอักษรระดับบนของข้อมูลตัวอักษรจากกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ.....	79
รูปที่ 4.6	แผนภูมิเปรียบเทียบอัตราการเรียนรู้จำตัวอักษรระดับบนของข้อมูลตัวอักษรจากกลุ่มผู้เขียนอิสระ.....	79
รูปที่ 4.7	แผนภูมิเปรียบเทียบอัตราการเรียนรู้จำตัวอักษรระดับล่างของข้อมูลตัวอักษรจากกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ.....	80
รูปที่ 4.8	แผนภูมิเปรียบเทียบอัตราการเรียนรู้จำตัวอักษรระดับล่างของข้อมูลตัวอักษรจากกลุ่มผู้เขียนอิสระ.....	80

บทที่ 1

บทนำ

1.1 ประวัติและความเป็นมา

การสื่อสารมีความสำคัญต่อมนุษย์อย่างมาก การที่มนุษย์จะทำความเข้าใจกันได้นั้นต้องอาศัยการสื่อสาร เช่น การพูด การเขียน เป็นต้น ในอดีตการสื่อสารนั้นจะเป็นการสื่อสารในระยะใกล้เท่านั้นเมื่อเวลาผ่านไปเทคโนโลยีของมนุษย์ได้เจริญก้าวหน้ามากขึ้นรวมถึงเทคโนโลยีที่ใช้ในการสื่อสาร จากการสื่อสารกันในระยะใกล้จนมีอุปกรณ์ที่ทำให้ระยะสื่อสารไกลเช่นโทรศัพท์ขึ้น จนถึงปัจจุบันซึ่งเป็นยุคแห่งข้อมูลข่าวสารไร้พรมแดนโดยมีคอมพิวเตอร์เป็นอุปกรณ์หนึ่งที่ใช้เข้ามาจัดการกับข้อมูลข่าวสารเหล่านี้ และมีแนวโน้มว่าคอมพิวเตอร์นั้นจะมีขนาดเล็กลงเรื่อย ๆ ในปัจจุบันมีเครื่องช่วยงานส่วนบุคคลแบบดิจิทัล (Personal Digital Assistants) ซึ่งมีขนาดเล็กและใช้ระบบรับข้อมูลผ่านทางหน้าจอ หรือใช้เสียงสั่งแทนการใช้แป้นพิมพ์ ดังนั้นจึงต้องมีการพัฒนาวิธีการเปลี่ยนรูปแบบข้อมูลเหล่านั้นให้เป็นข้อมูลที่คอมพิวเตอร์สามารถทำการจัดการได้ การรู้จำแบบรูป (Pattern Recognition) เช่น การรู้จำตัวอักษร การรู้จำเสียงพูด เป็นต้น จึงเข้ามามีบทบาทสำคัญ

งานทางด้านการรู้จำตัวอักษรแบ่งเป็น 2 ประเภทได้แก่ ตัวพิมพ์ และลายมือเขียน ซึ่งทั้งสองประเภทแบ่งได้เป็นแบบออฟไลน์ และแบบออนไลน์ ซึ่งทั้งสองแบบนี้มีข้อแตกต่างตรงที่แบบออนไลน์สามารถนำข้อมูลที่เกิดขึ้นในระหว่างการเขียน เช่น แนวทางเดินของปากกา, ความเร็ว, ความแรงของการเขียน แรงกด และลำดับในการเขียน เป็นต้น มาทำการรู้จำได้ ในขณะที่แบบออฟไลน์จะใช้เพียงจุดภาพที่ได้จากเครื่องสแกนเนอร์ หรือกล้องดิจิทัลมาทำการรู้จำ

ในอนาคตงานแบบออนไลน์จะเข้ามามีบทบาทอย่างมาก เช่น การเรียนการสอนที่ใช้กระดาษอิเล็กทรอนิกส์ (Electronic Paper) แทนกระดาษ การเขียนสมการ หรือข้อความใด ๆ ลงไป คอมพิวเตอร์ก็สามารถแปลงเป็นแฟ้มข้อความ (text file) โดยไม่ต้องเก็บเป็นแฟ้มภาพที่ใช้เนื้อที่มากกว่า

จากงานวิจัยที่ผ่านมาการรู้จำตัวอักษรทั้งแบบออนไลน์และออฟไลน์นั้น ลักษณะบ่งความต่าง (Distinctive Feature) การเข้าคู่แม่แบบ (Template matching) ฟัซซีโลจิก (Fuzzy Logic) และโครงข่ายประสาทเทียม (Artificial Neural Network) ถูกนำมาประยุกต์ใช้ในระบบ ในปัจจุบันจากความสำเร็จที่ได้ในการใช้แบบจำลองฮิดเดนมาร์คอฟ (Hidden Markov Model, HMM) ในการรู้จำเสียงพูด (Speech Recognition) [1] ทำให้แบบจำลองฮิดเดนมาร์คอฟถูกนำมาใช้กัน

อย่างกว้างขวางในงานด้านรู้จำตัวอักษรด้วย เนื่องด้วยความคล้ายคลึงในธรรมชาติของสัญญาณที่มีการเปลี่ยนแปลงเชิงเวลา (Time-variant) ซึ่งในงานรู้จำภาษาไทยที่ได้มีการประยุกต์ใช้แบบจำลองฮิดเดนมาร์คอฟกับพีซีโลจิกประสบความสำเร็จอย่างดี เช่น งานวิจัยของ R. Budsayaparakorn [2] ที่มีอัตราการรู้จำเกินร้อยละ 90 เป็นต้น

ปัจจุบันงานวิจัยด้านการรู้จำเสียงพูดได้มีการนำซัพพอร์ตเวกเตอร์แมชชีน (SVM) มาใช้ ได้ผลที่ดีขึ้นกว่า แบบจำลองฮิดเดนมาร์คอฟ เช่น งานวิจัยของ H. Shimodairay, K. Nomay M. Nakaiy และ S. Sagayama [3] ได้มีการปรับปรุงเสนอเคอร์เนลขึ้นใหม่ได้อัตราจำมากกว่าร้อยละ 98 เมื่อเปรียบเทียบกับงานเดียวกันที่ใช้แบบจำลองฮิดเดนมาร์คอฟพบว่ามีอัตราจำเพียงร้อยละ 93 เท่านั้น นอกจากนี้ยังมีการนำไปใช้ในการรู้จำผู้พูดใน [4] และยังมีมีการนำซัพพอร์ตเวกเตอร์แมชชีนไปประยุกต์ใช้กับแบบจำลองฮิดเดนมาร์คอฟใน [5] เพื่อรู้จำเสียงพูดอีกด้วย

ในงานรู้จำตัวอักษรได้มีการเริ่มใช้ซัพพอร์ตเวกเตอร์แมชชีนเช่นกัน เช่น งานวิจัยการรู้จำแบบออนไลน์ ของ C. Bahlmann B. Haasdonk และ H. Burkhardt [6] มีความผิดพลาดร้อยละ 7.84 ขณะที่แบบจำลองฮิดเดนมาร์คอฟให้ความผิดพลาดร้อยละ 7.9 ในงานวิจัยการรู้จำแบบออฟไลน์ของ A. R. Ahmad, M Khalid, and R. Yusof [7]

แต่โดยโครงสร้างมาตรฐานของซัพพอร์ตเวกเตอร์แมชชีนมีลักษณะเป็นตัวจำแนกประเภทแบบสถิตย์ (Static Classifier) จึงไม่เหมาะสมที่จะนำมาใช้กับข้อมูลแบบจลน์ (Dynamic data) อย่างเช่น ตัวอักษร หรือเสียงพูด ในขณะที่แบบจำลองฮิดเดนมาร์คอฟเหมาะสมอย่างยิ่งกับงานแบบจลน์ และจากความสำเร็จของงานวิจัยที่กล่าวมาข้างต้นรวมกับข้อได้เปรียบของทั้งซัพพอร์ตเวกเตอร์แมชชีน และแบบจำลองฮิดเดนมาร์คอฟ ดังเหตุผลที่กล่าวมาวิทยานิพนธ์ฉบับนี้จึงมีแนวความคิดที่จะนำซัพพอร์ตเวกเตอร์แมชชีนและแบบจำลองฮิดเดนมาร์คอฟมาประยุกต์ใช้ร่วมกันในระบบรู้จำตัวอักษรภาษาไทยแบบออนไลน์

1.2 วัตถุประสงค์

เพื่อพัฒนาวิธีการรู้จำตัวอักษรเขียนภาษาไทยแบบออนไลน์ที่เป็นตัวอักษรตัวเดียวที่ได้จากคำในภาษาไทย โดยใช้การทำงานร่วมกันระหว่างแบบจำลองฮิดเดนมาร์คอฟและซัพพอร์ตเวกเตอร์แมชชีน

1.3 เป้าหมายและขอบเขตของการวิจัย

1. สร้างระบบรู้จำลายมือเขียนตัวอักษรภาษาไทยแบบออนไลน์

2. แบ่งตัวอักษรภาษาไทยที่จะรู้จำตามระดับของตัวอักษรออกเป็น 3 กลุ่มดังตารางที่ 1.1
3. อัตราความถูกต้องเฉลี่ยในแต่ละกลุ่มไม่ต่ำกว่าร้อยละ 90 โดยเก็บตัวอย่างผู้เขียนอิสระจำนวน 65 คน

ตารางที่ 1.1 ตัวอักษรภาษาไทยแบ่งตามระดับ

ระดับของตัวอักษร	ตัวอักษร
ระดับบน	ั ำ ๅ ๆ ๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ +
ระดับกลาง	ก ข ค ฅ ง จ ฉ ช ซ ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ ท ธ น บ ป ผ ฝ พ ฟ ภ ม ย ร ล ว ศ ษ ส ห ฬ อ ฮ ะ ๅ ๖ ๗ ๘ ๙ ใ โ
ระดับล่าง	๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙

1.4 ขั้นตอนและวิธีการดำเนินงาน

1. ศึกษางานวิจัยที่เกี่ยวข้องกับการรู้จำตัวอักษร
2. ศึกษาการเขียนโปรแกรม Visual C++ และ MATLAB
3. เขียนโปรแกรมทดสอบระบบ
4. จัดเก็บตัวอย่างอักษรเขียนอักษรภาษาไทย
5. วิเคราะห์ผลที่ได้จากการทดสอบ และแก้ไขข้อผิดพลาด
6. สรุป และรวบรวมข้อมูลทั้งหมดพร้อมทั้งจัดทำรูปเล่มวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ทราบถึงกระบวนการที่เหมาะสมสำหรับระบบรู้จำลายมือเขียนภาษาไทยแบบออนไลน์
2. ทำให้สามารถวางแนวทางที่เหมาะสมต่อไปในการพัฒนาระบบรู้จำลายมือเขียนภาษาไทยแบบออนไลน์
3. สามารถนำไปประยุกต์ใช้กับคอมพิวเตอร์พกพาได้

บทที่ 2

ทฤษฎีและแนวคิด

ในบทนี้จะกล่าวถึงทฤษฎีที่เกี่ยวข้องกับการสร้างระบบรู้จำที่รวมข้อดีของการรู้จำด้วยแบบจำลองแบบเจเนอเรทีฟกับระบบรู้จำที่ถูกฝึกฝนแบบดิสคริมิเนทีฟไว้ด้วยกัน ซึ่งในที่นี้คือระบบรู้จำที่ประกอบด้วยระบบรู้จำ 2 ชนิด คือ แบบจำลองฮิดเดนมาร์คอฟซึ่งเป็นแบบจำลองแบบเจเนอเรทีฟและซัพพอร์ตเวกเตอร์แมชชีนซึ่งเป็นระบบรู้จำที่ถูกฝึกฝนแบบดิสคริมิเนทีฟโดยเริ่มจากการอธิบายถึงกระบวนการก่อนหน้า (Preprocessing) ทฤษฎีของระบบรู้จำทั้งสอง รวมถึงทฤษฎีของปริภูมิคะแนน (Score-spaces) ซึ่งเป็นปริภูมิที่ทำหน้าที่ส่ง (Mapping) ข้อมูลจากแบบจำลองฮิดเดนมาร์คอฟไปให้ซัพพอร์ตเวกเตอร์แมชชีน และเทคนิคการทำมัลติคลาส (Multi-class) เพื่อให้ซัพพอร์ตเวกเตอร์แมชชีนสามารถใช้งานกับการรู้จำแบบมัลติคลาสได้

2.1 กระบวนการก่อนหน้า

ข้อมูลของตัวอักษรแบบออนไลน์ซึ่งใช้ในวิทยานิพนธ์ฉบับนี้ คือ ลำดับของพิกัด (x, y) บนจอภาพที่ถูกดิจิทัลโดยอุปกรณ์ป้อนเข้า โดยจะสนใจเฉพาะพิกัดที่อยู่บนเส้นของตัวอักษรและลำดับที่ยกปากกาขึ้นเท่านั้น ไม่รวมถึงพิกัดขณะที่ยกปากกา แรกกดปากกา ฯลฯ โดยใช้ขั้นตอนของ Nair และ Leedham [8] ซึ่งมีขั้นตอนดังต่อไปนี้

1. ลบการยกปากกาโดยบังเอิญ
2. ทำเส้นให้เรียบ (Smoothing)
3. ทำการกรองระยะทางที่สั้น (Minimum Distance Filtering)
4. ลบส่วนที่เกิดขึ้นโดยไม่ตั้งใจเวลาลงปากกา หรือยกปากกาขึ้น (Serif Removal)
5. ทำนอร์มัลไลซ์ (Normalization)

สำหรับการทำนอร์มัลไลซ์ในภาษาไทยนั้นจะพบตัวอักษรที่มีหางเกินเส้นบรรทัดการขยายขนาดตัวอักษรให้มีความสูงเท่ากันนั้นควรจะสนใจความสูงของตัวอักษรที่ไม่รวมส่วนหาง ดังนั้นเราจึงหากรอบของตัวอักษรเดี่ยวโดยที่หางจะไม่อยู่ในกรอบเมื่อเราพบว่าตัวอักษรมีโอกาสเป็นหางหรือจุดปลายของตัวอักษรอยู่ด้านบนขวาเราจะพิจารณากรอบด้านบนให้สูงเท่ากับจุดที่สูงที่สุดในส่วนครึ่งด้านซ้ายของตัวอักษรแทนตัวอย่างการหากรอบตัวอักษรแสดงในรูปที่ 2.1



รูปที่ 2.1 ตัวอย่างตัวอักษรที่มีหาง และกรอบตัวอักษรที่ไม่รวมหาง

2.2 แบบจำลองฮิดเดนมาร์คอฟ

เมื่อพิจารณาถึงแบบจำลองมาร์คอฟระบบหนึ่ง ซึ่งไม่สามารถระบุได้ชัดเจนว่า ในขณะที่ระบบอยู่ในสถานะใด หากแต่เราสามารถสังเกตปรากฏการณ์อื่น ๆ เพื่อนำมาทำนายความน่าจะเป็นของสถานะในแบบจำลองมาร์คอฟที่เวลาใด ๆ ได้ เรียกแบบจำลองดังกล่าวเป็นแบบจำลองฮิดเดนมาร์คอฟ (Hidden Markov Model) [1] โดยเราจะพิจารณาถึง แบบจำลองฮิดเดนมาร์คอฟแบบไม่ต่อเนื่อง (Discrete HMM) ก่อน เพื่อทำความเข้าใจในหลักการการทำงาน แล้วจะกล่าวถึงแบบจำลองฮิดเดนมาร์คอฟแบบต่อเนื่อง (Continuous HMM) ในภายหลัง

2.2.1 องค์ประกอบของแบบจำลองฮิดเดนมาร์คอฟแบบไม่ต่อเนื่อง

1. จำนวนสถานะในแบบจำลอง (N) และกำหนดตัวแปรแทนสถานะในแบบจำลองเป็น $S = \{S_1, S_2, \dots, S_N\}$
2. จำนวนของสัญลักษณ์จากการสังเกตต่อสถานะ (M) และกำหนดตัวแปรแทนสัญลักษณ์เป็น $V = \{v_1, v_2, \dots, v_M\}$
3. ความน่าจะเป็นของการเปลี่ยนสถานะ $A = \{a_{ij}\}$ โดย

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i], \quad 1 \leq i, j \leq N \quad (2.1)$$

4. การกระจายความน่าจะเป็นของสัญลักษณ์ที่สังเกตเมื่อทราบสถานะ j , $B = \{b_j(k)\}$ โดย

$$b_j(k) = P(v_k \text{ at } t | q_t = S_j), \quad 1 \leq j \leq N \quad (2.2)$$

5. การกระจายความน่าจะเป็นของสถานะเริ่มต้น $\pi = \{\pi_i\}$ เมื่อ

$$\pi_i = P(q_1 = S_j), \quad 1 \leq i \leq N \quad (2.3)$$

การอธิบายแบบจำลองฮิตเดนมาร์คอฟต้องระบุ N และจำนวนสัญลักษณ์ M ของแบบจำลอง สามารถเขียนองค์ประกอบของความน่าจะเป็นรวมกันแทนแบบจำลองได้ดังนี้

$$\lambda = (A, B, \pi) \quad (2.4)$$

2.2.2 การใช้แบบจำลองฮิตเดนมาร์คอฟในการรู้จำ

แบบจำลองฮิตเดนมาร์คอฟนิยมนำมาประยุกต์ใช้ในการรู้จำรูปแบบ [9] โดยในการรู้จำรูปแบบที่ต้องการจำแนกประเภทของสิ่งที่เราสังเกตออกเป็น n ชนิด แบ่งเป็น $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n$ จากลำดับของค่าสังเกต \bar{O} สิ่งที่เราต้องการทราบคือความน่าจะเป็นของสิ่งที่เราสังเกตใกล้เคียงชนิดใด i , $P(\lambda_i | \bar{O})$ โดยตัดสินใจว่าสิ่งที่เราสนใจเป็นชนิดที่ i เมื่อ

$$P(\lambda_i | \bar{O}) \geq P(\lambda_j | \bar{O}), \quad \forall 1 \leq j \leq n \quad (2.5)$$

จากกฎของเบย์ (Baye's Rule) จะได้

$$P(\lambda_i | \bar{O}) = \frac{P(\bar{O} | \lambda_i)P(\lambda_i)}{P(\bar{O})} \quad (2.6)$$

หากตั้งสมมุติฐานว่าทุกประเภทมีโอกาสเกิดขึ้นเท่ากัน จะพบว่า

$$P(\lambda_i | \bar{O}) \propto P(\bar{O} | \lambda_i) \quad (2.7)$$

ดังนั้นสิ่งที่อยากทราบอันดับแรกคือ $P(\bar{O} | \lambda_i)$ เมื่อทราบแบบจำลองฮิตเดนมาร์คอฟแล้ว สิ่งที่น่าสนใจอันดับต่อไปคือลำดับของสถานะที่แท้จริงที่เกิดขึ้นตั้งแต่เวลาที่ 1 จนถึง เวลา T , $Q = q_1 q_2 \dots q_T$ เมื่อเราทราบลำดับของค่าสังเกต, $\bar{O} = \bar{o}_1 \bar{o}_2 \dots \bar{o}_T$ เนื่องจากเป็นส่วนที่ซ่อนอยู่ในแบบจำลอง อันดับสุดท้ายคือจะสร้างแบบจำลองอย่างไรเพื่อให้ $P(\bar{O} | \lambda_i)$ มีค่ามากที่สุด โดย

การแก้ปัญหาทั้ง 3 ประการจะใช้กระบวนการตามบทความของ L.R. Rabiner [1] ดังที่จะกล่าวใน ส่วนต่อไป

2.2.2.1 การคำนวณหาความน่าจะเป็นของลำดับของค่าสังเกตเมื่อทราบแบบจำลอง

เมื่อทราบลำดับของค่าสังเกต $\bar{O} = \bar{o}_1 \bar{o}_2 \dots \bar{o}_T$ และแบบจำลอง λ ตามลำดับ และ กำหนดให้มีลำดับของสถานะดังนี้

$$P(\bar{O} | Q, \lambda) = \prod_{t=1}^T P(\bar{o}_t | q_t, \lambda) \quad (2.8)$$

หากเราพิจารณาในกรณีที่ชุดสังเกตแต่ละชุดเป็นอิสระต่อกันทางสถิติจะได้

$$P(\bar{O} | Q, \lambda) = b_{q_1}(\bar{o}_1) \cdot b_{q_2}(\bar{o}_2) \cdots b_{q_T}(\bar{o}_T) \quad (2.9)$$

และ

$$P(Q | \lambda) = \pi_{q_1} \cdot a_{q_1 q_2} \cdot a_{q_2 q_3} \cdots a_{q_{T-1} q_T} \quad (2.10)$$

จาก

$$P(\bar{O}, Q | \lambda) = P(\bar{O} | Q, \lambda) P(Q | \lambda) \quad (2.11)$$

จะพบว่า

$$\begin{aligned} P(\bar{O} | \lambda) &= \sum_{\forall Q} P(\bar{O} | Q, \lambda) P(Q | \lambda) \\ &= \sum_{q_1, q_2, \dots, q_T} \pi_{q_1} b_{q_1}(\bar{o}_1) a_{q_1 q_2} b_{q_2}(\bar{o}_2) \cdots a_{q_{T-1} q_T} b_{q_T}(\bar{o}_T) \end{aligned} \quad (2.12)$$

เมื่อพิจารณาสมการที่ (2.12) จะพบว่าจะต้องการคูณกันถึง $(2T-1)N^T$ ครั้ง ดังนั้นจึง ไม่นิยมนำสมการ (2.12) ไปใช้กันในทางปฏิบัติโดยตรง เนื่องจากมีกระบวนการที่ทำงานได้เร็วกว่า คือ กระบวนการไปข้างหน้าและย้อนกลับ (Forward-Backward Procedure)

กำหนดตัวแปรไปข้างหน้า (Forward Variable: $\alpha_i(t)$) ดังนี้

$$\alpha_i(t) = P(\bar{o}_1 \bar{o}_2 \dots \bar{o}_t, q_t = S_i | \lambda) \quad (2.13)$$

กระบวนการไปข้างหน้าทำได้ดังนี้

1. เริ่มต้นด้วย

$$\alpha_i(1) = \pi_i b_i(\bar{o}_1), \quad 1 \leq i \leq N \quad (2.14)$$

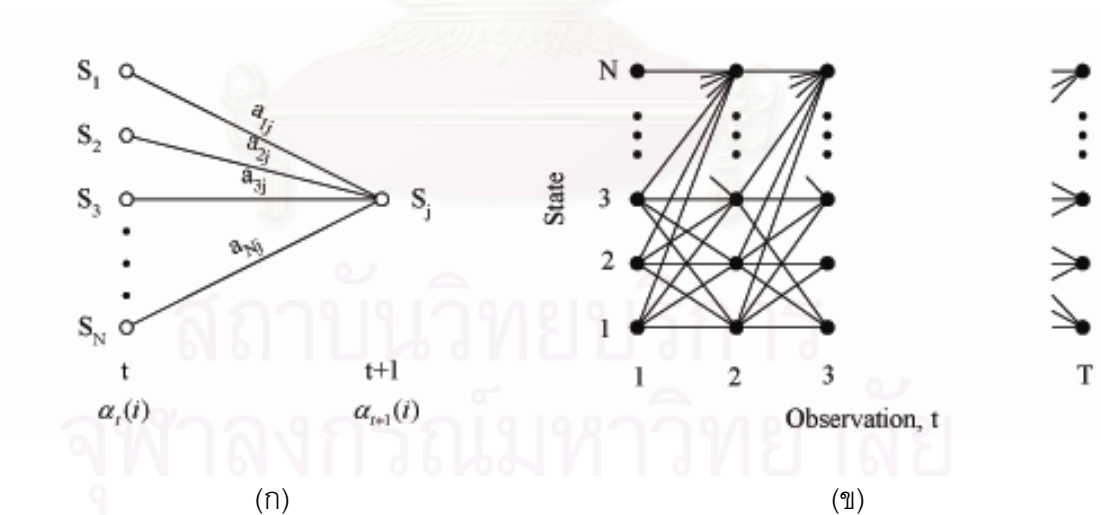
2. คำนวณหาค่าถัดไปตามสมการ

$$\alpha_j(t+1) = b_j(\bar{o}_{t+1}) \sum_{i=1}^N \alpha_i(t) a_{ij}, \quad 1 \leq t \leq T-1, 1 \leq j \leq N \quad (2.15)$$

3. จะได้ผลลัพธ์สุดท้ายคือ

$$P(\bar{O} | \lambda) = \sum_{i=1}^N \alpha_i(T) \quad (2.16)$$

จะพบว่าการคำนวณตามกระบวนการข้างต้นใช้การคำนวณทั้งหมดเพียง $N^2(T-1)$ ครั้ง โดยการคำนวณสามารถแสดงได้เป็นโครงสร้างผลึกตามรูปที่ 2.2



รูปที่ 2.2 (ก) แผนภูมิลำดับการคำนวณตัวแปรไปข้างหน้า (ข) แผนภูมิในการสร้างการคำนวณ

และในการทำงานเดียวกันกับกระบวนการไปข้างหน้า กระบวนการย้อนกลับจะกำหนดตัวแปรย้อนกลับ (Backward Variable: $\beta_i(t)$) ดังนี้

$$\beta_i(t) = P(\bar{o}_{t+1}\bar{o}_{t+2}\dots\bar{o}_T | q_t = S_i, \lambda) \quad (2.17)$$

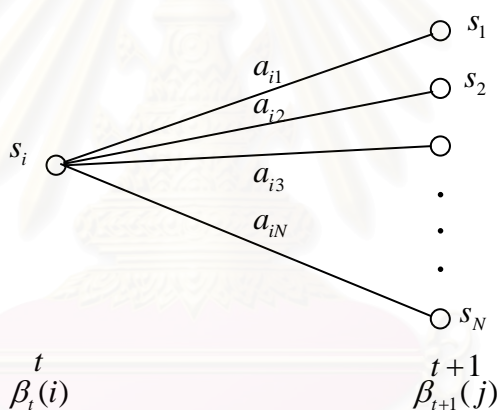
1. ตั้งค่าเริ่มต้น

$$\beta_i(T) = 1 \quad (2.18)$$

2. หาค่าตัวถัดไปจาก

$$\beta_i(t) = \sum_{j=1}^N a_{ij} b_j(\bar{o}_{t+1}) \beta_j(t+1), \quad t = T-1, T-2, \dots, 1; 1 \leq i \leq N \quad (2.19)$$

จะพบว่าการคำนวณตามกระบวนการข้างต้นใช้การคำนวณทั้งหมดเพียง N^2T ครั้ง โดย
การคำนวณสามารถแสดงได้เป็นโครงสร้างผลึกตามรูปที่ 2.3



รูปที่ 2.3 แผนภูมิลำดับการคำนวณตัวแปรย้อนกลับ

2.2.2.2 การหาลำดับของสถานะที่เหมาะสม

การเลือกสถานะที่เหมาะสมควรเลือกสถานะที่น่าจะเป็นไปได้ในทางสถิติมากที่สุด โดยจะตั้งตัวแปรขึ้นหนึ่งตัวซึ่งแสดงถึงความน่าจะเป็นที่จะเกิดสถานะ i ที่เวลา t เมื่อข้อมูลที่สังเกตได้คือ O ดังนี้

$$\gamma_i(t) = P(q_t = S_i | O, \lambda) \quad (2.20)$$

หรือจะเขียนได้ในอีกรูปหนึ่งเป็น

$$\gamma_i(t) = \frac{\alpha_i(t)\beta_i(t)}{P(O|\lambda)} = \frac{\alpha_i(t)\beta_i(t)}{\sum_{i=1}^N \alpha_i(t)\beta_i(t)} \quad (2.21)$$

และมีคุณสมบัติ

$$\sum_{i=1}^N \gamma_i(t) = 1 \quad (2.22)$$

เราจะตัดสินใจว่าที่เวลา t ระบบจะมีสถานะใดโดยเราจะใช้กฎในการตัดสินใจดังนี้

$$q_t = \arg \max[\gamma_i(t)], \quad 1 \leq t \leq T \quad (2.23)$$

การเลือกสถานะโดยใช้กฎตามสมการที่ (2.23) พบว่าในบางกรณีที่มีการเปลี่ยนจากสถานะหนึ่งไปอีกสถานะหนึ่งอาจเป็นไปได้ที่จะเกิดขึ้น ($a_{ij} = 0$ สำหรับบาง i หรือ j) ดังนั้นสถานะที่เลือกอาจไม่ใช่สถานะที่ถูกต้องเสมอไป ซึ่งแก้ปัญหานี้โดยใช้อัลกอริทึมของ Viterbi ทำได้ดังต่อไปนี้

ต้องการหาลำดับของสถานะที่ดีที่สุด $Q = \{q_1 q_2 \dots q_T\}$ โดยให้

$$\delta_i(t) = \max_{q_1 q_2 \dots q_{t-1}} P(q_1 q_2 \dots q_t = i, \bar{o}_1 \bar{o}_2 \dots \bar{o}_t | \lambda) \quad (2.24)$$

พบว่า

$$\delta_j(t+1) = b_j(\bar{o}_{t+1}) \max_i (\delta_i(t) a_{ij}) \quad (2.25)$$

การหาลำดับของสถานะที่ดีที่สุดตามขั้นตอนดังต่อไปนี้
เริ่มต้นโดย

$$\delta_i(1) = \pi_i b_i(\bar{o}_1), \quad 1 \leq i \leq N \quad (2.26)$$

ทำซ้ำตามสมการด้านล่างไปทุก ๆ ค่า t

$$\delta_j(t) = b_i(\bar{o}_t) \max_{1 \leq i \leq N} (\delta_i(t-1)a_{ij}), \quad 2 \leq t \leq T, 1 \leq j \leq N \quad (2.27)$$

$$\beta_j(t) = \arg \max_{1 \leq i \leq N} (\delta_i(t-1)a_{ij}), \quad 2 \leq t \leq T, 1 \leq j \leq N \quad (2.28)$$

จะได้ว่า

$$P^* = \max_{1 \leq i \leq N} (\delta_i(T)) \quad (2.29)$$

$$Q^* = \arg \max_{1 \leq i \leq N} (\delta_i(T)) \quad (2.30)$$

2.2.2.3 การหาพารามิเตอร์ของแบบจำลองที่เหมาะสม

การหาค่าพารามิเตอร์ที่เหมาะสมนั้นมีหลายกระบวนการ โดยในวิทยานิพนธ์นี้จะเสนอกระบวนการที่เข้าใจง่าย และเป็นที่ยอมรับใช้กระบวนการหนึ่งคือ การประมาณค่าซ้ำของ Baum และ Welch เนื่องจากมั่นใจได้ว่าสามารถสร้างแบบจำลองที่เหมาะสมได้เมื่อใช้การประมาณค่าซ้ำไปจนกระทั่งได้แบบจำลองที่ดีที่สุด

เพื่ออธิบายกระบวนการนี้เราจะกำหนด $\xi_{ij}(t)$ แสดงถึงความน่าจะเป็นที่จะเกิดสถานะ S_i ที่เวลา t และเกิดสถานะ S_j ที่เวลา $t+1$ เมื่อทราบแบบจำลองและลำดับของค่าสังเกต

$$\xi_{ij}(t) = P(q_t = S_i, q_{t+1} = S_j | \bar{O}, \lambda) \quad (2.31)$$

จะพบว่า $\xi_{ij}(t)$ มีความสัมพันธ์กับตัวแปรไปข้างหน้า ($\alpha_i(t)$) และตัวแปรย้อนกลับ ($\beta_j(t)$) ดังนี้

$$\begin{aligned} \xi_{ij}(t) &= \frac{\alpha_i(t)a_{ij}b_j(\bar{o}_{t+1})\beta_j(t+1)}{P(\bar{O} | \lambda)} \\ &= \frac{\alpha_i(t)a_{ij}b_j(\bar{o}_{t+1})\beta_j(t+1)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_i(t)a_{ij}b_j(\bar{o}_{t+1})\beta_j(t+1)} \end{aligned} \quad (2.32)$$

จากในหัวข้อที่แล้ว $\gamma_i(t)$ ซึ่งแสดงถึงความน่าจะเป็นที่จะเกิดสถานะ i ที่เวลา t มีความสัมพันธ์กับ $\xi_{ij}(t)$ ดังนี้

$$\gamma_i(t) = \sum_{j=1}^N \xi_{ij}(t) \quad (2.33)$$

เมื่อรวม $\gamma_i(i)$ ทุก ๆ เวลา t จะได้ปริมาณที่แสดงถึงค่าคาดหวัง (Expected value) ของจำนวนครั้งที่เกิดสถานะ S_i และการรวม $\xi_{ij}(t)$ ทุก ๆ เวลา t หมายถึงค่าคาดหวังของจำนวนครั้งที่เปลี่ยนสถานะจาก S_i ไปยังสถานะ S_j โดยจะสรุปได้ดังนี้

$$\sum_{t=1}^{T-1} \gamma_i(t) = \text{expected number of transitions from } S_i \quad (2.34)$$

$$\sum_{t=1}^{T-1} \xi_{ij}(t) = \text{expected number of transitions from } S_i \text{ to } S_j \quad (2.35)$$

จากสมการดังกล่าวจะสามารถประมาณค่าของพารามิเตอร์ของแบบจำลองฮิดเดนมาร์คอฟได้ดังนี้

$$\bar{\pi}_i = \text{expected frequency in state } S_i \text{ at time } (t=1) = \gamma_i(1) \quad (2.36)$$

$$\bar{a}_{ij} = \frac{\text{expected number of transition from } S_i \text{ to } S_j}{\text{expected number of transitions from } S_i} = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)} \quad (2.37)$$

$$\bar{b}_j(k) = \frac{\text{expected number of times in state } S_j \text{ and observing symbol } v_k}{\text{expected number of times in state } S_j} \quad (2.38)$$

$$= \frac{\sum_{t=1}^{T-1} \gamma_j(t)}{\sum_{t=1}^{T-1} \gamma_j(t)} \quad (2.38)$$

เมื่อกำหนดให้สถานะปัจจุบันเป็น $\lambda = (A, B, \pi)$ และคำนวณตามสมการที่ (2.36) ถึง (2.38) จะได้แบบจำลองที่เกิดจากการประมาณค่าเป็น $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ และหากนำแบบจำลองที่ได้มาใหม่ $\bar{\lambda}$ ไปแทนที่ λ และนำไปคำนวณตามสมการที่ (2.36) ถึง (2.38) ซ้ำไปเรื่อย ๆ จากการพิสูจน์ของ Baum และทีมงานพบว่าแบบจำลองที่ประมาณค่าได้ยิ่งคำนวณซ้ำหลายรอบจะมีค่ามากขึ้นจนเข้าใกล้จุดลิมิตค่าหนึ่งซึ่งเป็นจุดที่ $P(\bar{O} | \lambda)$ จะมีค่ามากที่สุด

2.2.3 แบบจำลองฮิดเดนมาร์คอฟแบบต่อเนื่อง

จากแบบจำลองฮิดเดนมาร์คอฟแบบไม่ต่อเนื่องที่ได้กล่าวมาข้างต้น ที่มีการพิจารณาเพียงสัญลักษณ์แบบไม่ต่อเนื่องที่เลือกมาจากชุดสัญลักษณ์ที่มีจำนวนจำกัด ในกรณีที่ข้อมูลสังเกตเป็นสัญญาณแบบต่อเนื่องจะนำข้อมูลนั้นมาควอนไทซ์ เพื่อให้ได้รูปแบบของสัญลักษณ์ที่จำกัด ในบางครั้งการควอนไทซ์นั้น ทำให้เกิดความผิดพลาดในการแทนสัญลักษณ์ของข้อมูลสังเกตได้ ดังนั้นการใช้แบบจำลองฮิดเดนมาร์คอฟที่มีความหนาแน่นของข้อมูลสังเกตแบบต่อเนื่องจึงให้ผลลัพธ์ที่ดีกว่า [15]

การนำความหนาแน่นของชุดข้อมูลสังเกตแบบต่อเนื่องมาใช้งาน โดยใช้รูปแบบของฟังก์ชันความหนาแน่นความน่าจะเป็น (Probability Density Function, PDF) จะต้องมีการหาพารามิเตอร์ที่สามารถประมาณค่าเข้าได้ตามวิธีการที่ได้กล่าวไปแล้ว รูปแบบของฟังก์ชันความหนาแน่นความน่าจะเป็นที่นิยมใช้คือฟังก์ชันความหนาแน่นผสม ที่ประกอบไปด้วยฟังก์ชันความหนาแน่นย่อยชนิดเดียวกันจำนวนจำกัด ดังแสดงได้ตามสมการต่อไปนี้

$$b_j(\vec{O}) = \sum_{m=1}^M c_{jm} \mathcal{N}[\vec{O}, \vec{\mu}_{jm}, \Sigma_{jm}], \quad 1 \leq j \leq N \quad (2.39)$$

เมื่อ \vec{O} คือเวกเตอร์สังเกตที่จะนำมาจำลองแบบ c_{jm} คือ สัมประสิทธิ์ความหนาแน่นย่อยของฟังก์ชันความหนาแน่นผสมตัวที่ m ในสถานะ j และ \mathcal{N} คือฟังก์ชันความหนาแน่นย่อยที่นิยมใช้ความหนาแน่นแบบเกาส์ โดยจะต้องมี c_{jm} ที่ตรงตามข้อกำหนดของกระบวนการเฟ้นสุ่ม (Stochastic) ดังนี้

$$\sum_{m=1}^M c_{jm} = 1, \quad 1 \leq j \leq N \quad (2.40)$$

$$c_{jm} \geq 0, \quad 1 \leq j \leq N, 1 \leq m \leq M \quad (2.41)$$

และฟังก์ชันความหนาแน่นความน่าจะเป็นจะถูกนอร์มัลไลซ์ให้เป็นอย่างนี้

$$\int_{-\infty}^{\infty} b_j(\vec{x}) d\vec{x} = 1, \quad 1 \leq j \leq N \quad (2.42)$$

สำหรับการประมาณค่าซ้ำเพื่อหาพารามิเตอร์ของฟังก์ชันความหนาแน่นผสมที่ k จะได้ c_{jk} , $\bar{\mu}_{jk}$ และ $\bar{\Sigma}_{jk}$ เป็นไปตามสมการเหล่านี้

$$\bar{c}_{jk} = \frac{\sum_{t=1}^T \gamma_{jk}(t)}{\sum_{t=1}^T \sum_{k=1}^M \gamma_{jk}(t)} \quad (2.43)$$

$$\bar{\mu}_{jk} = \frac{\sum_{t=1}^T \gamma_{jk}(t) \cdot \bar{o}_t}{\sum_{t=1}^T \gamma_{jk}(t)} \quad (2.44)$$

$$\bar{\Sigma}_{jk} = \frac{\sum_{t=1}^T \gamma_{jk}(t) \cdot (\bar{o}_t - \bar{\mu}_{jk})(\bar{o}_t - \bar{\mu}_{jk})^T}{\sum_{t=1}^T \gamma_{jk}(t)} \quad (2.45)$$

เมื่อ $\gamma_j(t)$ คือ ความน่าจะเป็นที่จะอยู่ในสถานะ j ที่เวลา t ด้วย ฟังก์ชันความหนาแน่นผสมที่อธิบาย \bar{o}_t

$$\gamma_{jk}(t) = \left[\frac{\alpha_j(t) \beta_j(t)}{\sum_{j=1}^N \alpha_j(t) \beta_j(t)} \right] \left[\frac{c_{jk} \mathcal{N}(\bar{o}_t, \bar{\mu}_{jk}, \bar{\Sigma}_{jk})}{\sum_{m=1}^M c_{jm} \mathcal{N}(\bar{o}_t, \bar{\mu}_{jm}, \bar{\Sigma}_{jm})} \right] \quad (2.46)$$

สำหรับการประมาณค่าซ้ำของ a_{ij} นั้นเหมือนกับในกรณีไม่ต่อเนื่อง

2.3 ซัพพอร์ตเวกเตอร์แมชชีน

ซัพพอร์ตเวกเตอร์แมชชีนเป็นการรู้จำแบบรูป (Pattern recognition) ซึ่งพยายามที่จะสร้างปริภูมิสมมุติฐาน (Hypothesis space) ของฟังก์ชันของกฎการจำแนก (Classification rules) $h: \bar{x} \rightarrow y$ [10], [11] ซึ่งในที่นี้เราจะกล่าวถึงกรณีที่มี 2 ชนิด (Class) $h: \mathcal{R}^N \rightarrow \{\pm 1\}$ กำหนดให้มี l ค่าสังเกต (Observation) แต่ละค่าสังเกตประกอบด้วยคู่ของเวกเตอร์ $\bar{x}_i \in \mathcal{R}^N$ และสัญลักษณ์แสดงชนิด (Class label) $y_i \in \{\pm 1\}$ ดังนั้นจะได้ว่า

$$(\bar{x}_1, y_1), \dots, (\bar{x}_l, y_l) \in \mathcal{R}^N \times \{\pm 1\} \quad (2.47)$$

เราจะกล่าวว่า h จำแนกตัวอย่าง (\bar{x}, y) ได้ถูกต้องเมื่อ

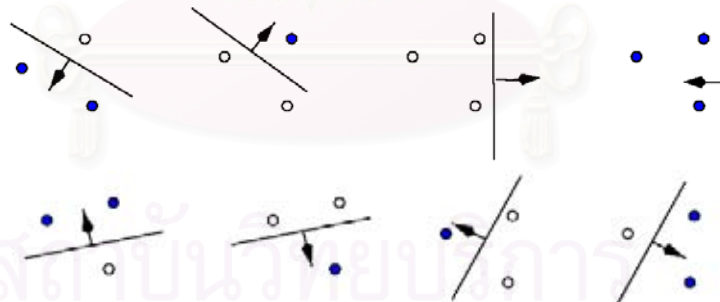
$$h(\bar{x}) = y \quad (2.48)$$

โดย $(\bar{x}, y) \sim P(\bar{x}, y)$ ซึ่งได้มาจากการฝึกฝน $(\bar{x}_1, y_1), \dots, (\bar{x}_l, y_l)$

2.3.1 มิติ Vapnik/Chervonenkis (VC-dimension)

ปริมาณที่ใช้วัดความสมบูรณ์ (Richness) หรือความเปลี่ยนแปลงได้ (Flexibility) ของชนิดฟังก์ชันซึ่งส่งผลถึงความจุของระบบรู้จำเรียกว่า มิติ VC การควบคุมความจุได้ย่อมทำให้ประสิทธิภาพของระบบดีขึ้น ตัวอย่างมิติ VC ของปริภูมิสมมุติฐานในกรณีมี 2 ชนิด จะเท่ากับจำนวนที่มากที่สุด, d ของตัวอย่างที่สามารถแบ่งเป็น 2 กลุ่มด้วย 2^d วิธีใด ๆ โดยใช้ปริภูมิสมมุติฐานนั้น

มิติ VC ของปริภูมิสมมุติฐานใน \mathcal{R}^N คือ $N+1$ เช่น ใน \mathcal{R}^2 จะได้ มิติ VC = 3 ดังรูปที่ 2.4 จากรูปพบว่า มีเพียงกรณี 3 ตัวอย่างเท่านั้นที่เป็นจำนวนมากที่สุดที่สามารถแบ่งออกได้ $2^3 = 8$ วิธี ส่วนกรณี 4 ตัวอย่างนั้นทำไม่ได้



(ก)



(ข)

รูปที่ 2.4 (ก) มิติ VC ใน \mathcal{R}^2 กรณี 3 ตัวอย่างใน \mathcal{R}^2 (ข) กรณี 4 ตัวอย่างใน \mathcal{R}^2

ถ้า $h \in H$ และ H เป็นปริภูมิสมมุติฐานซึ่งมีมิติ VC เท่ากับ d สำหรับทุก ๆ การกระจายความน่าจะเป็น D บน $X \times \{-1, 1\}$ ด้วยความน่าจะเป็น $1 - \delta$ บนตัวอย่างสุ่มจำนวน l ค่าความผิดพลาดจะมีค่าไม่เกิน

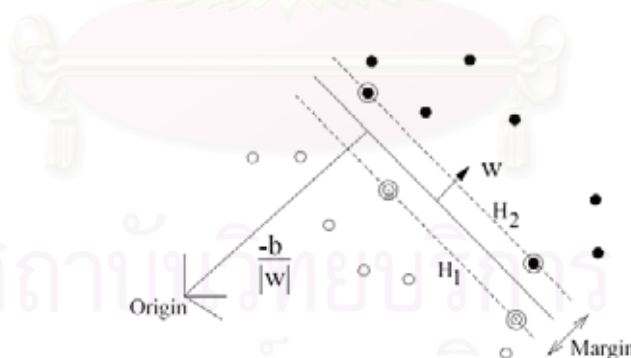
$$err_D(h) \leq \varepsilon(l, H, \delta) = \frac{2}{l} \left(d \log \frac{2el}{d} + \log \frac{2}{\delta} \right), \quad d \leq l, l > \frac{2}{\varepsilon} \quad (2.49)$$

2.3.2 ซัพพอร์ตเวกเตอร์แมชชีนแบบเชิงเส้น

ซัพพอร์ตเวกเตอร์แมชชีนแบบเชิงเส้น คือ ซัพพอร์ตเวกเตอร์แมชชีนที่มีไฮเปอร์เพลน (Hyperplane) ในลักษณะเป็นเชิงเส้น แบ่งออกได้ 2 กรณีคือ กรณีแยกกันได้ และกรณีแยกกันไม่ได้

2.3.2.1 กรณีแยกกันได้

เป็นการพิจารณาในกรณีที่ง่ายที่สุดเนื่องจากข้อมูลมีลักษณะแยกกันอย่างสมบูรณ์ซึ่งสามารถจำแนกประเภทชนิดได้ด้วยรูปแบบทางเรขาคณิต (ไฮเปอร์เพลน) ดังรูปที่ 2.5



รูปที่ 2.5 ไฮเปอร์เพลนแบบเชิงเส้นสำหรับกรณีแยกกันได้

โดยตำแหน่งของจุดตัวอย่างมีความสัมพันธ์ดังนี้

$$\tilde{w} \cdot \tilde{x} + b = 0 \quad (2.50)$$

โดย w เป็นเวกเตอร์น้ำหนักซึ่งเป็นเวกเตอร์ปกติของไฮเปอร์เพลนและ b เป็นไบแอส

ระยะห่างจากจุดกำเนิดไปตั้งฉาก ไฮเปอร์เพลน คือ $\frac{|b|}{\|w\|}$

ระยะห่างจากที่สั้นที่สุดจาก ไฮเปอร์เพลน ไปสู่จุดตัวอย่างทาง + และทาง - คือ d_+, d_-

ตามลำดับ

ระยะขอบเขต (Margin) คือ ระยะห่างระหว่าง d_+ และ d_-

เราสามารถกำหนดความสัมพันธ์ได้ดังนี้

$$h(\bar{x}) = \text{sign} \left[\sum_{i=1}^l \bar{w}_i \bar{x}_i + b \right] = \begin{cases} +1 & \text{if } \sum_{i=1}^l \bar{w}_i \bar{x}_i + b > 0 \\ -1 & \text{else} \end{cases} \quad (2.51)$$

จาก $y_i \in \{\pm 1\}$ และสมการ (2.51) เราจะได้สมการ

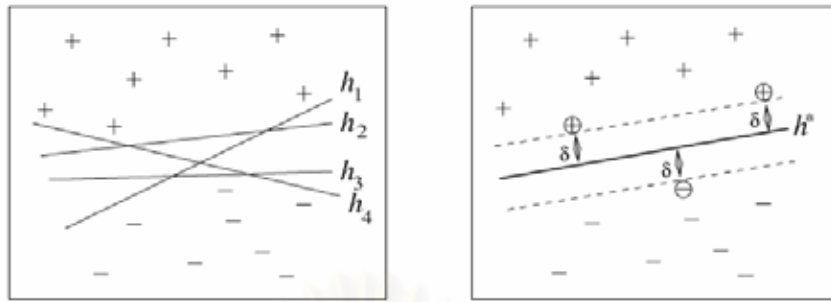
$$y_i (\bar{x}_i \cdot \bar{w} + b) - 1 \geq 0 \quad \forall i \quad (2.52)$$

จากรูปที่ 2.5 ไฮเปอร์เพลน $H_1 : \bar{x}_i \cdot \bar{w} + b = 1$ มีระยะห่างจากจุดกำเนิดคือ $\frac{|1-b|}{\|w\|}$

ไฮเปอร์เพลน $H_2 : \bar{x}_i \cdot \bar{w} + b = -1$ มีระยะห่างจากจุดกำเนิดคือ $\frac{|-1-b|}{\|w\|}$

ดังนั้น $d_+ = d_- = \frac{1}{\|w\|}$ และ Margin (δ) = $\frac{2}{\|w\|}$

ข้อสังเกต H_1 ขนานกับ H_2 เนื่องจากมีเวกเตอร์ปกติเดียวกันและไม่มีจุดตัวอย่างอยู่ระหว่างนั้น จุดตัวอย่างที่อยู่ในแนวของ H_1 และ H_2 เรียกว่า ซัพพอร์ตเวกเตอร์ (Support Vector) ลักษณะของไฮเปอร์เพลนที่เหมาะสมที่สุดเป็นดังรูปที่ 2.6



รูปที่ 2.6 ไฮเปอร์เพลนที่เหมาะสมที่สุด

เงื่อนไขที่ทำให้ได้ไฮเปอร์เพลนที่เหมาะสมที่สุด [12] ประกอบด้วย

1. ค่าความผิดพลาดในการฝึกฝนเป็นศูนย์ (Zero training error)
2. ระยะระหว่างซัพพอร์ตเวกเตอร์ของทั้ง 2 ชนิดห่างกันมากที่สุด (Maximum margin)

โดยค่าขอบเขต (Margin, δ)

$$\delta = \min_i \left| \frac{1}{\sqrt{\tilde{w} \cdot \tilde{w}}} [\tilde{w} \cdot \tilde{x}_i + b] \right| \quad (2.53)$$

และจากอสมการ (2.52) จะได้เงื่อนไขการหาค่าเหมาะสมดังนี้

$$\text{Maximize } \delta \quad \text{subject to} \quad y_i \left(\frac{1}{\sqrt{\tilde{w} \cdot \tilde{w}}} [\tilde{w} \cdot \tilde{x}_i + b] \right) \geq \delta, \quad \forall i \in [1..I] \quad (2.54)$$

แทนค่า $\delta = \frac{1}{\sqrt{\tilde{w} \cdot \tilde{w}}}$ ลงในอสมการ (2.54) จะได้ว่าเราต้อง

$$\text{Minimize } \tilde{w} \cdot \tilde{w} \quad \text{subject to} \quad y_i [\tilde{w} \cdot \tilde{x}_i + b] \geq 1, \quad \forall i \in [1..I] \quad (2.55)$$

ในการหาค่าที่เหมาะสมที่สุด (Optimization) โดยวิธีแบบ Lagrange สามารถอาศัยคุณสมบัติคู่อลิติ (Duality) เพื่อทำให้การแก้ไขปัญหานั้นทำได้ง่ายขึ้นคือ ทำได้โดยการกำหนดปัญหาที่เราต้องการ (Optimization Problem) ในที่นี้

Optimization Problem: Minimize $J(\bar{w}, b) = \frac{1}{2} \bar{w} \cdot \bar{w}$ (2.56)

Subject to $y_i[\bar{w} \cdot \bar{x}_i + b] \geq 1, \quad \forall i \in [1 \dots l]$

กำหนดตัวคูณ Lagrange บวก (α_i)

$$\alpha_i \geq 0, \quad i = 1, \dots, l \quad (2.57)$$

จาก (2.56) และ (2.57) เราจะได้ Primal Optimal Problem ดังนี้

Primal Optimal Problem:

Minimize $L(\bar{w}, b, \bar{\alpha}) = \frac{1}{2} \bar{w} \cdot \bar{w} - \frac{1}{2} \sum_i \alpha_i [y_i[\bar{w} \cdot \bar{x}_i + b] - 1]$ (2.58)

Subject to $\alpha_i \geq 0, \quad i = 1, \dots, l$

หาอนุพันธ์ย่อยของสมการที่ (2.58) เทียบกับ \bar{w} และ b จะได้

$$\frac{\partial L(\bar{w}, b, \bar{\alpha})}{\partial \bar{w}} = \bar{w} - \sum_{i=1}^l y_i \alpha_i \bar{x}_i$$

$$\frac{\partial L(\bar{w}, b, \bar{\alpha})}{\partial b} = \sum_{i=1}^l y_i \alpha_i$$
(2.59)

เนื่องจากค่า \bar{w} และ b ทำให้ $L(\bar{w}, b, \bar{\alpha})$ มีค่าน้อยที่สุดดังนั้นเราจะได้

$$\bar{w} = \sum_{i=1}^l y_i \alpha_i \bar{x}_i$$

$$0 = \sum_{i=1}^l y_i \alpha_i$$
(2.60)

เมื่อนำ (2.60) ไปแทนค่าลงใน (2.58) เราจะได้ Dual Optimal Problem ดังนี้

Dual Optimal Problem:

$$\text{Maximize } L(\bar{w}, b, \bar{\alpha}) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \bar{x}_i \cdot \bar{x}_j \quad (2.61)$$

$$\text{Subject to } \sum_i \alpha_i y_i = 0 \quad \alpha_i \geq 0, \quad i = 1, \dots, l$$

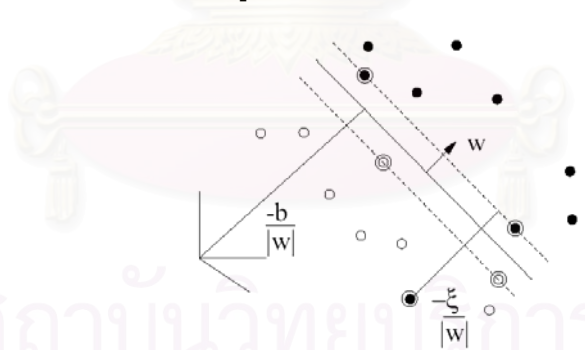
ผลตอบที่ได้จาก Primal Optimal Problem และ Dual Optimal Problem มีค่าเดียวกัน คือ

$$\bar{w}^\circ = \sum_i \alpha_i y_i \bar{x}_i \quad \text{และ} \quad b^\circ = \frac{1}{2} (\bar{w}_0 \cdot \bar{x}^{pos} + \bar{w}_0 \cdot \bar{x}^{neg}) \quad (2.62)$$

ผลตอบที่ได้มีหนึ่งเดียว และตัวอย่างที่มีค่า $\alpha_i > 0$ จะเป็นซัพพอร์ตเวกเตอร์

2.3.2.2 กรณีแยกกันไม่ได้

กรณีที่เรไม่สามารถแยกจุดตัวอย่างบางตัวอย่างได้ด้วยไฮเปอร์เพลน เราทำได้เพียงหาไฮเปอร์เพลนที่สามารถแยกจุดตัวอย่างออกจากกันให้ได้มากที่สุดและยอมให้มีจุดตัวอย่างส่วนน้อยเพียงบางจุดที่ผิดพลาดดังแสดงในรูปที่ 2.7



รูปที่ 2.7 ไฮเปอร์เพลนแบบแข็งเส้นสำหรับกรณีแยกกันไม่ได้

เราจึงไม่สามารถใช้ขอบเขตแบบแข็ง (Hard margin):

$$\text{Minimize } J(\bar{w}, b) = \frac{1}{2} \bar{w} \cdot \bar{w} \quad (2.63)$$

$$\text{Subject to } y_i [\bar{w} \cdot \bar{x}_i + b] \geq 1$$

ได้เหมือนในกรณีแยกกันได้ ดังนั้นเราจึงใช้ขอบเขตแบบอ่อน (Soft margin) แทน:

$$\text{Minimize } J(\bar{w}, b, \bar{\xi}) = \frac{1}{2} \bar{w} \cdot \bar{w} + C \sum_{i=1}^n \xi_i \quad (2.64)$$

$$\text{Subject to } y_i [\bar{w} \cdot \bar{x}_i + b] \geq 1 - \xi_i, \quad \xi_i \geq 0$$

C คือ ค่าความคุมการ Trade-off ระหว่างขอบเขตกับค่าผิดพลาด

$$\text{Dual Optimal Problem: Maximize } L(\bar{\alpha}) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \bar{x}_i \cdot \bar{x}_j \quad (2.65)$$

$$\text{Subject to } \sum_i \alpha_i y_i = 0 \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, l$$

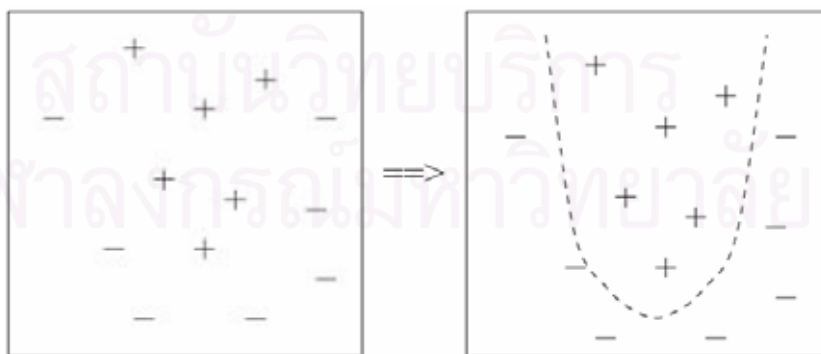
ผลตอบที่ได้มีหนึ่งเดียว

ตัวอย่างที่เป็น Support vector ที่ $\xi_i = 0$ ก็ต่อเมื่อ $0 < \alpha_i < C$

ตัวอย่างที่เป็น Support vector ที่ $\xi_i > 0$ ก็ต่อเมื่อ $\alpha_i = C$

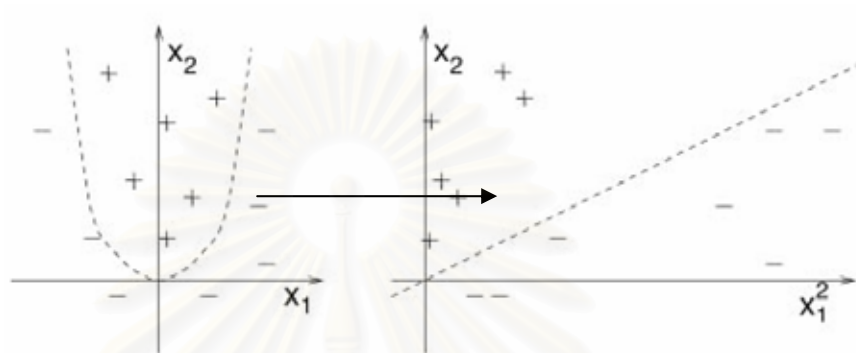
2.3.3 ซัพพอร์ตเวกเตอร์แมชชีนแบบไม่เชิงเส้น

ในงานบางประเภทมีลักษณะไม่เป็นเชิงเส้นจึงไม่สามารถใช้ไฮเปอร์เพลนมาจำแนกประเภทได้ดังรูปที่ 2.8



รูปที่ 2.8 โครงสร้างข้อมูลแบบไม่เป็นเชิงเส้น

วิธีประยุกต์ใช้ซัพพอร์ตเวกเตอร์แมชชีนกับกรณีไม่เชิงเส้นนี้ทำได้โดยการส่ง (Map) จากปริภูมิขาเข้า (Input space) ไปสู่ปริภูมิลักษณะ (Feature space) โดยผ่านฟังก์ชัน $\Phi(\vec{x})$ ดังรูปที่ 2.9



รูปที่ 2.9 การส่งปริภูมิขาเข้า (ซ้าย) ไปสู่ ปริภูมิลักษณะ (ขวา)

จากรูปเป็นตัวอย่างของการส่งจาก

Input Space: $\vec{x} = (x_1, x_2)$ (2 คุณลักษณะ)

Feature Space: $\Phi(\vec{x}) = (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, 1)$ (6 คุณลักษณะ)

หลังจากส่งไปสู่ปริภูมิลักษณะแล้วเราจะสามารถหาไฮเปอร์เพลนมาจำแนกประเภทได้

2.3.3.1 เคอร์เนล (Kernels)

เนื่องจากมีจำนวนของคุณลักษณะสำคัญ (Attribute) มากมายจากโพลีโนเมียลระดับชั้น p บน N คุณลักษณะสำคัญในปริภูมิขาเข้าไปเป็น $O(N^p)$ คุณลักษณะสำคัญในปริภูมิคุณลักษณะสำคัญ

Kernel functions: $K(\vec{x}_i, \vec{x}_j) = \Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j)$ (2.66)

จาก $\Phi(\vec{x}) = (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, 1)$ เราสามารถหาฟังก์ชันเคอร์เนลได้เป็น

$$K(\vec{x}_i, \vec{x}_j) = \Phi(\vec{x}_i) \cdot \Phi(\vec{x}_j) = [\vec{x}_i \cdot \vec{x}_j + 1]^2 \quad (2.67)$$

จากสมการ (2.65) เราปรับปรุงใหม่โดยใช้ฟังก์ชันเคอร์เนลได้ดังนี้

$$\text{Dual Optimal Problem: Maximize } L(\vec{\alpha}) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\vec{x}_i, \vec{x}_j) \quad (2.68)$$

$$\text{Subject to } \sum_i \alpha_i y_i = 0 \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, l$$

$$\text{Hyperplane: } h(\vec{x}) = \text{sign} \left[\sum_{x_i \in SV} \alpha_i y_i K(\vec{x}_i, \vec{x}_j) + b \right] \quad (2.69)$$

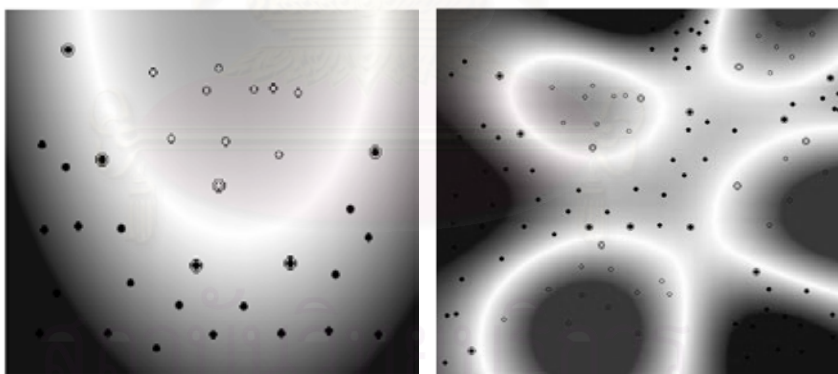
ฟังก์ชันเคอร์เนลมีหลากหลายขึ้นกับการใช้งาน เช่น

$$\text{Linear: } K(\vec{x}_i, \vec{x}_j) = \vec{x}_i \cdot \vec{x}_j \quad (2.70)$$

$$\text{Polynomial: } K(\vec{x}_i, \vec{x}_j) = [\vec{x}_i \cdot \vec{x}_j + 1]^d \quad (2.71)$$

$$\text{Radial Basis Function (RBF): } K(\vec{x}_i, \vec{x}_j) = \exp(-\|\vec{x}_i - \vec{x}_j\|^2 / (2\sigma^2)) \quad (2.72)$$

$$\text{Sigmoid: } K(\vec{x}_i, \vec{x}_j) = \tanh[\gamma(\vec{x}_i - \vec{x}_j) + c] \quad (2.73)$$



รูปที่ 2.10 ฟังก์ชันเคอร์เนลแบบ Polynomial (ซ้าย) และ แบบ RBF (ขวา)

2.4 ปริภูมิคะแนน (Score-spaces)

ปริภูมิคะแนน คือ ปริภูมิที่ลำดับเวกเตอร์การสังเกต (Observation vectors) ใด ๆ สามารถถูกส่ง (Map) มาเป็นคะแนน (Score) ซึ่งคะแนนที่ได้จะเป็นข้อมูลขาเข้าของตัวจำแนกประเภท (Classifier) ต่อไป เหตุผลที่ทำให้ปริภูมิคะแนนมีประโยชน์และน่าสนใจ [13] คือ

- ปริภูมิคะแนนถูกกำหนดให้ใช้กับแบบจำลองความน่าจะเป็นแบบเจเนอเรทีฟ (Generative) [13] เพราะฉะนั้นจึงเหมาะกับวิธีการแบบโดยนัย (Implicit method) เช่น โครงข่ายประสาทเทียม
- สามารถรวมเอาข้อดีของตัวจำแนกประเภทแบบจำลองแบบเจเนอเรทีฟกับตัวจำแนกประเภทที่ถูกล็อกแบบดิสคริมิเนทีฟ
- เนื่องจากข้อดีของแบบจำลองแบบเจเนอเรทีฟที่รองรับมิติที่หลากหลายของลำดับเวกเตอร์การสังเกตได้ ทำให้ปริภูมิคะแนนสามารถส่งลำดับเวกเตอร์การสังเกตที่มีมิติหลากหลายไปเป็นเวกเตอร์ที่มีมิติคงที่ได้ จึงเหมาะกับตัวจำแนกประเภทแบบสถิต เช่น ซัพพอร์ตเวกเตอร์แมชชีน ที่ถูกนำไปใช้กับงานการจำแนกประเภทแบบจลน์ ดังเช่น งานด้านการรู้เสียงพูด หรือ ตัวอักษร
- ความสามารถที่โดดเด่นในการจำแนกประเภทของตัวจำแนกประเภทที่ถูกล็อกแบบดิสคริมิเนทีฟ เช่น ซัพพอร์ตเวกเตอร์แมชชีน ในงานการจำแนกประเภทแบบสถิต เพราะจะทำการหาขอบเขตของชนิด (Class boundaries) โดยตรงโดยไม่ต้องผ่านขั้นตอนในการจำลองความหนาแน่นของข้อมูล

พิจารณาลำดับการสังเกต \bar{O} และ เซตของแบบจำลองความน่าจะเป็นแบบเจเนอเรทีฟที่เป็นไปได้ของ \bar{O} , $P = \{P_k(\bar{O} | \bar{\theta}_k)\}$ เมื่อ k คือ ตัวชี้สำหรับแต่ละแบบจำลองในเซต P และ $\bar{\theta}_k$ คือ เวกเตอร์พารามิเตอร์ของสมาชิกตัวที่ k ที่ประมาณจากแบบจำลองแบบเจเนอเรทีฟและให้ $\bar{\Theta}_k$ คือ เวกเตอร์พารามิเตอร์แท้จริงของสมาชิกตัวที่ k ที่สร้างจากแบบจำลองแบบเจเนอเรทีฟที่ถูกต้อง ในหลาย ๆ กรณีที่เราไม่สามารถหา $\bar{\Theta}_k$ ได้เราจึงจำเป็นต้องทำการประมาณค่า $\bar{\Theta}_k$ โดยใช้อนุกรมเทเลอร์ (Taylor series) และเนื่องจากค่าความน่าจะเป็นในเซต P มีค่าน้อยมากเราจึงนิยมใช้ค่า $\ln P$ แทนดังนั้น

$$\begin{aligned} \ln P_k(\bar{O} | \bar{\Theta}_k) &= \ln P_k(\bar{O} | \bar{\theta}_k) + (\bar{\Theta}_k - \bar{\theta}_k)^T \nabla_{\bar{\theta}_k} \ln P_k(\bar{O} | \bar{\theta}_k) \\ &+ \frac{1}{2} (\bar{\Theta}_k - \bar{\theta}_k)^T \left[\nabla_{\bar{\theta}_k}^2 \ln P_k(\bar{O} | \bar{\theta}_k) \right] (\bar{\Theta}_k - \bar{\theta}_k) + \dots \end{aligned} \quad (2.74)$$

หากเรากำหนดให้พจน์ที่มีระดับชั้นมากกว่า 2 ให้อยู่ในพจน์ $\Psi \left((\bar{\Theta}_k - \bar{\theta}_k)^{n+1} \right)$ ดังนั้น

$$\begin{aligned}
\ln P_k(\bar{O} | \bar{\Theta}_k) &= \ln P_k(\bar{O} | \bar{\theta}_k) + (\bar{\Theta}_k - \bar{\theta}_k)^T \nabla_{\bar{\theta}_k} \ln P_k(\bar{O} | \bar{\theta}_k) \\
&\quad + \frac{1}{2} (\bar{\Theta}_k - \bar{\theta}_k)^T \left[\nabla_{\bar{\theta}_k}^2 \ln P_k(\bar{O} | \bar{\theta}_k) \right] (\bar{\Theta}_k - \bar{\theta}_k) \\
&\quad + \Psi \left((\bar{\Theta}_k - \bar{\theta}_k)^{n+1} \right) \\
&= \bar{w}_{lik(k)}^T \varphi^{lik(i)}(\bar{O}) + \Psi \left((\bar{\Theta}_k - \bar{\theta}_k)^{n+1} \right)
\end{aligned} \tag{2.75}$$

ถ้า $n \rightarrow \infty$ พจน์ $\bar{w}_{lik(k)}^T \varphi^{lik(i)}(\bar{O})$ จะประมาณค่าจริงของ $\ln P_k(\bar{O} | \bar{\Theta}_k)$ ได้ใกล้เคียงที่สุด โดย $\varphi^{lik(i)}(\bar{O})$ คือ คะแนน (Score) ภายในปริภูมิคะแนนที่มีมิติอนันต์ซึ่งเกิดจากการหาอนุพันธ์อันดับอนันต์ในการประมาณค่าพารามิเตอร์ และ $\bar{w}_{lik(k)}$ คือเวกเตอร์น้ำหนักซึ่งถูกประมาณขึ้น

พิจารณาลำดับการสังเกต \bar{O} ที่ถูกส่งไปยังปริภูมิคะแนนจะเรียกว่าคะแนน (Score), $\varphi_{\hat{F}}^f(\bar{O})$ โดย

$$\varphi_{\hat{F}}^f(\bar{O}) = \varphi_{\hat{F}} f \left(\{P_k(\bar{O} | \bar{\theta}_k)\} \right) \tag{2.76}$$

โดย $f \left(\{P_k(\bar{O} | \bar{\theta}_k)\} \right)$ คือ Score-argument และเป็นฟังก์ชันของสมาชิกของ P

$\varphi_{\hat{F}}$ คือ ตัวส่งคะแนน (Score-mapping)

\hat{F} คือ ตัวดำเนินการคะแนน (Score-operator)

2.4.1 อารักิวเมนต์ของคะแนน (Score-argument)

อารักิวเมนต์ของคะแนนที่มักนิยมใช้ 3 แบบ คือ แบบ Likelihood แบบ Likelihood-ratio และแบบ posterior-ratio

- ปริภูมิคะแนนแบบ Likelihood เป็นปริภูมิคะแนนแบบดั้งเดิม

$$\varphi_{\hat{F}}^{lik}(\bar{O}) = \varphi_{\hat{F}} \ln P(\bar{O} | \bar{\theta}) \tag{2.77}$$

- ปริภูมิคะแนนแบบ Likelihood-ratio เป็นปริภูมิที่เหมาะสมกับการจำแนกประเภทแบบไบนารี

$$\varphi_{\hat{F}}^r(\bar{O}) = \varphi_{\hat{F}} \ln \left[\frac{P_1(\bar{O} | \bar{\theta}_1)}{P_2(\bar{O} | \bar{\theta}_2)} \right] \quad (2.78)$$

- ปริภูมิคะแนนแบบ Posterior-ratio เป็นปริภูมิที่ขยายจำนวนแบบจำลองเงื่อนไขที่แข่งขันกัน K ตัว โดยคะแนนที่ได้จะเป็นความน่าจะเป็นภายหลัง (Posterior probability) ของแบบจำลองที่ i เมื่อ $i \in 1 \dots K$

$$\varphi_{\hat{F}}^{post}(\bar{O}) = \varphi_{\hat{F}} \ln \left[\frac{P_1(\bar{O} | \bar{\theta}_1)}{\sum_{k=1}^K P_k(\bar{O} | \bar{\theta}_k)} \right] \quad (2.79)$$

2.4.2 ตัวดำเนินการคะแนน (Score-operator)

ตัวดำเนินการคะแนนที่ดีควรมีคุณสมบัติดังนี้

- ต้องทำให้มิติของปริภูมิคะแนนมีขนาดเหมาะสมกับตัวจำแนกประเภทที่ใช้ เช่น ซัพพอร์ตเวกเตอร์แมชชีนนั้นสามารถรองรับมิติจำนวนมากได้แต่ก็ทำให้เทคนิค Nearest-Neighbor ไม่มีประสิทธิภาพหากมิติที่มากเกินไป
- ต้องกำหนดลักษณะที่มีประโยชน์และจำแนกประเภทได้ในปริภูมิคะแนน
- ต้องมีการคำนวณที่ไม่ยุ่งยากมากเกินไป

ตัวดำเนินการคะแนนชนิดต่าง ๆ

- อนุพันธ์อันดับหนึ่ง: $\hat{F} = \bar{\nabla}_{\bar{\theta}}$
- อนุพันธ์อันดับหนึ่งและอาร์กิวเมนต์: $\hat{F} = [\bar{\nabla}_{\bar{\theta}}^T, 1]^T$
- อนุพันธ์อันดับหนึ่งและอันดับสอง: $\hat{F} = [\bar{\nabla}_{\bar{\theta}}, \text{vec}(\bar{\nabla}_{\bar{\theta}}^{2T})]^T$

2.4.3 การประยุกต์ใช้ปริภูมิคะแนนกับซัพพอร์ตเวกเตอร์แมชชีน

โดยในที่นี้จะสนใจเพียงตัวดำเนินการคะแนนแบบอนุพันธ์อันดับหนึ่งเท่านั้น เราใช้คะแนนเป็นข้อมูลเข้า จากสมการ (2.70), (2.71), (2.72) และ (2.73) จะได้

$$\text{Linear:} \quad K^{sc-lin}(\varphi(\bar{O}_i), \varphi(\bar{O}_j)) = \varphi(\bar{O}_i)^T \varphi(\bar{O}_j) \quad (2.80)$$

$$\text{Polynomial:} \quad K^{sc-pol}(\varphi(\bar{O}_i), \varphi(\bar{O}_j)) = [\varphi(\bar{O}_i)^T \varphi(\bar{O}_j) + p]^d \quad (2.81)$$

$$\text{RBF: } K^{sc-rbf}(\varphi(\bar{O}_i), \varphi(\bar{O}_j)) = \exp\left(-\frac{1}{2\sigma^2} [\varphi(\bar{O}_i) - \varphi(\bar{O}_j)]^T [\varphi(\bar{O}_i) - \varphi(\bar{O}_j)]\right) \quad (2.82)$$

$$\text{Sigmoid: } K^{sc-sig}(\varphi(\bar{O}_i), \varphi(\bar{O}_j)) = \tanh[\gamma(\varphi(\bar{O}_i) - \varphi(\bar{O}_j)) + c] \quad (2.83)$$

แต่ก่อนที่จะใช้คะแนนเป็นข้อมูลเข้าของเคอร์เนลใด ๆ จำเป็นต้องทำการทำให้ขาว (Whitening) และ นอร์มัลไลซ์ก่อน โดยคะแนนที่ถูกนอร์มัลไลซ์ $\varphi^N(\bar{O})$ มีค่าดังนี้

$$\varphi^N(\bar{O}) = \Sigma_{sc}^{-1/2} \varphi(\bar{O}) \quad (2.84)$$

โดย

$$\Sigma_{sc} = \int (\varphi(\bar{O}) - \bar{\mu}_{sc})(\varphi(\bar{O}) - \bar{\mu}_{sc})^T P(\bar{O} | \bar{\theta}) d\bar{O} \quad (2.85)$$

$$\bar{\mu}_{sc} = \int \varphi(\bar{O}) P(\bar{O} | \bar{\theta}) d\bar{O} \quad (2.86)$$

โดย Σ_{sc} คือ เมทริกซ์โควาเลี่ยนซ์ในปริภูมิคะแนน

$\bar{\mu}_{sc}$ คือ ค่าคาดหวัง (Expectation) ในปริภูมิคะแนน

2.4.4 ปริภูมิคะแนนที่ถูกนิยามด้วยตัวดำเนินการคะแนนแบบอนุพันธ์อันดับหนึ่ง

เมื่อพิจารณาปริภูมิคะแนนแบบ Likelihood และ ตัวดำเนินการคะแนนแบบอนุพันธ์อันดับหนึ่งพบว่าเป็นค่าเกรเดียนท์ของแบบจำลอง Likelihood นั้นเองดังสมการต่อไปนี้

$$\varphi_{\nabla_{\bar{\theta}}}^{lik}(\bar{O}) = \nabla_{\bar{\theta}} \ln P(\bar{O} | \bar{\theta}) \quad (2.87)$$

เมื่อพิจารณาปริภูมิคะแนนแบบ Likelihood-ratio และ ตัวดำเนินการคะแนนแบบอนุพันธ์อันดับหนึ่งแสดงดังสมการต่อไปนี้

$$\varphi_{\nabla_{\bar{\theta}}}^{lr}(\bar{O}) = \nabla_{\bar{\theta}} \ln \frac{P(\bar{O} | \bar{\theta}_1)}{P(\bar{O} | \bar{\theta}_2)} \quad (2.88)$$

เนื่องจากแบบจำลองทั้งสองอิสระต่อกันดังนั้นเราสามารถเขียนสมการ (2.88) ใหม่ได้เป็น

$$\begin{aligned}\phi_{\nabla_{\bar{\theta}}}^r(\bar{O}) &= \nabla_{\bar{\theta}} [\ln P(\bar{O} | \bar{\theta}_1) - \ln P(\bar{O} | \bar{\theta}_2)] \\ &= \begin{bmatrix} \nabla_{\bar{\theta}_1} \ln P(\bar{O} | \bar{\theta}_1) \\ \dots \\ -\nabla_{\bar{\theta}_2} \ln P(\bar{O} | \bar{\theta}_2) \end{bmatrix}\end{aligned}\quad (2.89)$$

สมมติว่าชนิด y_1 มี l_1 ลำดับการสังเกต และชนิด y_2 มี l_2 ลำดับการสังเกต และให้ $l = l_1 + l_2$ จากสมการ (2.84), (2.85) และ (2.86) คำนวณค่า Maximum Likelihood (ML) ในรูปแบบที่ถูกรวมแล้วของปริภูมิคะแนนแบบ Likelihood ภายใต้ตัวดำเนินการคะแนนแบบอนุพันธ์อันดับหนึ่ง, $\phi_{\nabla_{\bar{\theta}}}^{lik/N}(\bar{O})$ เป็น

$$\phi_{\nabla_{\bar{\theta}}}^{lik/N}(\bar{O}) = \sum_{sc}^{-1/2} \nabla_{\bar{\theta}} \ln P(\bar{O} | \bar{\theta}) \quad (2.90)$$

โดย

$$\hat{\Sigma}_{sc} = \sum_{i=1}^l (\nabla_{\bar{\theta}} \ln P(\bar{O}_i | \bar{\theta}) - \hat{\mu}_{sc}) (\nabla_{\bar{\theta}} \ln P(\bar{O}_i | \bar{\theta}) - \hat{\mu}_{sc})^T \quad (2.91)$$

$$\hat{\mu}_{sc} = \sum_{i=1}^l \nabla_{\bar{\theta}} \ln P(\bar{O}_i | \bar{\theta}) \quad (2.92)$$

โดย $\hat{\Sigma}_{sc}$ คือ ค่าประมาณ ML เมทริกซ์โควาเลนซ์ในปริภูมิคะแนน

$\hat{\mu}_{sc}$ คือ ค่าประมาณ ML ของค่าคาดหวังในปริภูมิคะแนน

2.4.5 นอร์แมลไลซ์เคอร์เนล (Normalized Kernel)

การเปรียบเทียบกันระหว่างคะแนนใด ๆ ในปริภูมิคะแนนที่เหมาะสม คือการเปรียบเทียบโดยใช้ผลคูณภายใน (Inner Product) โดยมีเวกเตอร์มูลฐาน (Basis Vector) ของปริภูมิคะแนนเป็นอนุพันธ์ของ $\ln P(\bar{O} | \bar{\theta})$ เทียบกับแต่ละพารามิเตอร์ใน $\bar{\theta}$ แต่เนื่องจากเวกเตอร์มูลฐานนี้ไม่ตั้งฉากกัน (Orthonormal) จึงทำให้ไม่สามารถใช้ผลคูณผลภายในได้โดยตรง ดังนั้นจึงต้องมีการแปลงให้ปริภูมิคะแนนไปเป็นปริภูมิที่มีเวกเตอร์มูลฐานที่ตั้งฉากกัน ตัวแปลงที่เหมาะสมในกรณีนี้คือรากที่สองของโควาเลี่ยนซ์เมทริกซ์ของปริภูมิคะแนน $\Sigma_{sc}^{-1/2}$ ดังนั้นผลคูณภายในที่ได้คือ

$$\langle \nabla_{\bar{\theta}} \ln P(\bar{O}_i | \bar{\theta}), \nabla_{\bar{\theta}} \ln P(\bar{O}_j | \bar{\theta}) \rangle = \nabla_{\bar{\theta}} \ln P(\bar{O}_i | \bar{\theta})^T \Sigma_{sc}^{-1} \nabla_{\bar{\theta}} \ln P(\bar{O}_j | \bar{\theta}) \quad (2.93)$$

ฟังก์ชันก่ใด ๆ ก็ตามที่สามารถเขียนอยู่ในรูปผลคูณภายในแบบสเกลาร์ได้เราจะเรียกพจน์นั้นว่า
เคอร์เนล ดังนั้นจาก (2.84) จะได้นอร์แมลไลซ์เคอร์เนลดังกล่าว

$$K_N(\bar{O}_i, \bar{O}_j) = \nabla_{\bar{\theta}} \ln P(\bar{O}_i | \bar{\theta})^T \Sigma_{sc}^{-1} \nabla_{\bar{\theta}} \ln P(\bar{O}_j | \bar{\theta}) \quad (2.94)$$

เมื่อนำไปประยุกต์กับเคอร์เนลอื่น ๆ จากเช่น (2.80) และ (2.82) เราจะได้

$$K_N^{sc-lin}(\bar{O}_i, \bar{O}_j) = [\Sigma_{sc}^{-1/2} \nabla_{\bar{\theta}} \ln P(\bar{O}_i | \bar{\theta})]^T [\Sigma_{sc}^{-1/2} \nabla_{\bar{\theta}} \ln P(\bar{O}_j | \bar{\theta})] = K_N(\bar{O}_i, \bar{O}_j) \quad (2.95)$$

$$\begin{aligned} K_N^{sc-RBF}(\bar{O}_i, \bar{O}_j) &= \exp \left\{ -\left\| \left[\Sigma_{sc}^{-1/2} \nabla_{\bar{\theta}} \ln P(\bar{O}_i | \bar{\theta}) - \Sigma_{sc}^{-1/2} \nabla_{\bar{\theta}} \ln P(\bar{O}_j | \bar{\theta}) \right] \right\|^2 / 2\sigma^2 \right\} \\ &= \exp \left\{ -[K_N(\bar{O}_i, \bar{O}_i) - 2K_N(\bar{O}_i, \bar{O}_j) + K_N(\bar{O}_j, \bar{O}_j)] / 2\sigma^2 \right\} \end{aligned} \quad (2.96)$$

2.4.6 ปริภูมิคะแนนสำหรับเวกเตอร์มิติคงที่

การส่งคะแนนทำได้โดยใช้พารามิเตอร์จากแบบจำลองแบบเจเนอเรทีฟต่าง ๆ ซึ่ง
แบบจำลองแบบเจเนอเรทีฟที่นิยมใช้ ได้แก่ แบบจำลองส่วนผสมแบบเกาส์แบบเจเนอเรทีฟ และ
แบบจำลองฮิดเดนมาร์คอฟ

2.4.6.1 แบบจำลองส่วนผสมแบบเกาส์แบบเจเนอเรทีฟ (Gaussian mixture generative model)

กำหนดให้แบบจำลองแบบเจเนอเรทีฟเป็นส่วนประกอบของแบบจำลองแบบเกาส์ ดังนั้น
เราจะเขียนความน่าจะเป็นแบบเจเนอเรทีฟที่เป็นไปได้ของลำดับการสังเกต \bar{O} ได้เป็น

$$P(\bar{O} | \bar{\theta}) = P(o_1, o_2, \dots, o_T | \bar{\theta}) \quad (2.97)$$

สมมุติว่าการสังเกตเป็นแบบ i.i.d. (independently independent distribution) และใช้นิยามสำหรับ
เกรเดียนท์

$$\begin{aligned}
[\nabla_{\bar{\theta}} \ln P(\bar{O} | \bar{\theta})]^T d\bar{\theta} &= d_{\bar{\theta}} \ln P(\bar{O} | \bar{\theta}) \\
&= d_{\bar{\theta}} \ln \prod_{t=1}^T P(\bar{o}_t | \bar{\theta}) \\
&= d_{\bar{\theta}} \sum_{t=1}^T \ln P(\bar{o}_t | \bar{\theta}) \\
&= \sum_{t=1}^T d_{\bar{\theta}} \ln P(\bar{o}_t | \bar{\theta}) \\
&= \sum_{t=1}^T \{[\nabla_{\bar{\theta}} \ln P(\bar{o}_t | \bar{\theta})]^T d\theta\} \\
&= \left[\sum_{t=1}^T [\nabla_{\bar{\theta}} \ln P(\bar{o}_t | \bar{\theta})]^T \right] d\bar{\theta}
\end{aligned} \tag{2.98}$$

จาก (2.98) เราได้ความสัมพันธ์สำหรับการส่งคะแนน (Score-mapping)

$$\nabla_{\bar{\theta}} \ln P(\bar{O} | \bar{\theta}) = \sum_{t=1}^T [\nabla_{\bar{\theta}} \ln P(\bar{o}_t | \bar{\theta})] \tag{2.99}$$

ดังนั้นสำหรับคะแนนเวกเตอร์การสังเกตมิติคงที่สามารถหาได้ด้วยผลรวมของคะแนนสำหรับแต่ละเวกเตอร์การสังเกตที่ลำดับนั้น ๆ จึงเป็นการเหมาะสมที่จะนำไปเป็นข้อมูลเข้าให้กับซัพพอร์ตเวกเตอร์แมชชีนซึ่งเป็นตัวจำแนกประเภทแบบสถิติ

2.4.6.2 แบบจำลองฮิดเดนมาร์คอฟแบบเจเนอเรทีฟ

ให้แบบจำลองแบบเจเนอเรทีฟของลำดับการสังเกตเป็นแบบจำลองฮิดเดนมาร์คอฟที่แต่ละสถานะในการกระจายขาออกเป็นการประกอบกันของส่วนประกอบเกาส์ การเปลี่ยนระหว่างสถานะถูกกำหนดโดยเมทริกซ์การเปลี่ยนสถานะ (State transition matrix) และเป็นแบบซ้ายไปขวา (Left-to-right) โดยไม่มีการข้ามสถานะ

ให้แบบจำลองฮิดเดนมาร์คอฟมี N สถานะ แต่ละสถานะถูกจำลองโดยการกระจายขาออกซึ่งมีความน่าจะเป็นสถานะขาออกของตัวสังเกต \bar{o}_t ณ สถานะ j เป็น $b_j(\bar{o}_t)$ ให้การกระจายขาออกถูกจำลองโดยส่วนผสมแบบเกาส์ K ส่วน และให้ jk คือ ดัชนีสำหรับส่วนผสมที่ k ที่สถานะ j ดังนั้นความน่าจะเป็นสถานะขาออกเป็น

$$b_j(\bar{o}_t) = \sum_{k=1}^K w_{jk} b_{jk}(\bar{o}_t) \quad (2.100)$$

โดยแต่ละส่วนประกอบเกาส์มีพารามิเตอร์ 3 ตัว คือ ตัวถ่วงน้ำหนัก w_{jk} เวกเตอร์ค่าเฉลี่ย $\bar{\mu}_{jk}$ และเมทริกซ์โควาเลี่ยนซ์ Σ_{jk} นอกจากนั้นความน่าจะเป็นในการเปลี่ยนสถานะจาก i ไปเป็น j คือ a_{ij} และพารามิเตอร์เวกเตอร์ $\bar{\theta}$ มีค่าดังนี้

$$\bar{\theta} = \{w_{jk}, \bar{\mu}, \Sigma_{jk}\}_{j \in 1 \dots N, k \in 1 \dots K}, \{a_{ij}\}_{i \in 1 \dots N, j \in 1 \dots N} \quad (2.101)$$

ความน่าจะเป็นในการเปลี่ยนสถานะสุดท้ายสามารถอนุมานได้โดยการนำผลรวมของความน่าจะเป็นในการเปลี่ยนสถานะอื่น ๆ จากสถานะสุดท้ายนี้มาลบออกจากหนึ่ง จึงไม่ได้ถูกนำมาพิจารณาแบบพารามิเตอร์อิสระ

การส่งคะแนน (Score-mapping)

ให้การสังเกตเป็นแบบ i.i.d. ผลรวมของทุกวิถี (Path, ϕ) เป็นดังนี้

$$P(\bar{O} | \bar{\theta}) = \sum_{\phi \in \Phi} \left[\prod_{t=1}^T P^\phi(\bar{o}_t | \bar{\theta}) \right] = \sum_{\phi \in \Phi} \left[\prod_{t=1}^T b_{s(t)}^\phi(\bar{o}_t) a^\phi(t, t+1) \right] \quad (2.102)$$

โดย $a(t, t+1)$ คือ ความน่าจะเป็นในการเปลี่ยนสถานะจากเวลา t ไป $t+1$
 $s(t)$ คือ สถานะที่เวลา t

ดังนั้นการส่งคะแนนด้วยพารามิเตอร์ต่าง ๆ ของลำดับการสังเกต \bar{O} สำหรับแต่ละส่วนประกอบ jk และการเปลี่ยน ij หาได้ดังนี้ [13], [14]

$$\nabla_{\bar{\mu}_{jk}} \ln P(\bar{O} | \bar{\theta}) = \sum_{t=1}^T \gamma_{jk}(t) S_{[t, jk]}^T \quad (2.103)$$

$$\nabla_{\text{vec}(\Sigma_{jk})} \ln P(\bar{O} | \bar{\theta}) = \sum_{t=1}^T \gamma_{jk}(t) \frac{1}{2} \left[-[\text{vec}(\Sigma_{jk}^{-1})]^T + \{S_{[t, jk]} \otimes S_{[t, jk]}\}^T \right] \quad (2.104)$$

$$\nabla_{w_{jk}} \ln P(\bar{O} | \bar{\theta}) = \sum_{t=1}^T \gamma_{jk}(t) \left[\frac{1}{w_{jk}} - \frac{\gamma(t)}{w_{j1} \gamma_{jk}(t)} \right] \quad (2.105)$$

$$\nabla_{a_{ii}} \ln P(\bar{O} | \bar{\theta}) = \sum_{t=1}^T \left[\frac{\gamma_i(t)}{a_{ii}} - \frac{1}{T a_{ii} (1 - a_{ii})} \right] \quad (2.106)$$

โดย

$$S_{[t,jk]} = (\bar{o}_t - \bar{\mu}_{jk})^T \Sigma_{jk}^{-1} \quad (2.107)$$

$$\hat{\gamma}_{jk}(t) = \frac{\gamma_j(t)}{\sum_{\tau=1}^T \gamma_j(\tau)} \left(\frac{w_{jk} \mathcal{N}(\bar{o}_t; \bar{\mu}_{jk}, \Sigma_{jk})}{\sum_{i=1}^K w_{ji} \mathcal{N}(\bar{o}_t; \bar{\mu}_{ji}, \Sigma_{ji})} \right) \quad (2.108)$$

$$\gamma_j(t) = P(s(t) = i | \bar{O}, \bar{\theta}) \quad (2.109)$$



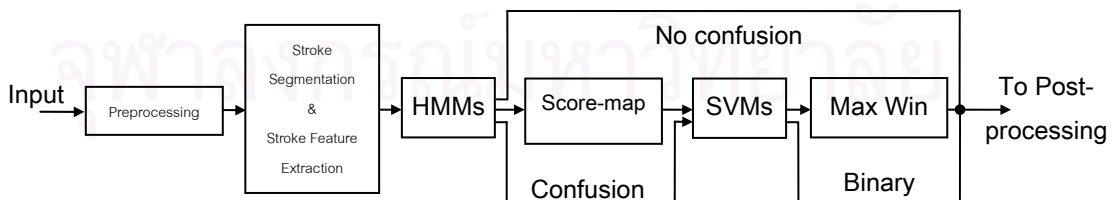
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

การรู้จำตัวอักษรลายมือเขียนภาษาไทยแบบออนไลน์ด้วยแบบจำลองฮิดเดน มาร์คอฟและซัพพอร์ตเวกเตอร์แมชชีน

ในบทนี้จะกล่าวถึงการรู้จำตัวอักษรลายมือเขียนภาษาไทยแบบออนไลน์ที่ใช้ระบบรู้จำที่ประกอบด้วยแบบจำลองฮิดเดนมาร์คอฟ และซัพพอร์ตเวกเตอร์แมชชีนซึ่งสามารถแสดงแผนภาพการทำงานได้ดังรูปที่ 3.1 เป็นระบบรู้จำหลัก

สำหรับการทำงานของแบบจำลองฮิดเดนมาร์คอฟ และซัพพอร์ตเวกเตอร์แมชชีนที่ใช้ในวิทยานิพนธ์ฉบับนี้จะอธิบายโดยละเอียดให้หัวข้อ 3.1 และ 3.2 ตามลำดับ ผลลัพธ์จากกระบวนการแรก (แบบจำลองฮิดเดนมาร์คอฟ) จะผ่านระเบียบวิธีที่เสนอขึ้นใหม่เพื่อคัดหาชนิดที่ใช้ในกระบวนการที่สอง (ซัพพอร์ตเวกเตอร์แมชชีน) ซึ่งได้แสดงรายละเอียดไว้ในหัวข้อ 3.2.3 เมื่อทดสอบวิธีที่กล่าวมากับตัวอักษรระดับกลางและระดับล่างพบว่าได้ผลน่าพอใจ แต่สำหรับกรณีสระและวรรณยุกต์ระดับบนผลไม่เป็นที่น่าพอใจเนื่องจากกลุ่มสระที่มีลักษณะคล้ายกัน คือ ิ, ี, ึ และ ื ซึ่งวิธีการแก้ปัญหาดังกล่าวจะแสดงรายละเอียดไว้ในหัวข้อ 3.3 โดยใช้ระบบรู้จำที่ประกอบด้วยแบบจำลองฮิดเดนมาร์คอฟ 2 ชั้นตอนดังรูปที่ 3.9 ส่วนวรรณยุกต์เอกที่มีเพียงเส้นตรงเส้นเดียวนั้นไม่สามารถฝึกฝนแบบจำลองฮิดเดนมาร์คอฟได้ ทำให้จำเป็นต้องคัดแยกด้วยวิธีอื่น โดยทั่วไปแล้วหากพบลายมือเขียนที่อยู่ระดับบนและมีเพียงเส้นตรงเพียงเส้นเดียวในแนวตั้งซึ่งไม่อาจแบ่งเป็นเส้นย่อยได้ จะตัดสินใจตอบเป็นวรรณยุกต์เอก แต่หากเส้นตรงนั้นอยู่ในแนวนอนจะตัดสินใจเป็นไม้หันอากาศ (~) [15]



รูปที่ 3.1 แผนภาพการทำงานของระบบรู้จำตัวอักษรลายมือเขียนภาษาไทยแบบออนไลน์ด้วยแบบจำลองฮิดเดนมาร์คอฟและซัพพอร์ตเวกเตอร์แมชชีน

3.1 การทำงานของแบบจำลองฮิดเดนมาร์คอฟในระบบรู้จำตัวอักษร

ข้อมูลของตัวอักษรซึ่งมีลักษณะเป็นลำดับของจุดพิกัดที่เรียงกันตามเวลาที่เขียนหลังจากผ่านกระบวนการก่อนหน้า ในขั้นตอนต่อไปคือการนำมาแบ่งออกเป็นเส้นย่อยหลายเส้น และสกัดคุณลักษณะสำคัญของแต่ละเส้นย่อยเพื่อใช้เป็นข้อมูลเข้าสำหรับแบบจำลองฮิดเดนมาร์คอฟ หากอยู่ในกระบวนการฝึกฝน ข้อมูลออกคือแบบจำลองของตัวอักษรนั้น ๆ (λ) แต่หากอยู่ในกระบวนการทดสอบ ข้อมูลออกคือความน่าจะเป็นภายหลัง ($P(\bar{O}|\lambda)$) ของทุก ๆ แบบจำลอง เมื่อ \bar{O} คือลำดับการสังเกต (Observation sequence) ที่ประกอบไปด้วยเวกเตอร์ของคุณลักษณะสำคัญจากแต่ละเส้นย่อยของข้อมูลของตัวอักษรเข้า

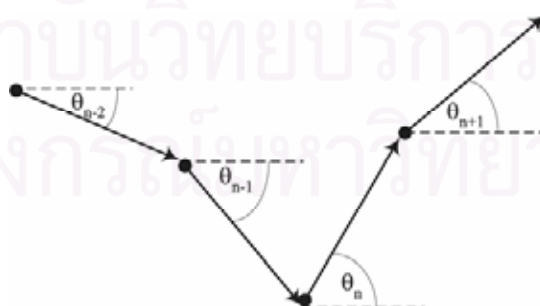
3.1.1 การแบ่งส่วนของเส้น (Stroke Segmentation)

ข้อมูลตัวอักษรที่ได้รับมาจะถูกแบ่งออกเป็นเส้นย่อย (Sub-stroke) โดยจุดที่ใช้แบ่งเส้นรวมออกเป็นเส้นย่อยจะมีลักษณะตามข้อกำหนดข้อใดข้อหนึ่งต่อไปนี้

- เมื่อจุดที่พิจารณาเป็นจุดที่เกิดมุมแหลม จากรูปที่ 3.2 จุดที่ n เป็นจุดดังกล่าวเมื่อ

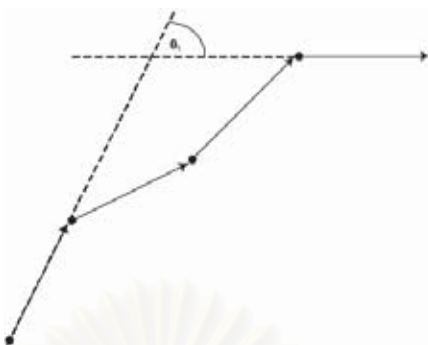
$$|\theta_{n-2} + \theta_{n-1} - \theta_n - \theta_{n+1}| > \theta_r \quad (3.1)$$

เมื่อ θ_r เป็นมุมคงที่ค่าหนึ่ง จากการทดลองใน [15] พบว่า $\theta_r = \frac{\pi}{3}$ เป็นค่าที่เหมาะสมกับตัวอักษรภาษาไทย



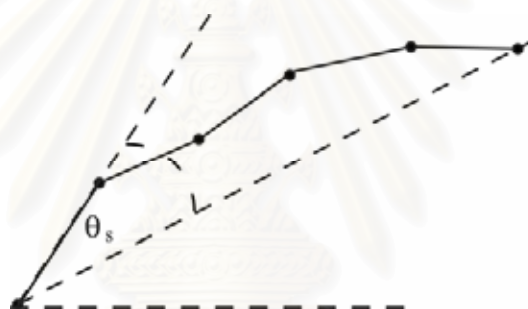
รูปที่ 3.2 มุมที่พบว่าเป็นมุมแหลม

- เมื่อจุดที่พิจารณาเป็นจุดที่มีเส้นสัมผัสตัดทำมุมกับเส้นสัมผัสของจุดเริ่มแรกของเส้นย่อย θ_r เกินค่ามุมที่กำหนดไว้ $\theta_{threshold,t}$ ดังแสดงในรูปที่ 3.3



รูปที่ 3.3 มุมระหว่างเส้นสัมผัสของจุดที่พิจารณากับเส้นสัมผัสของจุดแรกของเส้นย่อย

- เมื่อเส้นที่ลากจากจุดแรกของเส้นย่อยถึงจุดที่พิจารณาทำมุม θ_s กับเส้นสัมผัสที่จุดแรกของเส้นย่อยเกินค่ามุมที่กำหนดไว้ $\theta_{threshold,s}$ ดังแสดงในรูปที่ 3.4



รูปที่ 3.4 มุมระหว่างเส้นที่ลากจากจุดแรกของเส้นย่อยถึงจุดที่พิจารณากับเส้นสัมผัสที่จุดแรกของเส้นย่อย

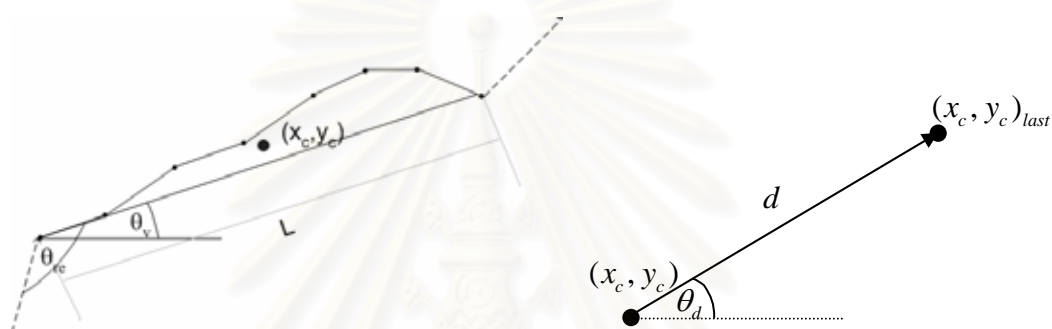
จากการทดลองใน [15] เมื่อทดลองใช้การแบ่งส่วนของเส้นด้วยกรรมวิธีดังที่กล่าวมากับตัวอักษรภาษาไทยพบว่าใช้งานได้ดี ดังแสดงให้เห็นได้ในตัวอย่างตัวอักษรในรูปที่ 3.5



รูปที่ 3.5 ตัวอย่างลายมือเขียนที่ถูกแบ่งส่วนของเส้นตามข้อกำหนดต่าง ๆ

3.1.2 การสกัดคุณลักษณะสำคัญ (Feature Extraction)

หลังจากการแบ่งส่วนของเส้นจะนำแต่ละเส้นมาสกัดคุณลักษณะสำคัญเพื่อนำไปเป็นเวกเตอร์ของคุณลักษณะและนำไปเป็นลำดับการสังเกตในแบบจำลองฮิดเดนมาร์คอฟต่อไปในวิทยานิพนธ์นี้ประมาณค่าเส้นย่อยเป็นเส้นตรงเส้นหนึ่งประกอบไปด้วยคุณลักษณะ 7 ชนิด ดังแสดงในรูปที่ 3.6 โดยคุณลักษณะ 5 ชนิดแรกนำมาจาก [15] และคุณลักษณะอีก 2 ชนิดปรับปรุงมาจาก [20] และสามารถอธิบายคุณลักษณะของเส้นย่อย i ได้ดังนี้



รูปที่ 3.6 คุณลักษณะสำคัญของเส้นย่อย

1. พิกัดตามแนวนอนของจุดศูนย์กลางของเส้นย่อย (x_c) หาได้จาก

$$x_c(i) = \frac{1}{N} \sum_{n=1}^N x(i) \quad (3.2)$$

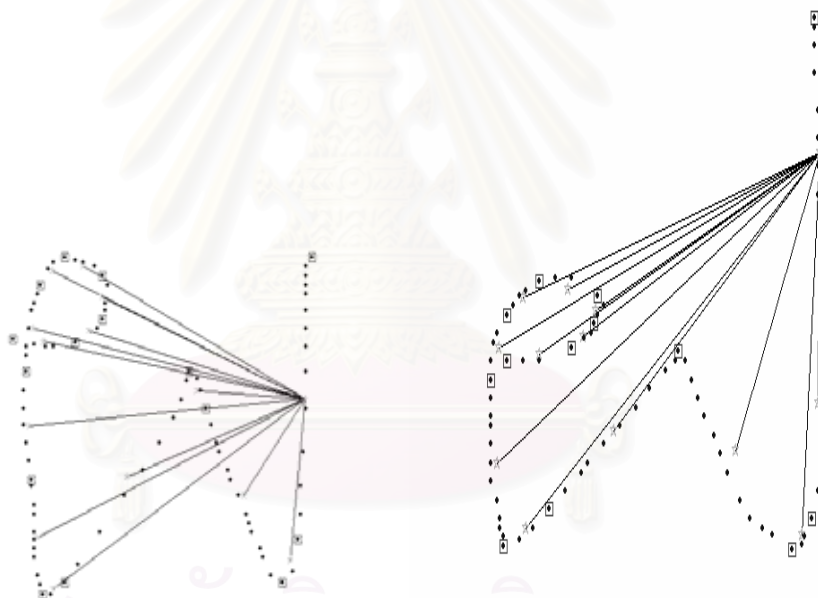
โดย $x(i)$ คือ พิกัดตามแนวนอนของจุดที่ i ของเส้นย่อย และ N จำนวนจุดทั้งหมดบนเส้นย่อยนั้น

2. พิกัดตามแนวตั้งของจุดศูนย์กลางของเส้นย่อย (y_c) หาได้จาก

$$y_c(i) = \frac{1}{N} \sum_{n=1}^N y(i) \quad (3.3)$$

โดย $y(i)$ คือ พิกัดตามแนวตั้งของจุดที่ i ของเส้นย่อย และ N จำนวนจุดทั้งหมดบนเส้นย่อยนั้น

3. มุมของเส้นที่ลากจากจุดต้นไปถึงจุดปลายของเส้นย่อยเมื่อเทียบกับเส้นแนวนอน ($\theta_v(i)$)
4. มุมระหว่างเส้นย่อยที่พิจารณากับเส้นย่อยก่อนหน้า ($\theta_{re}(i)$)
5. ความยาวของเส้นตรงที่ลากจากจุดแรกถึงจุดสุดท้ายของเส้นย่อยนั้น ($L(i)$)
6. ความยาวของเส้นตรงที่ลากจากจุดศูนย์กลางของเส้นย่อยนั้นถึงจุดศูนย์กลางของเส้นย่อยสุดท้ายของตัวอักษรนั้น (d) ดังแสดงในรูปที่ 3.6 และรูปที่ 3.7
7. มุมของเส้นตรงในข้อ 6 ที่กระทำกับแกนแนวนอน (θ_d)



รูปที่ 3.7 ตัวอย่างเส้นตรงที่ลากจากจุดศูนย์กลางของเส้นย่อยถึงจุดศูนย์กลางของเส้นย่อยสุดท้าย

3.1.3 การรู้จำตัวอักษรด้วยแบบจำลองฮิดเดนมาร์คอฟ

หลังจากสกัดคุณลักษณะสำคัญแล้วจะได้เวกเตอร์ลักษณะ (Feature vector) ซึ่งมี 2 มิติ มิติแรกคือจำนวนคุณลักษณะในหัวข้อ 3.1.2 ซึ่งในที่นี้จะมีขนาดเท่ากับ 7 เท่ากันหมดทุก ๆ ตัวอักษร ส่วนมิติที่สองมีขนาดเท่ากับจำนวนเส้นย่อยทั้งหมดในตัวอักษรนั้น ซึ่งแต่ละตัวอักษรจะมีจำนวนเส้นย่อยไม่เท่ากัน จากนั้นป้อนเวกเตอร์ลักษณะของตัวอักษรตัวที่ i ที่จะใช้ในการฝึกฝนทั้งหมดให้แบบจำลองฮิดเดนมาร์คอฟมาเพื่อทำการฝึกฝนหาพารามิเตอร์ของแบบจำลองของ

ตัวอักษรที่ i (λ_i) หลังจากฝึกฝนครบทุกชนิดตัวอักษรเราจะได้แบบจำลองฮิดเดนมาร์คอฟจำนวนเท่ากับจำนวนชนิดตัวอักษรทั้งหมด

ในกรณีที่มี K ชนิด การตัดสินใจชนิดของลำดับการสังเกต \bar{O} ต้องหาค่าความน่าจะเป็นภายหลัง $P(\bar{O} | \lambda_i)$ ของทุก ๆ แบบจำลอง ดังนั้นข้อมูลออกของแบบจำลองฮิดเดนมาร์คอฟคือความน่าจะเป็นภายหลังของทุก ๆ แบบจำลอง, $P(\bar{O} | \lambda_1), P(\bar{O} | \lambda_2), \dots, P(\bar{O} | \lambda_i), \dots, P(\bar{O} | \lambda_K)$ และเราจะตัดสินใจว่า \bar{O} เป็นตัวอักษรชนิดที่ i ถ้า $P(\bar{O} | \lambda_i)$ มีค่ามากที่สุดดังสมการ

$$i = \arg \max_{1 \leq i \leq K} P(\bar{O} | \lambda_i) \quad (3.4)$$

สำหรับในวิทยานิพนธ์ฉบับนี้จะไม่ตัดสินใจด้วยกระบวนการดังที่กล่าวมาโดยทันทีเนื่องจากคำตอบที่ได้จากแบบจำลองฮิดเดนมาร์คอฟมีการตัดสินใจผิดพลาดเป็นตัวอักษรที่มีลักษณะคล้ายกันค่อนข้างมาก ดังนั้นจึงนำเอาค่าพารามิเตอร์ของแบบจำลอง และ $P(\bar{O} | \lambda_i)$ ไปเพื่อสกัดเวกเตอร์คุณลักษณะสำคัญขึ้นมาใหม่ที่เรียกว่า คะแนน (Score) เพื่อใช้เป็นคุณลักษณะสำคัญป้อนเข้าของซัพพอร์ตเวกเตอร์แมชชีนซึ่งจะกล่าวถึงในหัวข้อถัดไป

3.2 การทำงานของซัพพอร์ตเวกเตอร์แมชชีนในระบบรู้จำตัวอักษร

เนื่องจากเวกเตอร์คุณลักษณะสำคัญที่ป้อนเข้าซัพพอร์ตเวกเตอร์แมชชีนต้องมีมิติเท่ากันในทุกมิติ ซึ่งต่างแบบจำลองฮิดเดนมาร์คอฟที่ต้องการแค่มิติของจำนวนคุณลักษณะสำคัญเท่านั้นเท่านั้น ดังนั้นเราจึงใช้ข้อได้เปรียบข้อนี้ของแบบจำลองฮิดเดนมาร์คอฟ มาช่วยในการสกัดคุณลักษณะสำคัญโดยส่งเวกเตอร์คุณลักษณะสำคัญไปในปริภูมิคะแนนดังนั้นเวกเตอร์คุณลักษณะสำคัญที่จะป้อนเข้าคือ คะแนน

3.2.1 การสกัดคุณลักษณะสำคัญ (Feature Extraction)

หลังจากเราได้เวกเตอร์คุณลักษณะสำคัญของ \bar{O} จากหัวข้อ 3.1.2 และแบบจำลองของทุกตัวอักษรแล้ว เพื่อทำให้มิติของเวกเตอร์ดังกล่าวมีขนาดคงที่เราจึงส่งเข้าไปในปริภูมิคะแนนดังสมการที่ (2.103) ถึง (2.109) เราจะได้เวกเตอร์ของคะแนนแบบ Likelihood $\phi^{lik}(\bar{O})$ ดังนี้

$$\varphi^{lik}(\bar{O}) = \nabla_{\bar{\theta}} \ln P(\bar{O} | \bar{\theta}) = \begin{bmatrix} \nabla_{\bar{\theta}_{s_1}} \ln P(\bar{O} | \bar{\theta}) \\ \vdots \\ \nabla_{\bar{\theta}_{s_j}} \ln P(\bar{O} | \bar{\theta}) \\ \vdots \\ \nabla_{\bar{\theta}_{s_N}} \ln P(\bar{O} | \bar{\theta}) \\ \vdots \\ \nabla_{\bar{\theta}_A} \ln P(\bar{O} | \bar{\theta}) \end{bmatrix} \quad (3.5)$$

ซึ่งมิติของเวกเตอร์นี้จะมีขนาดคงที่เท่ากับ $N[K(f^2 + f + 1) + 1]$ เมื่อ N คือจำนวนสถานะของแบบจำลองฮิดเดนมาร์คอฟ K คือจำนวนส่วนผสมของแบบจำลองแบบเกาส์ของแต่และสถานะ และ f คือจำนวนคุณลักษณะสำคัญ ซึ่งในวิทยานิพนธ์ฉบับนี้เราใช้ 5 สถานะ และ 5 ส่วนผสม ซึ่งได้จากการทดลองใน [15] ดังนั้นมิติของเวกเตอร์คะแนนแบบ Likelihood คือ 1430 เท่ากันทุกตัวอักษร

แต่ปริภูมิคะแนนแบบ Likelihood นั้นไม่เหมาะสมเนื่องจากเวกเตอร์พารามิเตอร์จะต้องอ้างจากแบบจำลองเดียว แต่ปัญหาคือเราไม่ทราบแบบจำลองของลำดับการสังเกตใด ๆ ที่เข้ามา ซึ่งเราอาจเลือกใช้เวกเตอร์พารามิเตอร์ได้ 2 วิธี ดังนี้

1. นำตัวอักษรทุกชนิดมาเพื่อฝึกฝนแบบจำลองใหม่เพียงหนึ่งเดียว แต่เนื่องจากภาษาไทยมีจำนวนชนิดของตัวอักษรมาก จึงนั้นทำได้ยาก และหากทำได้ แบบจำลองที่ได้จะมีลักษณะกว้างเกินไปจนกระทั่งคะแนนที่ได้ไม่มีนัยสำคัญในการตัดจำแนกประเภท
2. เลือกใช้แบบจำลองที่ฝึกฝนมาแล้วของแต่ละตัวอักษรมาหนึ่งแบบจำลอง โดยวิธีการเลือกนั้นอาจเลือกจากการตัดสินใจชนิดของแบบจำลองฮิดเดนมาร์คอฟ แต่เนื่องจากการตัดสินใจนี้ยังมีข้อผิดพลาดอยู่ ดังนั้นพารามิเตอร์ของแบบจำลองที่เลือกมาใช้นั้นอาจทำให้ได้คะแนนที่มีนัยสำคัญน้อย หรือผิดพลาดได้

จากเหตุผลที่กล่าวมา เราจึงไม่เลือกใช้ปริภูมิคะแนนแบบ Likelihood แต่จะทดลองใช้ปริภูมิคะแนนแบบ Likelihood ratio แทน ดังนั้นเวกเตอร์คะแนนจะมีลักษณะดังนี้

$$\varphi_{y_1, y_2}^{lr}(\bar{O}) = \begin{bmatrix} \varphi_{y_1}^{lik}(\bar{O}) \\ -\varphi_{y_2}^{lik}(\bar{O}) \end{bmatrix}, \quad \varphi_{y_2, y_1}^{lr}(\bar{O}) = \begin{bmatrix} \varphi_{y_2}^{lik}(\bar{O}) \\ -\varphi_{y_1}^{lik}(\bar{O}) \end{bmatrix} \quad (3.6)$$

เมื่อ $\varphi_{y_1}^{lik}(\bar{O})$ และ $\varphi_{y_2}^{lik}(\bar{O})$ คือคะแนนในปริภูมิแบบ Likelihood ของ \bar{O} ที่ได้จากพารามิเตอร์ของแบบจำลองตัวอักษรชนิดที่ y_1 และ y_2 ตามลำดับ จะเห็นได้ว่าการส่งลำดับการสังเกต \bar{O} ไปในปริภูมิคะแนนแบบนี้จะได้คะแนน 2 คะแนนในปริภูมิ ซึ่งในแต่ละคะแนนมีลักษณะไม่สมมาตร เนื่องจากครึ่งหนึ่งจะมีขั้วบวก และอีกครึ่งหนึ่งมีขั้วลบ

วิทยานิพนธ์ฉบับนี้จึงได้เสนอปริภูมิคะแนนขึ้นใหม่ เรียกว่า ปริภูมิคะแนนแบบ Symmetric likelihood ratio ดังแสดง

$$\varphi_{y_1, y_2}^{slr}(\bar{O}) = \begin{bmatrix} \varphi_{y_1}^{lik}(\bar{O}) + \varphi_{y_2}^{lik}(\bar{O}) \\ -\varphi_{y_1}^{lik}(\bar{O}) - \varphi_{y_2}^{lik}(\bar{O}) \end{bmatrix} \quad (3.7)$$

การส่งเวกเตอร์คุณลักษณะสำคัญของลำดับการสังเกตไปในปริภูมิคะแนนนี้จะมีคะแนนได้หนึ่งเดียวและมีคุณสมบัติแบบสมมาตรและมีความเหมาะสมในการใช้งานกับตัวอักษรภาษาไทยมากกว่า ดังผลการทดสอบใน [21]

3.2.2 การนอร์มัลไลซ์ปริภูมิคะแนน (Score space Normalization)

การนอร์มัลไลซ์ปริภูมิคะแนนเป็นสิ่งสำคัญ เนื่องจากลำดับการสังเกตแต่ละลำดับมีความแตกต่างกันทั้งในเรื่องของขนาดและช่วงระยะเวลา (Duration) เพื่อรองรับการใช้งานของซอฟต์แวร์ตัวอักษรแมชชีน เราใช้การนอร์มัลไลซ์ 4 ขั้นตอนดังนี้ การนอร์มัลไลซ์ความยาวลำดับ (Sequence length normalization) [18] การนอร์มัลไลซ์คะแนนเพื่อการฝึกฝนและทดสอบ [13] การทำปริภูมิคะแนนให้ขาว (Score-space whitening) [18] และ การนอร์มัลไลซ์เชิงทรงกลม (Spherical normalization) [17], [19] ตามลำดับ

3.2.2.1 การนอร์มัลไลซ์ความยาวลำดับ (Sequence length normalization)

ในการเขียนตัวอักษร ถึงแม้ตัวอักษรจะมีลักษณะคล้ายกันแต่อาจจะมีอัตราการเขียนไม่เท่ากันทำให้ช่วงระยะเวลาไม่เท่ากัน ทำให้พารามิเตอร์ของแบบจำลองบางพารามิเตอร์มีมิติไม่เท่ากัน เช่น เมตริกซ์ของความน่าจะเป็นภายหลังของแต่ละสถานะ $\gamma_j(t)$ มีความยาวต่างกัน เราจึงนอร์มัลไลซ์ด้วย ผลรวมของความน่าจะเป็นภายหลังทั้งหมดที่ครอบคลุมช่วงเวลาของแต่ละลำดับการสังเกตของแต่ละสถานะดังสมการ

$$\gamma_j^N(t) = \frac{\gamma_j(t)}{\sum_{\tau=1}^T \gamma_j(\tau)} \quad (3.8)$$

จากนั้นนำความน่าจะเป็นภายหลังที่ถูกลบออกแล้ว $\gamma_j^N(t)$ ไปแทน $\gamma_j(t)$ ในสมการ (2.103) ถึงสมการ (2.109)

3.2.2.2 การนอร์มัลไลซ์คะแนนเพื่อการฝึกฝนและทดสอบ

คะแนนที่ใช้ในการฝึกฝนและทดสอบล้วนมาจากลำดับการสังเกตต่าง ๆ กัน ซึ่งมีช่วงระยะเวลา (Duration) ยาวไม่เท่ากัน เพื่อเพิ่มความถูกต้อง คะแนนจะถูกนอร์มอลไลซ์ด้วยช่วงระยะเวลาของลำดับการสังเกตนั้น ๆ ดังสมการ

$$\tilde{\varphi}(\vec{O}) = \varphi(\vec{O}) / T \quad (3.9)$$

โดย T คือช่วงระยะเวลาของลำดับการสังเกต \vec{O}

3.2.2.3 การทำปริภูมิคะแนนให้ขาว (Score-space whitening)

จากเหตุผลในหัวข้อ 2.4.5 การเปรียบเทียบกันระหว่างคะแนนใด ๆ ในปริภูมิคะแนน เราทำได้โดยการหาผลคูณภายใน ดังนั้นจึงต้องการเวกเตอร์มูลฐานที่ตั้งฉากกัน (Orthonormal) โดยการนอร์มัลไลซ์เคอร์เนลด้วยส่วนกลับของรากที่สองของโควาเลนซ์เมตริกซ์ $\Sigma^{-1/2}$ ซึ่งเราอาจมองอีกอย่างหนึ่งได้ว่าเป็นการทำปริภูมิคะแนนให้ขาว (Score-space whitening) โดยการใช้ส่วนกลับของโควาเลนซ์เมตริกซ์ Σ_{sc}^{-1} ส่งปริภูมิคะแนนไปสู่ปริภูมิยูคลีเดียน (Euclidean space) ดังนั้นคะแนนที่ถูกทำให้ขาว $\varphi^N(\vec{O})$ แสดงได้ดังนี้

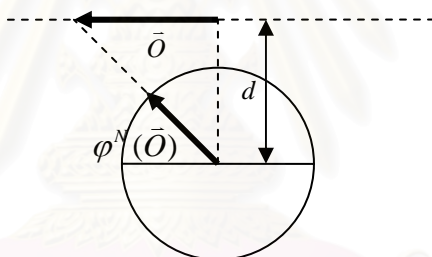
$$\varphi^N(\vec{O}) = \Sigma^{-1/2} \tilde{\varphi}(\vec{O}) \quad (3.10)$$

โดย $\Sigma_{sc} = E\{[\tilde{\varphi}(\vec{O}) - E(\tilde{\varphi}(\vec{O}))][\tilde{\varphi}(\vec{O}) - E(\tilde{\varphi}(\vec{O}))]^T\}$ (3.11)

ในทางปฏิบัติเนื่องจาก Σ_{sc} บางค่าไม่มีคุณสมบัติบวกแน่นอน (Positive definite) ทำให้การหาค่า Σ_{sc}^{-1} ทำไม่ได้และเพื่อลดการคำนวณลงเราจึงประมาณ Σ_{sc}^{-1} แบบการประมาณแนวทแยง (Diagonal approximation) และ $\tilde{\varphi}(\bar{O}_i)$ คือ คะแนนที่ถูกนอร์มัลไลซ์เพื่อการฝึกฝนและทดสอบ ดังอธิบายในหัวข้อก่อนหน้า

3.2.2.4 การนอร์มัลไลซ์เชิงทรงกลม (Spherical normalization)

เวกเตอร์คะแนนที่ได้จากการส่งของลำดับการสังเกตด้วยพารามิเตอร์ของแบบจำลองจะมีขนาดของแต่ละคะแนนต่าง ๆ กัน เพื่อให้มีการใช้คะแนนในการเปรียบเทียบที่ดี คะแนนควรมีขนาดในมาตราส่วนเดียวกัน ซึ่งทำได้โดยส่งเวกเตอร์คะแนนไปสู่ปริภูมิคะแนนที่มีมิติเพิ่มขึ้น ดังเช่นการส่งจากพื้นผิวระนาบไปบนพื้นผิวทรงกลม ซึ่งทำได้หลายวิธีแต่วิธีที่ใช้ภายในวิทยานิพนธ์ฉบับนี้แสดงดังรูปที่ 3.8



รูปที่ 3.8 การนอร์มัลไลซ์เชิงทรงกลม

การส่งนี้จะถูกประยุกต์ใช้หลังจากคะแนนที่ถูกทำให้ขาวแล้วดังแสดง

$$\varphi^N(\bar{O}) \rightarrow \Phi(\varphi^N(\bar{O})) = \frac{1}{\sqrt{\varphi^N(\bar{O}) \cdot \varphi^N(\bar{O}) + d^2}} \begin{bmatrix} \varphi^N(\bar{O}) \\ d \end{bmatrix} \quad (3.12)$$

ภายในวิทยานิพนธ์ฉบับนี้เรากำหนดให้แต่ละเวกเตอร์คะแนนมีขนาดและระยะห่างมีค่าหนึ่งเป็น d ดังนั้นแต่ละเวกเตอร์คะแนนจะมีจำนวนมิติเพิ่มขึ้นหนึ่งมิติคือ d

3.2.3 กลุ่มของชนิดที่มีการตัดสินใจผิดพลาด

เนื่องจากซอฟต์แวร์เวกเตอร์แมชชีนเป็นระบบรู้จำแบบไบนารี แต่งานรู้จำตัวอักษรภาษาไทยมีจำนวนชนิดตัวอักษรมากกว่าสองชนิด เราจึงต้องมีการเลือกกลุ่มของชนิดตัวอักษรเพื่อลดความสับสนและเพิ่มอัตราการรู้จำ ทั้งนี้เนื่องจากโอกาสในการเกิดความผิดพลาดจะแปรผันตรงกับจำนวนชนิด ยิ่งจำนวนชนิดน้อยลงเท่าไรโอกาสตัดสินใจก็ยิ่งมากขึ้น トラบดที่ภายในกลุ่มของชนิดนั้นมีชนิดที่ถูกต้องอยู่ สำหรับการหากรูมของชนิดตัวอักษร เราจะอาศัยแบบจำลองฮิดเดนมาร์คอฟเป็นบรรทัดฐานในการตัดสินใจเลือกชนิดตัวอักษร เนื่องด้วยแบบจำลองฮิดเดนมาร์คอฟสามารถเป็นตัวจำแนกประเภทแบบมัลติคลาสได้โดยไม่ต้องอาศัยเทคนิคอื่นใด ข้อดีของแบบจำลองฮิดเดนมาร์คอฟคือมีการเปรียบเทียบข้อมูล (แบบจำลอง) ของชนิดทั้งหมดและเรียงลำดับความน่าจะเป็นที่ลำดับการสังเกตจะเป็นชนิดนั้น ๆ ดังนั้นถ้ามีการตัดสินใจผิดพลาดเกิดขึ้นชนิดที่ถูกต้องมักจะอยู่ในอันดับถัดลงมาไม่มากนัก จากการทดลองกับกลุ่มตัวอักษรภาษาไทยระดับกลางที่ใช้ในการทดสอบในหัวข้อที่ 4.2 พบว่าทั้งค่าเฉลี่ยและค่าฐานนิยมของจำนวนอันดับที่ครอบคลุมชนิดที่ถูกต้อง (Top N) ของทุก ๆ ชนิดมีค่าประมาณ 4 อันดับ และค่าสูงสุดคือ 16 อันดับ ซึ่งมีเพียงค่าเดียว ในทางกลับกัน ถ้าการตัดสินใจถูกต้อง เราพบว่าความน่าจะเป็นอันดับถัดลงมาส่วนใหญ่จะเป็นชนิดที่มีการตัดสินใจผิดพลาด

การหากรูมของชนิดที่มีการตัดสินใจผิดพลาดของชนิดใด ๆ ในที่นี้จะพิจารณาจากเมตริกซ์ความสับสน (Confusion Matrix) ที่ได้จากการรู้จำด้วยแบบจำลองฮิดเดนมาร์คอฟด้วยชุดทดสอบชุดหนึ่ง โดยกรูมของชนิดที่มีการตัดสินใจผิดพลาดของชนิด i นี้ คือ ชนิด i และกรูมของชนิดที่มีชนิด i เป็นชนิดที่ตัดสินใจผิดเป็นอันดับแรก

สำหรับการหากรูมของชนิดที่มีการตัดสินใจผิดพลาดของลำดับการสังเกตใด ๆ ที่ไม่ทราบชนิดทำได้โดยการตัดสินใจด้วยแบบจำลองฮิดเดนมาร์คอฟว่าเป็นชนิดใด ดังนั้นกรูมของชนิดที่มีการตัดสินใจผิดพลาดของลำดับการสังเกตนี้ คือ กรูมของชนิดที่มีการตัดสินใจผิดพลาดของชนิดดังกล่าว ตัวอย่างเช่น ถ้าเมตริกซ์ความสับสนมีค่าดังตารางที่ 3.1

ตารางที่ 3.1 ตัวอย่างเมตริกซ์ความสับสน

ชนิด	1	2	3	4
1	83	3	4	0
2	6	81	0	3
3	11	0	70	9
4	0	0	4	86

ดังนั้นกลุ่มของชนิดที่มีการตัดสินสับสนของลำดับการสังเกตใด ๆ จะมี 4 กรณีตามจำนวนชนิด ดังนี้

กรณีที่ 1 ถ้าลำดับการสังเกตถูกตัดสินด้วยแบบจำลองฮิดเดนมาร์คอฟเป็นชนิด 1 แล้วกลุ่มของชนิดที่มีการตัดสินสับสน คือ ชนิด 1, 2 และ 3

กรณีที่ 2 ถ้าลำดับการสังเกตถูกตัดสินด้วยแบบจำลองฮิดเดนมาร์คอฟเป็นชนิด 2 แล้วกลุ่มของชนิดที่มีการตัดสินสับสน คือ ชนิด 2

กรณีที่ 3 ถ้าลำดับการสังเกตถูกตัดสินด้วยแบบจำลองฮิดเดนมาร์คอฟเป็นชนิด 3 แล้วกลุ่มของชนิดที่มีการตัดสินสับสน คือ ชนิด 1, 3 และ 4

กรณีที่ 4 ถ้าลำดับการสังเกตถูกตัดสินด้วยแบบจำลองฮิดเดนมาร์คอฟเป็นชนิด 4 แล้วกลุ่มของชนิดที่มีการตัดสินสับสน คือ ชนิด 4

แต่เพื่อการป้องกันความผิดพลาดที่อาจเกิดขึ้นในกรณีที่ลำดับการสังเกตที่นำมาทดสอบเพื่อเมตริกซ์สับสนเป็นลำดับการสังเกตที่ผิดปกติรูปแบบ และเพื่อลดจำนวนชนิดของกลุ่มที่มีการตัดสินสับสน จึงมีการกำหนดค่าคงที่ขึ้นเพื่อให้จำกัดความผิดพลาดดังกล่าว โดยค่าคงที่นี้จะถูกกำหนดให้มีค่าเป็นร้อยละของความผิดพลาด

3.2.4 การใช้ซอฟต์แวร์เวกเตอร์แมชชีนในการรู้จำตัวอักษร

หลังจากการส่งลำดับการสังเกตไปในปริภูมิคะแนนซึ่งมีมิติคงที่ เวกเตอร์คะแนนของแต่ละลำดับการสังเกตจะถูกป้อนเข้าสู่ซอฟต์แวร์เวกเตอร์แมชชีนเพื่อการฝึกฝน หรือทดสอบ ในการฝึกฝนนั้นจะทำในลักษณะเป็นคู่ กล่าวคือจะมีจำนวนซอฟต์แวร์เวกเตอร์แมชชีนเท่ากับจำนวนคู่ ซึ่งจำนวนคู่นี้ได้จากหัวข้อ 3.2.3 โดยกลุ่มของชนิดที่มีการตัดสินสับสนของแต่ละชนิดตัวอักษรจะมีจำนวนคงที่ แต่มีจำนวนไม่เท่ากันทุกตัวอักษร ส่วนชนิดตัวอักษรใดที่ไม่อยู่ในกลุ่มดังกล่าวจะใช้

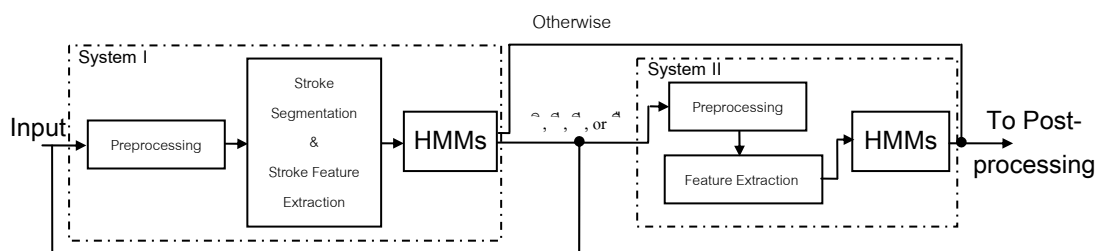
การตัดสินของแบบจำลองฮิดเดนมาร์คอฟเท่านั้น ไม่นำมาป้อนเข้าซอฟต์แวร์เวกเตอร์แมชชีน ถ้าจำนวนของกลุ่มของชนิดที่มีการตัดสินสับสนมีเพียงสองชนิดจะได้ผลการตัดสินใจสุดท้ายในขั้นตอนนี้ แต่หากมีมากกว่าสองชนิดต้องใช้เทคนิคในการทำมัลติคลาสเข้าช่วยดังจะกล่าวในหัวข้อถัดไป

3.2.4 ระเบียบวิธี Max Win

เนื่องจากซอฟต์แวร์เวกเตอร์แมชชีนเป็นตัวจำแนกประเภทแบบไบนารี คือพิจารณาเป็นคู่เท่านั้นดังนั้นการใช้งานในแบบมัลติคลาสต้องอาศัยเทคนิคอื่นเข้าช่วย ซึ่งในวิทยานิพนธ์ฉบับนี้ใช้ระเบียบวิธี Max Win โดยมีหลักการคือ จับคู่ของชนิดทั้งหมดที่เป็นไปได้ เช่นหากมีจำนวน N ชนิดจะมีจำนวนคู่ทั้งหมดที่เป็นไปได้เท่ากับ $N(N-1)/2$ คู่ ซึ่งจะเท่ากับจำนวนซอฟต์แวร์เวกเตอร์แมชชีนในระบบ โดยการตัดสินจะพิจารณาจากชนิดที่ซอฟต์แวร์เวกเตอร์แมชชีนแต่ละตัวตัดสินมา โดยจะเลือกชนิดที่ถูกตัดสินมากที่สุด หากชนิดที่ได้รับการตัดสินมากที่สุดมีจำนวนเท่ากันมากกว่าหนึ่งชนิดจะพิจารณาร่วมกับลำดับความน่าจะเป็นภายหลังที่ได้จากแบบจำลองฮิดเดนมาร์คอฟโดยจะตัดสินเป็นชนิดที่มีค่าของความน่าจะเป็นภายหลังมากกว่า

3.3 การรู้จำสระและวรรณยุกต์ระดับบน

ปัญหาของอักษรกลุ่มนี้แบ่งได้ 2 ปัญหาคือ กลุ่มสระที่มีลักษณะคล้ายกัน คือ ิ, ี, ื และ ุ และวรรณยุกต์เอก ่ ซึ่งการแก้ปัญหของวรรณยุกต์เอกทำได้ดังที่กล่าวไปแล้ว จากการวิเคราะห์ปัญหาของกลุ่มสระ ิ, ี, ื และ ุ พบว่าปัญหาเกิดจากความหลากหลายของรูปแบบและวิธีการเขียน การเขียนในลักษณะลายมือที่ไม่ใช่ตัวคัดจะมีลักษณะการเขียนและทิศทางแตกต่างกันไปตามแต่ละบุคคล ในอักษรภาษาไทย กลุ่มที่มีผลกระทบกับปัญหานี้มากที่สุดคือกลุ่มสระ ิ, ี, ื และ ุ ทำให้วิธีการแบ่งเส้นย่อยและการสกัดคุณลักษณะสำคัญในหัวข้อ 3.1 ไม่สามารถใช้ได้ภายในกลุ่มสระนี้แต่ยังสามารถใช้ได้ดีในการแยกกลุ่มสระทั้ง 4 ตัวนี้ออกจากสระและวรรณยุกต์ระดับบนอื่น ๆ ที่เหลือ เราจึงสามารถแยกกลุ่มสระ ิ, ี, ื และ ุ ออกมาเพื่อฝึกฝนแบบจำลองฮิดเดนมาร์คอฟใหม่ด้วยการออกแบบกระบวนการก่อนหน้าและคุณลักษณะสำคัญใหม่เพื่อใช้แก้ปัญหาดังกล่าวซึ่งมีโครงสร้างระบบดังรูปที่ 3.9



รูปที่ 3.9 แผนภาพการทำงานระบบรู้จำตัวอักษรลายมือเขียนภาษาไทยระดับบนแบบออนไลน์ ด้วยแบบจำลองฮิดเดนมาร์คอฟแบบ 2 ชั้นตอน

3.3.1 กระบวนการก่อนหน้า

เนื่องจากกลุ่มสระ ิ, ี, ื และ ุ มีขนาดและรูปแบบใกล้เคียงกันกระบวนการก่อนหน้าในหัวข้อที่ 2.1 จึงไม่เหมาะสมเพราะส่งผลให้การรู้จำไม่ดีดังผลการทดลองที่แสดงในตารางที่ 4.3 วิทยานิพนธ์นี้จึงทำการออกแบบกระบวนการก่อนหน้าขึ้นมาใหม่โดยพยายามกำจัดลักษณะที่จะทำให้ตัดคลื่นใจผิดพลาดดังนี้

1. นอร์มัลไลซ์ทางแนวแกนนอนให้มีขนาดทางแนวนอนคงที่
2. เปลี่ยนลำดับเส้นที่เขียนผิดวิธีให้ถูกต้องตามหลักการเขียนตัวอักษรภาษาไทย
3. การกำจัดเส้นย่อยที่ผิดรูปแบบการเขียนตามหลักการเขียนตัวอักษรภาษาไทย

3.3.2 การสกัดคุณลักษณะสำคัญ

หลังจากกระบวนการก่อนหน้าทำการสกัดคุณลักษณะสำคัญที่เหมาะสมในการจำแนกสระ ิ, ี, ื และ ุ โดยมีคุณลักษณะสำคัญที่ใช้ดังนี้

1. ระยะจากจุดเริ่มเขียนถึงจุดที่พิจารณา
2. ระยะจากจุดที่พิจารณาถึงจุดก่อนหน้า
3. ระยะจากจุดที่พิจารณาถึงจุดสุดท้าย
4. มุมของเส้นตรงที่เกิดจากจุดที่พิจารณาถึงจุดถัดไปหนึ่งจุดเทียบกับแกนนอน เพื่อใช้สังเกตการเปลี่ยนแปลงของทิศทางการเขียน
5. พิกัดของจุดทุกจุด

หลังจากการสกัดคุณลักษณะสำคัญดังกล่าวจะได้เวกเตอร์คุณลักษณะสำคัญของลำดับการสังเกตจะมีมิติเท่ากับ 6 ซึ่งน้อยกว่าเวกเตอร์คุณลักษณะสำคัญที่ได้จากหัวข้อ 3.1.2 ซึ่งมีขนาด 7 มิติ เพื่อให้แบบจำลองฮิดเดนมาร์คอฟทำงานได้ดีกับคุณลักษณะสำคัญใหม่ที่กล่าวมา เราต้องทดลองเพื่อหาจำนวนสถานะที่เหมาะสมซึ่งจะกล่าวถึงในบทต่อไป



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

ขั้นตอนการทดลอง ผลการทดลองและวิเคราะห์ผลการทดลอง

ในบทนี้จะกล่าวถึงการประเมินประสิทธิภาพของระบบการรู้จำตัวอักษรลายมือเขียนภาษาไทยแบบออนไลน์ในงานวิจัยนี้ โดยพิจารณาผลการทดลองการรู้จำด้วยวิธีที่เสนอ เทียบกับผลการทดลองการรู้จำด้วยแบบจำลองฮิดเดนมาร์คอฟพื้นฐาน รวมถึงการวิเคราะห์ผลที่ได้จากการทดลอง

4.1 เครื่องมือที่ใช้ในการทดสอบ

1. เครื่องคอมพิวเตอร์ส่วนบุคคล ใช้หน่วยประมวลผลกลางรุ่น Pentium IV ความเร็ว 2,400 เมกะเฮิร์ตซ์ และหน่วยความจำ 512 เมกะไบต์
2. กระดานอิเล็กทรอนิกส์ ซึ่งเป็นอุปกรณ์เชื่อมต่อในการรับข้อมูล
3. โปรแกรมการรู้จำตัวอักษรพัฒนาขึ้นบนโปรแกรม MATLAB เวอร์ชัน 6.5 ของบริษัท MathWorks ร่วมกับ Visual C++ เวอร์ชัน 6 ของบริษัทไมโครซอฟท์ ที่ทำงานภายใต้ระบบปฏิบัติการวินโดวส์ XP Service Pack 1 โดยใช้โปรแกรมแบบจำลองฮิดเดนมาร์คอฟจาก [21] และซอฟต์แวร์เวกเตอร์แมชชีนจาก [23] และ [24]

ข้อมูลที่ใช้ฝึกฝน และทดสอบเป็นตัวอักษรภาษาไทย 3 ระดับคือ ระดับบน กลาง และล่าง ดังแสดงในตารางที่ 1.1 ซึ่งเป็นข้อมูลชุดเดียวกับ [15] ซึ่งเก็บตัวอักษรผ่านทางกระดานอิเล็กทรอนิกส์ที่มีอัตราการสุ่มข้อมูล 100 จุดต่อวินาที และตั้งกระดานอิเล็กทรอนิกส์ให้ตรงไม่หมุนหรือเอียงจากผู้เขียน 2 กลุ่มที่อิสระต่อกัน จากนั้นคัดแยกตัวอักษรที่พิจารณาแล้วว่ามี การเขียนไม่ถูกต้องออก ตัวอักษรที่เขียนเสียนั้นคือตัวอักษรที่เขียนผิดเนื่องจากความเฉลหรือใช้กระดานอิเล็กทรอนิกส์ไม่ชำนาญ เช่น การเขียนตัวอักษรทับกัน เนื่องจากยังไม่ได้เก็บข้อมูลตัวเก่าก่อนเขียนตัวใหม่ การกดปากกาในกระดานอิเล็กทรอนิกส์เบาเกินไปทำให้บางเส้นนั้นหายไป เป็นต้น

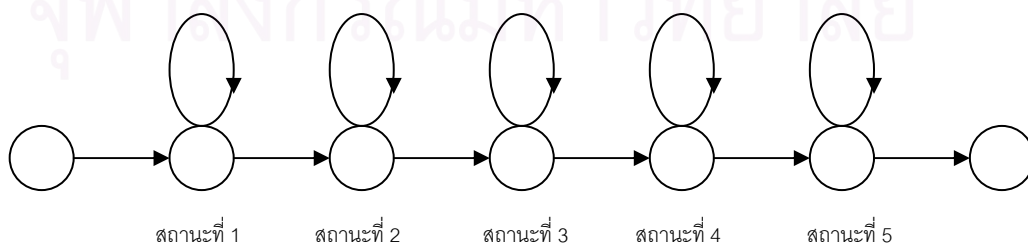
4.2 การจัดเก็บข้อมูล

ข้อมูลแบ่งออกเป็น 3 ชุด โดย 2 ชุดแรกมาจากกลุ่มผู้เขียนเดียวกัน ส่วนชุดสุดท้ายมาจากกลุ่มผู้เขียนอิสระ ดังนี้

1. ชุดที่ใช้ในการฝึกฝนระบบประกอบไปด้วย ตัวอักษรและสระระดับกลางจำนวน 14,557 ตัว สระและวรรณยุกต์ระดับบน 2,681 ตัว และ สระระดับล่างจำนวน 461 ตัว จากผู้เขียน 40 คน
2. ชุดที่ใช้ในการทดสอบได้จากกลุ่มผู้เขียนชุดเดิมประกอบไปด้วย ตัวอักษรและสระระดับกลางจำนวน 12,227 ตัว สระและวรรณยุกต์ระดับบน 2,493 ตัว และ สระระดับล่างจำนวน 461 ตัว
3. ชุดที่ใช้ในการทดสอบมาจากกลุ่มผู้เขียนอิสระอีกชุดหนึ่ง ประกอบไปด้วย ตัวอักษรและสระระดับกลางจำนวน 8946 ตัว สระและวรรณยุกต์ระดับบน 2079 ตัว และ สระระดับล่างจำนวน 378 ตัว จากผู้เขียน 65 คน

4.3 การทดสอบประสิทธิภาพการรู้จำด้วยแบบจำลองฮิดเดนมาร์คอฟ

จากการทดลองใน [15] พบว่าแบบจำลองฮิดเดนมาร์คอฟที่นำมาใช้กับภาษาไทยที่ผ่านกระบวนการก่อนหน้าในหัวข้อ 2.1 คือแบบจำลองฮิดเดนมาร์คอฟแบบต่อเนื่อง (CHMM) จำนวนสถานะ 5 สถานะ จำนวนส่วนผสมแบบเกาส์ในแต่ละสถานะ 5 ส่วนผสม และมีโครงสร้างแบบซ้ายไปขวาโดยไม่มีการข้ามสถานะดังรูปที่ 4.1 ได้อัตราการรู้จำเฉลี่ยดังตารางที่ 4.1 และผลการรู้จำโดยละเอียด หรือเมตริกซ์ความสับสน (Confusion Matrix) ดังตารางที่ 4.2 ถึงตารางที่ 4.7



รูปที่ 4.1 แบบจำลองฮิดเดนมาร์คอฟที่ใช้ในวิทยานิพนธ์

ตารางที่ 4.1 อัตราผู้จำเฉลี่ยของระบบผู้จำตัวอักษรภาษาไทยด้วยแบบจำลองฮิดเดนมาร์คอฟ

ชุดข้อมูล	อัตราการเรียนรู้ตัวอักษรระดับต่าง ๆ (ร้อยละ)		
	ระดับบน	ระดับกลาง	ระดับล่าง
ผู้เขียนที่ใช้ในการฝึกฝนระบบ	97.22	95.94	99.78
ผู้เขียนอิสระ	87.74	90.49	93.65

ตารางที่ 4.2 ผลการทดสอบระบบด้วยแบบจำลองฮิดเดนมาร์คอฟโดยใช้ข้อมูลจากสระและวรรณยุกต์ระดับบนของกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ

	อิ	อี	ือ	ือ	อ	ั	ิ	ึ	ึ	ึ	ึ	จำนวน	ร้อยละ
อิ	210	0	9	1	1	0	0	0	0	0	1	222	94.59
อี	7	204	3	12	1	0	0	0	2	0	0	229	89.08
ือ	1	0	213	11	0	0	0	0	0	0	0	225	94.67
ือ	0	2	4	226	0	0	0	0	0	0	0	232	97.41
อ	1	0	0	0	209	1	2	1	0	0	3	217	96.31
ั	0	0	0	0	0	228	0	0	1	0	0	229	99.56
ิ	0	0	0	0	0	0	226	0	0	0	0	226	100.00
ึ	0	0	0	0	0	0	2	222	0	0	0	224	99.11
ึ	0	0	0	0	0	0	0	0	219	2	0	221	99.10
ึ	0	0	0	0	0	0	0	0	0	230	0	230	100.00
ึ	0	0	0	0	0	0	0	0	0	1	237	238	99.58
	อิ	อี	ือ	ือ	อ	ั	ิ	ึ	ึ	ึ	ึ	2493	97.22

ตารางที่ 4.3 ผลการทดสอบระบบด้วยแบบจำลองฮิดเดนมาร์คอฟโดยใช้ข้อมูลจากสระและ
วรรณยุกต์ระดับบนของกลุ่มผู้เขียนอิสระ

	อิ	อี	อึ	อึ	อึ	อึ	อึ	อึ	อึ	อึ	อึ	จำนวน	ร้อยละ
อิ	150	1	22	4	5	1	4	0	0	0	2	189	79.37
อี	9	128	9	23	2	8	2	0	0	1	7	189	67.72
อึ	18	0	146	22	0	3	0	0	0	0	0	189	77.25
อึ	18	9	19	135	4	1	1	0	0	1	1	189	71.43
อึ	3	0	3	0	177	5	0	1	0	0	0	189	93.65
อึ	0	0	0	0	0	189	0	0	0	0	0	189	100.00
อึ	0	1	0	2	0	0	182	3	0	1	0	189	96.30
อึ	0	0	0	1	0	0	7	172	0	8	1	189	91.01
อึ	0	0	1	0	0	2	0	0	174	10	2	189	92.06
อึ	0	0	0	0	1	0	2	0	0	186	0	189	98.41
อึ	3	0	0	0	0	0	1	0	0	0	185	189	97.88
	อิ	อี	อึ	อึ	อึ	อึ	อึ	อึ	อึ	อึ	อึ	2079	87.73

ตารางที่ 4.4 ผลการทดสอบระบบด้วยแบบจำลองฮิดเดนมาร์คอฟโดยใช้ข้อมูลจากสระระดับล่าง
ของกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ

	อึ	อึ	จำนวน	ร้อยละ
อึ	232	0	232	100.00
อึ	1	228	229	99.56
	อึ	อึ	461	99.78

ตารางที่ 4.5 ผลการทดสอบระบบด้วยแบบจำลองฮิดเดนมาร์คอฟโดยใช้ข้อมูลจากสระระดับล่าง
ของกลุ่มผู้เขียนอิสระ

	ค	ค	จำนวน	ร้อยละ
ค	174	15	189	92.06
ค	9	180	189	95.24
	ค	ค	378	93.65



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.4 วิเคราะห์การทดสอบประสิทธิภาพของการรู้จำด้วยแบบจำลองฮิดเดนมาร์คอฟ

เนื่องจากการรู้จำด้วยแบบจำลองฮิดเดนมาร์คอฟคือการพยายามสร้างแบบจำลองของตัวอักษรแต่ละชนิด ดังนั้นแบบจำลองของตัวอักษรที่คล้ายกันย่อมมีลักษณะคล้ายกันด้วยการตัดสินผิดพลาดส่วนใหญ่จะเกิดขึ้นภายในกลุ่มของตัวอักษรที่คล้ายกัน โดยแบ่งลักษณะของกลุ่มที่คล้ายกันได้ดังนี้

- 4.4.1 กลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่ทิศทางการเขียนของหัว ได้แก่ ('ก' ผิดเป็น 'ถ' และ 'ภ') ('ร' ผิดเป็น 'ใ')
 - 4.4.2 กลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่ความกว้าง ได้แก่ ('ข' ผิดเป็น 'บ')
 - 4.4.3 กลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่รอยหยัก ได้แก่ ('ซ' ผิดเป็น 'ซ') ('ซ' ผิดเป็น 'ซ') ('ท' ผิดเป็น 'ท') ('ม' ผิดเป็น 'ม') ('ป' ผิดเป็น 'พ') ('ด' ผิดเป็น 'ด') ('ฎ' ผิดเป็น 'ฎ') ('ฎ' ผิดเป็น 'ฎ')
 - 4.4.4 กลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่วิธีการเขียนหาง ได้แก่ ('ฟ' ผิดเป็น 'ฟ') ('พ' ผิดเป็น 'พ') ('อ' ผิดเป็น 'ฮ') ('ล' ผิดเป็น 'ส') ('ท' ผิดเป็น 'ห') ('บ' ผิดเป็น 'ษ') ('ะ', 'เ', 'เ', 'เ')
 - 4.4.5 กลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันทั้งรอยหยักและวิธีการเขียนหาง ได้แก่ ('ป' ผิดเป็น 'พ') ('ษ' ผิดเป็น 'พ')
 - 4.4.6 กลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่สัดส่วนของหัว ได้แก่ ('น' ผิดเป็น 'ณ') ('ม' ผิดเป็น 'ณ') ('ร' ผิดเป็น 'ธ') ('ว' ผิดเป็น 'อ')
 - 4.4.7 กลุ่มตัวอักษรที่ตำแหน่งจุดต้น จุดปลาย และทิศทางการวนคล้ายกัน ได้แก่ ('ง' ผิดเป็น 'ฎ' และ 'ฎ') ('จ' ผิดเป็น 'ฉ') ('ล' ผิดเป็น 'ฉ') ('เ' ผิดเป็น 'ล' และ 'ะ')
- #### 4.5 กลุ่มของชนิดที่มีการตัดสินสับสนที่ได้จากผลการทดสอบ

กลุ่มของชนิดที่มีการตัดสินสับสนที่ได้จากการทดสอบด้วยแบบจำลองฮิดเดนมาร์คอฟแบ่งได้ 2 ชุด คือ ชุดที่ได้จากชุดทดสอบที่ได้จากกลุ่มผู้เขียนที่ใช้ในการฝึกฝน และชุดที่ได้จากชุดฝึกที่ได้มาจากกลุ่มผู้เขียนอิสระ

4.5.1 กลุ่มของชนิดที่มีการตัดสินสับสนที่ได้จากกลุ่มผู้เขียนที่ใช้ในการฝึกฝน

ด้วยระเบียบวิธีในหัวข้อ 3.2.3 และกำหนดให้ความผิดพลาดที่เกิดขึ้นจากการตัดสินผิด (Misclassification error) ของเฉพาะตัวที่ผิดมากที่สุดของแต่ละชนิดตัวอักษรต้องมีค่าเกินค่าคงที่ที่กำหนดไว้ ได้กลุ่มของชนิดที่มีการตัดสินสับสนของแต่ละชนิดตัวอักษรดังนี้

ตารางที่ 4.8 กลุ่มของชนิดที่มีการตัดสินสับสนที่ได้จากกลุ่มผู้เขียนที่ใช้ในการฝึกฝน

จำนวนชนิดภายในกลุ่ม	ตัวอักษรภายในกลุ่ม
1 ตัวอักษร (ไม่สับสน)	'ข' 'ค' 'ง' 'ฉ' 'ฃ' 'ญ' 'ฎ' 'ฐ' 'ฑ' 'ฒ' 'ณ' 'ด' 'ต' 'ป' 'ผ' 'ฝ' 'พ' 'ฟ' 'ภ' 'ม' 'ย' 'ร' 'ล' 'ว' 'ศ' 'ษ' 'ส' 'ห' 'ฬ' 'อ' 'ฮ' 'ะ' 'า' 'เ' 'ไ' 'ใ' 'เื'
2 ตัวอักษร (ไบนารีคลาส)	{'ก', 'ถ'}, {'ฌ', 'ฒ'}, {'จ', 'ฉ'}, {'ช', 'ซ'}, {'ข', 'ค'}, {'ฎ', 'ฐ'}, {'ด', 'ต'}, {'ท', 'ฑ'}, {'บ', 'ป'}, {'ภ', 'ม'}
มากกว่า 2 ตัวอักษร (มัลติคลาส)	{'พ', 'ฟ', 'ษ', 'ฬ'}

4.5.2 กลุ่มของชนิดที่มีการตัดสินลับสนที่ได้จากกลุ่มผู้เขียนอิสระ

เมื่อนำข้อมูลจากชุดทดสอบจากกลุ่มของผู้เขียนอิสระได้ผลดังตารางที่ 4.9

ตารางที่ 4.9 กลุ่มของชนิดที่มีการตัดสินลับสนที่ได้จากกลุ่มผู้เขียนอิสระ

จำนวนชนิดภายในกลุ่ม	ตัวอักษรภายในกลุ่ม
1 ตัวอักษร (ไม่ลับสน)	'ก' 'ข' 'ค' 'ง' 'ญ' 'ฐ' 'ด' 'ท' 'น' 'ป' 'พ' 'ภ' 'ย' 'ร' 'ล' 'ว' 'ศ' 'ห' 'า' 'เ' 'ใ' 'ไ' 'เ'
2 ตัวอักษร (โบราณวิคลาส)	{ 'ม', 'ม' }, { 'ณ', 'จ' }, { 'จ', 'ณ' }, { 'ช', 'ช' }, { 'ช', 'ช' }, { 'ท', 'ท' }, { 'ด', 'ด' }, { 'ก', 'ถ' }, { 'ร', 'ธ' }, { 'ข', 'บ' }, { 'ผ', 'ผ' }, { 'ป', 'พ' }, { 'ห', 'ม' }, { 'ว', 'อ' }, { 'อ', 'ฮ' }, { 'เ', 'ะ' }
มากกว่า 2 ตัวอักษร (มัลติคลาส)	กลุ่มที่มี 3 ตัวอักษร { 'ฎ', 'ฎ', 'ฐ' }, { 'ฎ', 'ง', 'ฎ' }, { 'ณ', 'ถ', 'น' }, { 'ผ', 'ผ', 'ย' }, { 'ช', 'ช', 'บ' } กลุ่มที่มี 5 ตัวอักษร { 'ณ', 'ญ', 'ณ', 'ณ', 'ภ' }, { 'ณ', 'ค', 'ณ', 'ณ', 'ด' }, { 'พ', 'พ', 'พ', 'ช', 'ส' } กลุ่มที่มี 6 ตัวอักษร { 'ส', 'ธ', 'ล', 'ศ', 'พ', 'ฮ' }

จากตารางที่ 4.8 และ ตารางที่ 4.9 พบว่าตัวอักษรที่มีรายละเอียดน้อยกว่ามักถูกตัดสินผิดเป็นตัวอักษรที่คล้ายกันแต่มีรายละเอียดน้อยกว่า เช่น ตัวอักษรที่ไม่มีรอยหยักถูกตัดสินผิดเป็นตัวที่มีรอยหยัก ยกตัวอย่างเช่น 'ท' ถูกตัดสินผิดเป็น 'ท' แต่ 'ท' ถูกตัดสินผิดเป็น 'ท' น้อยมาก

4.6 การทดสอบประสิทธิภาพการรู้จำด้วยระบบรู้จำที่นำเสนอ

จากข้อมูลตัวอักษรในตารางที่ 1.1 ซึ่งประกอบด้วยตัวอักษร 3 ระดับ ในหัวข้อ 4.5 นั้น ระบบรู้จำที่ใช้รู้จำตัวอักษรแต่ละระดับ คือ แบบจำลองฮิดเดนมาร์คอฟเหมือนกันทุกระดับอักษร แต่การรู้จำด้วยระบบที่นำเสนอแบ่งออกเป็น 2 ระบบย่อย กล่าวคือ ตัวอักษรระดับกลาง และล่าง นั้น ทำการรู้จำด้วยแบบจำลองฮิดเดนมาร์คอฟและซัพพอร์ตเวกเตอร์แมชชีนดังที่นำเสนอในหัวข้อ 3.1 และ 3.2 ตามลำดับ ส่วนตัวอักษรระดับบนรู้จำด้วยแบบจำลองฮิดเดนมาร์คอฟที่นำเสนอในหัวข้อ 3.3

4.6.1 การทดสอบประสิทธิภาพการรู้จำตัวอักษรระดับกลางและล่างด้วยระบบรู้จำที่นำเสนอ

เมื่อทดสอบประสิทธิภาพของระบบรู้จำลายมือเขียนภาษาไทยแบบออนไลน์ด้วยแบบจำลองฮิดเดนมาร์คอฟและซัพพอร์ตเวกเตอร์แมชชีนดังมีขั้นตอนการทำงานดังรูปที่ 3.1 โดยแบ่งการทดสอบออกเป็นสามการทดสอบโดยการทดสอบแรกเป็นการทดสอบตัวอักษรระดับกลาง โดยใช้กลุ่มชนิดที่มีการตัดสินใจตัดสินที่ได้จากกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบดังตารางที่ 4.8 และการทดสอบที่สองเป็นการทดสอบตัวอักษรระดับกลางโดยใช้กลุ่มชนิดที่มีการตัดสินใจตัดสินที่ได้จากกลุ่มผู้เขียนอิสระดังตารางที่ 4.9 และการทดสอบที่สามเป็นการทดสอบตัวอักษรระดับล่างซึ่งมีเพียง 2 ชนิดตัวอักษรจึงไม่จำเป็นต้องใช้กลุ่มชนิดที่มีการตัดสินใจตัดสิน

4.6.1.1 ทดสอบตัวอักษรระดับกลางด้วยระบบรู้จำที่เสนอโดยใช้กลุ่มชนิดที่มีการตัดสินใจตัดสินที่ได้จากกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ

จำนวนชนิดตัวอักษรของกลุ่มนี้มีจำนวนมากจึงจำเป็นต้องนำเทคนิคการทำมัลติคลาสมาใช้ ซึ่งในที่นี้คือ ระเบียบวิธี Max Win โดยจะพิจารณากลุ่มมัลติคลาสเป็นกลุ่มย่อยที่มีความสัมพันธ์กันโดยใช้ข้อมูลกลุ่มที่มักมีการตัดสินใจตัดสินจากตารางที่ 4.8 ซึ่งมาจากกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบโดยได้อัตราจำเฉลี่ยดังตารางที่ 4.10 และผลการทดสอบดังตารางที่ 4.11 และตารางที่ 4.12

ตารางที่ 4.10 อัตราผู้จำเฉลี่ยของระบบรู้จำตัวอักษรภาษาไทยระดับกลางด้วยระบบรู้จำที่นำเสนอ โดยใช้กลุ่มชนิดที่มีการตัดสินใจตัดสินที่ได้จากกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ

ชุดข้อมูล	อัตราการรู้จำตัวอักษรระดับกลาง (ร้อยละ)
ผู้เขียนที่ใช้ในการฝึกฝนระบบ	96.86
ผู้เขียนอิสระ	92.20



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.6.1.2 ทดสอบตัวอักษรระดับกลางด้วยระบบรู้จำที่เสนอโดยใช้กลุ่มชนิดที่มีการตัดสินใจ สับสนที่ได้จากกลุ่มผู้เขียนอิสระ

เช่นเดียวกันกับการทดสอบในหัวข้อ 4.6.1.1 แต่จะพิจารณากลุ่มมัลติคลาสเป็นกลุ่มย่อยที่มีความสับสนกันโดยใช้ข้อมูลกลุ่มที่มักที่การตัดสินใจสับสนจากตารางที่ 4.9 ซึ่งมาจากกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบโดยใช้อัตราจำเฉลี่ยดังตารางที่ 4.13 และผลการทดสอบดังตารางที่ 4.14 และ ตารางที่ 4.15

ตารางที่ 4.13 อัตราจำเฉลี่ยของระบบรู้จำตัวอักษรภาษาไทยระดับกลางด้วยระบบรู้จำที่นำเสนอ
โดยใช้กลุ่มชนิดที่มีการตัดสินใจสับสนที่ได้จากกลุ่มผู้เขียนอิสระ

ชุดข้อมูล	อัตราการรู้จำตัวอักษรระดับกลาง (ร้อยละ)
ผู้เขียนที่ใช้ในการฝึกฝนระบบ	96.65
ผู้เขียนอิสระ	92.94

4.6.1.3 ทดสอบตัวอักษรระดับล่างด้วยระบบรู้จำที่เสนอ

เนื่องจากจำนวนชนิดของตัวอักษรในกลุ่มนี้มีเพียง 2 ชนิด ดังนั้นเราไม่จำเป็นต้องใช้เทคนิคการทำมัลติคลาส ดังนั้นกลุ่มตัวอักษรระดับล่างจึงมีความเหมาะสมอย่างมากในการประยุกต์ใช้กับตัวจำแนกประเภทแบบไบนารี เช่น ซัพพอร์ตเวกเตอร์แมชชีน โดยอัตราการรู้จำที่ได้จากระบบที่นำเสนอมีมากกว่าระบบรู้จำที่ใช้แบบจำลองฮิดเดนมาร์คอฟพื้นฐานในทุกกลุ่มทั้งกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบดังตารางที่ 4.16 และกลุ่มผู้เขียนอิสระดังตารางที่ 4.17

ตารางที่ 4.16 ผลการทดสอบระบบรู้จำที่เสนอโดยใช้ข้อมูลจากสระระดับล่างของกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ

	คุ	คฺ	จำนวน	ร้อยละ
คุ	232	0	232	100
คฺ	0	229	229	100
	คุ	คฺ	461	100

ตารางที่ 4.17 ผลการทดสอบระบบรู้จำที่เสนอโดยใช้ข้อมูลจากสระระดับล่างของกลุ่มผู้เขียนอิสระ

	คุ	คฺ	จำนวน	ร้อยละ
คุ	185	4	189	97.88
คฺ	4	185	189	97.88
	คุ	คฺ	378	97.88

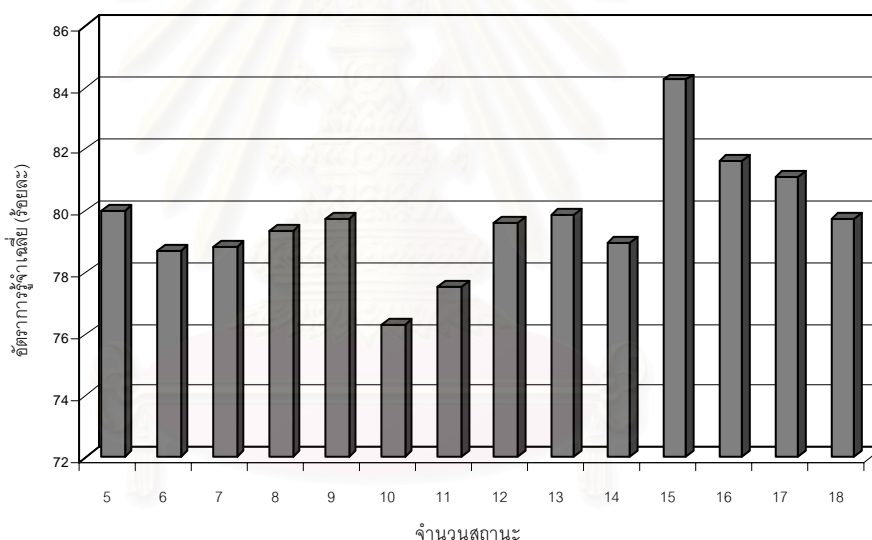
4.6.2 การทดสอบประสิทธิภาพการรู้จำตัวอักษรระดับบนด้วยระบบรู้จำที่นำเสนอ

การรู้จำตัวอักษรระดับบนด้วยวิธีเดียวกับที่เคยใช้กับตัวอักษรระดับกลาง และระดับล่างให้อัตราการรู้จำไม่ดีขึ้น เนื่องจากกลุ่มสระ $\bar{\cdot}$, $\bar{\cdot}$, $\bar{\cdot}$ และ $\bar{\cdot}$ มีลักษณะใกล้เคียงกันและรูปแบบการเขียนที่หลากหลาย ดังนั้นการรู้จำอักษรระดับบนจึงต้องใช้ระบบรู้จำด้วยแบบจำลองฮิดเดนมาร์คอฟ 2 ชั้นตอน ชั้นตอนแรกจะใช้กระบวนการก่อนหน้า คุณลักษณะสำคัญ และแบบจำลอง

ฮีดเดนมาร์คอฟ ที่ใช้กับตัวอักษรระดับกลางและล่างดังหัวข้อ 3.1 เพื่อแยกกลุ่มสระ \bar{a} , \bar{e} , \bar{i} และ \bar{u} ออกจากกลุ่ม ขึ้นตอนที่สองจะใช้กระบวนการก่อนหน้า คุณลักษณะสำคัญที่เสนอในหัวข้อที่ 3.3 และแบบจำลองฮีดเดนมาร์คอฟที่มีจำนวนสถานะมากขึ้น ซึ่งจำนวนสถานะที่เหมาะสมได้จากการทดลองในหัวข้อ 4.6.2.1 เพื่อรู้จำในกลุ่มสระ \bar{a} , \bar{e} , \bar{i} และ \bar{u} ในส่วนวรรณยุกต์เอกจะมีการคัดแยกออกจากระบบรู้จำตั้งแต่ตอนกระบวนการก่อนหน้าซึ่งได้กล่าวไปแล้ว

4.6.2.1 จำนวนสถานะที่เหมาะสมเพื่อการรู้จำกลุ่มสระ \bar{a} , \bar{e} , \bar{i} และ \bar{u}

เนื่องจากกระบวนการก่อนหน้าที่ใช้เปลี่ยนไป ทำให้ความยาวของลำดับสังเกตเพิ่มมากขึ้น ดังนั้นเพื่อความเหมาะสมจำนวนสถานะในแบบจำลองฮีดเดนมาร์คอฟต้องเพิ่มมากขึ้นตามโดยการทดสอบกลุ่มสระ \bar{a} , \bar{e} , \bar{i} และ \bar{u} ของกลุ่มทดสอบได้อัตราการเรียนรู้จำดังรูปที่ 4.2



รูปที่ 4.2 อัตราการเรียนรู้จำเฉลี่ยของกลุ่มสระ \bar{a} , \bar{e} , \bar{i} และ \bar{u} ของกลุ่มผู้เขียนอิสระเมื่อใช้จำนวนสถานะต่าง ๆ

จากรูปที่ 4.2 พบว่าที่จำนวน 15 สถานะให้อัตราการเรียนรู้จำสูงที่สุด ดังนั้นจำนวนสถานะของแบบจำลองฮีดเดนมาร์คอฟที่ใช้ในการรู้จำตัวอักษรระดับบนจึงกำหนดให้เป็น 15 สถานะ

4.6.2.2 อัตราการเรียนรู้จำตัวอักษรระดับบนด้วยระบบรู้จำที่นำเสนอ

จากหัวข้อ 4.6.2.1 พบว่าเมื่อจำนวนสถานะเท่ากับ 15 สถานะจะเหมาะสมที่สุดที่จะใช้กับกับระบบรู้จำดังได้ผลทดสอบดังตาราง

ตารางที่ 4.18 ผลการทดสอบระบบรู้จำที่เสนอโดยใช้ข้อมูลจากสระและวรรณยุกต์ระดับบนของกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ

	อิ	อ	อุ	อู	อื	อฺ	อึ	อึ	อึ	อึ	อึ	จำนวน	ร้อยละ
อิ	218	0	0	2	1	0	0	0	0	0	1	222	98.20
อ	0	214	0	12	1	0	0	0	2	0	0	229	93.45
อุ	0	1	223	1	0	0	0	0	0	0	0	225	99.11
อู	0	1	3	228	0	0	0	0	0	0	0	232	98.28
อื	1	0	0	0	209	1	2	1	0	0	3	217	96.31
อฺ	0	0	0	0	0	228	0	0	1	0	0	229	99.56
อึ	0	0	0	0	0	0	226	0	0	0	0	226	100.00
อึ	0	0	0	0	0	0	2	222	0	0	0	224	99.11
อึ	0	0	0	0	0	0	0	0	219	2	0	221	99.10
อึ	0	0	0	0	0	0	0	0	0	230	0	230	100.00
อึ	0	0	0	0	0	0	0	0	0	1	237	238	99.58
	อิ	อ	อุ	อู	อื	อฺ	อึ	อึ	อึ	อึ	อึ	2493	98.43

ตารางที่ 4.19 ผลการทดสอบระบบรู้จำที่เสนอโดยใช้ข้อมูลจากสระและวรรณยุกต์ระดับบนของกลุ่มผู้เขียนอิสระ

	อิ	อี	อึ	อื	อํ	อฺ	อิ	อี	อึ	อื	อํ	จำนวน	ร้อยละ
อิ	158	1	2	16	5	1	4	0	0	0	2	189	83.60
อี	3	154	0	12	2	8	2	0	0	1	7	189	81.48
อึ	1	2	162	21	0	3	0	0	0	0	0	189	85.71
อื	1	8	28	144	4	1	1	0	0	1	1	189	76.19
อํ	0	0	5	1	177	5	0	1	0	0	0	189	93.65
อฺ	0	0	0	0	0	189	0	0	0	0	0	189	100.00
อิ	0	3	0	0	0	0	182	3	0	1	0	189	96.30
อี	0	1	0	0	0	0	7	172	0	8	1	189	91.01
อึ	0	0	0	1	0	2	0	0	174	10	2	189	92.06
อื	0	0	0	0	1	0	2	0	0	186	0	189	98.41
อํ	1	2	0	0	0	0	1	0	0	0	185	189	97.88
	อิ	อี	อึ	อื	อํ	อฺ	อิ	อี	อึ	อื	อํ	2079	90.57

4.7 วิเคราะห์การทดสอบประสิทธิภาพของการรู้จำด้วยระบบรู้จำที่นำเสนอเมื่อเปรียบเทียบกับระบบรู้จำที่ใช้แบบจำลองฮิดเดนมาร์คอฟพื้นฐาน

จากการทดสอบจากหัวข้อที่ผ่านมาเมื่อนำผลการทดสอบของระบบรู้จำที่นำเสนอของตัวอักษรระดับต่าง ๆ มาเปรียบเทียบกับผลการทดสอบของระบบรู้จำที่ใช้แบบจำลองฮิดเดนมาร์คอฟพื้นฐานของทั้งกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบและกลุ่มผู้เขียนอิสระในลักษณะแผนภูมิรูปแท่งดังรูปที่ 4.3 ถึง รูปที่ 4.8 สามารถสรุปได้ดังตาราง

ตารางที่ 4.20 ผลการเปรียบเทียบอัตราผู้จำตัวอักษรระดับต่าง ๆ ระหว่างระบบผู้จำที่เสนอกับระบบผู้จำด้วยแบบจำลองฮิดเดนมาร์คอฟพื้นฐาน

ชนิดระบบผู้จำ	ชุดข้อมูล	อัตราการรู้จำเฉลี่ยตัวอักษรระดับต่าง ๆ (ร้อยละ)		
		ระดับบน	ระดับกลาง	ระดับล่าง
ระบบผู้จำที่นำเสนอ	ผู้เขียนที่ใช้ในการฝึกฝนระบบ	98.43	96.86	100
	ผู้เขียนอิสระ	90.57	92.20	97.88
ระบบผู้จำด้วยแบบจำลองฮิดเดนมาร์คอฟพื้นฐาน	ผู้เขียนที่ใช้ในการฝึกฝนระบบ	97.22	95.94	99.78
	ผู้เขียนอิสระ	87.74	90.49	93.65

โดยในที่นี้ระบบผู้จำที่นำเสนอสำหรับตัวอักษรระดับกลางจะหมายถึงระบบที่ใช้กลุ่มที่มีการตัดสินใจตัดสินจากกลุ่มผู้เขียนที่ใช้ในการฝึกฝน ในทุกกลุ่มการทดสอบระบบผู้จำที่นำเสนอให้อัตราการรู้จำเฉลี่ยสูงกว่าระบบผู้จำด้วยแบบจำลองฮิดเดนมาร์คอฟพื้นฐาน โดยระบบผู้จำที่เสนอมีร้อยละของอัตราการรู้จำเฉลี่ยเพิ่มขึ้นจากระบบผู้จำด้วยแบบจำลองฮิดเดนมาร์คอฟพื้นฐานดังแสดงดังตารางที่ 4.21

ตารางที่ 4.21 ร้อยละของอัตราการรู้จำเฉลี่ยของระบบผู้จำที่เสนอที่เพิ่มขึ้นจากระบบผู้จำด้วยแบบจำลองฮิดเดนมาร์คอฟพื้นฐาน

ชุดข้อมูล	อัตราการรู้จำเฉลี่ยตัวอักษรระดับต่าง ๆ ที่เพิ่มขึ้น (ร้อยละ)		
	ระดับบน	ระดับกลาง	ระดับล่าง
ผู้เขียนที่ใช้ในการฝึกฝนระบบ	1.21	0.92	0.22
ผู้เขียนอิสระ	2.83	1.71	4.23

เมื่อพิจารณาการตัดสินใจผิดพลาดส่วนใหญ่ที่เกิดขึ้นภายในกลุ่มของตัวอักษรที่คล้ายกันของระบบผู้จำด้วยแบบจำลองฮิดเดนมาร์คอฟพื้นฐานในกลุ่มต่าง ๆ ดังหัวข้อ 4.4 พบว่าระบบผู้จำที่

นำเสนอส่วนใหญ่่อัตรการรู้จำจะสูงขึ้นดังตารางที่ 4.22 ถึง ตารางที่ 4.28 โดยกำหนดให้ ระบบที่ 1 หมายถึง ระบบรู้จำที่ใช้แบบจำลองฮิดเดนมาร์คอฟพื้นฐาน ระบบที่ 2 หมายถึงระบบรู้จำที่นำเสนอ และหากเป็นการรู้จำตัวอักษรระดับกลางจะหมายถึงระบบใช้กลุ่มที่มีการตัดสินสับสนจากกลุ่มผู้เขียนที่ใช้ในการฝึกฝนจากตารางที่ 4.8 ชุดที่ 1 หมายถึงข้อมูลจากกลุ่มผู้เขียนที่ใช้ในการฝึกฝน ชุดที่ 2 หมายถึงข้อมูลจากกลุ่มผู้เขียนอิสระ และผลต่าง หมายถึงร้อยละของระบบที่ 2 ลบร้อยละของระบบที่ 1

4.7.1 กลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่ทิศทางกรเขียนของหัว ได้แก่ ('ก' ผิดเป็น 'ถ' และ 'ภ') ('ร' ผิดเป็น 'โร')

ตารางที่ 4.22 ร้อยละของอัตรการรู้จำของกลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่ทิศทางกรเขียนของหัว

ชนิดระบบรู้จำ	ระบบที่ 1		ระบบที่ 2		ผลต่าง	
	ชุดที่ 1	ชุดที่ 2	ชุดที่ 1	ชุดที่ 2	ชุดที่ 1	ชุดที่ 2
ก	94.07	82.26	96.19	87.10	2.12	4.84
ถ	97.32	97.31	97.32	96.24	0.00	-1.07
ภ	92.65	97.31	92.65	97.31	0.00	0.00
ร	98.07	91.94	98.07	91.94	0.00	0.00
โร	99.15	96.30	99.15	96.30	0.00	0.00
				รวม	2.12	3.77

จะเห็นได้ว่าระบบรู้จำที่นำเสนอสามารถเพิ่มอัตรการรู้จำของกลุ่มนี้โดยรวมได้ซึ่ง 'ก' เป็นตัวอักษรที่มีอัตรการรู้จำสูงขึ้นโดยไม่ส่งผลกระทบต่อ 'ภ' แต่ในชุดข้อมูลของกลุ่มผู้เขียนอิสระ 'ถ' อัตรการรู้จำลดลงไม่มากเมื่อเทียบกับอัตรการรู้จำของ 'ก' ที่เพิ่มขึ้น

4.7.2 กลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่ความกว้าง ได้แก่ ('ข' ผิดเป็น 'บ')

ตารางที่ 4.23 ร้อยละของอัตราการรู้จำของกลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่ความกว้าง

ชนิดระบบรู้จำ	ระบบที่ 1		ระบบที่ 2		ผลต่าง	
	ชุดที่ 1	ชุดที่ 2	ชุดที่ 1	ชุดที่ 2	ชุดที่ 1	ชุดที่ 2
ข	91.39	75.81	91.39	75.81	0	0
บ	84.53	70.43	93.21	86.56	8.68	16.13
				รวม	8.68	16.13

จะเห็นได้ว่าระบบรู้จำที่นำเสนอสามารถเพิ่มอัตราการรู้จำของกลุ่มนี้ได้เห็นได้จากอัตรา
การรู้จำ 'บ' ซึ่งมีการตัดสินใจผิดเป็น 'ข' มีค่าเพิ่มขึ้นมากแต่อัตราการรู้จำ 'ข' ไม่ลดลงเลย ดังนั้น
ระบบรู้จำที่นำเสนอสามารถแก้ไขปัญหาการรู้จำผิดของกลุ่มนี้ได้ดี

4.7.3 กลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่รอยหยัก ได้แก่ ('ซ' ผิดเป็น 'ซ') ('ซ' ผิดเป็น 'ซ') ('ท' ผิดเป็น 'ท') ('ม' ผิดเป็น 'ม') ('ป' ผิดเป็น 'พ') ('ด' ผิดเป็น 'ต') ('ฎ' ผิดเป็น 'ฎ') ('ฎ' ผิดเป็น 'ฎ')

ตารางที่ 4.24 ร้อยละของอัตราการเรียนรู้จำของกลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่รอยหยัก

ชนิดระบบรู้จำ	ระบบที่ 1		ระบบที่ 2		ผลต่าง	
	ชุดที่ 1	ชุดที่ 2	ชุดที่ 1	ชุดที่ 2	ชุดที่ 1	ชุดที่ 2
ซ	92	69.89	84.44	69.36	-7.56	-0.53
ซ	89.79	87.63	96.17	90.86	6.38	3.23
ท	93.09	68.28	94.55	74.73	1.46	6.45
ท	97.91	97.31	97.91	96.77	0	-0.54
ม	87.98	64.52	94.19	77.96	6.21	13.44
ม	93.52	90.32	94.33	93.01	0.81	2.69
ป	94.9	87.63	94.9	87.63	0	0
พ	93.73	84.41	96.86	89.79	3.13	5.38
ด	88.19	73.12	90.04	80.65	1.85	7.53
ต	97.41	94.62	97.41	94.62	0	0
ฎ	80.71	74.73	92.13	86.56	11.42	11.83
ฎ	95.08	86.56	95.08	86.56	0	0
				รวม	23.7	49.48

จะเห็นได้ว่าระบบรู้จำที่นำเสนอสามารถเพิ่มอัตราการเรียนรู้จำของกลุ่มนี้โดยรวมได้ซึ่ง 'ท', 'ม', 'พ', 'ด' และ 'ฎ' เป็นกลุ่มตัวอักษรที่มีอัตราการเรียนรู้จำสูงขึ้นโดยไม่ส่งผลกระทบต่อตัวอักษรในกลุ่มที่คล้ายกันยกเว้น 'ท' ที่มีการตัดสินใจผิดเป็น 'ท' เพิ่มขึ้นแต่น้อยกว่าเมื่อเทียบกับอัตราการเรียนรู้จำของ 'ท' ที่เพิ่มขึ้น แต่ในชุดข้อมูลของทั้ง 2 กลุ่มผู้เขียน 'ซ' มีอัตราการเรียนรู้จำลดลงเนื่องจากการตัดสินใจผิดเป็น 'ซ' เพิ่มขึ้น ในกลุ่มนี้ระบบรู้จำที่ใช้แบบจำลองฮิดเดนมาร์คอฟมีการตัดสินใจตัวอักษรที่ไม่มีรอยหยักเป็นตัวอักษรที่มีรอยหยัก ดังนั้นโดยรวมระบบรู้จำที่นำเสนอสามารถแก้ไขปัญหาดังกล่าวได้

4.7.4 กลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่วิธีการเขียนหาง ได้แก่ ('ฟ' ผิดเป็น 'ฟ') ('ฟ' ผิดเป็น 'ฟ') ('อ' ผิดเป็น 'ฮ') ('ล' ผิดเป็น 'ส') ('ท' ผิดเป็น 'ห') ('บ' ผิดเป็น 'ษ') ('ะ', 'เ', 'เ', 'เ')

ตารางที่ 4.25 ร้อยละของอัตราการรู้จำของกลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่วิธีการเขียนหาง

ชนิดระบบรู้จำ	ระบบที่ 1		ระบบที่ 2		ผลต่าง	
	ชุดที่ 1	ชุดที่ 2	ชุดที่ 1	ชุดที่ 2	ชุดที่ 1	ชุดที่ 2
พ	89.27	91.40	93.08	95.16	3.81	3.76
ฟ	93.73	84.41	96.86	89.79	3.13	5.38
ฟ	97.70	97.31	97.70	97.31	0.00	0.00
อ	97.08	94.62	97.08	94.62	0.00	0.00
ฮ	97.96	94.62	97.96	94.62	0.00	0.00
ล	97.74	89.78	97.74	89.79	0.00	0.00
ส	99.26	97.85	99.26	97.85	0.00	0.00
ท	93.09	68.28	94.55	74.73	1.46	6.45
ห	99.63	99.46	99.63	99.46	0.00	0.00
บ	84.53	70.43	93.21	86.56	8.68	16.13
ษ	95.38	96.77	96.15	96.24	0.77	-0.53
อึ	94.59	79.37	96.40	83.60	1.80	4.23
อึ	89.08	67.72	93.89	81.48	4.80	13.76
อึ	94.67	77.25	97.33	85.71	2.67	8.47
อึ	97.41	71.43	98.71	76.19	1.29	4.76
				รวม	28.41	62.41

จะเห็นได้ว่าระบบรู้จำที่นำเสนอสามารถเพิ่มอัตราการรู้จำของกลุ่มนี้โดยรวมได้ซึ่ง 'พ', 'ฟ', 'ท', 'บ', 'ะ', 'เ', 'เ' และ 'เ' เป็นกลุ่มตัวอักษรที่มีอัตราการรู้จำสูงขึ้นโดยไม่ส่งผลกระทบต่อตัวอักษรในกลุ่มที่คล้ายกันยกเว้น 'ษ' ที่มีการตัดสินใจผิดเป็น 'บ' เพิ่มขึ้นแต่น้อยกว่าเมื่อเทียบกับอัตราการรู้จำของ 'บ' ที่เพิ่มขึ้น ดังนั้นโดยรวมระบบรู้จำที่นำเสนอสามารถแก้ไขปัญหาดังกล่าวได้

4.7.5 กลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันทั้งรอยหยักและวิธีการเขียนหาง ได้แก่ ('ป' ผิดเป็น 'พ') ('ช' ผิดเป็น 'ฬ')

ตารางที่ 4.26 ร้อยละของอัตราการเรียนรู้จำของกลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันทั้งรอยหยักและวิธีการเขียนหาง

ชนิดระบบรู้จำ	ระบบที่ 1		ระบบที่ 2		ผลต่าง	
	ชุดที่ 1	ชุดที่ 2	ชุดที่ 1	ชุดที่ 2	ชุดที่ 1	ชุดที่ 2
ช	95.38	96.77	96.15	96.24	0.77	-0.53
ป	94.9	87.63	94.9	87.63	0	0
ฬ	97.7	97.31	97.7	97.31	0	0
				รวม	0.77	-0.53

จะเห็นได้ว่าระบบรู้จำที่นำเสนอสามารถเพิ่มอัตราการเรียนรู้จำของกลุ่มนี้ที่ใช้ข้อมูลจากกลุ่มผู้เขียนที่ใช้ในการฝึกฝนได้ แต่ข้อมูลของกลุ่มผู้เขียนอิสระมีอัตราการเรียนรู้จำลดลงเล็กน้อยทั้งนี้เนื่องจาก 'ช' มีการตัดสินสืบสนอยู่หลายกลุ่ม ดังนั้นโดยรวมระบบรู้จำที่นำเสนอไม่สามารถแก้ไขปัญหากลุ่มนี้ได้มากนักโดยเฉพาะในส่วนกลุ่มผู้เขียนอิสระแต่ก็ไม่ส่งผลกระทบต่ออัตราการเรียนรู้จำในกลุ่มตัวอักษรที่คล้ายกันกลุ่มอื่นที่มีตัวอักษรซ้ำกับกลุ่มนี้มากนัก

4.7.6 กลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่สัดส่วนของหัว ได้แก่ ('น' ผิดเป็น 'ณ') ('ม' ผิดเป็น 'ฒ') ('ร' ผิดเป็น 'ธ') ('ว' ผิดเป็น 'อ')

ตารางที่ 4.27 ร้อยละของอัตราการรู้จำของกลุ่มตัวอักษรที่คล้ายกันแต่ต่างกันที่สัดส่วนของหัว

ชนิดระบบรู้จำ	ระบบที่ 1		ระบบที่ 2		ผลต่าง	
	ชุดที่ 1	ชุดที่ 2	ชุดที่ 1	ชุดที่ 2	ชุดที่ 1	ชุดที่ 2
น	97.36	93.01	97.36	93.01	0.00	0.00
ณ	97.35	97.31	97.35	97.31	0.00	0.00
ม	93.52	90.32	94.33	93.01	0.81	2.69
ฒ	98.44	98.39	98.44	98.39	0.00	0.00
ร	98.07	91.94	98.07	91.94	0.00	0.00
ธ	97.77	93.55	97.77	93.55	0.00	0.00
ว	97.74	87.63	97.74	87.63	0.00	0.00
อ	97.08	94.62	97.08	94.62	0.00	0.00
				รวม	0.81	2.69

จะเห็นได้ว่าระบบรู้จำที่นำเสนอสามารถเพิ่มอัตราการรู้จำของกลุ่มนี้โดยรวมได้ซึ่ง 'ม' เป็นตัวอักษรที่มีอัตราการรู้จำสูงขึ้นโดยไม่ส่งผลกระทบต่อตัวอักษรในกลุ่มที่คล้ายกันทั้งข้อมูลจากกลุ่มผู้เขียนที่ใช้ในการฝึกฝนและกลุ่มผู้เขียนอิสระ ดังนั้นโดยรวมระบบรู้จำที่นำเสนอสามารถแก้ไขปัญหาดังกล่าวได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

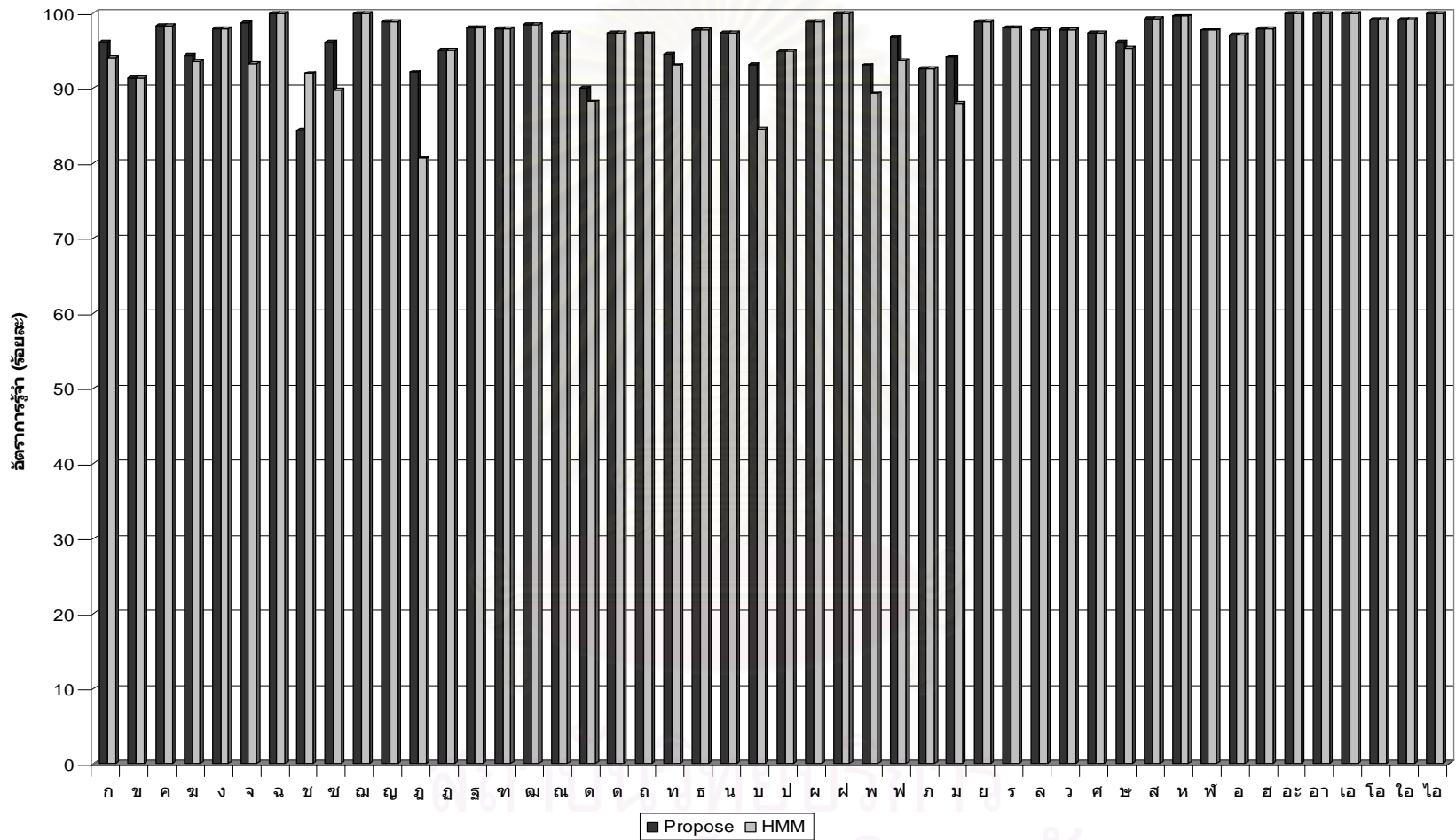
4.7.7 กลุ่มตัวอักษรที่ตำแหน่งจุดต้น จุดปลาย และทิศทางการวนคล้ายกัน ได้แก่ ('ง' ผิดเป็น 'ฏ' และ 'ฎ') ('จ' ผิดเป็น 'ฉ') ('ล' ผิดเป็น 'ฉ') ('เ' ผิดเป็น 'ล' และ 'ะ')

ตารางที่ 4.28 ร้อยละของอัตราการรู้จำของกลุ่มตัวอักษรที่ตำแหน่งจุดต้น จุดปลาย และทิศทางการวนคล้ายกัน

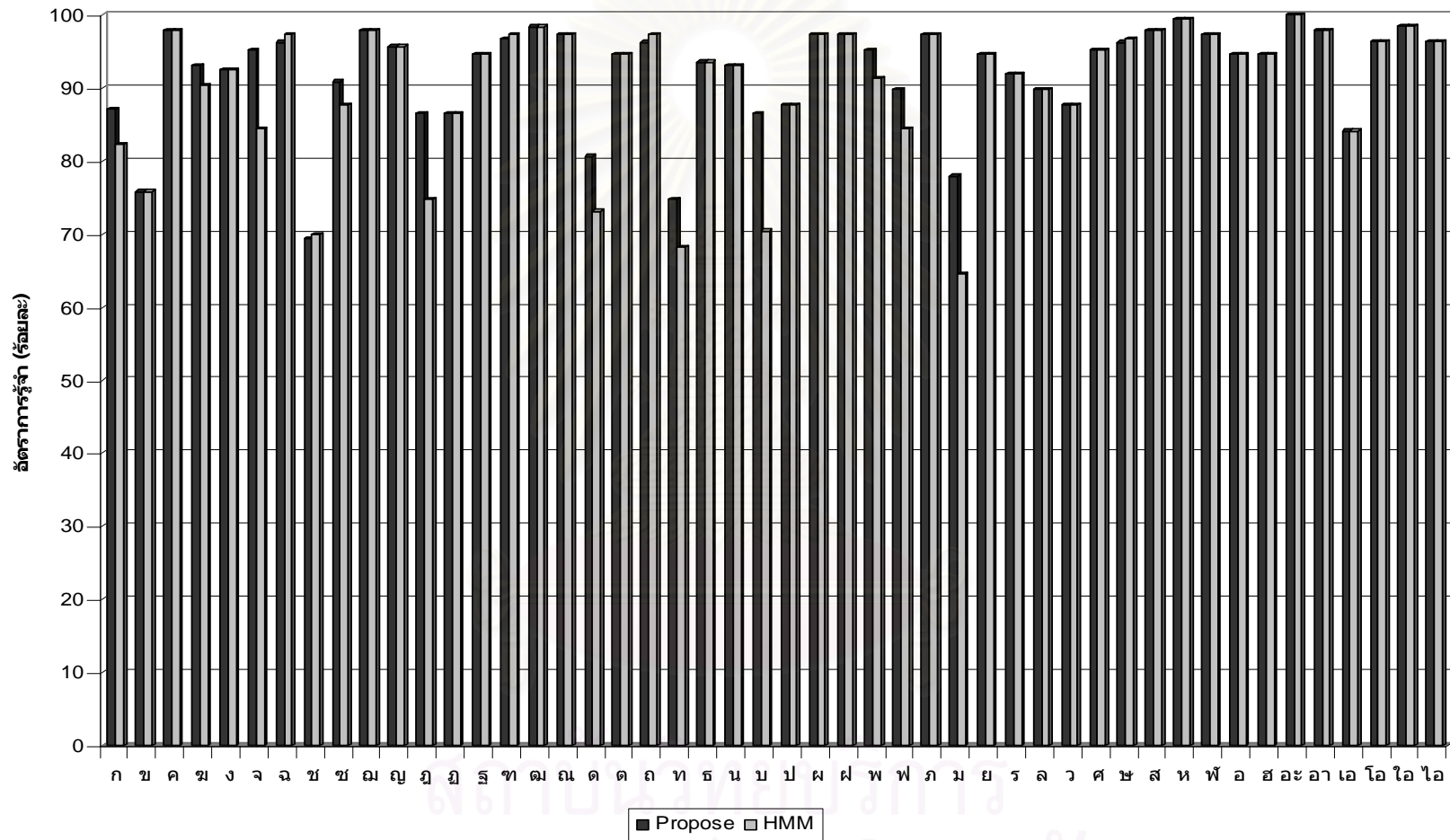
ชนิดระบบรู้จำ	ระบบที่ 1		ระบบที่ 2		ผลต่าง	
	ชุดที่ 1	ชุดที่ 2	ชุดที่ 1	ชุดที่ 2	ชุดที่ 1	ชุดที่ 2
ง	97.95	92.47	97.95	92.47	0.00	0.00
ฎ	80.71	74.73	92.13	86.56	11.42	11.83
ฏ	95.08	86.56	95.08	86.56	0.00	0.00
จ	93.28	84.41	98.74	95.16	5.46	10.75
ฉ	100.00	97.31	100.00	96.24	0.00	-1.07
ล	97.74	89.78	97.74	89.79	0.00	0.00
เ	100.00	84.13	100.00	84.13	0.00	0.00
อะ	100.00	100.00	100.00	100.00	0.00	0.00
				รวม	16.88	21.51

จะเห็นได้ว่าระบบรู้จำที่น่าเสนอสามารถเพิ่มอัตราการรู้จำของกลุ่มนี้ได้ซึ่ง 'ฎ' เป็นกลุ่มตัวอักษรที่มีอัตราการรู้จำสูงขึ้นโดยไม่ส่งผลกระทบต่อตัวอักษรในกลุ่มที่คล้ายกัน ส่วน 'ฉ' ที่มีการตัดสินผิดเป็น 'จ' เพิ่มขึ้นแต่น้อยกว่าเมื่อเทียบกับอัตราการรู้จำของ 'จ' ที่เพิ่มขึ้น ดังนั้นโดยรวมระบบรู้จำที่น่าเสนอสามารถแก้ไขปัญหาดังกล่าวได้

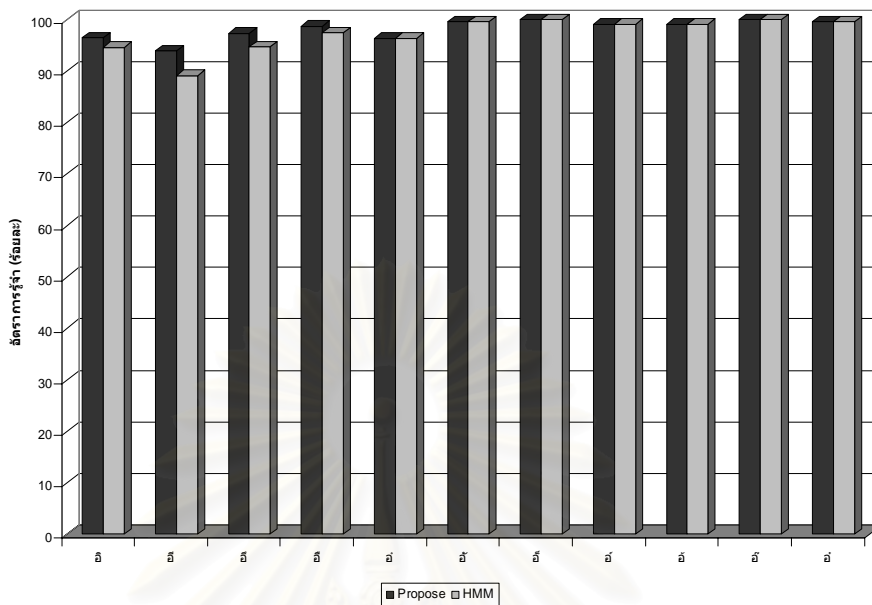
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



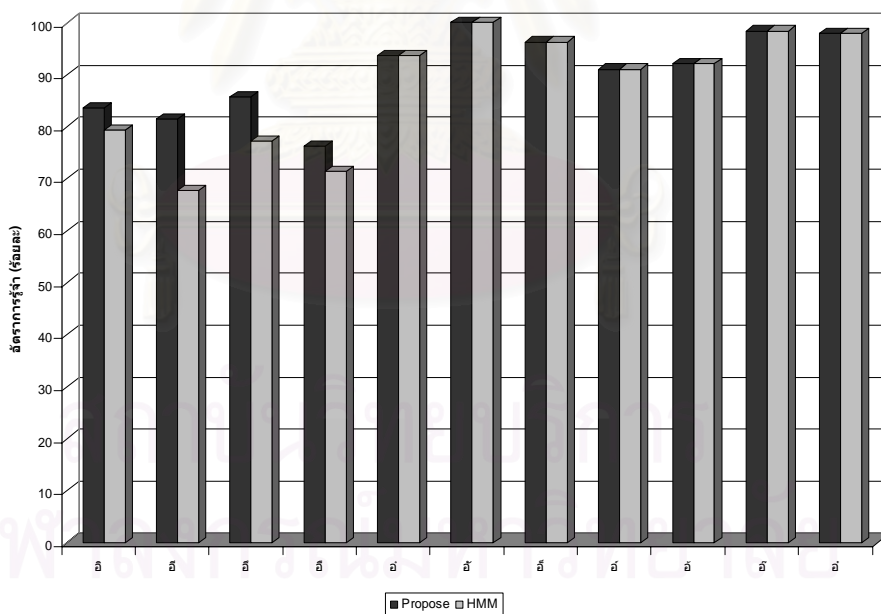
รูปที่ 4.3 แผนภูมิเปรียบเทียบอัตราความรู้จำตัวอักษรระดับกลางของข้อมูลตัวอักษรจากกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ



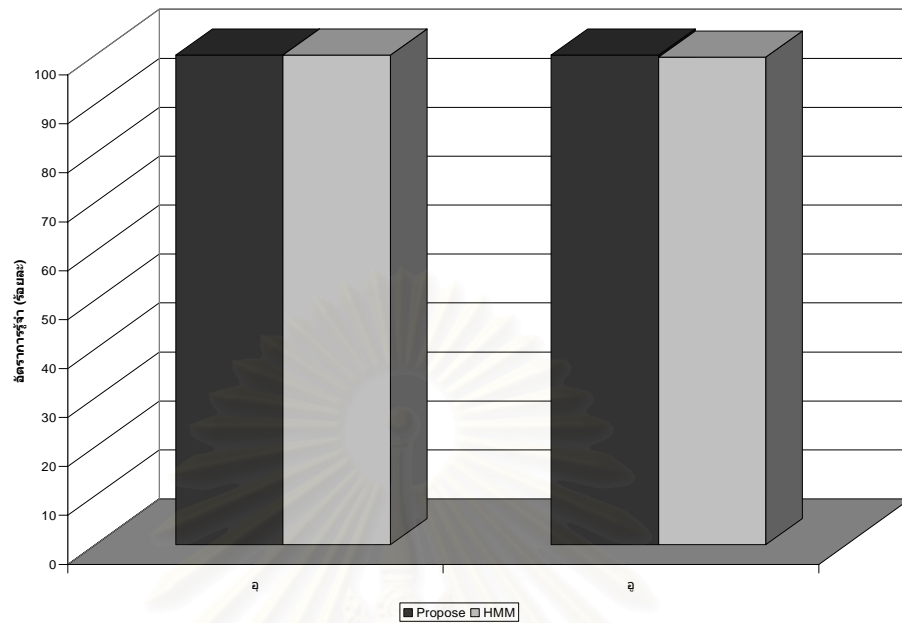
รูปที่ 4.4 แผนภูมิเปรียบเทียบอัตราการรู้จำตัวอักษรระดับกลางของข้อมูลตัวอักษรจากกลุ่มผู้เขียนอิสระ



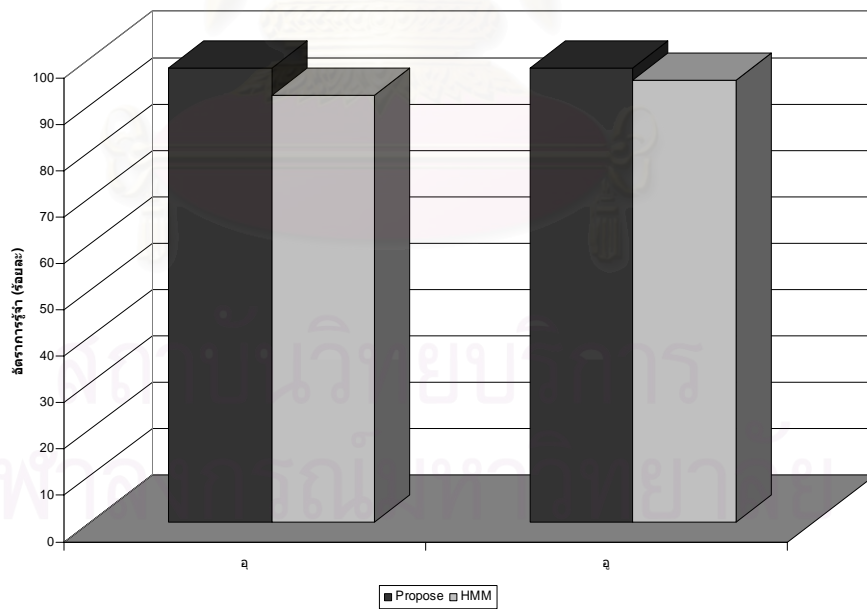
รูปที่ 4.5 แผนภูมิเปรียบเทียบอัตราการรู้จำตัวอักษรระดับบนของข้อมูลตัวอักษรจากกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ



รูปที่ 4.6 แผนภูมิเปรียบเทียบอัตราการรู้จำตัวอักษรระดับบนของข้อมูลตัวอักษรจากกลุ่มผู้เขียนอิสระ



รูปที่ 4.7 แผนภูมิเปรียบเทียบอัตราการรู้จำตัวอักษรระดับล่างของข้อมูลตัวอักษรจากกลุ่มผู้เขียนที่ใช้ในการฝึกฝนระบบ



รูปที่ 4.8 แผนภูมิเปรียบเทียบอัตราการรู้จำตัวอักษรระดับล่างของข้อมูลตัวอักษรจากกลุ่มผู้เขียนอิสระ

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 บทสรุป

จากการทดสอบระบบรู้จำที่นำเสนอโดยการนำการทำงานร่วมกันระหว่างแบบจำลองฮิดเดนมาร์คคอฟและซัพพอร์ตเวกเตอร์แมชชีนได้ยืนยันถึงแนวความคิดที่เสนอว่ามีความเป็นไปได้ในการประยุกต์ใช้กับตัวอักษรภาษาไทย โดยระบบรู้จำนี้ได้แยกระบบรู้จำออกเป็น 3 ระดับ ดังนี้

1. ระบบรู้จำตัวอักษรระดับบน โดยใช้แบบจำลองฮิดเดนมาร์คคอฟสองแบบที่มีโครงสร้างแบบจำลอง และคุณลักษณะสำคัญต่าง ๆ ทำการรู้จำสองขั้นตอน เนื่องจากคุณลักษณะสำคัญที่ใช้กับแบบจำลองฮิดเดนมาร์คคอฟขั้นตอนแรกสามารถทำการรู้จำได้ดี แต่มีเพียงกลุ่มสระ อี อี อี และ อี ที่มีการตัดสินสับสนกันภายในกลุ่ม จึงใช้แบบจำลองฮิดเดนมาร์คคอฟขั้นตอนแรกนี้ทำการแยกกลุ่มสระ ิ , ี , ึ และ ื ออกจากกลุ่มอื่น จากนั้นใช้กระบวนการก่อนหน้า และการสกัดคุณลักษณะสำคัญที่สามารถรู้จำกลุ่มสระดังกล่าวได้ดังได้นำเสนอในหัวข้อ 3.3 โดยการฝึกฝนแบบจำลองฮิดเดนมาร์คคอฟในขั้นตอนที่สองด้วยคุณลักษณะสำคัญใหม่เพื่อทำการรู้จำกลุ่มสระ ิ , ี , ึ และ ื
2. ระบบรู้จำตัวอักษรระดับกลาง วิทยานิพนธ์นี้ได้นำเสนอคุณลักษณะสำคัญขึ้นเพื่อแก้ไขปัญหาตัวอักษรที่มีการตัดสินผิดพลาดในหัวข้อ 4.4.4 โดยใช้แบบจำลองฮิดเดนมาร์คคอฟในการรู้จำกลุ่มตัวอักษรจากนั้นพิจารณาชนิดตัวอักษรที่ถูกตัดสิน หากพบว่าอยู่ในกลุ่มที่มีการตัดสินสับสนกับตัวอักษรอื่นซึ่งวิทยานิพนธ์นี้ได้นำเสนอวิธีการหาในหัวข้อ 3.2.3 แบบจำลองฮิดเดนมาร์คคอฟของกลุ่มตัวอักษรนี้จะถูกใช้เป็นพารามิเตอร์ในการส่งคะแนนเพื่อเป็นคุณลักษณะสำคัญป้อนเข้าซัพพอร์ตเวกเตอร์แมชชีน โดยในขั้นตอนสุดท้ายจะใช้ตัวจำแนกประเภทเป็นซัพพอร์ตเวกเตอร์แมชชีน ภายในกลุ่มที่มีการตัดสินสับสนที่มีตัวอักษรดังกล่าวอยู่โดยใช้เทคนิคมัลติคลาสแบบ Max Win ซึ่งกลุ่มที่มีการตัดสินสับสนได้จากการทดลองในหัวข้อที่ 4.5 โดยจะเห็นได้ว่าไม่ว่าจะใช้กลุ่มที่มีการตัดสินสับสนที่ได้จากการกลุ่มผู้เขียนที่ใช้ในการฝึกฝนจากตารางที่ 4.8 หรือกลุ่มผู้เขียนอิสระจากตารางที่ 4.9 ก็ตามอัตราการรู้จำเฉลี่ยมีค่าไม่ต่างกันมากนักดังเห็นได้จากตารางที่ 4.10 และ ตารางที่ 4.13

3. ระบบรู้จำตัวอักษรระดับล่าง เป็นระบบรู้จำลักษณะเดียวกับระบบรู้จำตัวอักษรระดับกลาง แต่เนื่องจากมีจำนวนชนิดเพียงสองชนิดคือสระและสระจึ่งไม่จำเป็นต้องมีเทคนิคการทำมัลติคลาสมาเกี่ยวข้องเนื่องจากซัพพอร์ตเวกเตอร์แมชชีนเป็นตัวจำแนกประเภทแบบนารีอยู่แล้ว

จากผลการทดสอบพบว่าอัตราการเรียนรู้จำถูกต้องเฉลี่ยของตัวอักษรและสระระดับกลางมีค่าร้อยละ 92.94 อัตราการเรียนรู้จำถูกต้องเฉลี่ยของสระและวรรณยุกต์ระดับบนมีค่าร้อยละ 90.57 และอัตราการเรียนรู้จำถูกต้องเฉลี่ยของสระระดับล่างมีค่าร้อยละ 97.88 ซึ่งจะเห็นได้ว่าตัวอักษรระดับล่างมีความเหมาะสมที่จะใช้กับระบบรู้จำด้วยแบบจำลองฮิดเดนมาร์คอปและซัพพอร์ตเวกเตอร์แมชชีนมากที่สุดเนื่องจากมีจำนวนชนิดตัวอักษรสองชนิดซึ่งเหมาะสมกับตัวแยกประเภทแบบไบนารี ดังเช่น ซัพพอร์ตเวกเตอร์แมชชีน

จุดเด่นของระบบรู้จำที่นำเสนอสามารถสรุปได้ดังนี้

1. มีอัตราการเรียนรู้จำสูงกว่าระบบรู้จำที่ใช้แบบจำลองฮิดเดนมาร์คอปพื้นฐาน
2. สามารถเพิ่มหรือลดตัวอักษรได้ โดยไม่ส่งผลกระทบต่อระบบส่วนใหญ่ และไม่ต้องฝึกฝนระบบใหม่ทั้งหมด
3. ไม่มีข้อกำหนดลักษณะบ่งความต่างที่ตายตัวทำให้สามารถรองรับลายมือของผู้เขียนได้หลากหลาย
4. ไม่ต้องอาศัยสมมุติฐานของมนุษย์ในการออกแบบลักษณะบ่งความต่าง

อย่างไรก็ตามระบบที่นำเสนอนี้ก็ยังมีข้อเสียอยู่ซึ่งสรุปได้ดังนี้

1. ต้องการพื้นที่ในการเก็บแบบจำลองฮิดเดนมาร์คอปและซัพพอร์ตเวกเตอร์แมชชีนค่อนข้างมาก
2. การคำนวณต้องการทรัพยากรและเวลาค่อนข้างมากกว่าระบบรู้จำที่ใช้แบบจำลองฮิดเดนมาร์คอปพื้นฐาน

5.2 ข้อเสนอแนะสำหรับงานในอนาคต

1. เนื่องจากภาษาไทยมีตัวอักษร 3 ระดับทำให้ระบบการรู้จำแต่ละระดับจำเป็นต้องแยกจากกัน และตัวอักษรบางตัวที่อยู่คนละระดับมีลักษณะเหมือนกัน เช่น ตัวอักษร 'บ' ซึ่งอยู่ระดับกลางกับสระ 'ู' ซึ่งจะอยู่ในระดับล่างทำให้สับสนได้หากมีระบบรู้จำระบบเดียว แต่ปัญหานี้สามารถแก้ไขได้โดยการรู้จำเป็นคำ หรือประโยคซึ่งจะสามารถประมวลแนวระดับได้
2. จากการทดสอบระบบพบว่ายังมีการรู้จำผิดอยู่มากในกลุ่มของตัวอักษรที่มีรอยหยักกับไม่มีรอยหยัก ซึ่งการแก้ปัญหานี้อาจทำได้โดยการปรับปรุงแก้ไขคุณลักษณะสำคัญ หรือการใช้เทคนิคอื่น ๆ เช่น ฟัซซีโลจิก
3. การนำไปประยุกต์ใช้จริงยังคงทำได้ยากอยู่เนื่องจากระบบที่นำเสนอเป็นเพียงการรู้จำในลักษณะอักษรเดี่ยวเท่านั้นการเขียนต่อเนื่องเป็นคำ หรือประโยคควรมีการศึกษารต่อไป
4. ระบบกระบวนการภายหลัง (Post Processing) จะช่วยเพิ่มประสิทธิภาพของระบบได้มากในส่วนของการเขียนเป็นคำหรือประโยค
5. ควรมีการทดลองหาค่า d ในหัวข้อ 3.2.2.4 ที่เหมาะสมที่สุด
6. การเพิ่มประสิทธิภาพของระบบรู้จำจะให้สูงขึ้นอาจทำได้โดยการวิเคราะห์แก้ไขและปรับปรุงคุณลักษณะสำคัญที่ใช้ เช่น อาจใช้เวกเตอร์ที่คำนวณจากจุดพิจารณาไปจุดสุดท้าย

รายการอ้างอิง

- [1] Lawrence R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceeding of the IEEE 77(Feb. 1989): 257-286.
- [2] Rud Budsayaparakorn, Widhayakorn Asdornwised, and Somchai Jitapunkul. On-line Thai handwritten characters recognition using Hidden Markov Model and Fuzzy logic. In IEEE International Workshop on Neural Networks for Signal Processing 13(Sep. 2003): 537-546.
- [3] Hiroshi Shimodaira, Ken-ichi Noma, Mitsuru Nakai, and Shigeki Sagayama. Support vector machine with dynamic time-alignment kernel for speech recognition. In Processing of Eurospeech (Sep. 2001).
- [4] William M. Campbell. A sequence kernel and its application to speaker recognition. In Neural Information Processing Systems. 14(Dec .2001): 1157-1163.
- [5] Aravind Ganapathiraju and Joseph. Picone. Hybrid SVM/HMM architectures for speech recognition. In Proceeding of International Conference on Spoken Language Processing 6(Oct. 2000): 504-507.
- [6] Claus Bahlmann, Bernard Haasdonk, and Hans Burkhardt. On-line handwriting recognition with Support Vector Machines—a kernel approach. In Proceeding of International Workshop on Frontiers in Handwriting Recognition. 8(Aug. 2002): 49-54.
- [7] A. R. Ahmad, M Khalid, and R. Yusof. Kernel Methods and Support Vector Machines for Handwriting Recognition. In Student Conference on Research and Development Proceedings. (2002).
- [8] A. Nair and C. G. Leedham. Preprocessing of line codes for online recognition purposes. Electronic Letters 12(Jan. 1991): 787-806.
- [9] Richard O. Duda, Peter E. Hart, and David G. Stork. Pattern Classification. New York: John Wiley & Sons, 2001.

- [10] Vladimir N. Vapnik. The Nature of Statistical Learning Theory. New York: Springer-Verlag, 2000.
- [11] Christopher J.C. Burges. A tutorial on Support Vector Machines for pattern recognition In DATA Mining and Knowledge Discovery. 2(1998): 121-167.
- [12] Thorsten Joachims. Tutorial Support Vector Machines. Technical report, Department of Computer Science, Cornell University.
- [13] Nathan Smith, Mark Gales, and Mahesan Niranjan. Data-dependent kernels in SVM classification of speech patterns. Technical report, Cambridge University, England, 2001.
- [14] Nathan Smith and Mark Gales. Speech recognition using SVM. In Neural Information Processing Systems. 14(Dec. 2001): 1197-1204
- [15] Rud Budsayaplakorn. Online Thai character recognition using Hidden Markov Model and Fuzzy logic. Master's thesis, Chulalongkorn University, Thailand, 2003.
- [16] Nathan Smith and Mark Gales. Using SVM and discriminative models for speech recognition. In Proceeding of IEEE International Conference on Speech, Acoustics, and Signal Processing. 1(May. 2002): I-77-I-80.
- [17] Vincent Wan and Steve Renals. Speaker verification using sequence discriminant Support Vector Machines. IEEE Transactions on Speech and Audio Processing. (2004).
- [18] Nathan Smith and Mark Gales. Using SVMs to classify variable length speech patterns. Technical Report, Cambridge University, England, 2002.
- [19] Vincent Wan. Speaker verification using Support Vector Machines. Ph.D. thesis, Sheffield University, June 2003.

- [20] Parinya Sanguansat, Widhayakorn Asdornwised, and Somchai Jitapunkul. Language-based hand-printed character recognition: a novel method using spatial and temporal informative features. In IEEE International Workshop on Neural Networks for Signal Processing 13(Sep. 2003): 527-536.
- [21] Parinya Sanguansat, Widhayakorn Asdornwised, and Somchai Jitapunkul. Online Thai handwritten character recognition using Hidden Markov Models and Support Vector Machines. In International Symposium on Communications and Information Technologies (Oct. 2004).
- [22] Paul M. Bagginstoss. Statistical modeling using Gaussian mixtures and HMMs with MATLAB. Technical Report, Naval Undersea Warfare Center, 2002.
- [23] S. Canu, Y. Grandvalet and A. Rakotomamonjy. SVM and Kernel Methods MATLAB Toolbox Perception. Systèmes et Information, INSA de Rouen, Rouen, France, 2003.
- [24] Schwaighofer. The SVM toolbox for MATLAB, Version 2.51. Available: <http://www.cis.tugraz.at/igi/aschwaig/software.html>, 2001.

ภาคผนวก

บทความทางวิชาการที่ได้รับการเผยแพร่

1. Parinya Sanguansat, Widhayakorn Asdornwised, and Somchai Jitapunkul. Online Thai handwritten character recognition using Hidden Markov Models and Support Vector Machines. To be published in International Symposium on Communications and Information Technologies (Oct. 2004).
2. Parinya Sanguansat, Widhayakorn Asdornwised, and Somchai Jitapunkul. Language-based hand-printed character recognition: a novel method using spatial and temporal informative features. In IEEE International Workshop on Neural Networks for Signal Processing 13(Sep. 2003): 527-536.
3. Widhayakorn Asdornwised, Pittichat Sootaroj, Parinya Sanguansat, Somchai Jitapunkul, and Chedsada Chinrungrueng. Context Partitions in Lossless Image Compression: An Enumerative Experiment. To be published in IEEE Region 10 International Conference on Global Connectivity in Energy, Computer, Communication & Control. (Nov. 2004).

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Online Thai Handwritten Character Recognition Using Hidden Markov Models and Support Vector Machines

Parinya Sanguansat, Widhyakorn Asdornwised, and Somchai Jitapunkul

Department of Electrical Engineering
Faculty of Engineering, Chulalongkorn University
Bangkok, 10330, Thailand
Email: {Widhyakorn.A, Somchai.J}@chula.ac.th

Abstract— In this paper, we present a proposed method for online Thai handwritten character recognition using Hidden Markov Models (HMMs) and Support Vector Machines (SVMs) with score-space kernels. Score-space kernels generalize Fisher kernels and are based on underlying generative models such as Gaussian mixture models (GMMs), which are output distribution of each state in HMMs. Our system combines the advantages of both generative and discriminative classifiers. The first phase, HMMs are used for multi-classification then SVMs are applied to resolve the certainty remaining after the first-pass HMM-based recognizer but did not apply for all of classes because the results of some classes are worse. We consider the confusion matrix of HMMs to find the confused candidates in each class. If the candidate is one then it means that no confusion in this class, alone HMMs are sufficient enough to classify. SVMs will be applied to system if the candidates are more than one. If there are more than two, the multi-class method will be applied to system. On account of the basic score-spaces, likelihood and likelihood ratio score-space are not symmetry. In case of likelihood score-space, the parameters refer to only one generative model from two class models. In case of likelihood ratio score-space, the parameters refer to both of them but in different position thus one observation sequence can map to two score-vectors. Here, we proposed a new score-space with symmetric property, called *symmetric likelihood ratio score-space*. In this way, one observation sequence is mapped to only one score-vector. The experimental results were shown an average recognition rate improved from 89.9%, using baseline HMM, to 92.5% using our proposed method.

I. INTRODUCTION

Many state-of-the-art online handwritten character recognition systems use Hidden Markov Models (HMMs) with continuous density output distributions modeled by Gaussian mixture models (GMMs) for many languages such as English, Arabic, Chinese and Japanese even Thai. Recently, in Thai language, Rud *et al.* [1] have developed online Thai handwritten character recognition by using HMMs and Fuzzy logic. This is done by using HMMs to find the posterior probability of train set and then use it for creating Fuzzy's rule with respect to this probability and human's assumption. We present alternative way to recognize the confused classes without making any assumption as Fuzzy logic. The binary

classifier is used for this task. In pattern recognition, Support Vector Machine (SVM) is binary classifier which has been successfully in many tasks for example in speech recognition [2] [3], speaker verification [4], and character recognition [5].

The SVM has many desirable properties including the ability to make both linear and nonlinear boundaries via kernel function and to classify data by using a small number of the training. However, it has some constrains for using SVM. The first constrain is the feature vectors must have fixed dimension but our features are variable length sequences. To overcome this constrain, Smith *et al.* [6] have defined a spaces, which variable length sequences are mapped to fixed dimensional vector, called *score-spaces*. This is done by using parameters of generative models such as HMMs or GMMs. The brief description of score-spaces appears in Section III.A. The second constrain is how to apply SVM in multi-classification. Because some sets of confused classes in Thai characters did not consist of two classes, the correct class is not the first and second order in N-best list from the result of HMMs phase. Many multi-class methods were applied to SVM such as Max Win algorithm, Decision Directed Acyclic Graph (DDAG), Error Correcting Output Codes (ECOC). In this paper, we choose Max Win algorithm for multi-class problem, as discussed in Section II.E because the number of class are not large. As general constrain for all recognizers or classifiers is the input data must be normalized for increasing performance. For normalizing score vectors, we found four normalizations, sequence length normalization [7], normalizing the score-maps for training and testing [6], score-space whitening [7], and spherical normalization [4], as briefly discuss in Section III.B.

The structure of this paper is as follow: the next section provides an overview of the online handwritten character system; Section III reviews score-spaces, our proposed score-space and methods for normalizing them; Section IV discusses about our proposed system, training and testing algorithm; Experimental evaluation and results are presented in Section V; Section VI concludes the papers.

II. ONLINE THAI HANDWRITTEN CHARACTER RECOGNITION

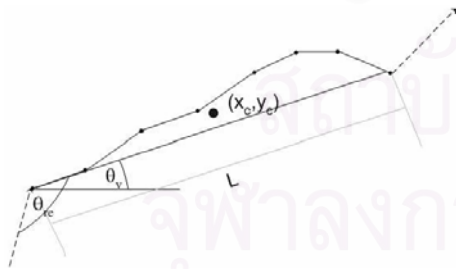
In this section, we will show our overall system; preprocessing, feature extraction, Hidden Markov Models, Support Vector Machines, and Max Win algorithm for multi-class problem.

A. Preprocessing

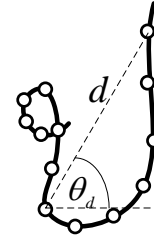
In online task, one of the informative could be coordinate sequence of pen moving, which could be from input device such as tablet. However, not all coordinates of the sequence are informative. In fact, they are very redundant. The preprocessing process removes these redundancies and normalizes sequences. From [8], the order of process is removal of accident pen-lifts, smoothing, minimum distance filtering, serif removal, and normalization. After all these process, the sequences are smaller and more efficient in classification.

B. Features Extraction

After preprocessing process, the sequence of each character is divided into sub-strokes by stroke segmentation. We hold on to the algorithm presented in [1] for representing sub-stroke, features are extracted for every sub-stroke. As shown in Fig. 1a, the feature vector of the i^{th} sub-stroke consists of center of sub-stroke coordinate (x_c, y_c) , angle from a starting to end point of sub-stroke based on x-axis (θ_s) , relative angle (θ_{re}) --a different between a considered sub-stroke angle and a previous sub-stroke angle, and a length of sub-stroke. Note that all of these features have local information that concerns only the current sub-stroke and previous one. We add two new features that provide global sub-stroke features. In Thai last sub-stroke contains more discriminative information, i.e., many confused pairs such as “จ” and “ช”, “จ” and “ญ”, and “ญ” and “จ”. As shown in Fig. 1b, the new features consist of distance from center of considered sub-stroke to center of the last sub-stroke (d) and angle of this distance line based on x-axis (θ_d) .



a.) sub-stroke features



b.) proposed features

Fig. 1. Sub-stroke features and proposed features.

C. Hidden Markov Models

HMM describes a stochastic process characterized by a finite state Markov chain and a set of output probability functions.

Given \bar{O} is an observation sequence and M_i is the HMM trained for the class i , the classification of an unknown pattern requires the computation the posteriori probability $P(\bar{O} | M_i)$ by using forward-backward algorithm. For K classes, if

$$i = \arg \max_{1 \leq i \leq K} P(\bar{O} | M_i) \quad (1)$$

then we will expect that \bar{O} belong to i^{th} class. For more detail about HMM are available in [9].

The topology of HMM in our system is left-to-right model with no skip and each state output distribution is a Gaussian components purpose of the score-map which relative to transition probabilities require this topology [6].

D. Support Vector Machines

The SVM is a binary linear classifier in its basic form. In the case of linearly separable two-class data, a separating hyperplane is chosen so as to maximize the margin between two classes.

Generally, the online handwritten character data are not separable and nonlinear boundary. In this case, SVM need a new parameter C for control trade-off between margin error and the kernel functions K for mapping input space, in which the data are observed, to a manifold in higher dimension feature space. Given training data $\bar{x}_i \in \mathcal{R}^N$ and their labels $y_i \in \{-1, +1\}$, the objective function is

$$\text{Maximize: } L(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\bar{x}_i, \bar{x}_j) \quad (2)$$

$$\text{Subject to: } \sum_i \alpha_i y_i = 0 \quad 0 \leq \alpha_i \leq C \quad (3)$$

in which α is the Lagrange multiplier of the constraints in the primal optimization problem. For test observation sequences \bar{x} are classified as

$$y = \text{sgn}\left(\sum_i y_i \alpha_i K(\bar{x}, \bar{x}_i)\right) + b \quad (4)$$

where b is the bias of the hyperplane that results from the constraints of the cost function. More details about SVMs are described by Vapnik [10] and in Burges' tutorial [11].

E. Max Win Algorithms

The SVMs are binary classifiers so we need an algorithm for combining these binary classifiers to multi-class classifier. This algorithm uses all pairs among all class candidates, which consider from HMMs confusion matrix. The number of SVMs is equal to the number of pairs and proportional to recognition time. Each SVM casts one vote, and the recognition result is the class which gets the maximum number of votes. According to the method, the rank for the results can be obtained by the proportion of the votes for each class.

III. SCORES AND SCORE-SPACES

T. Jaakkola *et al.* [12] suggest the *Fisher scores* as a means to map variable length observation sequences into fixed dimension vectors after that Smith *et al.* [6] generalize the Fisher kernel to score-space kernels and apply in speech recognition,

A. Score-spaces definition

Generally, the lengths of sequences are variable but the input of SVMs must be fixed-dimension. The score-space is a fixed-dimension space which variable length sequences are mapped explicitly onto single points. Score-space kernels [6], which generalize Fisher kernels [12], enable SVMs to classify whole sequences by exploiting a set of parametric generative models. Consider an observation sequence, $\bar{O} = [\bar{o}_1, \bar{o}_2, \bar{o}_3, \dots, \bar{o}_T]$ where T is duration of \bar{O} and a set of possible generative probability models for that sequence $P(\bar{O} | \bar{\theta}_i)$ where the vector of parameter, $\bar{\theta}$, for HMM with K Gaussian mixture components and N states is defined as following,

$$\bar{\theta} = \{w_{jk}, \mu_{jk}, \Sigma_{jk}, a_{ij}\}, \quad i, j \in [1, N], k \in [1, K] \quad (5)$$

where $w_{jk}, \mu_{jk}, \Sigma_{jk}$ are parameters of Gaussian component which are the weight, mean vector, and covariance matrix respectively for component k of state j , and a_{ij} is the transition probability from state i to state j .

The observation sequence \bar{O} is transformed to a score-space using the mapping defined below

$$\varphi_{\hat{F}}^f(\bar{O}) = \varphi_{\hat{F}}^f\left(\{p_k(\bar{O} | \bar{\theta}_k)\}\right) \quad (6)$$

where $f(\{p_k(\bar{O} | \bar{\theta}_k)\})$ is the *score-argument*, $\varphi_{\hat{F}}$ is the *score-mapping*, \hat{F} is the *score-operator*, and $\varphi_{\hat{F}}^f(\bar{O})$ is *score*.

In this section, we will demonstrate two particular score-spaces, the likelihood score-space and the likelihood-ratio score-space.

Likelihood score-space: likelihood score-space is defined on models with parameters estimated by Maximum Likelihood (ML) criterion and score-argument is the log likelihood of single generative models as shown below

$$\varphi_{\hat{F}}^{lik}(\bar{O}) = \varphi_{\hat{F}} \ln P(\bar{O} | \bar{\theta}) \quad (7)$$

Likelihood-ratio score-space: the score-argument of this score-space, motivated by discriminative binary classification schemes, is log likelihood-ratio between two competing models.

Consider two generative probability models $P(\bar{O} | \bar{\theta}_1)$ and $P(\bar{O} | \bar{\theta}_2)$ then

$$\varphi_{\hat{F}}^{lr}(\bar{O}) = \varphi_{\hat{F}} \ln \left[\frac{P(\bar{O} | \bar{\theta}_1)}{P(\bar{O} | \bar{\theta}_2)} \right] \quad (8)$$

Smith *et al.* [6] derived the score-space which defined from first-order derivative operators as shown below

$$\varphi(\bar{O}) = \nabla_{\bar{\theta}} \ln P(\bar{O} | \bar{\theta}) = \begin{bmatrix} \nabla_{\mu} \ln P(\bar{O} | \bar{\theta}) \\ \nabla_{\text{vec}(\Sigma)} \ln P(\bar{O} | \bar{\theta}) \\ \nabla_w \ln P(\bar{O} | \bar{\theta}) \\ \nabla_a \ln P(\bar{O} | \bar{\theta}) \end{bmatrix} \quad (9)$$

where

$$\nabla_{\bar{\mu}_k} \ln P(\bar{O} | \bar{\theta}) = \sum_{t=1}^T \gamma_{jk}(t) S_{[t, jk]}^T \quad (10)$$

$$\begin{aligned} & \nabla_{\text{vec}(\Sigma_k)} \ln P(\bar{O} | \bar{\theta}) \\ &= \sum_{t=1}^T \gamma_{jk}(t) \frac{1}{2} \left[-[\text{vec}(\Sigma_{jk}^{-1})]^T + \{S_{[t, jk]} \otimes S_{[t, jk]}\}^T \right] \end{aligned} \quad (11)$$

$$\nabla_{w_k} \ln P(\bar{O} | \bar{\theta}) = \sum_{t=1}^T \gamma_{jk}(t) \left[\frac{1}{w_{jk}} - \frac{\gamma(t)}{w_{j1} \gamma_{jk}(t)} \right] \quad (12)$$

$$\nabla_{a_{ij}} \ln P(\bar{O} | \bar{\theta}) = \sum_{t=1}^T \left[\frac{\gamma_i(t)}{a_{ii}} - \frac{1}{T a_{ii} (1 - a_{ii})} \right] \quad (13)$$

$$S_{[t, jk]} = (\bar{o}_t - \bar{\mu}_{jk})^T \Sigma_{jk}^{-1} \quad (14)$$

$$\gamma_j(t) = P(s(t) = i | \bar{O}, \bar{\theta}) \quad (15)$$

$$\gamma_{jk}(t) = \gamma_j(t) \times \left(\frac{w_{jk} \mathcal{N}(\bar{o}_t; \bar{\mu}_{jk}, \Sigma_{jk})}{\sum_{i=1}^K w_{ji} \mathcal{N}(\bar{o}_t; \bar{\mu}_{ji}, \Sigma_{ji})} \right) \quad (16)$$

$\gamma_j(t)$ is the state posterior probability for observation \bar{o}_i , being generated by state j which supplied by HMM recognizer and N is a Gaussian distribution for observation \bar{o}_i , of output distributions models in HMM.

Symmetric likelihood ratio score-space: the above score-spaces, likelihood and likelihood ratio score-space are not symmetry. Consider two classes problem, C1 and C2. In case of likelihood score-space, the parameters refer to only one generative model from two class models. The likelihood scores, which defined by parameters of model of class C1 and C2, are denoted as $\varphi_{C1}^{ik}(\bar{O})$ and $\varphi_{C2}^{ik}(\bar{O})$ respectively. In case of likelihood ratio score-space, the parameters refer to both of them but in different position thus one observation sequence can map to two score-vectors, $\varphi_{C1,C2}^{lr}(\bar{O})$ or $\varphi_{C2,C1}^{lr}(\bar{O})$ as shown below

$$\varphi_{C1,C2}^{lr}(\bar{O}) = \begin{bmatrix} \varphi_{C1}^{ik}(\bar{O}) \\ -\varphi_{C2}^{ik}(\bar{O}) \end{bmatrix}, \quad \varphi_{C2,C1}^{lr}(\bar{O}) = \begin{bmatrix} \varphi_{C2}^{ik}(\bar{O}) \\ -\varphi_{C1}^{ik}(\bar{O}) \end{bmatrix} \quad (17)$$

We proposed a new score-space with symmetric property, called *Symmetric likelihood ratio score-space* is defined as following,

$$\varphi_{C1,C2}^{lr}(\bar{O}) = \varphi_{C2,C1}^{lr}(\bar{O}) = \begin{bmatrix} \varphi_{C1}^{ik}(\bar{O}) + \varphi_{C2}^{ik}(\bar{O}) \\ -\varphi_{C1}^{ik}(\bar{O}) - \varphi_{C2}^{ik}(\bar{O}) \end{bmatrix} \quad (18)$$

In this way, one observation sequence is mapped to only one score-vector.

B. Score-space normalization

Normalization of the feature vectors is desirable because the observation sequences are variable, different duration, and SVMs are not invariant to linear transformations in feature space. We used four stages of normalization: sequence length normalization [7], normalizing the score-maps for training and testing [6], score-space whitening [7], and spherical normalization [4]. After scores were normalized the classifiers have more reliable and better performance.

Sequence length normalization: In handwritten character recognition, two observation sequences are similar content but at different rates of variation, project into two different regions of score-space. As proposed in [7], each state posterior is normalized by the total state occupancy over the observation sequence.

$$\gamma_j^N(t) = \frac{\gamma_j(t)}{\sum_{\tau=1}^T \gamma_j(\tau)} \quad (19)$$

The normalized component posteriors $\gamma_i^N(t)$ should be substituted for in the derivatives in Equations (10) to (16).

Normalizing the score-maps for training and testing:

It is important to normalize the scores correctly in the training and test phases. Because of our observation sequences are difference duration so SVM should be trained and tested on the mean average score per observation,

$$\tilde{\varphi}(\bar{O}_i) = \varphi(\bar{O}_i) / T_i \quad (20)$$

where $\tilde{\varphi}(\bar{O}_i)$ is normalized score and T_i is duration of \bar{O}_i .

Score-space whitening: If the score-space is not whitened then the Euclidean distance metric would be invalid. The inverse of the covariance matrix of the score-vectors, Σ_{sc}^{-1} are used to map the non-Euclidean vector to a Euclidean space so the whitened score $\varphi^N(\bar{O})$ becomes

$$\varphi^N(\bar{O}) = \Sigma_{sc}^{-1} \tilde{\varphi}(\bar{O}) \quad (21)$$

where,

$$\Sigma_{sc} = E\{[\tilde{\varphi}(\bar{O}) - E(\tilde{\varphi}(\bar{O}))][\tilde{\varphi}(\bar{O}) - E(\tilde{\varphi}(\bar{O}))]^T\} \quad (22)$$

However, in this paper, we used a diagonal approximation for Σ_{sc}^{-1} since the inversion of the full matrix is problematic, some of them are not positive definite.

Spherical normalization: The dimension of score vector is large thus dot products between score vectors may lead to an ill-conditioned Hessian even when individual component of the vectors are whitened. To solve this problem, spherical vector-length normalization or spherical normalization for short is applied. The concept idea is mapping score vector to a higher dimension vector and no information lost. Mapping a plane onto a sphere's surface may be achieved by many different projections but in this paper we use the modified stereographic projection as shown in Fig. 2. This mapping applied to the whitened score-vectors is

$$\varphi^N(\bar{O}) \rightarrow \phi(\varphi^N(\bar{O})) = \frac{1}{\sqrt{\varphi^N(\bar{O}) \cdot \varphi^N(\bar{O}) + d^2}} \begin{bmatrix} \varphi^N(\bar{O}) \\ d \end{bmatrix} \quad (23)$$

This spherical normalization is developed in the context of SVMs using high order polynomial kernels for discussion but we try to apply with Gaussian kernel, in greater depth and in the context of many kernels in [13].

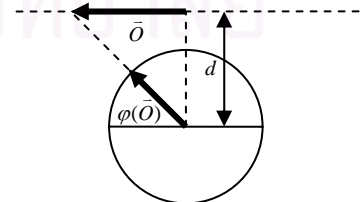


Fig. 2. Spherical normalization: The modified stereographic projection.

- [1] R. Budsayaparakorn, W. Asdornwised, and S. Jitapunkul, "On-line Thai handwritten characters recognition using hidden Markov model and Fuzzy logic," in *2003 IEEE International Workshop on Neural Networks for Signal Processing*, pp. 537-546, France, 2003.
- [2] N. Smith and M. Gales, "Speech recognition using SVM," in *Advances in the 14th Neural Information Processing Systems*, MIT Press, 2002.
- [3] N. Smith and M. Gales, "Using SVM and Discriminative Models for Speech Recognition," in *Proceeding of IEEE ICASSP*, 2002.
- [4] V. Wan and S. Renals, "Speaker verification using sequence discriminant Support Vector Machines," *IEEE Transactions on Speech and Audio Processing*, 2002.
- [5] C. Bahlmann, B. Haasdonk and H. Burkhardt, "On-line handwriting recognition with Support Vector Machines—A kernel approach," in *Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition*, pp.49-54, 2002.
- [6] N. Smith, M. Gales, and M. Niranjan, "Data-dependent kernels in SVM classification of speech patterns," Technical Report, University of Cambridge, England, 2001.
- [7] N. Smith, M. Gales, "Using SVMs to classify variable length speech patterns," Technical Report, University of Cambridge, England, 2002.
- [8] A. Nair and C. G. Leedham, "Preprocessing of line codes for online recognition purposes," *Electronic Letters*, Vol. 27, No. 1, 1991.
- [9] L.R. Rabiner, "Tutorial on Hidden Markov Models and selected applications in speech recognition," in *Proceeding of the IEEE*, vol. 77, no.2, Feb. 1989, pp. 257 -286.
- [10] V. N. Vapnik, *Statistical Learning Theory*, Wiley, 1998.
- [11] C.J.C. Burges, "A tutorial on Support Vector Machines for pattern recognition," submitted to *DATA Mining and Knowledge Discovery*, 1998.
- [12] T. S. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Advances in Neural Information Processing System 11*, M. S. Kearns, S. A. Solla, and D. A. Cohn, Eds. MIT Press, 1998.
- [13] V. Wan, *SpeakerVerification using Support Vector Machines*, Ph.D. thesis, University of Sheffield, June 2003.
- [14] P. M. Baggenstoss, "Statistical modeling using Gaussian mixtures and HMMs with MATLAB," Technical Report, Naval Undersea Warfare Center, 2002.
- [15] S. Canu and Y. Grandvalet and A. Rakotomamonjy, *SVM and Kernel Methods Matlab Toolbox Perception*, Systèmes et Information, INSA de Rouen, Rouen, France, 2003.
- [16] A. Schwaighofer, *The SVM toolbox for MATLAB, Version 2.51*, Available: <http://www.cis.tugraz.at/igi/aschwaig/software.html>, 2001.

ประวัติผู้เขียนวิทยานิพนธ์

นายปริญญา สงวนสัตย์ เกิดวันที่ 14 มิถุนายน พ.ศ. 2523 ที่กรุงเทพมหานคร เข้าศึกษาในหลักสูตรวิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2541 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ที่ห้องปฏิบัติการวิจัย สัญญาณดิจิทัล ภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2545



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย