การใช้ข้อมูลจำลองในการปรับโมเดลสำหรับการรู้จำเสียงพูดแบบคงทน

ROBUST SPEECH RECOGNITION USING SIMULATED-DATA IN MODEL ADAPTATION

ณัฐนันท์ ทัดพิทักษ์กุล
NATTANUN THATPHITHAKKUL

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2551

KMITL-2008-EW-D-018-223

การใช้ข้อมูลจำลองในการปรับโมเดลสำหรับการรู้จำเสียงพูดแบบคงทน

ROBUST SPEECH RECOGNITION USING SIMULATED-DATA IN MODEL ADAPTATION

ณัฐนันท์ ทัดพิทักษ์กุล

NATTANUN THATPHITHAKKUL

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2551

KMITL-2008-EW-D-018-223

ROBUST SPEECH RECOGNITION USING SIMULATED-DATA IN MODEL ADAPTATION

NATTANUN THATPHITHAKKUL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

DOCTOR OF ENGINEERING IN ELECTRICAL ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2008

KMITL-2008-EW-D-018-223

หัวข้อวิทยานิพนธ์	การใช้ข้อมูลจำลองในการปรับโมเดลสำหรับการรู้จำเสียงพูดแบบคงทน
นักศึกษา	นายณัฐนันท์ ทัดพิทักษ์กุล
รหัสประจำตัว	46060008
ปริญญา	วิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2551
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร. บุญธีร์ เครือตราชู

บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอเทคนิคใหม่ ในการปรับโมเดลเสียงพูดแบบออนไลน์ สำหรับการรู้จำเสียงพูดแบบคงทน โดยการปรับโมเดลด้วยข้อมูลจำลอง (Simulated-data Adaptation) ร่วมกับการประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดล (Hidden Markov model (HMM) Interpolation) วิธีที่นำเสนอแบ่งเป็นสองขั้นตอน ขั้นตอนที่หนึ่ง เป็นการสร้างโมเดลเสียงพูดขึ้นใหม่ จากการรวมโมเดลเสียงพูดที่มีเสียงรบกวนหลายๆ ชนิดเข้าด้วยกันด้วยวิธีการประมาณค่าในช่วง (Interpolation) วิธีนี้ต่างจากวิธีก่อนหน้านี้ที่มีการสร้างโมเดลเสียงผสมไว้ล่วงหน้า เนื่องจากในการใช้งานจริงต้องการความเร็ว และลดพื้นที่ในการจัดเก็บ รวมทั้งต้องการโมเดลที่ครอบคลุมเสียงรบกวนที่เกิดขึ้นจริง เป็นผลให้โมเดลที่สร้างขึ้นมาใหม่โดยวิธีที่นำเสนอมีความใกล้เคียงกับเสียงพูดที่เข้ามามากกว่า และใช้พื้นที่ในการจัดเก็บน้อยกว่า ขั้นตอนที่สอง เป็นการปรับโมเดลที่ได้จากขั้นตอนที่หนึ่งด้วยข้อมูลจำลอง โดยการผสมเสียงรบกวนจากเสียงพูดที่เข้ามากับเสียงพูดสะอาดที่เตรียมไว้ซึ่งสามารถรู้คำอ่าน (Transcript) ก่อนล่วงหน้า และสามารถเพิ่มข้อมูลให้เพียงพอต่อการปรับโมเดล เป็นผลให้โมเดลที่ได้จากขั้นตอนที่สองมีความใกล้เคียงกับเสียงพูดที่เข้ามายิ่งขึ้นกว่าโมเดลที่ได้จากขั้นตอนที่หนึ่ง

การทดลองวิธีที่นำเสนอ ใช้ข้อมูลเสียงพูดภาษาไทยที่มีเสียงรบกวน นอกเหนือจากเสียงรบกวนที่อยู่ในชุดฝึกสอน นอกจากนี้ ยังได้ทดลองกับคลังข้อมูลมาตรฐานออโรราทูเจ (AURORA-2J) เพื่อยืนยันว่าวิธีการที่นำเสนอสามารถใช้กับเสียงพูดภาษาอื่นได้ ผลการทดลองแสดงให้เห็นว่าวิธีที่นำเสนอให้ผลการรู้จำเสียงพูดที่ดีกว่าวิธีพื้นฐานที่มีการเลือกโมเดลที่สร้างเตรียมไว้ก่อน แล้วนำไปปรับด้วยเสียงพูดที่เข้ามาเพียงอย่างเดียว

Thesis	Robust Speech Recognition Using Simulated-Data in Model Adaptation
Student	Mr.Nattanun Thatphithakkul
Student ID.	46060008
Degree	Doctor of Engineering
Program	Electrical Engineering
Year	2008
Thesis Advisor	Assoc.Prof.Dr.Boontee Kruatrachue

ABSTRACT

This thesis proposes a novel technique for online acoustic-model adaptation in robust speech recognition. The technique is based on model adaptation using simulated data and hidden Markov model (HMM) interpolation. Two major steps are constructed in the proposed system. The first step is to build on-the-fly an acoustic model by interpolating several existing acoustic-models trained in different noisy environments. As opposed to previous algorithms that based mainly on selecting one of those existing acoustic-models, the proposed interpolation method lowers the model-selection time and storage. Moreover, it allows a possibility to generate a model for unseen environmental noise. The second step is to adapt the model created in the first step by using simulated data. Adaptation data can be simulated by mixing noise extracted from the input speech with pre-recorded clean speech whose phoneme transcriptions are already known. This avoids an adaptation error made by incorrect transcription and simultaneously helps increasing adaptation data. The acoustic model created in the second step will better match to the input speech than the one created in the first step.

Evaluations are performed mainly on Thai speech data collected in noisy environments unseen in the training set. For sake of language generalization, the method is also tested with a standard AURORA-2J corpus. Results clearly show that the proposed method outperforms the baseline system where a simple adaptation technique is applied on a selected acoustic-model.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สามารถประสบความสำเร็จลุล่วงมาได้อย่างดี โดยได้รับความกรุณาจาก รองศาสตราจารย์ ดร. บุญธีร์ เครือตราชู อาจารย์ผู้ควบคุมวิทยานิพนธ์ ซึ่งได้แนะแนวทางในการทำวิจัย ให้คำปรึกษาและให้ความช่วยเหลือในเรื่องต่างๆ เสมอมา ผู้วิจัยรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณกรรมการสอบวิทยานิพนธ์ทุกท่านที่ได้กรุณาให้คำแนะนำในเรื่องต่างๆ ทั้งวิธีการแก้ไขปัญหาที่เกิดขึ้นในวิทยานิพนธ์และมุมมองความรู้ในด้านอื่นๆ ทำให้ผู้วิจัยมีวิสัยทัศน์ที่กว้างไกลยิ่งขึ้น

ขอขอบคุณ ดร. ชัย วุฒิวิวัฒน์ชัย และ ดร. สรรพฤทธิ์ มฤคทัต ที่คอยให้คำปรึกษาที่มีประโยชน์ในการทำงานวิจัย

ขอขอบคุณเพื่อน ๆ ร่วมงานทุกคน สำหรับกำลังใจ และความห่วงใยที่มีให้ รวมทั้งความช่วยเหลือในเรื่องต่างๆ ทำให้ผู้วิจัยมีแรงใจที่จะมุมานะพยายามทำวิทยานิพนธ์ให้สำเร็จลงได้

คุณความดีอันใดที่บังเกิดจากวิทยานิพนธ์ฉบับนี้ ขอมอบแด่บิดาและมารดาของผู้วิจัย ผู้ซึ่งได้ให้โอกาสทางการศึกษาและสนับสนุนทั้งแรงกาย แรงใจ เป็นผู้คอยห่วงใย เอาใจใส่ตลอดมา ทำให้ผู้วิจัยได้รับโอกาสที่จะแสวงหาความรู้และพัฒนาตนเอง จนเกิดวิทยานิพนธ์ฉบับนี้ขึ้นมาได้ ผู้วิจัยสำนึกถึงพระคุณในข้อนี้เป็นอย่างสูง

สุดท้ายนี้ หากวิทยานิพนธ์ฉบับนี้มีข้อผิดพลาดแต่ประการใด ข้าพเจ้าขอน้อมรับไว้แต่เพียงผู้เดียว

ณัฐนันท์ ทัดพิทักษ์กุล

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

รูปที่ หน้า

สารบัญรูป (ต่อ)

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันมีการใช้งานระบบการรู้จำเสียงพูด (Speech Recognition System) มากขึ้น แต่ยังมีปัญหาที่ทำให้ประสิทธิภาพของระบบรู้จำเสียงพูดให้ผลได้ไม่ดี เพราะเสียงพูดที่ต้องการรู้จำและเสียงพูดที่ใช้ในการสร้างระบบการรู้จำมีความแตกต่างกัน โดยระดับความแตกต่างของเสียงพูดจะแปรผกผันกับประสิทธิภาพการรู้จำ และสิ่งที่ทำให้เสียงพูดมีความแตกต่างกันสามารถแบ่งออกได้เป็น 2 ประเภทใหญ่ๆ [1] คือ 1) ความแตกต่างของสภาพแวดล้อม ซึ่งทำให้มีเสียงรบกวน (Noise) ที่แตกต่างกัน และ 2) ความแตกต่างของผู้พูดที่ใช้งาน ดังนั้นจึงมีการพัฒนาระบบการรู้จำเสียงพูดแบบคงทน (Robust Speech Recognition System) ขึ้น โดยระบบนี้จะทำให้เสียงพูดที่เข้ามามีความใกล้เคียงกับเสียงพูดที่ใช้ฝึกสอน หรือทำให้โมเดลเสียงพูด (Acoustic Model) ที่มีสามารถใช้ได้กับเสียงพูดที่เข้ามา และในวิทยานิพนธ์นี้สนใจที่จะแก้ไขปัญหาเสียงรบกวนที่มาจากสภาพแวดล้อมที่ใช้งาน เนื่องจากเป็นปัญหาสำคัญในการนำไปใช้งานจริง ที่ทำให้ประสิทธิภาพการรู้จำเสียงพูดลดลง และยังคงเป็นปัญหาที่มีการวิจัยอยู่อย่างต่อเนื่อง

ระบบรู้จำเสียงพูดแบบคงทนที่ใช้ในการแก้ปัญหาเสียงรบกวนที่มาจากสภาพแวดล้อมสามารถแบ่งออกได้เป็น 3 แบบ [1] คือ 1) การหาลักษณะสำคัญของเสียงพูดแบบคงทน (Robust Speech Feature) เป็นการหาลักษณะสำคัญ (Feature) ของเสียงพูดที่มีความต้านทานต่อเสียงรบกวน 2) การปรับปรุงเสียงพูด (Speech Enhancement) เป็นการกรองเสียงรบกวนออกจากเสียงพูดที่มีเสียงรบกวน ทำให้เสียงพูดที่ได้จากการปรับปรุงเสียงพูดมีความใกล้เคียงกับเสียงพูดสะอาด (Clean Speech) และ 3) การปรับโมเดล (Model Adaptation) เป็นการปรับพารามิเตอร์ (Parameter) ของโมเดลด้วยเสียงพูดที่มีเสียงรบกวน (Noisy Speech) ทำให้โมเดลที่ได้จากการปรับสามารถใช้ได้ในสภาพแวดล้อมที่มีเสียงรบกวนเหมือนกับข้อมูลที่ใช้ปรับ โดยการหาลักษณะสำคัญของเสียงพูดแบบคงทนและการปรับปรุงเสียงพูดจะให้ผลการรู้จำเสียงพูดที่ดีกับเสียงรบกวนที่อยู่ในชุดฝึกสอน (Training Set) เท่านั้น แต่การปรับโมเดลสามารถให้ผลการรู้จำเสียงพูดที่ดีกับเสียงพูดซึ่งไม่ได้อยู่ในชุดฝึกสอนด้วย จึงทำให้การปรับโมเดลมีประสิทธิภาพการรู้จำเสียงพูดได้ดีกว่าการแก้ปัญหาแบบอื่น [1],[2],[3],[4] ดังนั้นวิทยานิพนธ์นี้ จึงสนใจในการพัฒนาระบบรู้จำเสียงพูดแบบคงทนแบบปรับโมเดล

ปกติแล้วระบบรู้จำเสียงพูดแบบคงทน แบบปรับโมเดล เป็นการแก้ปัญหาโดยใช้วิธีสร้างโมเดลจากเสียงพูดที่มีเสียงรบกวนขึ้นมาเก็บไว้ก่อน ด้วยการปรับโมเดลที่ถูกสร้างจากเสียงพูดสะอาด ให้เข้ากับสภาพแวดล้อมแบบต่างๆ และเรียกวิธีการนี้ว่าการปรับโมเดลแบบออฟไลน์ (Offline Model Adaptation) แต่ในความเป็นจริง ไม่สามารถสร้างโมเดลจากทุกเสียงรบกวนได้ ทำให้เกิดวิธีการที่เรียกว่าการปรับโมเดลแบบออนไลน์ (Online Model Adaptation) คือ การปรับโมเดลตามสภาพแวดล้อมจริงของเสียงพูดที่เข้ามาในแต่ละครั้ง เป็นผลให้วิธีการนี้สามารถแก้ปัญหาเสียงรบกวนที่ไม่มีในชุดฝึกสอนได้ วิธีที่ให้ผลการรู้จำเสียงพูดสูงมากวิธีหนึ่ง คือ การแปลงเชิงเส้นแบบแบ่งส่วน (Piecewise-linear Transformation, PLT) [3],[4],[5] ซึ่ง PLT เป็นการนำเทคนิคการปรับโมเดล ที่เรียกว่า การถดถอยแบบเชิงเส้นตามความเป็นไปได้สูงสุด (Maximum Likelihood Linear Regression, MLLR) [6],[7] มาใช้ร่วมกับวิธีการเลือกโมเดล (Model Selection) [5],[8] โดย PLT จะเลือกโมเดลเสียงพูดที่มีเสียงรบกวนที่มีความใกล้เคียงกับเสียงพูดที่เข้ามามากที่สุด จากกลุ่มของโมเดลเสียงพูดที่มีเสียงรบกวนที่มีการสร้างไว้ล่วงหน้า แล้วนำโมเดลที่เลือกได้ไปปรับแบบ MLLR ซึ่งวิธีการเลือกโมเดลที่ให้ได้ผลดีที่สุด คือ การเลือกโมเดลที่มีการจัดกลุ่มแบบโครงสร้างต้นไม้ (Model Selection based Tree-Structured Cluster, MSTC) [5] เนื่องจากบางโมเดลในโครงสร้างนี้มีการสร้างจากการผสมเสียงรบกวนมากกว่าหนึ่งชนิด ทำให้เกิดโมเดลที่สามารถรองรับเสียงรบกวนที่ไม่มีอยู่ในชุดฝึกสอนได้ ซึ่งโมเดลที่เกิดจากการผสมในที่นี้จะเรียกว่า “โมเดลแบบผสม” แต่อย่างไรก็ตาม การปรับโมเดลแบบออนไลน์ด้วย PLT มีปัญหาสำคัญซึ่งยังไม่สามารถแก้ได้อยู่ 3 ประการ คือ 1) การปรับโมเดลด้วย MLLR เป็นการปรับโมเดลตามคำอ่าน (Transcript) ของเสียงพูดที่ใช้ปรับโมเดล เนื่องจากเราไม่ทราบคำอ่านของเสียงพูดที่เข้ามา ทำให้ต้องมีการหาคำอ่านของเสียงพูดแบบอัตโนมัติ ดังนั้นถ้าใช้คำอ่านที่มีความถูกต้องต่ำ อาจทำให้มีการปรับโมเดลไม่ตรงกับความเป็นจริง ทำให้ผลการรู้จำเสียงพูดลดลง 2) การมีปริมาณข้อมูลเสียงพูดที่ใช้ในการปรับข้อมูลไม่เพียงพอ ถึงแม้ว่าวิธี MLLR จะใช้ข้อมูลในการปรับน้อยกว่าวิธีอื่นๆ แล้วก็ตาม แต่ก็ยังต้องมีข้อมูลที่มากเพียงพอด้วยถึงจะทำให้ได้โมเดลที่มีคุณภาพ และ 3) ปัญหาที่เกิดจากการเลือกโมเดลแบบ MSTC ที่มีการสร้างโมเดลแบบผสมเฉพาะจากเสียงรบกวนที่มีลักษณะคล้ายกันเท่านั้น ทำให้สามารถสร้างโมเดลแบบผสมได้มากสุดเท่ากับ N-1 โมเดล เมื่อ N คือ จำนวนโมเดลเสียงรบกวนพื้นฐานทั้งหมด ซึ่งอาจทำให้โมเดลที่เตรียมไว้ไม่ครอบคลุมเสียงรบกวนทั้งหมดที่เป็นไปได้

ดังนั้นเป้าหมายของวิทยานิพนธ์นี้ จึงเป็นการพัฒนาแนวคิดใหม่ ในการปรับโมเดลแบบออนไลน์ ให้ได้โมเดลที่มีคุณภาพดี เมื่อใช้กับเสียงพูดที่มีขนาดเล็กและไม่ทราบคำอ่านของเสียงพูด และใช้โมเดลตั้งต้นของการปรับโมเดลด้วยการสร้างโมเดลแบบผสมขึ้นมาใหม่แบบออนไลน์ ให้มีความใกล้เคียงกับเสียงพูดที่เข้ามา โดยที่ไม่ต้องสร้างโมเดลแบบผสมเตรียมไว้ล่วงหน้า

1.2 วัตถุประสงค์ของการศึกษา

1. เพื่อศึกษาและทำความเข้าใจระบบรู้จำเสียงพูดแบบคงทน แบบปรับโมเดลที่ใช้แก้ปัญหาเสียงรบกวนที่มาจากสภาพแวดล้อมที่ใช้งาน ในงานวิจัยก่อนหน้านี้
2. เพื่อสร้างระบบรู้จำเสียงพูดแบบคงทน แบบปรับโมเดลได้อย่างมีประสิทธิภาพ
3. เพื่อคิดค้นวิธีการใหม่ในการปรับโมเดล นำมาใช้สำหรับระบบรู้จำเสียงพูดแบบคงทน โดยวิทยานิพนธ์นี้มุ่งเน้นแก้ปัญหาเสียงรบกวนที่มาจากสภาพแวดล้อมที่ใช้งาน และเป็นเสียงรบกวนที่ไม่อยู่ในชุดฝึกสอน
4. วิเคราะห์เปรียบเทียบข้อดีข้อเสีย ระหว่างวิธีใหม่ที่นำเสนอในวิทยานิพนธ์กับวิธีที่มีการนำเสนอไปแล้ว
5. เพื่อเป็นแนวทางในการศึกษาวิจัยระบบรู้จำเสียงพูดแบบคงทน แบบการปรับโมเดล ให้สามารถใช้ในการแก้ปัญหาระบบรู้จำเสียงพูดแบบอื่นได้ต่อไปในอนาคต

1.3 สมมุติฐานของการศึกษา

ในวิทยานิพนธ์นี้สนใจการสร้างระบบรู้จำเสียงพูดแบบคงทน แบบปรับโมเดลแบบออนไลน์ ที่สามารถใช้งานในสภาพแวดล้อมที่มีเสียงรบกวนที่ไม่ได้อยู่ในชุดฝึกสอนได้ ซึ่งการปรับโมเดลมีอยู่ด้วยกัน 2 ขั้นตอน คือ 1) การสร้างโมเดลเสียงพูดที่มีเสียงรบกวนขึ้นมาใหม่ จากการรวมกันของโมเดลเสียงพูดที่มีเสียงรบกวนเข้าด้วยกัน ให้มีความใกล้เคียงกับเสียงพูดที่เข้ามา ซึ่งต่างจากวิธีก่อนหน้าที่มีการสร้างโมเดลเสียงพูดแบบผสมไว้ล่วงหน้า เนื่องจากในการใช้งานจริงต้องการความเร็วและลดพื้นที่ในการจัดเก็บ ทำให้โมเดลที่เตรียมไว้อาจจะไม่ครอบคลุมเสียงรบกวนที่เกิดขึ้นจริง เป็นผลทำให้โมเดลที่สร้างขึ้นมาใหม่มีความใกล้เคียงกับเสียงพูดที่เข้ามามากกว่า จากนั้นนำโมเดลที่ได้มาใช้เป็นโมเดลตั้งต้นของการปรับโมเดลในขั้นตอนถัดไป และ 2) การนำโมเดลตั้งต้นที่ได้ไปปรับโมเดลแบบออนไลน์ เพื่อให้ได้โมเดลที่มีความใกล้เคียงกับเสียงพูดที่เข้ามามากยิ่งขึ้น เนื่องจากในการปรับโมเดลแบบออนไลน์จะไม่ทราบคำอ่านของเสียงพูดที่เข้ามา อาจทำให้ปรับโมเดลได้ไม่ตรงตามเสียงพูดที่เข้ามา และถ้าเสียงพูดที่เข้ามามีขนาดเล็กก็จะไม่เพียงพอต่อการปรับโมเดลให้มีประสิทธิภาพได้ ดังนั้นในวิทยานิพนธ์นี้จึงนำเสนอการจำลองเสียงพูดสะอาดที่ทราบคำอ่าน ให้เป็นเสียงพูดที่มีเสียงรบกวนแบบเดียวกับเสียงรบกวนในเสียงพูดที่เข้ามา แล้วนำข้อมูลจำลองที่ทราบคำอ่านไปใช้ในการปรับโมเดล นอกจากนี้ ยังทำให้มีจำนวนข้อมูลที่มากเพียงพอต่อการปรับโมเดล เป็นผลทำให้โมเดลที่ได้จากขั้นตอนที่ 2 ให้ผลการรู้จำเสียงพูดได้ดีกว่าโมเดลที่ได้จากขั้นตอนที่ 1

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

ระบบรู้จำเสียงพูดในปัจจุบันนิยมใช้ สัมประสิทธิ์เซปสตรอลบนความถี่เมล (Mel Frequency Cepstral Coefficient, MFCC) เป็นลักษณะสำคัญของเสียงพูด และใช้ฮิดเดนมาร์คอฟโมเดล (Hidden Markov Model, HMM) เป็นโมเดลเสียงพูด [2],[3],[9] ดังนั้นโมเดลเสียงพูดที่ใช้ในการปรับโมเดล สำหรับระบบรู้จำเสียงพูดแบบคงทน คือ HMM และวิธีการปรับโมเดลแบ่งออกเป็น 2 ประเภท [3]

1. เทคนิคที่ใช้ในการปรับโมเดลเสียงพูด คือ การปรับค่าพารามิเตอร์ของโมเดลเสียงพูด สามารถแบ่งได้เป็น 2 แบบ คือ 1) แบบการแยกและการรวมโมเดล (Model Composition and Decomposition) เป็นการดึงเสียงรบกวนที่ได้จากเสียงพูดที่เข้ามา จากนั้นนำเสียงรบกวนที่ได้ไปผสมกับโมเดลเสียงพูดที่มีอยู่ เช่น การรวมโมเดลขนาน (Parallel Model Combination, PMC) [2],[10],[11],[12] และ โครงข่ายประสาทเทียม (Neural Network) [13] เป็นต้น และ 2) แบบการปรับพารามิเตอร์โมเดล (Model Parameter Adaptation) เป็นการปรับพารามิเตอร์ของโมเดลด้วยการใช้เสียงพูด ซึ่งเทคนิคที่ใช้ในการปรับพารามิเตอร์มีอยู่ด้วยกันหลายแบบ เช่น การฝึกสอนซ้ำ (Retrain) [14], การพิจารณาจากค่าประสบการณ์สูงสุด (Maximum a Posteriori , MAP) [15],[16] และ MLLR [6],[7] เป็นต้น และเนื่องจากการปรับพารามิเตอร์โมเดลเป็นการปรับโมเดลตามเสียงพูด ทำให้ต้องมีการเตรียมคำอ่านของเสียงพูด เพื่อใช้ในการระบุโมเดลเสียงพูดที่จะมีการปรับด้วย
2. ระบบการปรับโมเดลสำหรับการรู้จำเสียงพูดแบบคงทน คือ ระบบการปรับโมเดลเสียงพูดที่มีการนำเทคนิคในส่วนแรกมาใช้ แบ่งออกได้เป็น 2 ระบบ คือ 1) ระบบการปรับโมเดลแบบออฟไลน์ คือ ระบบที่มีการปรับโมเดลไว้ล่วงหน้าตามข้อมูลเสียงพูดที่มีเสียงรบกวนที่เตรียมไว้ เช่น วิธีการฝึกสอนแบบหลากสภาวะ (Multi-condition Training, MULTI) [14] และวิธีการเลือกโมเดล [20],[21],[22],[23] เป็นต้น และ 2) ระบบการปรับโมเดลแบบออนไลน์ เป็นการปรับโมเดลในขณะที่ใช้งานระบบรู้จำเสียงพูด เช่น PLT [3],[4],[5] เป็นต้น

วิทยานิพนธ์นี้สนใจ ระบบรู้จำเสียงพูดแบบคงทน แบบการปรับโมเดลแบบออนไลน์ ด้วยวิธี PLT เพราะสามารถใช้แก้ปัญหาเสียงรบกวนที่ไม่มีอยู่ในชุดฝึกสอนได้ดี ซึ่งวิธี PLT มีการทำงาน 2 ขั้นตอน คือ ขั้นตอนแรกเป็นการเลือกโมเดลจาก MSTC ที่ทำให้ได้โมเดลที่ใกล้เคียงกับเสียงรบกวนในเสียงพูดที่เข้ามามากที่สุด และขั้นตอนที่สองเป็นการนำโมเดลที่ได้จากขั้นแรกไปปรับโมเดลแบบ MLLR ด้วยเสียงพูดที่เข้ามาในเวลานั้นอีกครั้งหนึ่ง ซึ่งทำให้วิธี PLT มีประสิทธิภาพการรู้จำเสียงพูดดีกว่า PMC, โครงข่ายประสาทเทียม และ MLLR [3],[4] แต่อย่างไรก็ตาม PLT ยังมีปัญหาสำคัญ คือ ปัญหาที่เกิดจาก MLLR ที่ต้องการคำอ่านของเสียงพูดและขนาดเสียงพูดที่มากพอจึงจะทำให้ได้โมเดลที่มีคุณภาพ และอีกปัญหาหนึ่งเกิดจาก โครงสร้างแบบต้นไม้ของ MSTC ที่มี

จำนวนโมเดลแบบผสมที่สร้างจากเสียงพูดที่มีเสียงรบกวนมากกว่าหนึ่งชนิดได้อย่างจำกัด ซึ่งข้อดีของโมเดลแบบผสม คือ สามารถรองรับเสียงรบกวนที่ไม่ได้อยู่ในชุดฝึกสอนได้ดีกว่าโมเดลที่สร้างมาจากเสียงพูดที่มีเสียงรบกวนเพียงชนิดเดียว และถึงแม้จะหาโครงสร้างที่มีโมเดลแบบผสมได้มากขึ้น ก็ยังคงเกิดปัญหาเรื่องพื้นที่การจัดเก็บโมเดลแบบผสมที่เพิ่มขึ้นอยู่ดี

วิทยานิพนธ์นี้จึงนำเสนอ วิธีการปรับโมเดลด้วยการประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดลที่มีการจัดกลุ่มเสียงรบกวน (Noise Cluster HMM Interpolation, NCHI) ร่วมกับการปรับโมเดลด้วยข้อมูลจำลอง (Simulated-data Adaptation) [17] ซึ่งเป็นวิธีการปรับโมเดลแบบออนไลน์แบบใหม่ ซึ่งมีความแตกต่างจากแบบเดิม คือ 1) โมเดลตั้งต้นของการปรับโมเดล ซึ่งได้มาจากการเลือกโมเดลเสียงพูดที่มีเสียงรบกวนที่ใกล้เคียงเสียงพูดที่เข้ามากที่สุด จากโมเดลเสียงพูดที่มีเสียงรบกวนที่สร้างเก็บไว้ล่วงหน้า ส่วนโมเดลที่วิทยานิพนธ์นี้นำเสนอ เป็นการสร้างโมเดลเสียงพูดที่มีเสียงรบกวนขึ้นมาใหม่ ด้วยการประมาณค่าในช่วง (Interpolation) [24],[25] ของโมเดลเสียงรบกวนพื้นฐานที่สร้างเก็บไว้ล่วงหน้า ให้ได้โมเดลที่มีความใกล้เคียงกับเสียงพูดที่เข้ามามากที่สุด และ 2) การจำลองข้อมูลเสียงพูดที่วิทยานิพนธ์นี้นำเสนอ เป็นการผสมเสียงพูดสะอาดและเสียงรบกวนที่สร้างมาจากเสียงรบกวนของเสียงพูดที่เข้ามา และนำไปปรับพารามิเตอร์โมเดล ซึ่งวิธีนี้เป็นการรวมข้อดีของวิธีการปรับพารามิเตอร์โมเดล ในเรื่องความเร็วและประสิทธิภาพการปรับโมเดล และข้อดีของวิธีการรวมและการแยกโมเดล ที่ใช้หลักการผสมเสียงรบกวนและเสียงพูดสะอาดเข้าด้วยกัน ทำให้ไม่ต้องทราบคำอ่านของเสียงพูดที่เข้ามา และมีจำนวนข้อมูลเพียงพอสำหรับการปรับโมเดล

วิธีการปรับโมเดลของ NCHI ร่วมกับการปรับโมเดลด้วยข้อมูลจำลอง คือ การใช้การประมาณค่าในช่วงของโมเดลในการสร้างโมเดลเสียงพูดตั้งต้นขึ้นมาใหม่ ให้มีความใกล้เคียงกับเสียงพูดที่เข้ามา ด้วยการประมาณค่าในช่วงของโมเดลเสียงพูดที่มีเสียงรบกวนหลายๆ ชนิดเข้าด้วยกัน และนำโมเดลที่ได้จาก NCHI มาปรับด้วยข้อมูลจำลอง โดยข้อมูลที่จำลองขึ้นมาเป็นเสียงรบกวนแบบบวก (Additive Noise) เกิดจากการบวกเสียงรบกวนพื้นหลัง (Background Noise) ที่ได้มาจากส่วนเสียงเงียบ (Silence) ของเสียงพูดที่เข้ามาและเสียงพูดสะอาดที่เตรียมไว้ล่วงหน้า ทำให้มีข้อมูลมากเพียงพอต่อการปรับโมเดล ส่งผลให้ผลรู้จำเสียงพูดดีขึ้น และเห็นได้ว่าวิธีที่วิทยานิพนธ์นี้นำเสนอสามารถแก้ปัญหาของการปรับโมเดลแบบออนไลน์ ได้ครบทั้ง 3 ประการ คือ 1) การไม่ทราบคำอ่านของเสียงพูดที่เข้ามา 2) เสียงพูดที่เข้ามามีปริมาณน้อย ซึ่งทั้งสองปัญหาแก้ไขได้ด้วยการปรับโมเดลด้วยข้อมูลจำลอง และ 3) การมีโมเดลแบบผสมที่มีได้อย่างจำกัด และพื้นที่ในการเก็บโมเดล ซึ่งแก้ไขได้ด้วย NCHI

วิธีที่วิทยานิพนธ์นี้นำเสนอไม่สามารถแก้ปัญหาของผู้พูดที่ไม่ได้อยู่ในชุดฝึกสอนได้ เนื่องจากไม่ได้ใช้ส่วนที่เป็นเสียงพูดของเสียงพูดที่เข้ามาในการจำลองข้อมูล ดังนั้นเพื่อลดปัญหาที่อาจจะเกิดจากผู้พูด จึงมีการใช้เสียงพูดที่เข้ามาร่วมกับการจำลองข้อมูลด้วย นอกจากนี้ ยังมีการทดลองเปรียบเทียบผลการรู้จำเสียงพูด ระหว่างโมเดลที่ได้จากการปรับโมเดลด้วยข้อมูลจำลองเพียงอย่างเดียว และโมเดลที่ได้จากการปรับโมเดลด้วยข้อมูลจำลองร่วมกับเสียงพูดที่เข้ามา

1.5 ขอบเขตการวิจัย

วิทยานิพนธ์นี้เป็นการศึกษาและนำเสนอวิธีการสร้างระบบรู้จำเสียงพูดแบบคงทน โดยกำหนดขอบเขตปัญหาให้เฉพาะเจาะจง ซึ่งมีขอบเขตวิทยานิพนธ์ ดังนี้

1. วิทยานิพนธ์นี้เป็นการสร้างระบบรู้จำเสียงพูดแบบคงทน แบบการปรับโมเดล เพื่อแก้ปัญหาเสียงรบกวนที่มาจากสภาพแวดล้อมที่ใช้งาน ที่ไม่อยู่ในชุดฝึกสอน
2. เสียงพูดที่มีเสียงรบกวนที่ใช้ในวิทยานิพนธ์นี้ เป็นเสียงพูดภาษาไทยแบบคำโดด (Isolated Word) โดยเสียงพูดที่มีเสียงรบกวนที่ใช้ในการฝึกสอนเป็นเสียงพูดที่มีเสียงรบกวนแบบบวก แต่ในการทดลองจะเป็นเสียงพูดที่มีเสียงรบกวนแบบบวก และเสียงรบกวนจากสภาพแวดล้อมจริง (Real Noise)
3. ระบบรู้จำเสียงพูดใช้ MFCC เป็นลักษณะสำคัญของเสียงพูด และใช้ HMM เป็นโมเดลเสียงพูด โดยโมเดลเสียงพูดหนึ่งโมเดลแทนด้วยหนึ่งหน่วยเสียง (Phoneme) ดังนั้นสำหรับระบบรู้จำเสียงพูดภาษาไทยมีโมเดลเสียงพูดทั้งหมดเท่ากับ 76 โมเดล [26] (รวมเสียงเงียบเข้าไปด้วย) และไม่นับรวมวรรณยุกต์ (Tone)
4. ทำการเปรียบเทียบกับระบบรู้จำเสียงพูดแบบคงทน แบบการปรับโมเดลวิธีอื่น พร้อมทั้งวิเคราะห์ข้อดีข้อเสีย ในที่นี้จะเปรียบเทียบกับวิธีการปรับโมเดล คือ PMC [11], MLLR [7] และเปรียบเทียบกับระบบในการปรับโมเดล คือ MULTI [14], MSTC [5] และ PLT [5]

1.6 ขั้นตอนของการศึกษา

1. ศึกษาทฤษฎีและความรู้พื้นฐานที่เกี่ยวข้องกับระบบรู้จำเสียงพูดแบบคงทน แบบการปรับโมเดล
2. ศึกษาวิธีการสร้างระบบรู้จำเสียงพูดแบบคงทน แบบการปรับโมเดล โดยเรียนรู้จากงานวิจัยที่เกี่ยวข้อง ซึ่งมีผู้นำเสนอมาก่อนหน้านี้
3. พัฒนาวิธีการปรับโมเดล เพื่อใช้ในระบบรู้จำเสียงพูดแบบคงทน โดยนำเสนอแนวคิดใหม่ที่พัฒนาขึ้นเอง

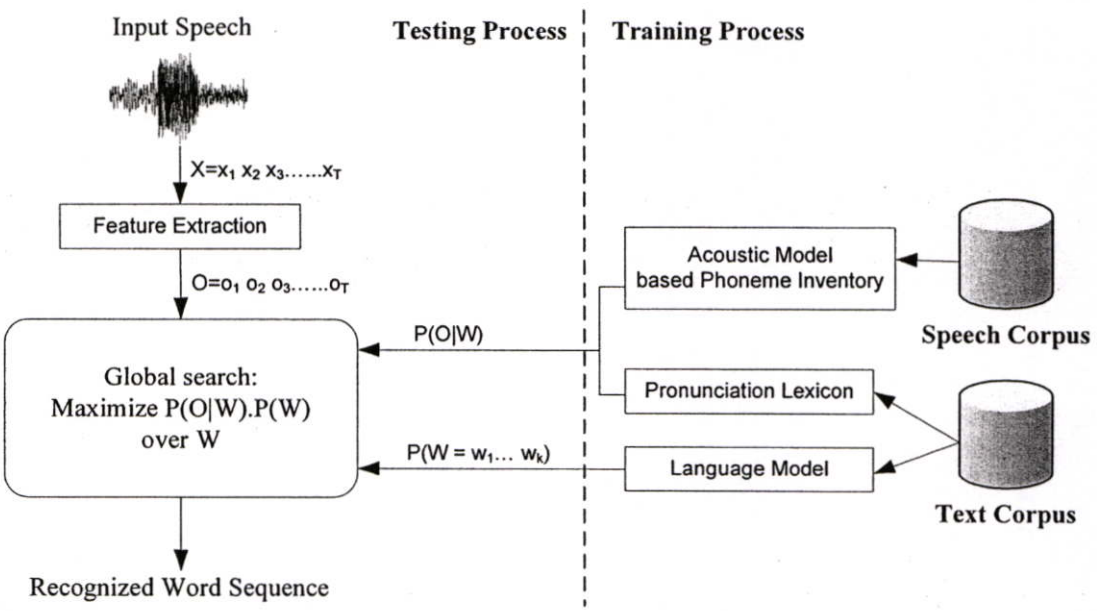
4. ทดลองสร้างระบบรู้จำเสียงพูดแบบคงทน แบบการปรับโมเดล ตามแนวคิดที่วางไว้ และวัดผลการรู้จำเสียงพูดของโมเดลเสียงพูดที่ได้จากการปรับโมเดล
5. ปรับปรุงการปรับโมเดลให้มีผลการรู้จำเสียงพูดเพิ่มขึ้น โดยการปรับข้อกำหนดต่างๆ พร้อมทั้งวัดผลการรู้จำเสียงพูดและวิเคราะห์ผลที่ได้ เพื่อนำมาพัฒนาต่อไป
6. ทดลองเปรียบเทียบผลการรู้จำเสียงพูดของการปรับโมเดล ระหว่างวิธีใหม่ที่นำเสนอกับวิธีอื่นๆ เพื่อศึกษาข้อดีข้อเสีย
7. สรุปผลการสร้างระบบรู้จำเสียงพูดแบบคงทน แบบการปรับโมเดลด้วยวิธีใหม่ที่พัฒนาขึ้น พร้อมวิเคราะห์ผลที่ได้
8. จัดทำเอกสารประกอบวิทยานิพนธ์

วิทยานิพนธ์นี้นำเสนอเทคนิคใหม่ ในการปรับโมเดลเสียงพูดแบบออนไลน์ สำหรับการรู้จำเสียงพูดแบบคงทน โดยการปรับโมเดลด้วยข้อมูลจำลอง ร่วมกับการประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดล วิธีที่นำเสนอมี 2 ขั้นตอน ขั้นตอนที่หนึ่ง คือ การสร้างโมเดลเสียงพูดขึ้นใหม่ ด้วยการประมาณค่าในช่วงของโมเดลที่มีเสียงรบกวนหลายๆ ชนิดเข้าด้วยกัน ขั้นตอนที่สอง คือ การปรับโมเดลด้วยข้อมูลจำลอง เป็นการปรับโมเดลที่ได้จากขั้นตอนที่หนึ่งด้วยข้อมูลจำลอง โดยข้อมูลจำลองได้จากการผสมเสียงรบกวนจากเสียงพูดที่เข้ามา และเสียงพูดสะอาดที่เตรียมไว้ล่วงหน้า

วิทยานิพนธ์นี้แบ่งออกเป็น 7 บท ได้แก่ บทที่ 1 บทนำ กล่าวถึง ความเป็นมาและความสำคัญของปัญหาระบบรู้จำเสียงพูด ตลอดจนนำเสนอแนวคิดและกำหนดขอบเขตของวิทยานิพนธ์ บทที่ 2 ระบบรู้จำเสียงพูด กล่าวถึงระบบรู้จำเสียงพูดที่นิยมใช้ในปัจจุบัน และปัญหาที่เกิดขึ้นในระบบรู้จำเสียงพูด และวิธีการแก้ปัญหาที่เกิดขึ้นด้วยระบบรู้จำเสียงพูดแบบคงทน บทที่ 3 งานวิจัยที่เกี่ยวข้อง กล่าวถึงงานวิจัยที่เกี่ยวข้องกับระบบรู้จำเสียงพูดแบบคงทน แบบการปรับโมเดลเสียงพูด บทที่ 4 การปรับโมเดลด้วยข้อมูลจำลอง และบทที่ 5 การประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดล กล่าวถึงวิธีการปรับโมเดลแบบออนไลน์แบบใหม่ที่วิทยานิพนธ์นี้นำเสนอ บทที่ 6 การทดลองและผลการทดลอง กล่าวถึงระบบรู้จำเสียงพูดและข้อมูลที่ใช้ในการทดลอง ผลการทดลอง และการวิเคราะห์ผลการทดลองของวิธีที่วิทยานิพนธ์นี้นำเสนอและวิธีต่างๆ ซึ่งมีความใกล้เคียงกับวิธีที่วิทยานิพนธ์นี้นำเสนอ และบทที่ 7 สรุปผลการทดลองและข้อเสนอแนะ

บทที่ 2
ระบบรู้จำเสียงพูด

ในบทนี้ นำเสนอระบบรู้จำเสียงพูด [9],[27] ซึ่งมีโครงสร้างเป็นดังรูปที่ 2.1 ระบบการรู้จำเสียงพูดนี้มีขั้นตอนการทำงาน คือ เมื่อมีเสียงพูดเข้ามาก็จะมีการดึงลักษณะสำคัญ (Feature Extraction) ของเสียงพูด โดยค่าที่ได้จากการดึงลักษณะสำคัญจะเรียกว่า ลำดับของค่าสังเกต (Observation, O) จากนั้นระบบจะนำลำดับของค่าสังเกตที่ได้ไปหาค่าความน่าจะเป็นที่จะเป็นเสียงพูดอะไรจากโมเดลเสียงพูด (λ) และโมเดลทางภาษา (Language Model) โดยที่โมเดลเสียงพูดใช้ในการหาค่าความน่าจะเป็นของลำดับของค่าสังเกตสำหรับคำ (W) ใดๆ (P(O|W)) แต่โดยปกติโมเดลเสียงหนึ่งโมเดลแทนด้วยหนึ่งหน่วยเสียง ดังนั้น W จะประกอบด้วยลำดับของหน่วยเสียงที่ประกอบเป็นคำ W นี้มาต่อกัน ตามการออกเสียงในพจนานุกรม (Pronunciation Lexicon) และ P(O|W) จะมีค่าเท่ากับผลคูณของ $P(O|\lambda)$ ของแต่ละหน่วยเสียงที่มาต่อกันเป็น W สำหรับโมเดลภาษาที่ใช้ในการหาค่าความน่าจะเป็น (P(W)) นั้น ใช้บอกความน่าจะเป็นของการเกิดลำดับของคำ หรือค่าความน่าจะเป็นที่คำใดๆ จะสามารถพูดต่อกันได้ และยังสามารถบอกได้ด้วยว่าประโยคที่เกิดจากคำที่มาต่อเรียงกันนั้น ($W = w_1\ w_2 ... w_k$ โดย w แทนคำแต่ละคำ) มีโอกาสเกิดขึ้นได้ด้วยความน่าจะเป็นเท่าไหร่ ดังนั้นในการเลือกผลการรู้จำเสียงพูด จะเลือก W ที่มีค่า P(O|W)P(W) มากที่สุด



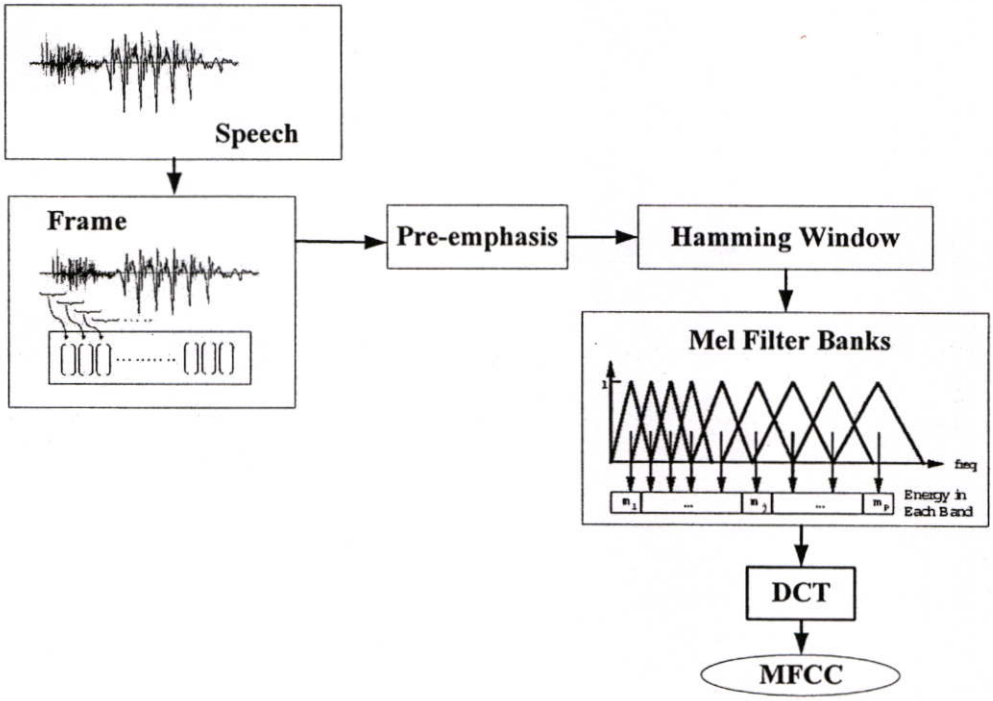
รูปที่ 2.1 ระบบการรู้จำเสียงพูด

ในการนำระบบรู้จำเสียงพูดไปใช้งานจริง มักจะเกิดปัญหาเสียงพูดที่ต้องการรู้จำและเสียงพูดที่ใช้ในการสร้างระบบการรู้จำมีความแตกต่างกัน ทำให้ประสิทธิภาพการรู้จำเสียงพูดลดลงไปมากจนไม่สามารถใช้งานได้ ทำให้ต้องมีการพัฒนาระบบรู้จำเสียงพูดแบบคงทน ซึ่งระบบสามารถแก้ปัญหาเรื่องความแตกต่างของเสียงพูดให้มีความใกล้เคียงกัน ในบทนี้กล่าวถึงระบบรู้จำเสียงพูดอย่างละเอียด อันประกอบด้วย หัวข้อ 2.1 การดึงลักษณะสำคัญของเสียงพูด หัวข้อ 2.2 โมเดลเสียงพูด หัวข้อ 2.3 โมเดลภาษา และหัวข้อ 2.4 ระบบรู้จำเสียงพูดแบบคงทน

2.1 การดึงลักษณะสำคัญของเสียงพูด

การวิเคราะห์และวัดค่าลักษณะสำคัญเป็นการวิเคราะห์เสียงพูด เพื่อเก็บรวบรวมลักษณะสำคัญของเสียงพูดแต่ละเสียง สำหรับการฝึกสอนระบบให้รู้ถึงความแตกต่างของเสียงพูดแต่ละเสียง เพื่อใช้ในการเปรียบเทียบแบ่งแยกความแตกต่างของเสียงพูดแต่ละเสียงออกจากกัน วิทยานิพนธ์นี้เลือกใช้สัมประสิทธิ์เซปสตรอลบนความถี่เมล (MFCC) เป็นลักษณะสำคัญ เพราะให้ผลการรู้จำที่ดีเมื่อนำมาใช้รวมกับฮิดเดนมาร์คอฟโมเดล (HMM) [9]

MFCC เป็นการนำเสียงพูดมาผ่านวงจรกรองแบบผ่านแถบความถี่ (Band Pass Filter) หลายวงจร ซึ่งมีช่วงความถี่ที่ผ่านได้แตกต่างกัน โดยเลียนแบบตามการได้ยินของมนุษย์ เป็นดังรูปที่ 2.2



รูปที่ 2.2 โครงสร้างการดึงลักษณะสำคัญของเสียงพูด

การดึงค่าลักษณะสำคัญด้วย MFCC มีขั้นตอนดังนี้

1) การแบ่งเสียงพูดออกเป็นเฟรม (Frame) เหตุที่ต้องแบ่งเป็นเฟรมนั้น ก็เนื่องจากทำให้เสียงพูดในแต่ละเฟรมมีความคงที่ (Stationary) และไม่แปรเปลี่ยนตามเวลา โดยแต่ละเฟรมมีช่วงเวลาเท่ากัน 25 มิลลิวินาที และมีช่วงเวลาในการซ้อนทับของเฟรมเท่ากับ 10 มิลลิวินาที

2) การเน้นล่วงหน้า (Pre-emphasis) เป็นการบีบอัดช่วงพิสัยพลวัต (Dynamic Range) ของเสียงพูด โดยการทำให้ความลาดเอียงในเชิงความถี่แบนราบลง ซึ่งส่งผลให้อัตราส่วนสัญญาณต่อสัญญาณรบกวน (Signal to Noise Ratio, SNR) มีค่าสูงขึ้น ด้วยการใช้ตัวกรองเชิงเลขอันดับหนึ่ง (First Order Digital Filter) ดังสมการ

$$\tilde{s}_i(n) = s_i(n) - 0.95 s_i(n-1) \tag{2.1}$$

เมื่อ

i คือ ลำดับเฟรมเสียงพูด

n คือ ลำดับข้อมูลในเฟรมที่ i

$\tilde{s}_i(n)$ คือ ค่าเสียงพูดขาออกของข้อมูลที่ n ในเฟรมที่ i

$s_i(n)$ คือ ค่าเสียงพูดขาเข้าของข้อมูลที่ n ในเฟรมที่ i

3) การทำหน้าต่างแบบแฮมมิ่ง (Hamming Window) เป็นการลดทอนแอมปลิจูด (Amplitude) อย่างช้าๆ ที่บริเวณปลายแต่ละข้างของเฟรมข้อมูลเสียงพูด เพื่อป้องกันการเปลี่ยนแปลงอย่างกะทันหันที่จุดปลาย ด้วยค่าฟังก์ชันหน้าต่าง (Window Function) ดังสมการ

$$x_i(n) = \left\{0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right)\right\}\tilde{s}_i(n) \quad ; \begin{matrix} i = 0,1,....,I-1 \\ n = 0,1,....,N-1 \end{matrix} \tag{2.2}$$

เมื่อ

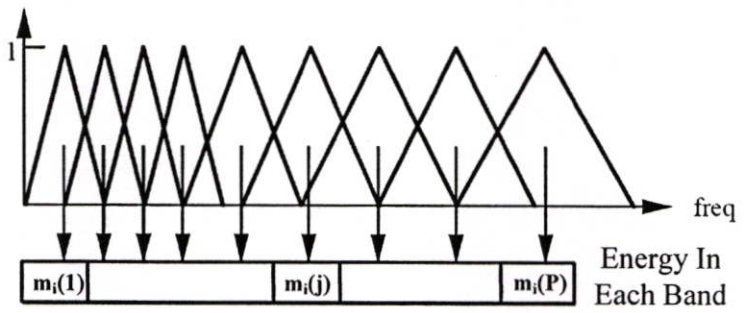
I คือ จำนวนเฟรมเสียงพูด

N คือ จำนวนข้อมูลในเฟรมเสียงพูด

$x_i(n)$ คือ เสียงพูดที่ผ่านการทำหน้าต่างแบบแฮมมิ่งของข้อมูลที่ n ในเฟรมที่ i

4) การนำเสียงพูดในแต่ละเฟรมไปผ่านวงจรกรองความถี่เมล (Mel Filter Bank) ที่เลียนแบบตามการได้ยินของมนุษย์ โดยวงจรกรองความถี่ที่ง่ายที่สุด และให้ผลที่มีประสิทธิภาพดี คือ

การแปลงฟูริเยร์แบบเร็ว (Fast Fourier Transform) ของเสียงพูด แล้วนำไปหาค่าลอการิทึม (Logarithm) ของพลังงาน (Energy) ของแต่ละวงจรกรองความถี่เมล [27] ดังรูปที่ 2.3



รูปที่ 2.3 วงจรกรองความถี่เมล [27]

เมื่อ

j คือ ลำดับวงจรของวงจรกรองความถี่เมล

P คือ จำนวนวงจรของวงจรกรองความถี่เมล

$m_i(j)$ คือ ค่าลอการิทึมของพลังงานของวงจรกรองความถี่วงจรที่ j ในเฟรมที่ i

5) การหาค่า MFCC คือ การนำ m_j ไปแปลงด้วยการแปลงดิสครีตโคไซต์ (Discrete Cosine Transform , DCT) ดังสมการ

$$C_i(k) = \sqrt{\frac{2}{P}} \sum_{j=1}^{P} m_i(j) \cos\left(\frac{\pi k}{P}(j-0.5)\right) \quad ;k = 1,2,...K \tag{2.3}$$

เมื่อ

K คือ จำนวนของ MFCC

$C_i(k)$ คือ ค่า MFCC ของข้อมูลที่ k ในเฟรมที่ i

2.2 โมเดลเสียงพูด

ระบบรู้จำเสียงพูดทั่วไปนิยมใช้ HMM เป็นโมเดลของเสียงพูด [9],[27] เนื่องจาก HMM เป็นโมเดลที่ใช้ในการจำแนกรูปแบบที่ดีที่สุดวิธีการหนึ่ง โดยเฉพาะสำหรับข้อมูลที่อยู่ในรูปลำดับ HMM เป็นการอาศัยวิธีการทางสถิติในการเก็บข้อมูลการกระจายที่สมบูรณ์ของลักษณะสำคัญที่มีอยู่ในข้อมูลที่ใช้ฝึกสอน จึงสามารถจำแนกความแตกต่างระหว่างเสียงพูดได้ดียิ่งขึ้น อีกทั้งขั้นตอนวิธีการนี้ยังอาศัยการโปรแกรมแบบพลวัต (Dynamic Programming) ทำให้มีความรวดเร็วในการประมวลผลมากยิ่งขึ้น

2.2.1 องค์ประกอบของฮิดเดนมาร์คอฟโมเดล

องค์ประกอบของ HMM ประกอบไปด้วยตัวแปรต่างๆ ดังนี้

1) O_t คือ ลำดับค่าสังเกต ที่เวลา t ซึ่งเซ็ตของลำดับค่าสังเกตแทนด้วย $O = \{O_1 O_2 \ldots O_T\}$ เมื่อ T คือ จำนวนลำดับของค่าสังเกต
2) N คือ จำนวนสเตท (State) ในโมเดล ซึ่งเซ็ตของสเตทแสดงได้ด้วย $S = \{S_1, S_2, \ldots, S_N\}$ โดยกำหนดให้ q_t คือ เสตทที่เวลา T และ $q_t \in S$
3) M คือ จำนวนค่าสังเกตที่สามารถเป็นไปได้ต่อสเตท และแทนเซ็ตของจำนวนค่าสังเกตด้วย $V = \{v_1, v_2, v_3, \ldots\ldots, v_M\}$
4) $A = \{ a_{ij} \}$ คือ ความน่าจะเป็นในการเปลี่ยนแปลงสเตท (State Transition Probability) โดย a_{ij} แทนการย้ายสเตทจากสเตท i ไปสเตท j
5) $B = \{b_j(O_t)\}$ คือ ความน่าจะเป็นของค่าสังเกตในสเตทที่ j โดยมีการกระจายความน่าจะเป็นของค่าสังเกตเป็นแบบเกาส์เซียนมิกเจอร์ (Gaussian Mixture) ซึ่งมีลักษณะการกระจายแบบต่อเนื่อง (Continuous Distribution) ดังสมการ

$$b_j(O_t) = \sum_{m=1}^{\hat{M}} c_{jm} \tilde{N}(O_t; \mu_{jm}, \Sigma_{jm}) \tag{2.4}$$

$$\tilde{N}(O; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(O-\mu)' \Sigma^{-1} (O-\mu)} \tag{2.5}$$

เมื่อ

$\hat{M}$ คือ จำนวนของมิกซ์เจอร์

c_{jm} คือ ค่าถ่วงน้ำหนักที่ส่วนประกอบมิกซ์เจอร์ m และสเตท j

μ คือ เวกเตอร์ค่าเฉลี่ย

Σ คือ เมตริกซ์ความแปรปรวนร่วม

$\tilde{N}(O; \mu, \Sigma)$ คือ การกระจายเกาส์เซียนซึ่งประกอบด้วย μ และ Σ

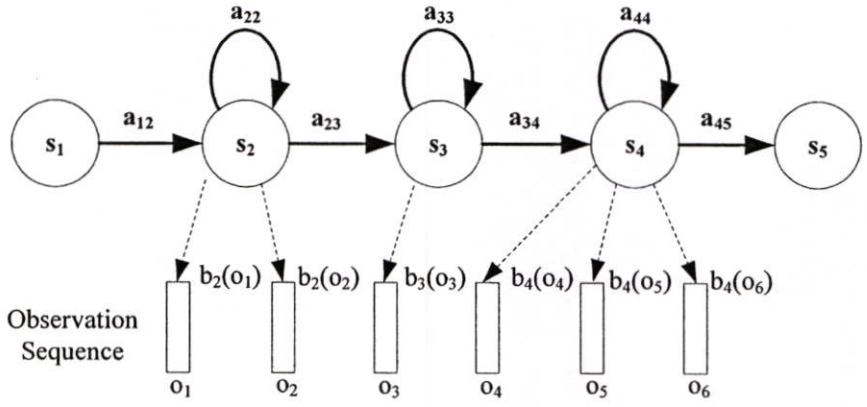
n คือ จำนวนมิติของเวกเตอร์ค่าสังเกต O

6) ค่าความน่าจะเป็นของสเตทเริ่มต้น $\pi = \{\pi_i\}$ และกำหนดให้ π_i มีค่าเท่ากับ 0 เมื่อ $i \neq 1$ และ π_i มีค่าเท่ากับ 1 เมื่อ $i = 1$

การกำหนดคุณสมบัติเฉพาะของ HMM ต้องการคุณสมบัติเฉพาะของพารามิเตอร์โมเดลสองค่า คือ N และ M ส่วนคุณสมบัติเฉพาะของค่าสังเกต และคุณสมบัติเฉพาะของการวัดค่าความน่าจะเป็น ได้แก่ A, B, π โดยเขียนอยู่ในรูปแบบย่อ เพื่อบ่งบอกชุดของพารามิเตอร์ที่สมบูรณ์ของโมเดล HMM คือ $\lambda = (A, B, \pi)$

2.2.2 คุณสมบัติการย้ายสเตทของฮิดเดนมาร์คอฟโมเดล

วิทยานิพนธ์นี้เลือกใช้การย้ายสเตทของ HMM แบบซ้าย - ขวา ดังรูปที่ 2.4 ซึ่งเป็นโมเดลแบบที่เหมาะกับสัญญาณที่มีลักษณะเปลี่ยนแปลงตามเวลาอย่างต่อเนื่อง เช่น สัญญาณเสียงพูด และในวิทยานิพนธ์นี้ได้ใช้ HMM แบบซ้าย – ขวาที่มีลักษณะของ สเตทแรกและสเตทสุดท้าย ที่ไม่มีการวนซ้ำอยู่ในสเตทเดิม ถ้าลำดับค่าสังเกตตกอยู่ในสเตทสุดท้ายเป็นอันสิ้นสุดกระบวนของการรู้จำเสียงพูดของโมเดลนี้



รูปที่ 2.4 ฮิดเดนมาร์คอฟโมเดลแบบซ้าย – ขวาที่มี 5 สเตท

2.2.3 ปัญหาพื้นฐานสามประการของฮิดเดนมาร์คอฟโมเดล

การประยุกต์ใช้งาน HMM นั้น ในทางปฏิบัติ คือ การแก้ปัญหาพื้นฐานทั้งสามประการ โดยมีรายละเอียดของปัญหาดังนี้

ปัญหาที่ 1 เมื่อมีลำดับของค่าสังเกต O และมีโมเดล λ จะคำนวณหาค่าความน่าจะเป็นของลำดับค่าสังเกต $P(O|\lambda)$ ตามโมเดลที่กำหนดให้ได้อย่างไร

ปัญหาที่ 2 เมื่อมีลำดับของค่าสังเกต O และโมเดล λ จะคำนวณหาลำดับสเตท $q = \{q_1 q_2 \cdots q_T\}$ ที่เหมาะสมกับลำดับค่าสังเกตนั้นได้อย่างไร

ปัญหาที่ 3 การปรับค่าพารามิเตอร์ของโมเดล λ อย่างไร เพื่อให้ได้ค่าความน่าจะเป็นของลำดับค่าสังเกต $P(O|\lambda)$ ที่มีค่าสูงสุดได้

2.2.4 การแก้ไขปัญหาพื้นฐานสามประการของฮิดเดนมาร์คอฟโมเดล

ในการแก้ไขปัญหาพื้นฐานสามประการของ HMM ตามปัญหาที่ได้กำหนดนั้นสามารถอธิบายวิธีแก้ไขในแต่ละปัญหาดังต่อไปนี้

การแก้ปัญหาที่ 1 ทำได้โดยระบุสเตทให้กับลำดับของค่าสังเกตที่มีจำนวนเท่ากับ T ซึ่งสามารถเป็นไปได้ถึง N^T แบบ โดยให้สเตทต่างๆ แทนด้วย q เมื่อ q_1 เป็นสเตทเริ่มต้นที่เวลา t = 1 ความน่าจะเป็นของ O ได้มาจากผลรวมของความน่าจะเป็น O และ q เกิดขึ้นพร้อมกัน โดยคิดจากทุก สเตท q ที่จะเป็นไปได้ดังนี้

$$P(O|\lambda) = \sum_{allq} P(O|q,\lambda)P(q|\lambda) \tag{2.6}$$

$$P(O|\lambda) = \left(\sum_{q_1 q_2 \cdots q_T} \pi_{q_1} . b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_T) \ldots a_{q_{T-1} q_T} b_{q_T}(O_T) \right) \tag{2.7}$$

จากสมการที่ 2.7 มีการคำนวณที่ยุ่งยาก เนื่องจากมีการคูณกันเป็นจำนวนมากในรูปของลำดับ $2TxN^T$ ดังนั้นจึงมีการพัฒนาวิธีอื่นมาช่วย ซึ่งแบ่งออกเป็น

1) กระบวนการไปข้างหน้า

ตัวแปรของการไปข้างหน้า ($\alpha_t(i) = P(O_1 O_2 \ldots O_t, q_t = S_i|\lambda)$) คือ ความน่าจะเป็นของการเกิดลำดับค่าสังเกตบางส่วนจากเวลา 1 ถึงเวลา t และสเตท S_i ที่เวลา t โดยมีโมเดลเป็น λ และหา $\alpha_t(i)$ ได้ดังนี้

ขั้นตอนที่ 1 การเริ่มต้น

$$\alpha_1(i) = \pi_i b_i(O_1) \quad ;1 \le i \le N \tag{2.8}$$

ขั้นตอนที่ 2 การเหนี่ยวนำ (Induction)

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} a_{ij} \alpha_t(i) \right] b_j(O_{t+1}) \quad ; \begin{matrix} 1 \le t \le T-1 \\ 1 \le j \le N \end{matrix} \tag{2.9}$$

ขั้นตอนที่ 3 การสิ้นสุด

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{2.10}$$

2) กระบวนการย้อนกลับ

ตัวแปรของการย้อนกลับ ($\beta_t(i) = P(O_{t+1}O_{t+1}.....O_T \mid q_t = S_i, \lambda)$) คือ ความน่าจะเป็นของการเกิดลำดับค่าสังเกตบางส่วนจาก t + 1 จนถึง T และสเตท S_i ที่เวลา t โดยมีโมเดลเป็น λ และหา $\beta_t(i)$ ได้ดังนี้

ขั้นตอนที่ 1 การเริ่มต้น

$$\beta_T(i) = 1 \;\; ;1 \le i \le N \tag{2.11}$$

ขั้นตอนที่ 2 การเหนี่ยวนำ

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \;\; ; \begin{matrix} t = T-1, T-2, ..., 1 \\ 1 \le i \le N \end{matrix} \tag{2.12}$$

ขั้นตอนที่ 3 การสิ้นสุด

$$P(O \mid \lambda) = \sum_{i=1}^{N} \pi_i b_i(O_1) \beta_1(i) \tag{2.13}$$

การแก้ปัญหาที่ 2 ด้วย Viterbi Algorithm เพื่อหาลำดับสเตทที่ดีที่สุดเพียงลำดับเดียว $Q = \{q_1 q_2 .....q_T\}$ สำหรับลำดับของค่าสังเกต $O = \{O_1 O_2 \ldots O_T\}$ โดยกำหนดตัวแปรดังนี้

$$\delta_t(i) = \max_{q_1, q_2, ..., q_{t-1}} P[q_1 q_2 .....q_{t-1}, q_t = i, O_1 O_2 ....O_t \mid \lambda] \tag{2.14}$$

โดยที่ $\delta_t(i)$ คือ ค่าความน่าจะเป็นที่มีค่าสูงที่สุดของเส้นทางเดี่ยวที่เวลา t ซึ่งเป็นค่าสังเกต t ค่าแรก และสิ้นสุดในสเตท S_i ด้วยวิธีการเหนี่ยวนำดังนี้

$$\delta_{t+1}(j) = [\max_i \;\; \delta_t(i) a_{ij}] b_j(O_{t+1}) \tag{2.15}$$

ในการเรียกใช้ค่าลำดับสเตทจำเป็นต้องติดตามค่าอาร์กิวเมนต์ที่ทำให้สมการที่ (2.15) มีค่ามากที่สุด สำหรับแต่ละค่าเวลา t และสเตท j โดยอาศัยแถวลำดับ $\psi_t(j)$ ซึ่งมีขั้นตอนในการหาลำดับสเตทที่ดีที่สุดเพียงลำดับเดียว ดังนี้

ขั้นตอนที่ 1 การเริ่มต้น

$$\delta_1(i) = \pi_i b_i(O_1) \;; 1 \le i \le N \tag{2.16}$$

$$\psi_1(i) = 0 \tag{2.17}$$

ขั้นตอนที่ 2 การเหนี่ยวนำ

$$\delta_t(j) = \max_{1 \le i \le N}[a_{ij}\delta_{t-1}(i)]b_j(O_t) \;; \begin{matrix} 2 \le t \le T \\ 1 \le j \le N \end{matrix} \tag{2.18}$$

$$\psi_t(j) = \arg \max_{1 \le i \le N}[a_{ij}\delta_{t-1}(i)] \;; \begin{matrix} 2 \le t \le T \\ 1 \le j \le N \end{matrix} \tag{2.19}$$

ขั้นตอนที่ 3 การสิ้นสุด

$$P^* = \max_{1 \le i \le N}[\delta_T(i)] \tag{2.20}$$

$$q_T^* = \arg \max_{1 \le i \le N}[\delta_T(i)] \tag{2.21}$$

ขั้นตอนที่ 4 เส้นทางเดินย้อนกลับ (Backtracking)

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \;; t = T-1, T-2, \ldots\ldots, 1 \tag{2.22}$$

การแก้ปัญหาที่ **3** ด้วยกระบวนการวนซ้ำของ Baum-Welch ซึ่งเป็น EM (Expectation-Maximization Method) อัลกอริทึมแบบหนึ่ง กำหนดให้ $\overline{\lambda} = (\overline{A}, \overline{B}, \overline{\pi})$ คือ โมเดลเสียงพูด $\lambda = (A, B, \pi)$ที่ถูกปรับพารามิเตอร์ ให้มีค่าความน่าจะเป็นของลำดับค่าสังเกตมากที่สุด ได้ดังนี้

$$\overline{\pi}_i = \gamma_1(i)) \tag{2.23}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \zeta_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{2.24}$$

$$\bar{b}_j(k) = \frac{\sum_{t=1, O_t = v_k}^{T-1} \gamma_t(j)}{\sum_{t=1}^{T-1} \gamma_t(j)} \tag{2.25}$$

เมื่อ

$\zeta_t(i,j)$ คือ ค่าความน่าจะเป็นในการอยู่สเตท S_i ที่เวลา t และสเตท S_j ที่เวลา t+1

$$\delta_t(i,j) = P(q_t = S_i, q_{t+1} = S_j \mid O, \lambda) \tag{2.26}$$

$$\zeta_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \tag{2.27}$$

$\gamma_t(i)$ คือ ความน่าจะเป็นของการอยู่สเตท S_i ที่เวลา t

$$\gamma_t(i) = \sum_{j=1}^{N} \zeta_t(i,j) \tag{2.28}$$

2.3 โมเดลภาษา

โมเดลทางภาษาใช้ความรู้ทางสถิติในการหาค่าความน่าจะเป็นทางภาษา (P(W)) ของคำที่มาต่อเรียงกัน ($W = w_1 w_2 ... w_k$ และ k คือ จำนวนคำที่มาต่อกันเป็น W) โดยใช้ข้อมูลจากคลังข้อความขนาดใหญ่ และมีสมการ คือ

$$P(W) = \prod_{i=1}^{k} P(w_i \mid w_1 w_2 .... w_{i-1}) \tag{2.29}$$

และมีการประมาณค่าของ P(W) ให้มีจำนวนคำเท่ากับ N และเรียกการประมาณค่านี้ว่า โมเดลภาษาแบบ N-Gram ซึ่งมีสมการ คือ

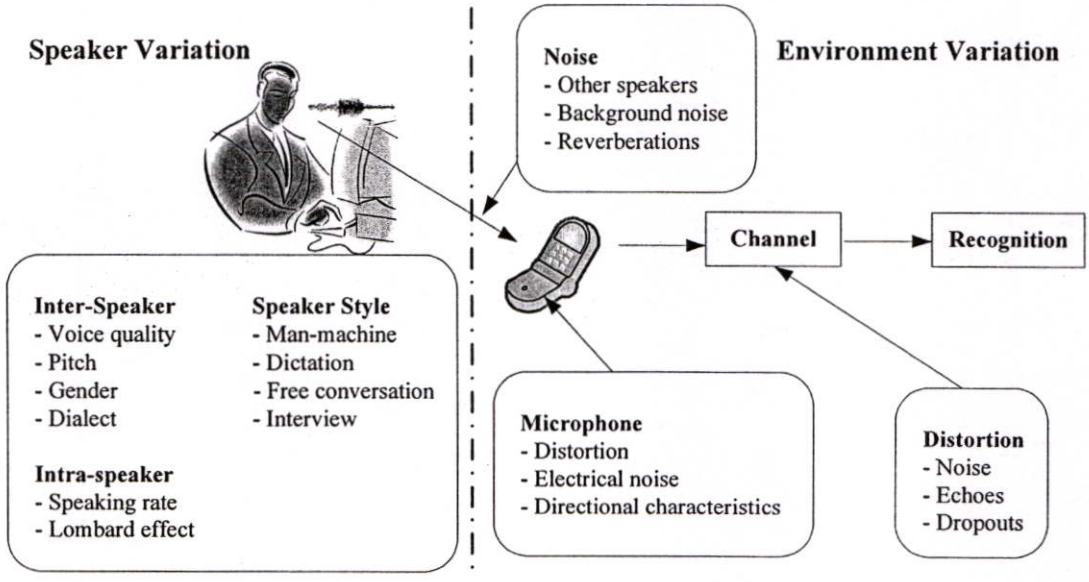
$$P_N(W) = \prod_{i=1}^{k} P(w_i \mid w_1 w_2 .... w_{i-N+1}) \tag{2.30}$$

$$P(w_i \mid w_1 w_2 .... w_{i-N+1}) = \frac{F(w_1 w_2 .... w_{i-N+1})}{F(w_{i-1} w_{i-2} .... w_{1+N+1})} \tag{2.31}$$

โดยที่ F คือ จำนวนเหตุการณ์ที่มีอยู่ในชุดสอนของคลังข้อความ และเรียกโมเดลภาษาที่ N เท่ากับสองว่า Bi-Gram และเรียกโมเดลภาษาที่ N เท่ากับสามว่า Tri-Gram แต่เนื่องจากในวิทยานิพนธ์นี้เป็นการรู้จำเสียงพูดแบบคำโดดทำให้ P(W) ของคำทุกคำมีค่าเท่ากันหมด ดังนั้นในการเลือกผลการรู้จำเสียงพูดจะพิจารณาเฉพาะค่าของ P(O|W) เท่านั้น

2.4 ระบบการรู้จำเสียงพูดแบบคงทน

ปัญหาที่ทำให้ระบบการรู้จำเสียงพูดยังไม่ได้รับการนำไปใช้อย่างแพร่หลาย เป็นเพราะเสียงพูดที่ต้องการรู้จำ และเสียงพูดที่ใช้ในการสร้างระบบการรู้จำมีความแตกต่างกัน จึงทำให้ประสิทธิภาพของระบบรู้จำเสียงพูดลดลง สาเหตุหลักของความแตกต่างของเสียงพูด ดังรูปที่ 2.5



รูปที่ 2.5 สาเหตุหลักของความแตกต่างของเสียงพูด [3]

สิ่งที่ทำให้เสียงพูดมีความแตกต่างกันสามารถแบ่งออกได้เป็น 2 ประเภทใหญ่ๆ คือ 1) ความแตกต่างของสภาพแวดล้อมที่ใช้งาน เช่น ตัวรับสัญญาณเสียง, วิธีการส่งสัญญาณเสียง และสภาพแวดล้อมที่มีการใช้งาน เป็นต้น และ 2) ความแตกต่างของผู้พูดที่ใช้งาน เช่น สำเนียงของชาวต่างประเทศ, ภาษาถิ่น, อายุ และ เพศ เป็นต้น โดยระดับความแตกต่างของเสียงพูดจะแปรผกผันกับประสิทธิภาพการรู้จำเสียงพูด ดังนั้นจึงมีการพัฒนาระบบการรู้จำเสียงพูดแบบคงทนขึ้น เพื่อลดระดับความแตกต่างที่เกิดขึ้นของเสียงพูดที่เข้ามาให้ใกล้เคียงกับเสียงพูดที่ใช้ฝึกสอน หรือปรับโมเดลเสียงพูดที่สร้างมาจากเสียงพูดที่ใช้ฝึกสอนให้มีความใกล้เคียงกับเสียงพูดที่เข้ามาใหม่

วิทยานิพนธ์นี้สนใจในการแก้ไขปัญหาเสียงรบกวนที่มาจากสภาพแวดล้อมที่ใช้งาน เนื่องจากเป็นปัญหาสำคัญในการนำไปใช้งานจริง แล้วทำให้ประสิทธิภาพการรู้จำเสียงพูดลดลง ปัญหานี้ยังคงเป็นปัญหาที่มีการวิจัยอยู่อย่างต่อเนื่อง โดยระบบรู้จำเสียงพูดแบบคงทน ที่ใช้ในการแก้ปัญหาเสียงรบกวนที่มาจากสภาพแวดล้อมสามารถแบ่งออกได้เป็น 3 แบบ [1],[2],[3] คือ

1) การหาลักษณะสำคัญเสียงพูดแบบคงทน คือ การหาลักษณะสำคัญของเสียงพูดที่มีความต้านทานต่อเสียงรบกวน ซึ่งวิธีนี้จะทำให้ลักษณะสำคัญของเสียงพูดที่มีเสียงรบกวน มีความใกล้เคียงกับลักษณะสำคัญของเสียงพูดสะอาด วิธีการดึงลักษณะสำคัญที่ใช้ในระบบรู้จำเสียงพูดแบบคงทน ได้แก่ Linear Prediction Coefficients (LPC) [9], Short-term Modified Coherence (SMC) [28], Linear Discriminant Analysis (LDA) [29], MFCC [9], Perceptual Linear Predictive (PLP) [30] และ Relative Spectral (RASTA) [31] เป็นต้น นอกจากนี้ ยังมีวิธีการทำให้ลักษณะสำคัญเสียงพูดมีความคงทนต่อเสียงรบกวนมากขึ้นด้วยการกรองเสียงรบกวนออกจากลักษณะสำคัญที่หาได้ ซึ่งวิธีที่นิยมใช้ คือ Cepstral Mean Normalization (CMN) [32] วิธีหาลักษณะสำคัญเสียงพูดแบบคงทน ให้ผลการรู้จำเสียงพูดที่ดีกับเสียงรบกวนที่มีลักษณะคงที่และมีความเฉพาะเจาะจง เช่น เสียงรบกวนจากช่องสัญญาณ (Channel Noise) และต้องเป็นเสียงรบกวนที่อยู่ในชุดสอนด้วย [1],[2],[3],[4]
2) การปรับปรุงเสียงพูด คือ การกรองเสียงรบกวนออกจากเสียงพูดที่มีเสียงรบกวน ทำให้เสียงพูดที่ได้จากปรับปรุงเสียงพูดมีความใกล้เคียงกับเสียงพูดสะอาด ทำให้ประสิทธิภาพของระบบรู้จำเสียงพูดที่ฝึกสอนมาจากเสียงพูดสะอาด มีความคงทนต่อเสียงรบกวนมากยิ่งขึ้น และวิธีการปรับปรุงเสียงพูดสำหรับระบบรู้จำเสียงพูดแบบคงทน ได้แก่ Spectral Subtraction (SS) [33], Probabilistic Optimal Filtering (POF) [34], Code-book Dependent Cepstral Normalization (CDCN) [35], Wiener Filter [36] และ Kalman Filter [37] เป็นต้น การปรับปรุงเสียงพูด ให้ผลการรู้จำเสียงพูดที่ดีกับเสียงพูดที่มีเสียงรบกวนที่อยู่ในชุดฝึกสอน แต่เมื่อนำทดสอบกับเสียงพูดที่มีเสียงรบกวนที่ไม่ได้อยู่ในชุดฝึกสอน พบว่ายังให้ผลการรู้จำเสียงพูดได้ไม่ดีนัก [1],[2],[3],[4]

3) การปรับโมเดล คือ การปรับพารามิเตอร์ของโมเดลเสียงพูด ด้วยเสียงพูดที่มีเสียงรบกวน ทำให้โมเดลที่ได้จากการปรับสามารถใช้ได้ในสภาพแวดล้อม ที่มีเสียงรบกวนเหมือนกับข้อมูลที่ใช้ปรับโมเดล และวิธีการปรับโมเดลที่ใช้ในระบบรู้จำเสียงพูดแบบคงทน ได้แก่ Stochastic Matching [38], PMC [2],[10],[11],[12], MAP [15],[16], MLLR [6],[7], โครงข่ายประสาทเทียม [13] และ PLT [3],[4],[5] เป็นต้น การปรับโมเดล ให้ผลการรู้จำเสียงพูดที่ดีทั้งกับเสียงพูดที่มีเสียงรบกวนที่อยู่ในชุดฝึกสอนและไม่ได้อยู่ในชุดฝึกสอน [1],[2],[3],[4]

วิทยานิพนธ์นี้จึงสนใจพัฒนาระบบรู้จำเสียงพูดแบบคงทน แบบการปรับโมเดล เพื่อใช้ในการแก้ปัญหาเสียงรบกวนที่มาจากสภาพแวดล้อมที่ใช้งาน และเป็นเสียงรบกวนที่ไม่ได้อยู่ในชุดฝึกสอนด้วย

บทที่ 3
งานวิจัยที่เกี่ยวข้อง

ในบทนี้ เป็นการนำเสนองานวิจัยที่เกี่ยวข้องกับการปรับโมเดลเสียงพูด สำหรับระบบรู้จำเสียงพูดแบบคงทน โดยมีเนื้อหาที่สำคัญ 2 ส่วน ได้แก่หัวข้อ 3.1 เทคนิคที่ใช้ในการปรับโมเดลเสียงพูด ซึ่งกล่าวถึง เทคนิคที่ใช้ในการปรับค่าพารามิเตอร์ของโมเดลเสียงพูด เช่น PMC [3],[11], การฝึกสอนซ้ำ [14], MAP [16] และ MLLR [7] เป็นต้น และหัวข้อ 3.2 ระบบการปรับโมเดลสำหรับการรู้จำเสียงพูดแบบคงทน ซึ่งกล่าวถึงระบบการปรับโมเดลเสียงพูดที่มีการนำเทคนิคในส่วนแรกมาใช้ เช่น การเลือกโมเดล [20], การฝึกสอนแบบหลากสภาวะ [14], และ PLT [4] ปืนต้น

3.1 เทคนิคในการปรับโมเดล

เทคนิคในการปรับโมเดลในปัจจุบันสามารถแบ่งตามวิธีปรับพารามิเตอร์ของโมเดลเสียงพูดเป็น 2 แบบ คือ 1) แบบการรวมและการแยกโมเดล และ 2) แบบการปรับพารามิเตอร์โมเดล ซึ่งเทคนิคทั้งสองแตกต่างกันที่วิธีการปรับพารามิเตอร์ของ HMM โดยหัวข้อ 3.1.1 กล่าวถึง การรวมและการแยกโมเดล และหัวข้อที่ 3.1.2 กล่าวถึง การปรับพารามิเตอร์โมเดล

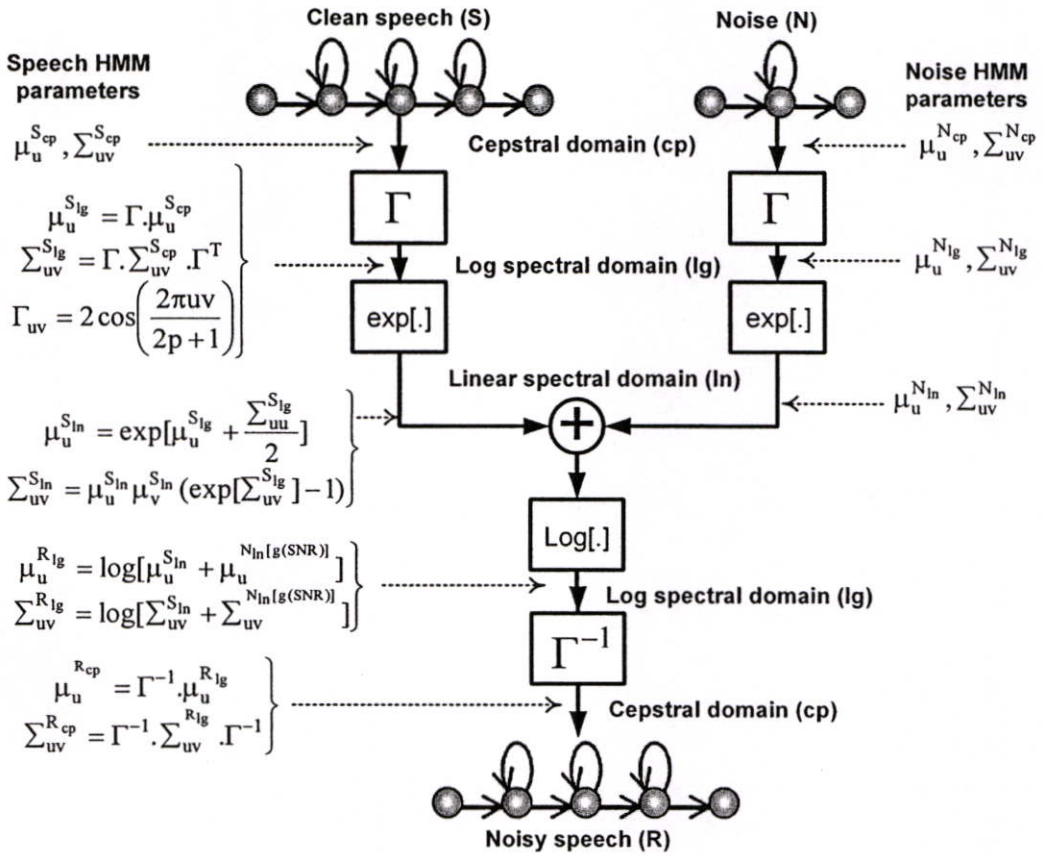
3.1.1 การรวมและการแยกโมเดล

การปรับโมเดลแบบการรวมและการแยกโมเดล เป็นการดึงเสียงรบกวนจากเสียงพูดที่เข้ามาด้วยการหาส่วนที่เป็นเสียงพูดและส่วนที่ไม่เป็นเสียงพูด (Speech/Non-speech Detection) โดยเสียงรบกวนที่ต้องการนั้นอยู่ในส่วนที่ไม่เป็นเสียงพูดหรือส่วนเสียงเงียบ จากนั้นนำเสียงรบกวนที่ได้ไปสร้างโมเดลเสียงรบกวน แล้วนำไปผสมกับโมเดลเสียงพูดสะอาดที่มีอยู่ ตัวอย่างของการปรับโมเดลด้วยวิธีนี้ได้แก่ PMC และโครงข่ายประสาทเทียม เป็นต้น และวิธีที่ใกล้เคียงกับวิธีที่วิทยานิพนธ์นี้นำเสนอ คือ PMC ซึ่งจะกล่าวถึงในหัวข้อถัดไป

3.1.1.1 การรวมโมเดลขนาน

การรวมโมเดลขนาน (PMC) [11] เป็นวิธีการปรับเฉพาะค่าเฉลี่ยและความแปรปรวนร่วมในพารามิเตอร์ B ของ HMM เพื่อให้ได้โมเดลเสียงพูดที่มีเสียงรบกวน โดยการผสมโมเดลเสียงพูดสะอาด และโมเดลเสียงรบกวนเข้าด้วยกัน มีโครงสร้างดังรูปที่ 3.1 โดยมีการผสม HMM เข้าด้วยกันในโดเมนเสปคตรอลเชิงเส้น (Linear Spectral Domain) ดังนั้นก่อนที่จะมีการผสมกันต้องมีการแปลงจากโดเมนเซปสตรอล (Cepstral Domain) ไปเป็นโดเมนเสปคตรอลเชิงเส้นเสียก่อน และเมื่อผสมกันเสร็จต้องมีการแปลงกลับจากโดเมนเสปคตรอลเชิงเส้นกลับไปเป็นโดเมนเซปสตรอลเหมือนเดิม ซึ่งการแปลงโดเมนไปกลับนี้เองที่ทำให้ PMC ใช้เวลาในการปรับโมเดลนาน จึงไม่เป็น

ที่นิยมใช้งาน [3],[4],[13] อย่างไรก็ตาม วิธีนี้มีข้อดีที่ไม่ต้องหาคำอ่านของเสียงพูดที่เข้ามา เพราะวิธีนี้ไม่ได้ใช้ประโยชน์จากสัญญาณช่วงที่เป็นเสียงพูดในการปรับโมเดล



รูปที่ 3.1 โครงสร้างการทำงานของการรวมโมเดลขนาน [3]

เมื่อ

μ คือ เวกเตอร์ค่าเฉลี่ย

Σ คือ เมตริกซ์ค่าความแปรปรวนร่วม

Γ คือ เมตริกซ์การแปลงดิสครีตโคไซต์

3.1.2 การปรับพารามิเตอร์โมเดล

การปรับพารามิเตอร์โมเดลเป็นการปรับพารามิเตอร์ของ HMM ด้วยการใช้เสียงพูดในการปรับโมเดลโดยตรง ซึ่งเทคนิคที่ใช้ในการปรับพารามิเตอร์ของโมเดลมีอยู่ด้วยกันหลายวิธี เช่น การฝึกสอนซ้ำ [14], MAP [16] และ MLLR [7] เป็นต้น และเนื่องจากการปรับพารามิเตอร์โมเดลเป็นการปรับโมเดลตามเสียงพูดที่ใช้ในการปรับ ทำให้ต้องมีการเตรียมคำอ่านของเสียงพูด เพื่อใช้ในการระบุโมเดลที่จะมีการปรับด้วย โดยกล่าวถึงวิธีที่ใกล้เคียงกับวิธีที่วิทยานิพนธ์นี้นำเสนอ คือ การฝึกสอนซ้ำ, MAP และ MLLR ในหัวข้อ 3.2.1 หัวข้อ 3.2.2 และหัวข้อ 3.2.3 ตามลำดับ

3.1.2.1 การฝึกสอนซ้ำ

การฝึกสอนซ้ำ เป็นการปรับพารามิเตอร์ A, B และ π ของ HMM ด้วยการนำโมเดลเสียงพูดสะอาดหรือโมเดลที่ผ่านการปรับมาแล้ว มาเป็นโมเดลตั้งต้น แล้วนำข้อมูลเสียงพูดที่มีเสียงรบกวนมาใช้เป็นข้อมูลในการฝึกสอนซ้ำอีกครั้งหนึ่ง ด้วยวิธี Baum-Welch [9],[27] โมเดลที่ได้จากการปรับด้วยวิธีนี้ให้ผลการรู้จำเสียงพูดที่สูง อย่างไรก็ตาม วิธีการฝึกสอนซ้ำต้องการข้อมูลขนาดใหญ่ในการปรับโมเดล

3.1.2.2 การพิจารณาจากค่าประสบการณ์สูงสุด

การพิจารณาจากค่าประสบการณ์สูงสุด (MAP) สำหรับระบบการรู้จำเสียงพูดนั้นมีการเลือกปรับเฉพาะค่าเฉลี่ยในพารามิเตอร์ B ของ HMM เท่านั้น [16],[27] โดยใช้ข้อมูลที่มีร่วมกับความรู้เบื้องต้น (Priori Knowledge) ซึ่งได้มาจากโมเดลตั้งต้นที่ต้องการปรับ โดยเวกเตอร์ค่าเฉลี่ยของโมเดลที่ผ่านการปรับ ได้มาจากการบวกกันของเวกเตอร์ค่าเฉลี่ยที่มีการคูณด้วยค่าถ่วงน้ำหนัก ซึ่งเวกเตอร์ค่าเฉลี่ยที่ใช้ในการบวกกัน คือ เวกเตอร์ค่าเฉลี่ยของข้อมูลที่ต้องการปรับโมเดล และเวกเตอร์ค่าเฉลี่ยของโมเดลตั้งต้น ส่วนค่าถ่วงน้ำหนักคำนวณได้จากความน่าจะเป็นของค่าสังเกตที่ใช้ในการปรับโมเดลตั้งต้น และค่าถ่วงน้ำหนักของความรู้เบื้องต้น ซึ่งมีวิธีการปรับโมเดลเป็นดังนี้

กำหนดให้

k และ j คือ ลำดับสเตท

m คือ ส่วนประกอบมิกซ์เจอร์

$\overline{T}$ คือ จำนวนลำดับของค่าสังเกต

λ คือ โมเดลเสียงพูดตั้งต้น

N คือ จำนวนสเตทใน λ

O(t) คือ ค่าสังเกตที่เวลา t, $1 \le t \le \overline{T}$

μ_{jm} คือ เวกเตอร์ค่าเฉลี่ยของ λ ที่ส่วนประกอบมิกซ์เจอร์ m และสเตท j

$\hat{\mu}_{jm}$ คือ เวกเตอร์ค่าเฉลี่ยของ λ ที่ผ่านการปรับโมเดล ที่ส่วนประกอบมิกซ์เจอร์ m และ สเตท j

$\tilde{\mu}_{jm}$ คือ เวกเตอร์ค่าเฉลี่ยของค่าสังเกตที่ใช้ในการปรับโมเดล ที่ส่วนประกอบมิกซ์เจอร์ m ที่สเตท j

ρ คือ ค่าถ่วงน้ำหนักของความรู้เบื้องต้น

N_{jm} คือ ความน่าจะเป็นของค่าสังเกตที่ใช้ในการปรับโมเดล λ ที่ส่วนประกอบมิกซ์เจอร์ m และสเตท j

L_{jm} คือ ความน่าจะเป็นของ λ ที่ส่วนประกอบมิกซ์เจอร์ m และ สเตท j

a_{ij} คือ ค่าความน่าจะเป็นในการเปลี่ยนแปลงสเตทจากสเตท i ไปสเตท j

c_{jm} คือ ค่าถ่วงน้ำหนักที่ส่วนประกอบมิกซ์เจอร์ m และสเตท j

$b_{jm}(O)$ คือ ความน่าจะเป็นของค่าสังเกต ที่ส่วนประกอบมิกซ์เจอร์ m และสเตท j

$\alpha_i(t)$ คือ ความน่าจะเป็นของการเกิด O จากเวลา 1 ถึง t ของสเตท i

$\beta_i(t)$ คือ ความน่าจะเป็นของการเกิด O จากเวลา t + 1 ถึง $\overline{T}$ ของสเตท i

$$\hat{\mu}_{jm} = \frac{N_{jm}}{N_{jm}+\rho}\tilde{\mu}_{jm} + \frac{\rho}{N_{jm}+\rho}\mu_{jm} \tag{3.1}$$

เมื่อ

$$N_{jm} = \sum_{t=1}^{\overline{T}} L_{jm}(t) \tag{3.2}$$

$$\overline{\mu}_{jm} = \frac{\sum_{t=1}^{\overline{T}} L_{jm}(t)O(t)}{\sum_{t=1}^{\overline{T}} L_{jm}(t)} \tag{3.3}$$

$$L_{jm}(t) = \frac{1}{P(O|\lambda)} U_j(t) c_{jm} b_{jm}(O(t)) \beta_j(t) b_j^*(O(t)) \tag{3.4}$$

$$U_j(t) = \begin{cases} a_{1j} & \text{if} \quad t=1 \\ \sum_{k=2}^{N-1} \alpha_k(t-1)a_{kj} & ; \text{otherwise} \end{cases} \tag{3.5}$$

$$b_j^*(O(t)) = \prod b_j(O(t)) \tag{3.6}$$

ในการปรับโมเดลด้วยข้อมูลที่จำกัด MAP จะให้ผลการรู้จำเสียงพูดได้ดีกว่าวิธีการฝึกสอนซ้ำ แต่อย่างไรก็ตาม ทั้งการฝึกสอนซ้ำและ MAP จะมีการปรับเฉพาะโมเดลเสียงพูดที่มีอยู่ในข้อมูลเสียงพูดที่ใช้ปรับเท่านั้น [27] ดังนั้นอาจมีบางโมเดลเสียงพูดที่ไม่ถูกปรับได้

3.1.2.3 การถดถอยแบบเชิงเส้นตามความเป็นไปได้สูงสุด

การถดถอยแบบเชิงเส้นตามความเป็นไปได้สูงสุด (MLLR) [7] เป็นวิธีการปรับโมเดลเฉพาะค่าเฉลี่ยและความแปรปรวนร่วมในพารามิเตอร์ B ของ HMM ตามข้อมูลที่ใช้ในการปรับโมเดลนั้น โดย MLLR เป็นเทคนิคในการปรับพารามิเตอร์ของโมเดลด้วยการแปลงเชิงเส้น (Linear Transformation) เป็นผลให้พารามิเตอร์ของโมเดลตั้งต้นมีความใกล้เคียงกับข้อมูลในการปรับโมเดลมากขึ้น การแปลงเชิงเส้นหาได้จากวิธีการถดถอยแบบเชิงเส้น โดยเป็นการหาค่าของสมการ

เส้นตรงที่ทำให้เกิดความเป็นไปได้สูงสุด ที่ทำให้โมเดลที่ปรับได้มีความใกล้เคียงกับข้อมูลที่ใช้ในการปรับโมเดล ซึ่งทำให้สามารถลดความแตกต่างของผลกระทบจากเสียงรบกวนจากช่องสัญญาณและจากเสียงรบกวนแบบบวกได้

กำหนดให้

$\overline{M}$ คือ จำนวนส่วนประกอบมิกซ์เจอร์

$\xi(t)$ คือ ค่าสังเกตที่ถูกขยายที่เวลา t, $1 \le t \le \overline{T}$

μ_m คือ เวกเตอร์ค่าเฉลี่ยของโมเดลตั้งต้นของส่วนประกอบมิกซ์เจอร์ที่ m

ξ_m คือ เวกเตอร์ค่าเฉลี่ยที่ถูกขยายของส่วนประกอบมิกซ์เจอร์ที่ m

$\sum_m$ คือ เมตริกซ์ความแปรปรวนร่วมของส่วนประกอบมิกซ์เจอร์ที่ m

$\sigma^2_{m,i}$ คือ ความแปรรวนของส่วนประกอบมิกซ์เจอร์ที่ m แถวที่ i

n คือ มิติ (Dimension) ของเวกเตอร์ค่าสังเกต

W คือ เมตริกซ์การแปลงมีขนาดเท่ากับ n x (n+1)

b คือ เวกเตอร์ไบแอส (Bias)

A คือ เมตริกซ์การแปลงมีขนาดเท่ากับ n x n

w_i คือ แถวที่ i ของ W

a_i คือ เวกเตอร์แถวที่ i ของ A

c_i คือ เวกเตอร์แถวที่ i ของโคแฟคเตอร์ (Cofactors) ของ A

1) การปรับค่าเฉลี่ย

การปรับค่าเฉลี่ยของโมเดล HMM ด้วยวิธี MLLR จะให้เมตริกซ์การแปลงในการปรับค่าเฉลี่ย ดังสมการต่อไปนี้

$$\hat{\mu}_m = A\mu_m + b = W\xi_m \tag{3.6}$$

เมื่อ $\xi_m = [1 \quad \mu_m(1) \quad \mu_m(2).....\mu_m(n)]^T$ และ $W = [b \quad A]$ และสามารถหาค่า W ได้จาก

$$w_i = k_i G_i^{-1} \tag{3.7}$$

$$G_i = \sum_{m=1}^{\overline{M}} \frac{1}{\sigma^2_{m,i}} \xi_m \xi_m^T \sum_{t=1}^{\overline{T}} L_m(t) \tag{3.8}$$

$$k_i = \sum_{m=1}^{\overline{M}} \sum_{t=1}^{\overline{T}} L_m(t) \frac{1}{\sigma^2_{m,i}} O_i(t) \zeta_m^T \tag{3.9}$$

2) การปรับค่าความแปรปรวนร่วม

การปรับค่าความแปรปรวนร่วมของโมเดล HMM ด้วยวิธี MLLR จะเป็นการใช้เมตริกซ์การแปลง ในการปรับค่าความแปรปรวนร่วม ดังสมการต่อไปนี้

$$\hat{\Sigma}_m = H \Sigma_m H^T \tag{3.10}$$

$$A = H^{-1} \tag{3.11}$$

และสามารถหาค่า A ได้จาก

$$a_i = c_i G_i^{-1} \sqrt{\left(\frac{\beta}{c_i G_i^{-1} c_i^T} \right)} \tag{3.12}$$

$$c_{ij} = \mathrm{cof}(A_{ij}) \tag{3.13}$$

$$\beta = \sum_{m=1}^{\bar{M}} \sum_{t=1}^{\bar{T}} L_m(t) \tag{3.14}$$

$$G_i = \sum_{m=1}^{\bar{M}} \frac{1}{\sigma_{m,i}^2} \sum_{t=1}^{\bar{T}} L_m(t)(O(t) - \hat{\mu}_m)(O(t) - \hat{\mu}_m)^T \tag{3.15}$$

ในการปรับโมเดลด้วยข้อมูลที่จำกัด MLLR จะให้ผลการรู้จำเสียงพูดได้ดีกว่าวิธี MAP ส่วนในกรณีที่มีข้อมูลการปรับโมเดลมาก MAP จะให้ผลการรู้จำเสียงพูดที่ดีกว่า ทั้งนี้เป็นเพราะ MAP มีการปรับโมเดลอย่างเฉพาะเจาะจงในแต่ละส่วนประกอบของโมเดล ทำให้มีจำนวนส่วนประกอบที่ต้องการปรับโมเดลมากกว่าของ MLLR ที่มีการวิเคราะห์ส่วนประกอบรวมกัน จึงทำให้ MAP ต้องการข้อมูลที่มากและครบถ้วนมากกว่า MLLR ถึงจะให้ผลการรู้จำเสียงพูดได้ดีกว่า [27],[39] นอกจากนี้ MLLR ยังไม่เกิดปัญหาบางโมเดลที่ไม่ถูกปรับ เนื่องจาก MLLR มีการแบ่งแยกโมเดลออกเป็นกลุ่มๆ ซึ่งโมเดลภายในกลุ่มเดียวกันจะใช้เมตริกซ์การแปลงเดียวกัน ดังนั้นถึงแม้มีข้อมูลในการปรับโมเดลไม่ครบ แต่ MLLR ก็ยังสามารถปรับโมเดลได้ครบทุกโมเดล [27]

นอกจากนี้ ยังมีการเสนอวิธีการปรับโมเดลที่ใช้ MLLR และ MAP [39] ร่วมกัน (MLLR-MAP) โดยจะทำการปรับโมเดลด้วย MLLR ก่อน เพื่อแก้ปัญหาบางโมเดลไม่โดนปรับ แล้วก็นำโมเดลที่ได้ไปทำการปรับโมเดลด้วย MAP อีกครั้ง ซึ่งก็จะทำให้ได้ผลการรู้จำเสียงพูดสูงขึ้น

3.2 ระบบการปรับโมเดล

ระบบการปรับโมเดลในปัจจุบันสามารถแบ่งตามการสร้างโมเดลด้วยการปรับโมเดลได้เป็น 2 ระบบ ได้แก่ 1) ระบบการปรับโมเดลแบบออฟไลน์ คือ ระบบที่มีการปรับโมเดลไว้ล่วงหน้าตามข้อมูลเสียงพูดที่มีเสียงรบกวนที่เตรียมไว้ และ 2) ระบบการปรับโมเดลแบบออนไลน์ คือ ระบบที่มีการปรับโมเดลในขณะที่ใช้งานระบบรู้จำเสียงพูด โดยได้แสดงรายละเอียดของแต่ละระบบในหัวข้อ 3.3.1 และ 3.3.2 ตามลำดับ

3.2.1 ระบบการปรับโมเดลแบบออฟไลน์

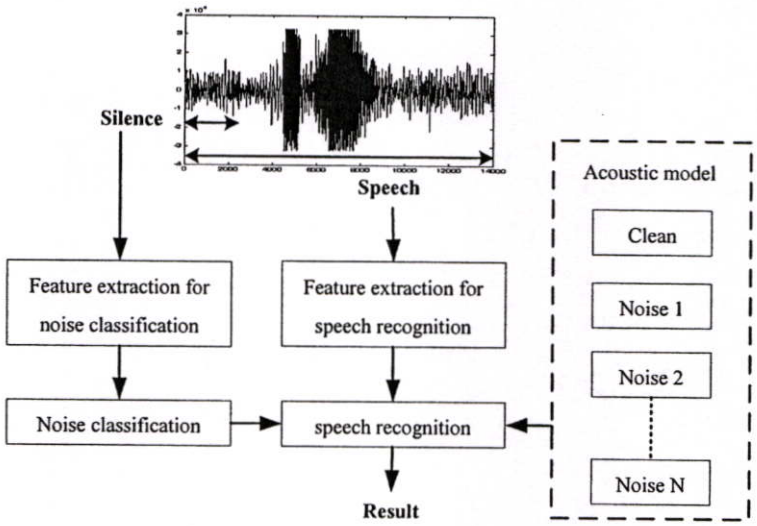
ระบบการปรับโมเดลแบบออฟไลน์ คือ ระบบที่มีการเตรียมโมเดลที่มีการปรับไว้ล่วงหน้า โดยสร้างมาจากข้อมูลเสียงพูดที่มีเสียงรบกวนที่ทราบชนิดของเสียงรบกวน และรู้คำอ่านของเสียงพูด ซึ่งเมื่อนำไปรู้จำเสียงพูดจะไม่มีการปรับโมเดลด้วยเสียงพูดที่เข้ามาอีก สำหรับวิธีการปรับโมเดลแบบออฟไลน์ที่มีใช้ในปัจจุบัน คือ วิธีการฝึกสอนแบบหลากสภาวะ และวิธีการเลือกโมเดล โดยจะแสดงรายละเอียดในหัวข้อ 3.2.1.1 และ 3.2.1.2 ตามลำดับ

3.2.1.1 วิธีการฝึกสอนแบบหลากสภาวะ

โมเดลที่ได้จากวิธีการฝึกสอนแบบหลากสภาวะ (MULTI) [14] เกิดจากการนำเสียงพูดสะอาด และเสียงพูดที่มีเสียงรบกวนหลายชนิดมาสร้างโมเดลเสียงพูดขึ้นใหม่ ด้วยการปรับโมเดลเสียงพูด โดยโมเดลเสียงพูดที่ใช้ตั้งต้น คือ โมเดลเสียงพูดสะอาด การสร้างโมเดลลักษณะนี้มีข้อดี คือ มีการจัดเก็บโมเดลเสียงพูดที่ใช้งานเท่ากับโมเดลเสียงพูดสะอาด ทำให้ไม่ต้องมีการจัดเก็บเพิ่มขึ้น แต่อย่างไรก็ตาม เนื่องจากค่าคุณลักษณะของเสียงรบกวนสำหรับการปรับโมเดลมีความแตกต่างกันมาก ทำให้โมเดลเสียงพูดที่ได้เป็นโมเดลแบบผสม ซึ่งมีข้อเสีย คือ ทำให้เสียงพูดที่มีเสียงรบกวนบางชนิดมีผลการรู้จำที่ดีขึ้น แต่บางชนิดอาจให้ผลรู้จำเสียงพูดที่ลดลงหรือดีขึ้นไม่มาก

3.2.1.2 วิธีการเลือกโมเดล

วิธีการเลือกโมเดล คือ การสร้างโมเดลเสียงพูดที่มีเสียงรบกวนเตรียมไว้ล่วงหน้า ซึ่งจะมีจำนวนเท่ากับชนิดของเสียงรบกวน หรือจำนวนเท่ากับรูปแบบการผสมเสียงรบกวนตามโครงสร้างที่ใช้งาน โดยจะเลือกโมเดลที่มีเสียงรบกวนใกล้เคียงกับเสียงรบกวนในเสียงพูดที่เข้ามามากที่สุด ดังรูปที่ 3.2



รูปที่ 3.2 โครงสร้างของการเลือกโมเดล

วิธีการนี้ต้องสร้างโมเดลขึ้น 2 แบบต่อเสียงรบกวนหนึ่งชนิด คือ แบบแรกเป็นการสร้างโมเดลเสียงรบกวน ซึ่งในที่นี้เรียกว่า Noise Cluster HMM เพื่อใช้ในการจำแนกชนิดของเสียงรบกวน (Noise Classification) และแบบที่สองเป็นการสร้างโมเดลเสียงพูดที่มีเสียงรบกวน ในที่นี้เรียกว่า Noisy Speech HMM เพื่อใช้ในการรู้จำเสียงพูด โดยวิธีการเลือกโมเดลมี 2 ขั้นตอน คือ ขั้นตอนแรกเป็นการจำแนกชนิดของเสียงรบกวน และขั้นตอนที่สองเป็นการรู้จำเสียงพูดด้วยโมเดลเสียงพูดที่มีการสร้างมาจากเสียงรบกวนชนิดเดียวกับที่จำแนกได้จากขั้นตอนแรก ทำให้โมเดลเสียงพูดที่ได้มีความใกล้เคียงกับเสียงพูดที่เข้ามามากกว่าโมเดลเสียงพูดของ MULTI ซึ่งเป็นผลให้มีผลการรู้จำเสียงพูดได้ดีกว่า

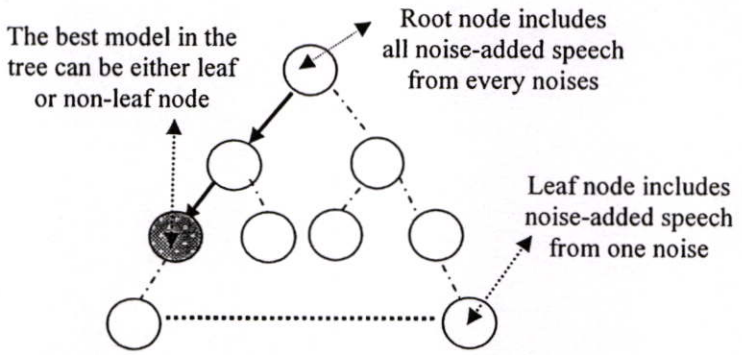
การสร้างโมเดลการจำแนกเสียงรบกวนมีอยู่ด้วยกันหลายวิธี เช่น

- การใช้ Line Spectral Frequency (LSF) ในการดึงลักษณะสำคัญ และใช้ซัพพอร์ทเวกเตอร์แมชชีน (Support Vector Machines, SVM) [21] หรือ โครงข่ายประสาทเทียมในการจำแนกเสียงรบกวน เพื่อนำมาใช้ในการเลือกโมเดลเสียงพูด [41]
- การใช้ LPC [41] หรือ MFCC [42] ในการดึงลักษณะสำคัญ และใช้ HMM ในการจำแนกเสียงรบกวน เพื่อนำมาใช้ในการเลือกโมเดลพูด
- การใช้การวิเคราะห์องค์ประกอบหลัก (Principal Component Analysis, PCA) ร่วมกับ LPC หรือ MFCC หรือ นอร์มอไลซ์ลอการิทึมสเปกตรัม (Normalized Logarithm Spectrums, NLS) ในการดึงลักษณะสำคัญ และใช้ SVM หรือโครงข่ายประสาทเทียมในการจำแนกเสียงรบกวน เพื่อนำมาใช้ในการเลือกโมเดลเสียงพูด [21]

- การใช้การวิเคราะห์องค์ประกอบหลักแบบเคอร์เนล (Kernel Principal Component Analysis, KPCA) ร่วมกับ NLS ในการดึงลักษณะสำคัญ และใช้ SVM ในการจำแนกเสียงรบกวน เพื่อนำมาใช้ในการเลือกโมเดลเสียงพูด [22],[23]

จากวิธีการจำแนกเสียงรบกวนที่กล่าวมาทั้งหมด การใช้ KPCA ร่วมกับ NLS ในการดึงลักษณะสำคัญ และการใช้ SVM ในการจำแนกเสียงรบกวน ให้ผลการจำแนกเสียงรบกวนที่มีอยู่ในชุดฝึกสอนได้ดีที่สุด นอกจากนี้ ยังให้ผลการรู้จำเสียงพูดจากโมเดลที่เลือกได้ดีกว่าวิธีอื่น แต่ก็ดีกว่าไม่มากนัก อย่างไรก็ตาม เมื่อนำมาใช้กับการทดสอบกับเสียงรบกวนที่ไม่ได้อยู่ในชุดฝึกสอน พบว่าให้ผลการรู้จำเสียงพูดที่น้อยกว่าการใช้ MFCC ในการดึงลักษณะสำคัญ และการใช้ HMM ในการจำแนกเสียงรบกวนอย่างเห็นได้ชัด ดังนั้นในการเลือกโมเดลในวิทยานิพนธ์นี้จึงเลือกใช้ MFCC ในการดึงลักษณะสำคัญ และใช้ HMM ในการจำแนกเสียงรบกวน

การเลือกโมเดลเสียงพูดด้วย MFCC ในการดึงลักษณะสำคัญ และการใช้ HMM ในการจำแนกเสียงรบกวนนี้ ได้มีการพัฒนารูปแบบการเลือกโมเดลเสียงพูดให้มีประสิทธิภาพสูงขึ้น ด้วยการเลือกโมเดลจากโมเดลที่มีการจัดกลุ่มแบบโครงสร้างต้นไม้ (MSTC) [5],[8] ดังรูปที่ 3.3 โดย MSTC มีขั้นตอนการสร้าง และขั้นตอนการค้นหาดังนี้



รูปที่ 3.3 โครงสร้างการเลือกโมเดลแบบการจัดกลุ่มแบบโครงสร้างต้นไม้

ขั้นตอนการสร้างโมเดลของ MSTC คือ

1) สร้างโมเดลเสียงรบกวน ให้ได้ครบตามจำนวนเสียงรบกวนที่เตรียมไว้ โดยจำนวนของเสียงรบกวนขึ้นอยู่กับชนิดของเสียงรบกวนและระดับ SNR
2) แบ่งกลุ่มโมเดลเสียงรบกวน โดยพิจารณาจากค่า $D(\lambda_i) = [P(O_1|\lambda_i), P(O_2|\lambda_i)..... P(O_N|\lambda_i)]$ เมื่อ O คือ ค่าสังเกตของเสียงรบกวน, N คือจำนวนค่าสังเกตของเสียงรบกวนทั้งหมดที่ใช้สร้าง MSTC และ λ_i คือ โมเดลรบกวน i ซึ่งมีขั้นตอนดังนี้
   2.1) หาค่า $D(\lambda_i)$ ของทุกเสียงรบกวน i ที่อยู่ในกลุ่มที่ต้องการแบ่งกลุ่ม

2.2) หาค่าจุดอ้างอิงสองจุด (D_1 และ D_2) โดยที่ D_1 = Mean(T) – SD(T) และ D_2 = Mean(T) + SD(T) เมื่อ Mean(T) คือ ฟังก์ชันค่าเฉลี่ย, SD(T) คือ ฟังก์ชันค่าเบี่ยนเบนมาตรฐาน และ T คือ $D(\lambda_i)$ ทั้งหมดที่อยู่ในกลุ่มที่ต้องการแบ่งกลุ่ม

2.3) แบ่งโมเดลเสียงรบกวนเป็น 2 กลุ่ม โดยมีเงื่อนไขการแบ่ง คือ

- ถ้า $\|D(\lambda_i) - D_1\| \leq \|D(\lambda_i) - D_2\|$ กำหนดให้ λ_i เป็นสมาชิกในกลุ่มแรก
- ถ้า $\|D(\lambda_i) - D_1\| > \|D(\lambda_i) - D_2\|$ กำหนดให้ λ_i เป็นสมาชิกในกลุ่มที่สอง

และ λ_i ในแต่ละกลุ่มไม่จำเป็นต้องมีจำนวนสมาชิกเท่ากัน

2.4) แบ่งกลุ่มโมเดลเสียงรบกวนเป็นทีละ 2 กลุ่มไปเรื่อยๆ จนกว่าจำนวน λ_i ที่จะถูกแบ่งในแต่ละกลุ่มเหลือตามที่กำหนด ซึ่งโครงสร้างของการแบ่งกลุ่มที่ได้ คือ โครงสร้างต้นไม้แบบไบนารี (Binary Tree Structure) และกลุ่มแต่ละกลุ่มที่แบ่งได้ในแต่ละครั้ง คือ โหนด (Node) แต่ละโหนดในโครงสร้างต้นไม้

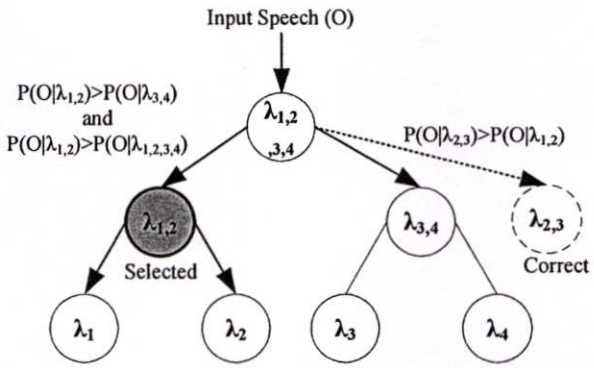
3) สร้างโมเดลของ Noise Cluster HMM เพื่อใช้ในการจำแนกเสียงรบกวน และโมเดลของ Noisy Speech HMM เพื่อใช้ในการรู้จำเสียงพูด สำหรับแต่ละโหนดตามโครงสร้างต้นไม้ในข้อที่ 2 โดยจะเห็นได้ว่าที่โหนดกลางๆ ของโครงสร้างต้นไม้นั้นเกิดจากการผสมเสียงรบกวนหลายชนิด ซึ่งทำให้ MSTC มีโมเดลเพิ่มขึ้นทั้งหมดเท่ากับ 2(N-1) โมเดล เมื่อ N คือจำนวนชนิดของเสียงรบกวนทั้งหมด

ขั้นตอนการค้นหาโมเดลของ MSTC

1) ค้นหาโมเดลที่มีความใกล้เคียงกับเสียงรบกวนในเสียงพูดที่เข้ามามากที่สุด กำหนดให้ O_{input} คือ ค่าสังเกตของเสียงพูดที่เข้ามา การค้นหาโมเดลของ Noise Cluster HMM เป็นการค้นหาแบบไบนารี โดยการเปรียบเทียบค่า $P(O_{input}|\lambda)$ ทีละคู่โมเดลที่อยู่ทางด้านซ้ายและด้านขวา หากโมเดลไหนมีค่า $P(O|\lambda)$ มากกว่าก็เลือกโมเดลนั้น แล้วค้นหาต่อไปจนกว่าโมเดลที่เลือกมีค่า $P(O|\lambda)$ น้อยกว่าโมเดลในโหนดก่อนหน้านี้ จึงหยุดการค้นหาและเลือกโมเดลที่มีค่า $P(O|\lambda)$ มากที่สุดเป็นคำตอบ
2) เลือกโมเดลที่ใช้รู้จำเสียงพูดจากโมเดลของ Noisy Speech HMM โดยเลือกโมเดลเดียวกับที่เลือกได้ในโมเดลของ Noise Cluster HMM ในขั้นตอนที่หนึ่ง

MSTC มีข้อดี คือ ให้ผลการรู้จำเสียงพูดที่มีเสียงรบกวนที่อยู่นอกชุดฝึกสอนได้ดีขึ้น เพราะมีโมเดลที่ถูกสร้างมาจากเสียงรบกวนหลายๆ ชนิด ทำให้ได้โมเดลแบบผสมที่มีความใกล้เคียงกับเสียงรบกวนที่อยู่นอกชุดฝึกสอน มากกว่าโมเดลที่สร้างมาจากเสียงรบกวนเพียงชนิดเดียว และยังใช้เวลาที่น้อยลงในการเลือกโมเดล เพราะใช้โครงสร้างต้นไม้ในการค้นหาโมเดล ทำให้มีจำนวนครั้งที่มากที่สุดในการค้นหาโมเดล คือ $2\log_2(N)$ อย่างไรก็ตาม MSTC ก็มีข้อเสีย คือ ต้องการข้อมูลที่มากเพียงพอในการสร้างโมเดลในแต่ละโหนด ซึ่งถ้ามีจำนวนข้อมูลไม่เพียงพอ MSTC ก็จะให้ผลการรู้จำเสียงพูดที่ใกล้เคียงหรือน้อยกว่าโมเดลแบบ MULTI นอกจากนี้ การใช้โครงสร้างแบบ

ต้นไม้ ทำให้สร้างโมเดลแบบผสมได้เฉพาะจากเสียงรบกวนที่มีลักษณะคล้ายกันเท่านั้น จึงทำให้สามารถสร้างโมเดลแบบผสมได้มากที่สุดเท่ากับ N-1 โมเดล ซึ่งไม่ครอบคลุมเสียงรบกวนทั้งหมดที่เป็นไปได้ ทำให้เกิดเป็นข้อจำกัดในการมีโมเดลที่เกิดจากการผสมของเสียงรบกวนต่างชนิดกัน ดังรูปที่ 3.4

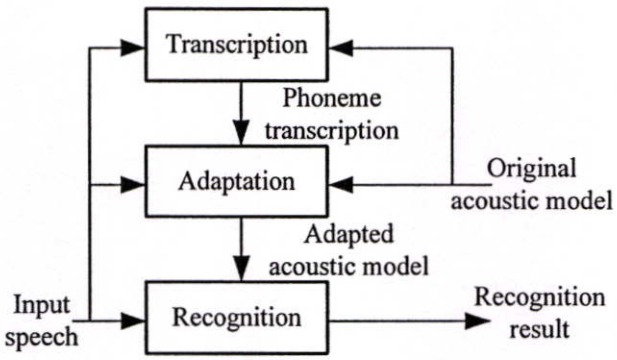


รูปที่ 3.4 ปัญหาข้อจำกัดของการเลือกโมเดลแบบการจัดกลุ่มแบบโครงสร้างต้นไม้

เมื่อ λ_i คือ โมเดลที่สร้างมาจากเสียงพูดที่มีเสียงรบกวน i และ $\lambda_{i,j}$ คือ โมเดลแบบผสมที่สร้างมาจากเสียงพูดที่มีเสียงรบกวน i และเสียงรบกวน j จากรูปที่ 3.4 แสดงให้เห็นว่าโมเดลแบบผสมที่ให้ผลดีที่สุดไม่ได้ถูกสร้างขึ้น เนื่องจากเสียงรบกวนที่ต้องนำมาผสมกันถูกแยกออกจากกัน อาจทำให้ไม่มีโอกาสที่จะได้โมเดลที่สามารถรองรับเสียงพูดที่เข้ามาใหม่ได้ อย่างไรก็ตาม แม้ว่าสามารถหาโครงสร้างที่มีโมเดลแบบผสมได้มากกว่านี้ ก็ยังต้องแลกกับพื้นที่ในการเก็บโมเดลแบบผสมของ Noise Cluster HMM และ Noisy Speech HMM ที่เพิ่มขึ้นตามไปด้วย ทำให้สิ้นเปลืองหน่วยความจำ และไม่สามารถนำไปใช้งานจริงได้ ซึ่งปัญหานี้เป็นปัญหาที่สำคัญของวิธีการเลือกโมเดลที่ยังไม่สามารถแก้ปัญหาได้ นอกจากนี้ แม้ว่า MSTC สามารถแก้ไขเสียงรบกวนที่ไม่ได้อยู่ในชุดฝึกสอนได้ดีขึ้น แต่ในความเป็นจริงแล้วไม่สามารถหาเสียงรบกวนได้ครอบคลุมทั้งหมด เป็นผลให้ระบบการปรับแบบออฟไลน์ ยังคงมีปัญหากับเสียงรบกวนที่ไม่ได้อยู่ในชุดฝึกสอน

3.2.2 ระบบการปรับโมเดลแบบออนไลน์

ระบบการปรับแบบออนไลน์ คือ ระบบที่มีการปรับโมเดลตามผู้พูดและเสียงรบกวนจากเสียงพูดที่เข้ามาในแต่ละครั้ง โดยเทคนิคที่นิยมใช้ในการปรับโมเดล คือ MLLR เพราะใช้ข้อมูลที่มีขนาดเล็กในการปรับโมเดล และใช้เวลาในการปรับโมเดลน้อย แต่การใช้ MLLR ต้องมีส่วนที่ใช้หาคำอ่านแบบอัตโนมัติด้วย ซึ่งมีโครงสร้างการทำงาน ดังรูป 3.5 โดยการปรับโมเดลแบบนี้มีข้อดี คือ สามารถแก้ปัญหาเสียงรบกวนที่ไม่อยู่ในชุดฝึกสอนได้ดีกว่าระบบการปรับโมเดลแบบออฟไลน์ เพราะมีการปรับโมเดลตามเสียงรบกวนที่เข้ามาในขณะนั้น ซึ่งวิธีที่นิยมใช้ในการปรับโมเดลแบบออนไลน์ คือ PLT ซึ่งจะกล่าวถึงรายละเอียดในหัวข้อถัดไป

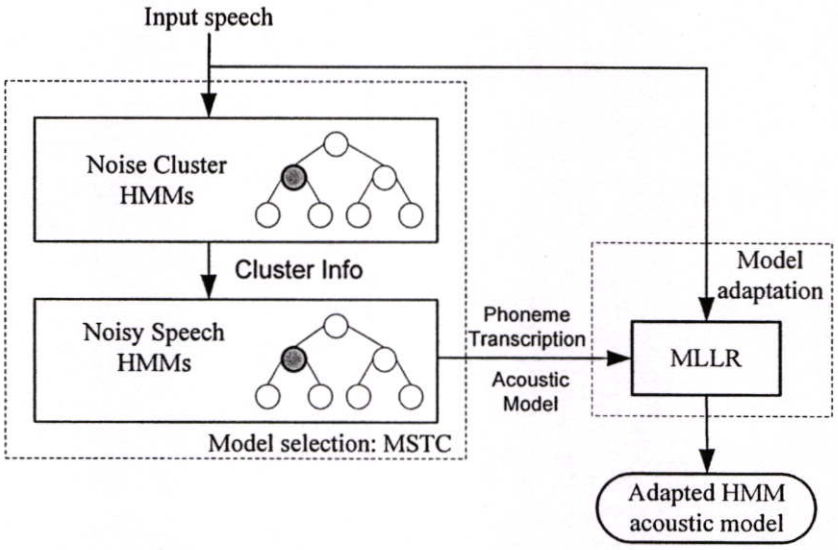


รูปที่ 3.5 ระบบการปรับโมเดลแบบออนไลน์

3.2.2.1 วิธีการแปลงเชิงเส้นแบบแบ่งส่วน

วิธีการแปลงเชิงเส้นแบบแบ่งส่วน (PLT) [5] เป็นการแก้ปัญหาของเสียงรบกวนที่มีระดับ SNR และเสปคตรอลที่หลากหลาย ทำให้เสียงรบกวนมีความไม่เป็นเชิงเส้นสูง ซึ่งทำให้โมเดลที่มีการปรับแบบ MLLR ที่ใช้หลักการของการแปลงเชิงเส้น ไม่สามารถปรับโมเดลตั้งต้นให้ได้ผลดีกับเสียงรบกวนทุกชนิด โดยขึ้นอยู่กับความแตกต่างระหว่างเสียงรบกวนที่ใช้ในการสร้างโมเดลตั้งต้นกับเสียงรบกวนที่ใช้ในการปรับ ส่งผลให้การรู้จำเสียงพูดยังไม่ดีเท่าที่ควร PLT ได้เสนอการใช้โมเดลตามการแบ่งกลุ่มโมเดลเสียงพูดที่มีเสียงรบกวน และเลือกใช้โมเดลเสียงพูดที่มีเสียงรบกวนใกล้เคียงกับเสียงรบกวนในเสียงพูดที่เข้ามามากที่สุด เพื่อใช้เป็นโมเดลตั้งต้นในการปรับโมเดล ทำให้สามารถลดความแตกต่างระหว่างเสียงรบกวนที่ใช้ในการสร้างโมเดลตั้งต้น และเสียงรบกวนที่ใช้ในการปรับโมเดลให้น้อยลง เป็นผลให้การรู้จำเสียงพูดดีขึ้น เห็นได้ว่า PLT สามารถแก้ปัญหาความหลากหลายของเสียงรบกวนของการปรับโมเดลแบบ MLLR ได้ โดย PLT มีโครงสร้างการทำงานดังรูปที่ 3.6 และมีขั้นตอนในการปรับโมเดล 3 ขั้นตอน คือ

1) การค้นหาโมเดลเสียงพูดที่มีเสียงรบกวนใกล้เคียงกับเสียงรบกวนในเสียงพูดที่เข้ามาให้ได้มากที่สุด โดยขั้นตอนนี้ใช้การเลือกโมเดลแบบ MSTC เพื่อให้ได้โมเดลตั้งต้นของการปรับโมเดล
2) การหาคำอ่านของเสียงพูดที่เข้ามา ด้วยการนำเสียงพูดที่เข้ามาไปรู้จำเสียงพูด ด้วยโมเดลที่ใช้รู้จำเสียงพูด คือ โมเดลเสียงพูดที่เลือกได้จากขั้นตอนที่หนึ่ง
3) การปรับโมเดลให้มีความใกล้เคียงเสียงรบกวนที่มีอยู่ในเสียงพูดมากยิ่งขึ้น ด้วยการปรับโมเดลที่ได้จากขั้นตอนที่หนึ่ง โดยใช้เสียงพูดที่เข้ามาและใช้คำอ่านที่ได้จากขั้นตอนที่สอง ซึ่งในขั้นตอนนี้ใช้ MLLR ในการปรับโมเดล ซึ่งโมเดลที่ได้จากขั้นตอนนี้จะถูกใช้ในการรู้จำเสียงพูดต่อไป



รูปที่ 3.6 โครงสร้างของการแปลงเชิงเส้นแบบแบ่งส่วน

ระบบการปรับแบบออนไลน์ด้วยวิธี PLT มีข้อดี คือ สามารถรองรับเสียงรบกวนที่ไม่อยู่ในชุดฝึกสอนได้ดีกว่าระบบการปรับแบบออฟไลน์ เพราะระบบการปรับแบบออนไลน์มีการใช้เสียงพูดที่เข้ามาในการปรับโมเดล จึงทำให้โมเดลที่ได้มีความใกล้เคียงกับเสียงรบกวนในเสียงพูดที่เข้ามามากกว่า ซึ่งทำให้มีผลการรู้จำเสียงพูดได้ดีขึ้นนั่นเอง ดังนั้นในวิทยานิพนธ์นี้จึงเลือกใช้ระบบการปรับโมเดลเสียงพูดแบบออนไลน์ด้วยวิธี PLT ในการเปรียบเทียบประสิทธิภาพกับวิธีที่นำเสนอ

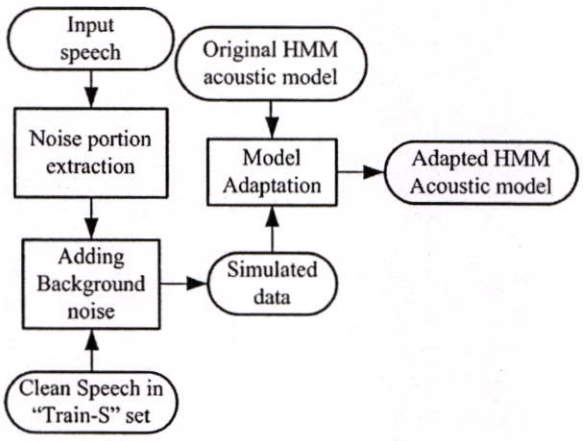
อย่างไรก็ตาม ระบบการปรับโมเดลเสียงพูดแบบออนไลน์ด้วยวิธี PLT ยังคงมีปัญหาสำคัญที่ไม่สามารถแก้ได้ 3 ปัญหาดังนี้ ปัญหาที่ 1 คือ การไม่ทราบคำอ่านของเสียงพูดที่เข้ามา ทำให้ต้องมีการหาคำอ่านของเสียงพูดแบบอัตโนมัติ เพื่อนำคำอ่านมาใช้ในการปรับโมเดลแบบ MLLR ดังนั้นถ้าใช้คำอ่านที่มีความถูกต้องต่ำ ย่อมทำให้ปรับโมเดลไม่ตรงกับความเป็นจริง ทำให้ผลการรู้จำเสียงพูดลดลง และปัญหาที่ 2 คือ ปริมาณข้อมูลเสียงพูดที่ใช้ในการปรับโมเดลมีไม่เพียงพอ ถึงแม้ว่าวิธี MLLR จะใช้ข้อมูลในการปรับโมเดลน้อยกว่าวิธีอื่นๆ แล้วก็ตาม แต่ก็ยังต้องการข้อมูลที่มากเพียงพอด้วยจึงจะทำให้โมเดลที่ปรับได้มีคุณภาพ และปัญหาที่ 3 คือ ปัญหาที่เกิดจากการเลือกโมเดลแบบ MSTC ที่มีการสร้างโมเดลแบบผสม โดยพิจารณาจากเสียงรบกวนที่มีลักษณะคล้ายกันเท่านั้น ทำให้มีโมเดลที่สามารถรองรับเสียงรบกวนที่ไม่มีอยู่ในชุดฝึกสอนได้น้อยลง

วิทยานิพนธ์นี้จึงนำเสนอแนวคิดใหม่ ในการปรับโมเดลแบบออนไลน์ด้วยวิธีการปรับโมเดลด้วยข้อมูลจำลอง ร่วมกับการประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดล โดยบทที่ 4 กล่าวถึงการปรับโมเดลด้วยข้อมูลจำลอง และบทที่ 5 กล่าวถึง การประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดล

บทที่ 4
การปรับโมเดลด้วยข้อมูลจำลอง

ปัญหาของระบบการปรับโมเดลแบบออนไลน์ คือ การไม่ทราบคำอ่านของเสียงพูดที่เข้ามา ทำให้ต้องมีการหาคำอ่านแบบอัตโนมัติ ด้วยการนำเสียงพูดที่เข้ามามารู้จำเสียงพูดก่อน เพื่อให้ได้คำอ่านมาใช้เลือกปรับโมเดลได้ถูกต้อง ดังนั้นถ้าคำอ่านของเสียงพูดที่เข้ามามีความถูกต้องต่ำ ย่อมทำให้ปรับโมเดลผิดได้ เช่น เสียงพูดที่เข้ามามีคำอ่านที่ถูกต้อง คือ n a t^ (นัด) แต่คำอ่านที่ได้จากการรู้จำเสียงพูดแล้วนำไปใช้ปรับโมเดล คือ n a p^ (นับ) เห็นได้ว่ามีการปรับโมเดลผิดอยู่ 1 โมเดล คือ โมเดลของหน่วยเสียง p^ เป็นผลให้โมเดลที่ได้มีผลการรู้จำเสียงพูดที่ไม่ดีนัก นอกจากนี้ ปริมาณข้อมูลเสียงพูดที่ใช้ในการปรับโมเดลยังต้องมากเพียงพอ เพราะแม้ว่าวิธี MLLR จะใช้ข้อมูลในการปรับโมเดลน้อยกว่าวิธีอื่นๆ แต่ก็ต้องมีข้อมูลที่มากพอจึงจะทำให้โมเดลที่ปรับได้มีคุณภาพดี [39]

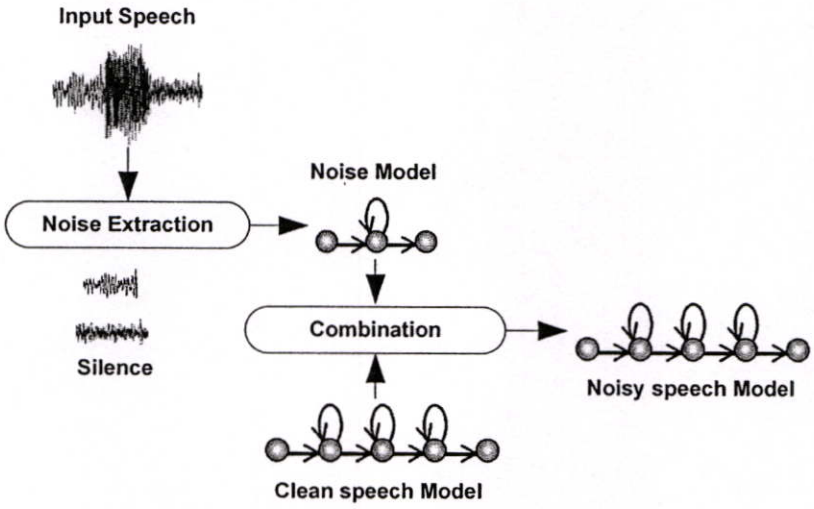
วิทยานิพนธ์นี้นำเสนอวิธีการปรับโมเดลด้วยข้อมูลจำลอง [17],[18],[19] ซึ่งเป็นวิธีการใหม่ มีโครงสร้างดังรูปที่ 4.1 วิธีการนี้เป็นการสร้างข้อมูลจำลอง (Simulated-data) ของเสียงพูดที่มีเสียงรบกวน เพื่อใช้ในการปรับโมเดล โดยข้อมูลที่จำลองขึ้นมาเป็นเสียงพูดที่มีเสียงรบกวนแบบบวก ที่เกิดจากการนำเสียงรบกวนพื้นหลัง (Background Noise Addition) ที่ได้มาจากการดึงเสียงรบกวน (Noise Portion Extraction) ของเสียงพูดที่เข้ามามาบวกกับเสียงพูดสะอาดที่ทราบคำอ่านและมีการเตรียมไว้ล่วงหน้า ซึ่งในที่นี้เรียกเสียงพูดสะอาดที่ใช้ในการสร้างข้อมูลจำลองว่า “Train-S”



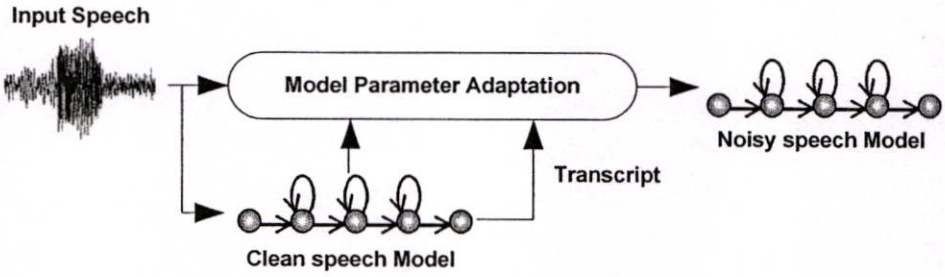
รูปที่ 4.1 กระบวนการปรับโมเดลด้วยข้อมูลจำลอง

การปรับโมเดลด้วยข้อมูลจำลองสามารถแก้ปัญหาการไม่ทราบคำอ่านของเสียงพูดที่เข้ามาได้ โดยการใช้ข้อมูลจำลองเสียงพูดที่ทราบคำอ่าน และสามารถแก้ปัญหาเสียงพูดที่ใช้ในการปรับโมเดลที่มีจำนวนน้อยได้ โดยสามารถเพิ่มจำนวนเสียงพูดใน Train-S ให้ได้จำนวนที่เพียงพอต่อ

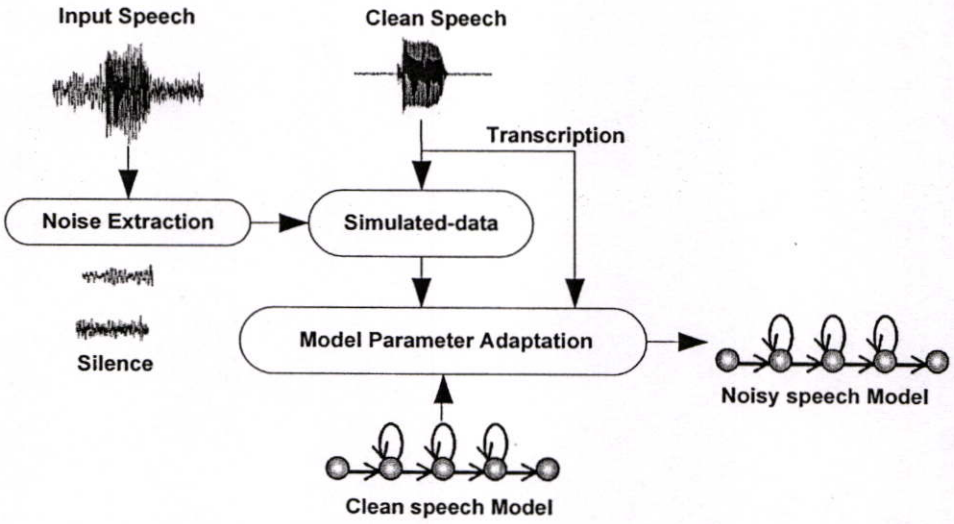
ความต้องการ ฉะนั้นเห็นได้ว่าวิธีการที่นำเสนอสามารถแก้ปัญหาพื้นฐานของระบบการปรับโมเดลแบบออนไลน์แบบ MLLR ที่นิยมใช้ในปัจจุบันได้



(a) Model Composition and Decomposition



(b) Model Parameter Adaptation



(c) Propose Model Adaptation

รูปที่ 4.2 การเปรียบเทียบการปรับโมเดลระหว่างวิธีแบบอื่น และวิธีในวิทยานิพนธ์นี้

ความแตกต่างของการปรับโมเดลในวิทยานิพนธ์นี้ และการปรับโมเดลแบบทั่วไป เป็นดังรูปที่ 4.2 โดยรูปที่ 4.2 (a) เป็นการปรับโมเดลแบบการรวมและการแยกโมเดล วิธีนี้เป็นการผสมโมเดลเสียงพูดสะอาดและโมเดลเสียงรบกวนที่สร้างจากเสียงรบกวนของเสียงพูดที่เข้ามา ซึ่งการปรับโมเดลวิธีนี้มีข้อดี คือ ไม่ต้องใช้คำอ่านในการระบุโมเดลที่ต้องการปรับ แต่ยังต้องมีการหาตำแหน่งส่วนที่เป็นเสียงพูดและส่วนที่เป็นเสียงเงียบของเสียงพูดที่เข้ามา เพื่อดึงเฉพาะเสียงรบกวนที่อยู่ในส่วนที่เป็นเสียงเงียบออกมา อย่างไรก็ตามการปรับโมเดลด้วยวิธีนี้ใช้เวลาในการปรับนาน [4],[13] ทำให้ไม่เหมาะกับระบบการปรับโมเดลแบบออนไลน์ ส่วนรูปที่ 4.2 (b) เป็นการปรับโมเดลแบบการปรับพารามิเตอร์โมเดล ซึ่งเป็นการปรับพารามิเตอร์โมเดลด้วยเสียงพูดที่เข้ามาโดยตรง วิธีนี้จำเป็นต้องใช้คำอ่านในการระบุโมเดล แล้วนำโมเดลที่ระบุในคำอ่านไปปรับโมเดลด้วยเสียงพูดที่เข้ามา การปรับโมเดลด้วยวิธีนี้ใช้เวลาไม่มาก และให้ผลการรู้จำเสียงพูดที่ดีกว่าการปรับโมเดลแบบการรวมและการแยกโมเดล ซึ่งทำให้เหมาะกับระบบการปรับโมเดลแบบออนไลน์มากกว่า อย่างไรก็ตาม วิธีการนี้ต้องการคำอ่านที่ถูกต้อง เพื่อใช้ในการปรับโมเดลให้ถูกต้อง และข้อมูลที่ใช้ในการปรับโมเดลที่เพียงพอ รูปที่ 4.2 (c) เป็นการปรับโมเดลที่วิทยานิพนธ์นี้นำเสนอ โดยการผสมเสียงรบกวนที่ได้จากส่วนที่เป็นเสียงเงียบของเสียงพูดที่เข้ามา และเสียงพูดสะอาดที่เตรียมไว้ล่วงหน้าเข้าด้วยกัน ทำให้ได้เสียงพูดที่มีเสียงรบกวนใกล้เคียงกับเสียงรบกวนในเสียงพูดที่เข้ามา และเป็นเสียงพูดที่ทราบคำอ่าน ซึ่งเรียกเสียงพูดที่มีเสียงรบกวนที่ได้จากการผสมนี้ว่า ข้อมูลจำลอง จากนั้นนำข้อมูลจำลองที่ได้ไปปรับโมเดลแบบการปรับพารามิเตอร์โมเดล เช่น วิธี MAP และ วิธี MLLR เป็นต้น เห็นได้ว่าวิธีที่นำเสนอไม่จำเป็นต้องใช้คำอ่านของเสียงพูดที่เข้ามาในการปรับโมเดล และสามารถสร้างข้อมูลให้เพียงพอต่อการปรับโมเดลแบบการปรับพารามิเตอร์โมเดลได้ วิธีที่นำเสนอเป็นการรวมเอาข้อดีของการปรับพารามิเตอร์โมเดล ในเรื่องความเร็ว และโมเดลที่ปรับได้นี้ให้ผลการรู้จำเสียงพูดที่ดี กับข้อดีของการรวมและการแยกโมเดล ในเรื่องการไม่ต้องใช้คำอ่านในการปรับโมเดล ทำให้ไม่เกิดปัญหาการปรับผิดโมเดล จากที่กล่าวมานี้แสดงให้เห็นว่า วิธีที่นำเสนอมีความเหมาะสมกับระบบการปรับโมเดลแบบออนไลน์มากกว่าวิธีที่มีการนำเสนอมาก่อนหน้านี้

ในบทนี้ อธิบายถึงการปรับโมเดลด้วยข้อมูลจำลอง ซึ่งเป็นวิธีการที่ได้พัฒนาขึ้นใหม่อย่างละเอียด โดยประกอบด้วย หัวข้อ 4.1 การคัดเลือกเสียงพูดสะอาด, หัวข้อ 4.2 การดึงส่วนเสียงรบกวน, หัวข้อ 4.3 การบวกเสียงรบกวนพื้นหลัง, หัวข้อ 4.4 รูปแบบการใช้งานข้อมูลจำลอง, หัวข้อ 4.5 การเลือกผลการรู้จำเสียงพูด (Recognition Result Selection) และหัวข้อ 4.6 การประยุกต์ใช้การปรับโมเดลด้วยข้อมูลจำลองกับเทคนิคอื่นๆ

4.1 การคัดเลือกเสียงพูดสะอาด

การคัดเลือกเสียงพูดสะอาด เพื่อใช้เป็นเสียงพูดตั้งต้นในการจำลองข้อมูล โดยการคัดเลือกเสียงพูดสะอาด ซึ่งในที่นี้เรียกว่า "Train-S" มีขั้นตอนดังนี้

1. เลือกคำที่มีเสียงพูดสะอาด และเมื่อนำไปรู้จำเสียงพูดด้วยโมเดลเสียงพูดแบบสะอาดแล้วได้ผลการรู้จำเสียงพูดที่ถูกต้อง
2. เลือกคำที่ได้จากขั้นตอนแรก ให้มีจำนวนคำที่น้อยที่สุด และครอบคลุมจำนวนหน่วยเสียงของการสร้างโมเดลเสียงพูด สำหรับระบบรู้จำเสียงพูดภาษาไทยมีจำนวนหน่วยเสียงที่เป็นโมเดลเสียงพูดทั้งสิ้น 76 หน่วยเสียง (รวมหน่วยเสียงของเสียงเงียบด้วย) ดังตารางที่ 4.1

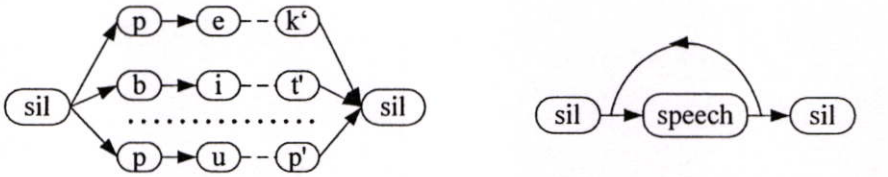
ตารางที่ 4.1 หน่วยเสียงภาษาไทยซึ่งอิงกับสัทอักษรสากล (International Phonetic Alphabet, IPA) [43]

Type		IPA symbol
Initial consonants (C_i)	Single	p, t, ɕ, k, ʔ, p^h, t^h, $ɕ^h$, k^h,
	Cluster	Pr, p^hr, tr, kr, k^hr, pl, p^hl, t^hr, kl, k^hl, kw, k^hw, br, bl, fr, fl, dr
Vowels (V)	Single	a, aː, i, iː, ɯ, ɯː, u, uː,
	Diphthong	ia, iːa, ɯa, ɯːa, ua, uːa
Final consonants (C_f)		p^, t^, k^, n^, m^, ŋ^, j^, ʋ^, f^, l^, s^, t$ɕ^h$^, ʔ^
Silence		Sil

4.2 การดึงส่วนเสียงรบกวน

การดึงส่วนเสียงรบกวน คือ การดึงเสียงรบกวนออกมาจากเสียงพูดที่เข้ามา โดยนิยามส่วนของเสียงรบกวนที่ต้องการ คือ ส่วนเสียงเงียบด้านหน้าและด้านหลังของเสียงพูดที่เข้ามา สาเหตุที่ต้องนิยามเช่นนี้เป็นเพราะการแยก (Separation) เสียงรบกวนและเสียงพูดสะอาดให้ออกจากกันอย่างสมบูรณ์ทำได้ค่อนข้างยาก แต่การแบ่งแยกส่วนระหว่างส่วนที่เป็นเสียงพูดและส่วนที่เป็นเสียงเงียบทำได้ง่ายกว่า และแน่ใจมากกว่าว่าส่วนที่ตัดออกมีเพียงเสียงรบกวนเท่านั้นไม่ได้มีเสียงพูดเข้ามาด้วย โดยมีข้อจำกัดว่าเสียงรบกวนที่เกิดขึ้นในส่วนที่เป็นเสียงเงียบ และส่วนที่เป็นเสียงพูดจะต้องเป็นเสียงรบกวนชนิดเดียวกัน

สำหรับการหาส่วนที่เป็นเสียงเงียบ ในวิทยานิพนธ์นี้ใช้ MFCC ในการดึงลักษณะสำคัญ และใช้ HMM ในการแบ่งแยกตำแหน่ง โดยโครงสร้างของ HMM ที่ใช้ในการแบ่งแยกมีอยู่ด้วยกัน 2 รูปแบบ ดังรูปที่ 4.3 กล่าวคือแบบแรกมีโครงสร้างแบบการอิงกับหน่วยเสียง (Phoneme-based) ซึ่งประกอบด้วยโมเดลหน่วยเสียงที่ใช้ในการรู้จำเสียงพูดและโมเดลเสียงเงียบ สำหรับภาษาไทยมีจำนวนโมเดลเท่ากับ 76 โมเดล สิ่งที่ได้จากโครงสร้างนี้ คือ ผลการรู้จำเสียงพูด ซึ่งทำให้ทราบทั้งตำแหน่งของเสียงเงียบและคำอ่านของเสียงพูดที่เข้ามา ดังรูป 4.3.(a) แบบที่สองมีโครงสร้างแบบการหาส่วนที่เป็นเสียงพูดและส่วนที่ไม่เป็นเสียงพูด โดยโครงสร้างนี้มีจำนวนโมเดลเท่ากับ 2 โมเดล ประกอบด้วยโมเดลเสียงพูดและโมเดลเสียงเงียบ สิ่งที่ได้จากโครงสร้างนี้จะมีแค่ตำแหน่งของเสียงเงียบเท่านั้น ดังรูป 4.3.(b) จากโครงสร้างทั้งสองแบบแสดงให้เห็นว่า โครงสร้างแบบที่สองใช้เวลาในการแบ่งแยกตำแหน่งน้อยกว่าโครงสร้างแบบที่หนึ่ง เพราะมีจำนวนโมเดลน้อยกว่า แต่ในการทดลอง [17] พบว่าโครงสร้างแบบที่หนึ่งให้ผลการแบ่งแยกตำแหน่งได้ดีกว่าโครงสร้างแบบที่สอง ทำให้เสียงรบกวนของข้อมูลจำลองที่ได้จากโครงสร้างแบบที่หนึ่ง มีความใกล้เคียงกับเสียงรบกวนของเสียงพูดที่เข้ามามากกว่าโครงสร้งแบบที่สอง เป็นผลให้การปรับโมเดลด้วยข้อมูลจำลองที่ใช้โครงสร้างแบบที่หนึ่งมีผลการรู้จำเสียงพูดมากกว่าการใช้โครงสร้างแบบที่สอง



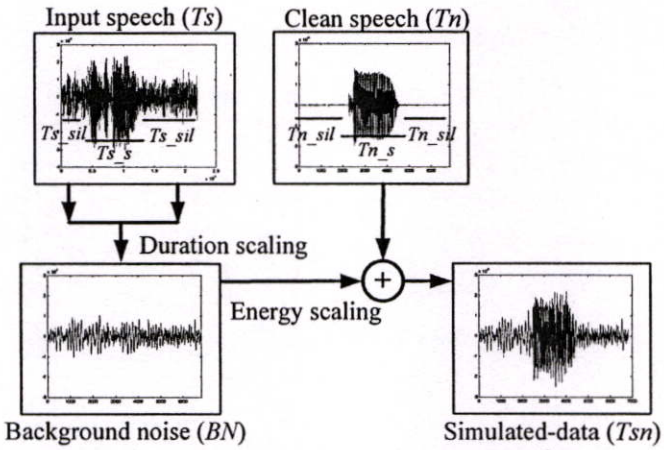
(a) Phoneme-base HMM (b) Speech/non-speech Detection

รูปที่ 4.3 รูปแบบของการดึงส่วนของเสียงรบกวน

4.3 การบวกเสียงรบกวนพื้นหลัง

การบวกเสียงรบกวนพื้นหลัง คือ การนำเสียงรบกวนที่ดึงออกมาได้มารวมกับเสียงพูดสะอาดจาก Train-S ซึ่งการบวกเสียงรบกวนพื้นหลังมีปัญหาอยู่ด้วยกัน 2 ปัญหา ปัญหาแรก คือ การทำให้ความยาวเสียงรบกวนที่ดึงออกมา มีความยาวเท่ากับความยาวเสียงสะอาดจาก Train-S กล่าวคือโดยปกติแล้วความยาวของเสียงรบกวนที่ดึงออกมาได้นั้น ยาวน้อยกว่าเสียงสะอาดจาก Train-S และปัญหาที่สอง คือ การทำให้ระดับ SNR ของข้อมูลจำลองเท่ากับระดับ SNR ของเสียงพูดที่เข้ามา เนื่องจากความดังของเสียงพูดที่เข้ามาไม่เท่ากับความดังของเสียงพูดสะอาดจาก Train-S ทำให้ถ้าบวกเสียงรบกวนเข้าไปตรงๆ ก็จะทำให้ได้เสียงพูดที่มีเสียงรบกวนเดียวกันแต่มีระดับ SNR ที่แตกต่างกัน และการที่ระดับ SNR ที่ได้ไม่ตรงกันก็ส่งผลให้โมเดลที่ปรับได้มีผลการรู้จำเสียงพูดลดลงไปด้วย ดังนั้นในวิทยานิพนธ์นี้จึงได้นำเสนอวิธีการแก้ปัญหาทั้งสอง

การแก้ปัญหาความยาวของเสียงรบกวนพื้นหลัง (BN) และเสียงสะอาด ทำได้โดยการนำเสียงรบกวนส่วนหน้าและเสียงรบกวนส่วนหลังของเสียงพูดที่เข้ามา มาต่อเรียงสลับไปมาระหว่างเสียงรบกวนส่วนหน้าและเสียงรบกวนส่วนหลังไปเรื่อยๆ จนได้ความยาวเท่ากับความยาวของเสียงพูดสะอาด ซึ่งเรียกเสียงรบกวนนี้ว่า "เสียงรบกวนพื้นหลัง" ดังรูปที่ 4.4 และเรียกกระบวนการนี้ว่า "การสเกลความยาว (Duration Scaling)" วิธีการต่อเสียงรบกวนส่วนหน้าและส่วนหลังในกระบวนการสเกลความยาวเป็นการต่อกันแบบธรรมดา อาจทำให้เสปคตรอลของเสียงรบกวนเปลี่ยนไปได้ อย่างไรก็ตาม ในงานวิทยานิพนธ์นี้ยังไม่มีการทำให้เสปคตรอลเรียบ (Spectral Smoothing) เพราะต้องการลดเวลาในการประมวลผล และจากการทดลองเบื้องต้นในการนำเสียงรบกวนแบบเดียวกับเสียงรบกวนในเสียงพูดที่เข้ามามาใช้ในการสร้างข้อมูลจำลอง แล้วนำข้อมูลที่ได้ไปใช้ในการปรับโมเดล พบว่าให้ผลการรู้จำเสียงพูดที่ใกล้เคียงกับวิธีที่วิทยานิพนธ์นี้นำเสนอ



รูปที่ 4.4 กระบวนการบวกเสียงรบกวนพื้นหลัง

สำหรับปัญหาเรื่องค่าความดังของเสียงพูดที่เข้ามา และค่าความดังของเสียงพูดสะอาดที่เตรียมไว้ล่วงหน้ามีค่าไม่เท่ากัน ทำให้ไม่สามารถบวกเสียงรบกวนพื้นหลังเข้ากับเสียงพูดสะอาดได้แบบตรงๆ เพราะจะทำให้ระดับ SNR ของข้อมูลจำลองและระดับ SNR เสียงพูดที่เข้ามามีค่าไม่เท่ากัน วิทยานิพนธ์นี้จึงได้นำเสนอวิธีการคำนวณค่าการสเกลเสียง (scale_speech) อย่างง่าย เพื่อใช้ในการปรับค่าของเสียงรบกวนพื้นหลังให้มีค่าที่ทำให้ระดับ SNR ของข้อมูลจำลองมีค่าใกล้เคียงกับระดับ SNR ของเสียงพูดที่เข้ามา โดยเรียกกระบวนการนี้ว่า "การสเกลพลังงาน (Energy Scaling)"

โดยกำหนดให้

Ts คือ เสียงพูดที่เข้ามา

Tn คือ เสียงพูดสะอาดใน Train-S ซึ่งเป็นเสียงพูดที่รู้คำอ่าน

Tsn คือ ข้อมูลจำลอง

Ln คือ ความยาวของ Tn_s_i

Ls คือ ความยาวของ Ts_s_i

Tn_s_i ; (i=1,2,….,Ln) คือ ส่วนที่เป็นเสียงพูดของ Tn

Ts_s_i; (i=1,2….,Ls) คือ ส่วนที่เป็นเสียงพูดของ Ts

Tsn_s_i; (i=1,2 … Ln) คือ ส่วนที่เป็นเสียงพูดของ Tns

Tn_sil คือ ส่วนที่เป็นเสียงเงียบของ Tn

Ts_sil คือ ส่วนที่เป็นเสียงเงียบของ Ts

Tsn_sil คือ ส่วนที่เป็นเสียงเงียบของ Tsn

BN คือ เสียงรบกวนพื้นหลัง

การหา Tsn_sil ที่ทำให้สมการ SNR ของ Ts เท่ากับ SNR ของ Tsn เป็นจริง ซึ่งสามารถเขียนเป็นสมการได้ คือ

$$20\log\left(\frac{E(Ts\_s_i)}{E(Ts\_sil)}\right) = 20\log\left(\frac{E(Tsn\_s_i)}{E(Tsn\_sil)}\right) \tag{4.1}$$

$$\frac{E(Ts\_s_i)}{E(Ts\_sil)} = \frac{E(Tsn\_s_i)}{E(Tsn\_sil)} \tag{4.2}$$

โดย E(T) คือ ฟังก์ชันของค่าเฉลี่ยพลังงานของสัญญาณ T และพิจารณาการแก้ปัญหาของเสียงรบกวนแบบบวก ดังนั้น

$$Ts = Ts\_s_i + Ts\_sil \tag{4.3}$$

$$Tsn = Tsn\_s_i + Tsn\_sil \tag{4.4}$$

เมื่อ

Tsn_sil คือ BN ที่สร้างจาก Ts_sil ที่ได้จาก Ts

Tsn_sil คือ Tn_s_i ที่ได้มาจาก Tn

และเนื่องจาก Tn เป็นเสียงพูดสะอาด ดังนั้นจึงไม่มีเสียงรบกวนผสมอยู่เลย ทำให้สมการ Tsn จะมีค่าเท่ากับ

$$Tsn = Tn\_s_i + BN \tag{4.5}$$

ดังนั้นเมื่อแทนด้วยสมการ (4.3) และ (4.5) ลงในสมการที่ (4.2) จะได้เป็น

$$\frac{E(Ts\_s_i)}{E(Ts\_sil)} = \frac{E(Tn\_s_i)}{E(BN)} \tag{4.6}$$

$$\frac{E(Tn\_s_i)}{E(Ts\_s_i)} = \frac{E(BN)}{E(Ts\_sil)} \tag{4.7}$$

$$E(BN) = \frac{E(Tn\_s_i)}{E(Ts\_s_i)} \times E(Ts\_sil) \tag{4.8}$$

กำหนดให้

$$scale\_factor = \frac{E(Tn\_s_i)}{E(Ts\_s_i)} \tag{4.9}$$

$$= \frac{\dfrac{\sum_{i=1}^{Ln} \left|Tn\_s_i\right|}{Ln}}{\dfrac{\sum_{i=1}^{Ls} \left|Ts\_s_i\right|}{Ls}}$$

$$E(BN) = scale\_factor \times E(Ts\_sil) \tag{4.10}$$

$$E(BN) = E(scale\_factor \times Ts\_sil) \tag{4.11}$$

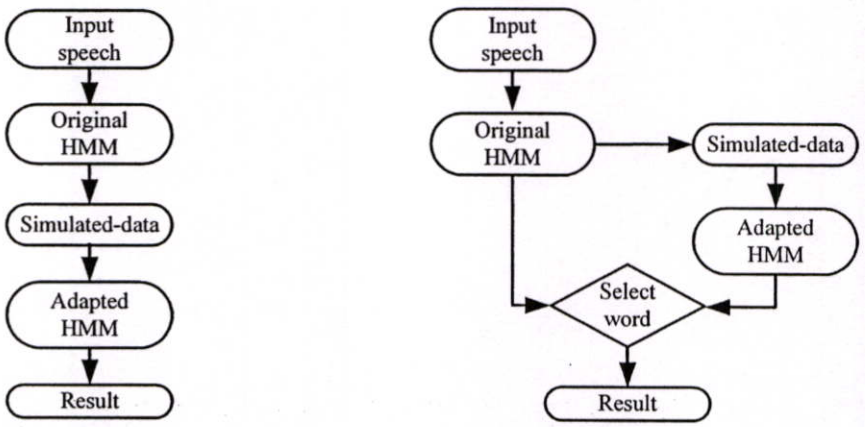
จากสมการที่ 4.11 เห็นได้ว่า Tsn_sil ที่จะทำให้สมการ SNR ของ Tn เท่ากับ SNR ของ Tsn เกิดจากผลคูณของ scale_factor และ Ts_sil นั่นเอง อย่างไรก็ตาม Ts_s_i ที่หาได้ด้วยการดึงส่วนเสียงรบกวนในหัวข้อ 4.2 ยังมีส่วนที่เป็น Ts_sil ติดเข้ามาด้วย จึงทำให้ scale_factor มีค่าน้อยกว่าความเป็นจริง และทำให้ SNR ที่ได้มีค่ามากกว่าความเป็นจริง แต่จากผลการทดลองเบื้องต้นพบว่าเมื่อวัด SNR ของ Tn และ Tsn ก็พบว่า SNR สูงๆ มีค่า SNR ของข้อมูลจำลองและเสียงพูดที่เข้ามาใกล้เคียงกัน แต่ SNR ที่ต่ำๆ ยังมีความแตกต่างของ SNR อยู่บ้าง เนื่องจากวิธีการหาส่วนที่เป็นเสียงพูดและส่วนที่ไม่เป็นเสียงพูดที่ได้นำเสนอ ยังไม่สามารถแก้ปัญหานี้ได้ โดยปัญหาดังกล่าวเป็นปัญหาที่ต้องแก้ด้วยวิธีการแยกเสียงพูด เพื่อแยก Ts_sil ออกจาก Ts_s_i ซึ่งเป็นสิ่งที่ต้องพัฒนาต่อไป

4.4 รูปแบบการใช้งานข้อมูลจำลอง

ในการปรับโมเดลก่อนหน้ามีการใช้เฉพาะเสียงพูดที่เข้ามา ช่วยให้การปรับโมเดลให้ผลการรู้จำเสียงพูดดีขึ้นได้ ดังนั้นในวิทยานิพนธ์นี้จึงยังใช้เสียงพูดที่เข้ามาในการปรับโมเดลด้วย แต่การนำเสียงพูดที่เข้ามามาใช้ ต้องมีการหาคำอ่านของเสียงพูดนี้ด้วย จึงทำให้ในขั้นตอนการดึงเสียงรบกวนจะใช้ได้เฉพาะโครงสร้างแบบการอิงกับหน่วยเสียง เพราะโครงสร้างนี้สามารถให้ คำอ่านของเสียงพูดที่เข้ามาได้ด้วย ดังนั้นในการปรับโมเดลด้วยวิธีการปรับโมเดลด้วยข้อมูลจำลองแบ่งออกเป็น 2 รูปแบบ คือ รูปแบบแรก เป็นการใช้เสียงพูดจากข้อมูลจำลองเพียงอย่างเดียว ทำให้รู้คำอ่านของเสียงพูดที่ใช้ปรับโมเดลทุกเสียงพูด ซึ่งในที่นี้เรียกว่า การปรับโมเดลด้วยข้อมูลจำลองแบบ “Supervised” ซึ่งการปรับโมเดลแบบนี้สามารถใช้การดึงเสียงรบกวนได้ทั้งโครงสร้างการอิงกับหน่วยเสียง และโครงสร้างการหาส่วนที่เป็นเสียงพูดและส่วนที่ไม่เป็นเสียงพูด ส่วนในรูปแบบที่สอง เป็นการใช้ทั้งเสียงพูดจากข้อมูลจำลองและเสียงพูดที่เข้ามาในการปรับโมเดล ทำให้มีทั้งข้อมูลที่ทราบคำอ่านที่ถูกต้อง (ข้อมูลจำลอง) และมีข้อมูลที่ต้องหาคำอ่าน (เสียงพูดที่เข้ามา) ด้วยการนำไปรู้จำเสียงพูด ซึ่งในที่นี้เรียกว่า การปรับโมเดลด้วยข้อมูลจำลองแบบ “Semi-supervised” เห็นได้ว่าโครงสร้างนี้ต้องการตำแหน่งของเสียงเงียบสำหรับการสร้างข้อมูลจำลอง และคำอ่านสำหรับเสียงพูดที่เข้ามา ทำให้การปรับโมเดลแบบนี้สามารถใช้ได้เฉพาะการดึงเสียงรบกวนด้วยโครงสร้างแบบการอิงกับหน่วยเสียง เพราะเป็นโครงสร้างที่บอกครบทั้งตำแหน่งของเสียงเงียบ และคำอ่านของเสียงพูดที่เข้ามาในเวลาเดียวกัน ส่วนโครงสร้างการหาส่วนที่เป็นเสียงพูดและส่วนที่ไม่เป็นเสียงพูดจะบอกเฉพาะตำแหน่งของเสียงเงียบเท่านั้น

4.5 การเลือกผลการรู้จำเสียงพูด

การปรับโมเดลด้วยข้อมูลจำลอง เป็นการแก้ปัญหาที่เกิดจากเสียงรบกวนจากสภาพแวดล้อม ที่มีลักษณะของเสียงรบกวนที่มีความคงที่ตลอดทั้งเสียงพูด อย่างไรก็ตาม ในการนำไปใช้งานจริงยังมีหลายปัจจัยที่ส่งผลต่อประสิทธิภาพของระบบการรู้จำเสียงพูดแบบคงทน เช่น ความไม่คงที่ของเสียงรบกวนในเสียงพูด, เสียงรบกวนจากช่องสัญญาณ หรือแม้กระทั่งเกิดจากความแตกต่างของผู้พูดที่ใช้งานเอง เป็นต้น ซึ่งเป็นปัจจัยที่วิธีการที่นำเสนอยังไม่สามารถแก้ไขปัญหาเหล่านี้ได้ ดังนั้นอาจมีความเป็นไปได้ที่โมเดลที่ได้จากการปรับโมเดลให้ผลการรู้จำเสียงพูดได้ไม่ดีเท่ากับโมเดลที่ไม่ได้มีการปรับโมเดล จึงเป็นเหตุผลที่ทำให้ไม่สามารถละเลยผลการรู้จำเสียงพูดที่ได้จากโมเดลที่ไม่มีการปรับโมเดลได้ ดังนั้นในวิทยานิพนธ์นี้จึงมีความสนใจที่จะแบ่งการเลือกคำตอบของการปรับโมเดลด้วยข้อมูลจำลองเป็น 2 แบบ แบบแรก คือ แบบหนึ่งขั้นตอน ดังรูป 4.5 (a) วิธีนี้จะเลือกคำตอบของโมเดลที่มีการปรับโมเดล และแบบที่สอง คือ แบบสองขั้นตอน ดังรูป 4.5 (b) วิธีนี้จะเลือกคำตอบระหว่างโมเดลที่ไม่มีการปรับโมเดลและโมเดลที่มีการปรับโมเดล ซึ่งเกณฑ์ในการเลือกคำตอบจะดูจากค่าความน่าจะเป็นของคำตอบที่ได้จากโมเดลทั้งสอง โดยคำตอบของโมเดลไหนมีค่าความน่าจะเป็นมากกว่าก็เลือกคำตอบของโมเดลนั้นมาเป็นคำตอบของระบบรู้จำเสียงพูด



a) One-step Method b) Two-step Method

รูปที่ 4.5 วิธีการเลือกคำตอบของการปรับโมเดลด้วยข้อมูลจำลอง

ตารางที่ 4.2 การปรับโมเดลด้วยข้อมูลจำลองทั้ง 4 รูปแบบ

Configuration	Description
1	One-step method with supervised adaptation
2	One-step method with semi-supervised adaptation
3	Two-step method with supervised adaptation
4	Two-step method with semi-supervised adaptation

จากตารางที่ 4.2 เป็นการสรุปรูปแบบของการปรับโมเดลด้วยข้อมูลจำลองในการทดลองในวิทยานิพนธ์ ซึ่งแบ่งออกได้เป็น 4 รูปแบบ แบบที่หนึ่ง คือ การเลือกคำตอบจากการปรับโมเดลแบบ Supervised และเลือกคำตอบแบบหนึ่งขั้นตอน แบบที่สอง คือ การเลือกคำตอบจากการปรับโมเดลแบบ Semi-Supervised และเลือกคำตอบแบบหนึ่งขั้นตอน แบบที่สาม คือ การเลือกคำตอบจากการปรับโมเดลแบบ Supervised และเลือกคำตอบแบบสองขั้นตอน และแบบที่สี่ คือ การเลือกคำตอบจากการปรับโมเดลแบบ Semi-Supervised และเลือกคำตอบแบบสองขั้นตอน

4.6 การประยุกต์ใช้การปรับโมเดลด้วยข้อมูลจำลองร่วมกับเทคนิคอื่นๆ

การปรับโมเดลด้วยข้อมูลจำลองเป็นการแก้ปัญหาการไม่ทราบคำอ่านของเสียงพูดที่เข้ามา และทำให้มีจำนวนข้อมูลในการปรับโมเดลเพิ่มมากขึ้น ด้วยการบวกส่วนที่เป็นเสียงรบกวนที่ได้จากเสียงพูดที่เข้ามาไปยังเสียงพูดสะอาดที่มีการอัดเสียงพูดเตรียมไว้ก่อน วิธีที่วิทยานิพนธ์นี้นำเสนอสามารถนำไปใช้งานร่วมกับเทคนิคอื่นๆ ที่มีการนำเสนอก่อนหน้านี้ได้ ซึ่งเทคนิคที่ใช้ร่วมกับวิธีการปรับโมเดลด้วยข้อมูลจำลองแบ่งได้เป็น 2 ส่วน คือ ส่วนแรกเป็นการเตรียมโมเดลตั้งต้นในการปรับโมเดลด้วยข้อมูลจำลอง ซึ่งสามารถใช้กับโมเดลที่ได้จากวิธีอื่นๆ ได้ โดยไม่จำเป็นต้องเป็นโมเดลเสียงพูดสะอาด (Clean Speech Model หรือ Baseline) เพียงอย่างเดียว เช่น โมเดลแบบ MULTI [14] และโมเดลแบบ MSTC [5] เป็นต้น ส่วนที่สองเป็นวิธีการปรับโมเดลด้วยข้อมูลจำลองที่ไม่จำเป็นต้องเป็นวิธี MLLR [7] เพียงอย่างเดียว แต่อาจเป็นวิธีการปรับโมเดลวิธีอื่นๆ ได้ เช่น วิธี MAP [16] และ วิธี MLLR-MAP [39] เป็นต้น

การปรับโมเดลด้วยข้อมูลจำลองมีตัวแปรที่ต้องนำมาพิจารณา คือ จำนวนของคำ และจำนวนคนพูดใน Train-S ซึ่งตัวแปรเหล่านี้มีการทดลองปรับค่า เพื่อให้ได้ผลการรู้จำเสียงพูดที่ดีที่สุด ซึ่งมีการทดลองในบทที่ 6

บทที่ 5
การประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดล

ปัญหาของการเลือกโมเดลแบบ MSTC คือ ต้องสร้างโมเดลแบบผสมเตรียมไว้ล่วงหน้าสำหรับใช้ในการรองรับเสียงรบกวน และถึงแม้ว่า การเลือกโมเดลแบบโครงสร้างต้นไม้ทำให้หาโมเดลที่ใกล้เคียงเสียงพูดที่เข้ามาได้เร็วขึ้น แต่การแบ่งกลุ่มในการสร้างโครงสร้างต้นไม้มีรูปแบบของโมเดลแบบผสมที่พิจารณาจากความใกล้เคียงกันของเสียงรบกวน ซึ่งทำให้มีจำนวนโมเดลแบบผสมได้จำกัด โดยการเลือกโมเดลแบบ MSTC มีรูปแบบของโมเดลแบบผสมได้สูงสุดเท่ากับ (N – 1) แบบ เมื่อ N คือ จำนวนชนิดเสียงรบกวนทั้งหมด อย่างไรก็ตาม การที่มีโมเดลแบบผสมเพิ่มขึ้นมา ก็ทำให้ต้องมีการจัดเก็บโมเดลของ Noise Cluster HMM และ Noisy Speech HMM เพิ่มขึ้นตามไปด้วย รวมทั้งสิ้นเท่ากับ 2(N - 1) โมเดล การเพิ่มของจำนวนโมเดลแบบผสมนี้ ทำให้มีโมเดลที่สามารถรองรับเสียงรบกวนที่ไม่ได้อยู่ในชุดฝึกสอนได้มากขึ้น แต่ก็ต้องแลกมากับพื้นที่ในการเก็บโมเดลแบบผสมที่เพิ่มขึ้นตามไปด้วย

วิทยานิพนธ์นี้นำเสนอ การประยุกต์ใช้งานเทคนิคการประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดล (Hidden Markov Model Interpolation) [24],[25] ของโมเดลเสียงพูด ซึ่งเป็นเทคนิคในการสร้างโมเดลเสียงพูดขึ้นจากการประมาณค่าในช่วงของโมเดลเสียงพูดหลายๆ โมเดลเข้าด้วยกัน ซึ่งในที่นี้เรียกว่า "การประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดลที่มีการจัดกลุ่มเสียงรบกวน (NCHI)" สำหรับการประมาณค่าในช่วง ใช้ข้อมูลเสียงพูดที่เข้ามา ในการหาค่าถ่วงน้ำหนัก (Weight) และจำนวนชนิดของโมเดล เพื่อใช้ในการสร้างโมเดลเสียงพูดขึ้นใหม่ จุดเด่นของวิธีนี้ คือ สามารถทำให้มีรูปแบบของโมเดลแบบผสมได้มากกว่า MSTC และไม่ต้องสร้างโมเดลแบบผสมไว้ล่วงหน้าด้วย เป็นผลให้วิธีที่วิทยานิพนธ์นี้นำเสนอสามารถแก้ไขข้อจำกัดของการเลือกโมเดลแบบ MSTC ได้

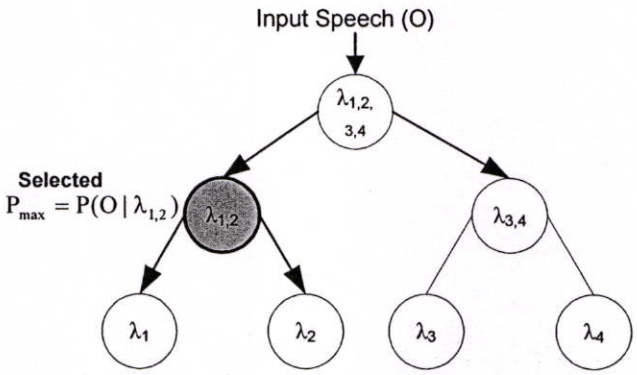
ความแตกต่างของวิธีที่วิทยานิพนธ์นี้นำเสนอ และการเลือกโมเดลแบบทั่วไป แสดงดังรูปที่ 5.1 โดยรูปที่ 5.1 (a) คือ การเลือกโมเดลแบบทั่วไป และรูปที่ 5.1 (b) คือ NCHI จากรูปที่ 5.1 ความแตกต่างของทั้งสองวิธี คือ

1) การจัดเก็บโมเดล สำหรับการเลือกโมเดลแบบทั่วไปจะต้องมีการจัดเก็บโมเดลเสียงรบกวนพื้นฐาน (λ_1, λ_2 λ_3 และ λ_4) และโมเดลเสียงรบกวนแบบผสม ($\lambda_{1,2}$, $\lambda_{3,4}$ และ $\lambda_{1,2,3,4}$) ในการสร้างโมเดลเสียงรบกวนแบบผสมมีรูปแบบการผสมที่พิจารณาจากความใกล้เคียงกันของโมเดลเสียงรบกวนพื้นฐาน โดยโมเดลที่มีความใกล้เคียงกันจะถูกผสมเข้าด้วยกัน ซึ่งอาจจะทำให้มีรูปแบบการผสมไม่ครบทุกรูปแบบที่เป็นไปได้ ส่วน NCHI จะมีการจัดเก็บเฉพาะโมเดลเสียงรบกวนพื้นฐานเท่านั้น

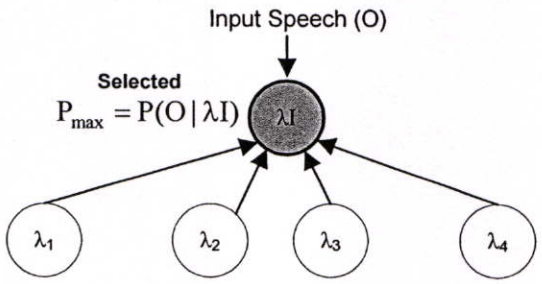
2) โมเดลที่ใช้ในการรู้จำเสียงพูด สำหรับการเลือกโมเดลแบบทั่วไปเป็นการเลือกโมเดลที่มีความใกล้เคียงกับเสียงพูดที่เข้ามามากที่สุด จากรูปที่ 5.1 เห็นได้ว่าโมเดลแบบทั่วไปจะเลือก $\lambda_{1,2}$ ที่มีค่า $P(O|\lambda_{1,2})$ สูงสุด ส่วนโมเดลที่ได้จากวิทยานิพนธ์นี้นำเสนอ เป็นการเลือกกลุ่มของโมเดลเสียงพูดที่มีเสียงรบกวนที่มีการสร้างเก็บไว้ แล้วนำมาประมาณค่าในช่วงให้ได้โมเดล λI ที่มีค่า $P(O|\lambda I)$ สูงสุด (กำหนดให้ λI คือ โมเดลที่ได้จากการประมาณค่าในช่วง)

กล่าวได้ว่า NCHI สามารถสร้างโมเดลเสียงพูดขึ้นมาใหม่ได้ โดยไม่ต้องมีการสร้างโมเดลแบบผสมขึ้นมาเก็บไว้ก่อน ดังนั้นจำนวนของโมเดลที่ต้องจัดเก็บไว้ใช้ในการเลือกโมเดลจึงน้อยกว่า และยังสามารถมีรูปแบบของโมเดลแบบผสมได้มากกว่าด้วย

ในบทนี้ จะกล่าวถึงการประมาณค่าในช่วงของ HMM จำนวน 2 หัวข้อ ได้แก่ หัวข้อ 5.1 อธิบายวิธีการในการประมาณค่าในช่วงของ HMM หัวข้อ 5.2 อธิบายวิธีการในการประมาณค่าในช่วงของ HMM สำหรับระบบรู้จำเสียงพูด ซึ่งประกอบด้วย 3 หัวข้อ ได้แก่ หัวข้อ 5.2.1 การหาค่าถ่วงน้ำหนักและจำนวนชนิดของโมเดล ด้วยวิธีการค้นหาแบบโครงสร้างต้นไม้ (Tree Structure Search) หัวข้อ 5.2.2 การหาค่าถ่วงน้ำหนักและจำนวนชนิดของโมเดล ด้วยวิธีการค้นหาแบบตรง (Direct Search) และหัวข้อ 5.2.3 อธิบายการใช้งาน NCHI ร่วมกับการปรับโมเดลด้วยข้อมูลจำลอง



(a) Basic model Selection

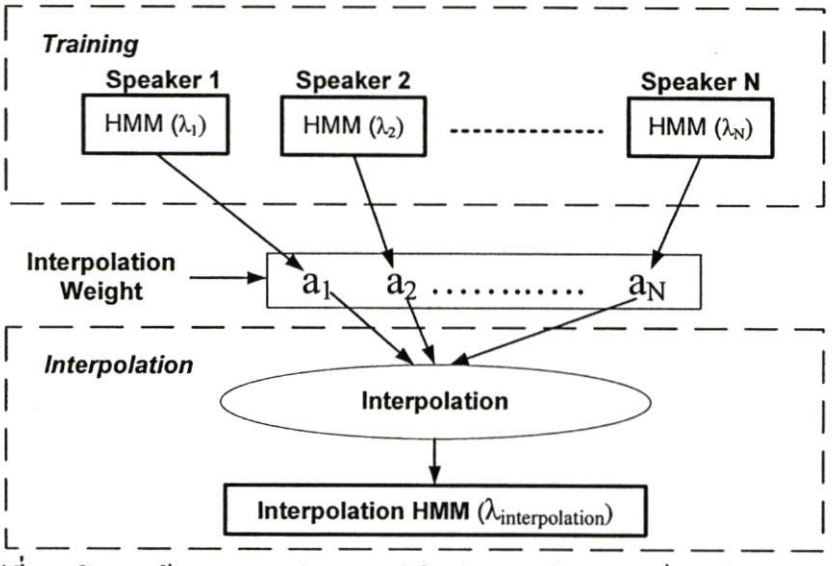


(b) Noise Cluster HMM Interpolation

รูปที่ 5.1 การเปรียบเทียบการเลือกโมเดลระหว่างวิธีแบบปกติ และวิธีในวิทยานิพนธ์นี้

5.1 การประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดล

โดยปกติแล้วการประมาณค่าในช่วงของ HMM เป็นเทคนิคที่มีใช้อยู่แล้วในการปรับผู้พูด (Speaker Adaptation) ของระบบการแปลงข้อความเป็นเสียงพูด (Text to Speech System, TTS) หรือระบบการสังเคราะห์เสียงพูด (Speech Synthesis System) ด้วยโมเดลเสียงพูด [24],[25] ซึ่งโมเดลเสียงพูดที่ใช้ในระบบการสังเคราะห์เสียงพูด คือ HMM มีโครงสร้างดังรูปที่ 5.2



รูปที่ 5.2 โครงสร้างของการประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดล [24]

โมเดลเสียงพูดที่นำมาประมาณค่าในช่วงนั้นต้องมีโครงสร้างแบบเดียวกันไม่ว่าจะเป็นรูปแบบการย้ายสเตท, จำนวนสเตท และโมเดลความน่าจะเป็นของแต่ละสเตท สำหรับการประมาณค่าในช่วงของ HMM เป็นการประมาณค่าในช่วงของความน่าจะเป็นของค่าสังเกตในแต่ละสเตทของ HMM ซึ่งในวิทยานิพนธ์นี้ใช้การกระจายความน่าจะเป็นของค่าสังเกตแบบเกาส์เซียนมิกเจอร์

การประมาณค่าในช่วงของ HMM จะประมาณค่าในช่วงทีละเกาส์เซียนมิกเจอร์ของแต่ละสเตทที่อยู่ในแต่ละโมเดล ทำให้พารามิเตอร์ของการประมาณค่าในช่วง ในการทำแต่ละครั้งลดลงไปเหลือแค่ $p_k = \tilde{N}(O;\mu_k,\Sigma_k)$ เมื่อ O คือ ค่าสังเกต, k คือ ลำดับของชนิดเสียงรบกวนที่นำมาสร้างโมเดล, μ_k คือ เวกเตอร์ค่าเฉลี่ยของโมเดล k และ Σ_k คือ เมตริกซ์ความแปรปรวนร่วมของโมเดล k โดยสมการที่ใช้ในการประมาณค่าในช่วงที่มีใช้อยู่ใน TTS มีด้วยกัน 3 สมการ [24],[25] ดังนี้

กำหนดให้

N คือ จำนวนโมเดลทั้งหมดที่ใช้ในการประมาณค่าในช่วง

μ คือ เวกเตอร์ค่าเฉลี่ยที่เกิดจากการประมาณค่าในช่วง

Σ คือ เมตริกซ์ความแปรปรวนร่วมที่เกิดจากการประมาณค่าในช่วง

a_k คือ ค่าถ่วงน้ำหนักของโมเดลที่สร้างจากเสียงรบกวน k

สมการแบบที่หนึ่ง ตั้งสมมุติฐานให้ค่าสังเกตที่ใช้สร้างโมเดลใหม่ ได้จากการประมาณค่าในช่วงค่าสังเกตของโมเดลต่างๆ ดังสมการ

$$O = \sum_{k=1}^{N} a_k O_k \tag{5.1}$$

เมื่อ a_k เป็นค่าถ่วงน้ำหนักสำหรับโมเดลที่ k และ $\sum_{k=1}^{N} a_k = 1$ โดยที่ μ และ Σ ของโมเดลใหม่หาได้จากสมการที่ (5.2) และ (5.3) ตามลำดับ

$$\mu = \sum_{k=1}^{N} a_k \mu_k \tag{5.2}$$

$$\Sigma = \sum_{k=1}^{N} a_k^2 \Sigma_k \tag{5.3}$$

สมการแบบที่สอง ตั้งสมมุติฐานให้ค่าของ μ และ Σ ของโมเดลใหม่ได้จากการประมาณค่าในช่วงของค่า μ_k และ Σ_k ของโมเดลต่างๆ จากการฝึกสอนเวกเตอร์ลักษณะสำคัญ ที่มีจำนวนเท่ากับ γ_k ตัวอย่าง การสร้างโมเดลใหม่ด้วยการประมาณค่าในช่วงของสมการนี้ขึ้นอยู่กับจำนวน γ_k ของแต่ละเสียงรบกวน ดังสมการ

$$\begin{aligned} \mu &= \frac{\sum_{k=1}^{N} \gamma_k \mu_k}{\gamma} \\ &= \sum_{k=1}^{N} a_k \mu_k \end{aligned} \tag{5.4}$$

$$\begin{aligned} \Sigma &= \frac{\sum_{k=1}^{N} \gamma_k \Sigma_k}{\gamma} - \mu\mu^T \\ &= \sum_{k=1}^{N} a_k (\Sigma_k + \mu_k \mu_k^T) - \mu\mu^T \end{aligned} \tag{5.5}$$

เมื่อ $\gamma = \sum_{k=1}^{N} \gamma_k$, $a_k = \frac{\gamma_k}{\gamma}$ และ $\sum_{k=1}^{N} a_k = 1$

สมการแบบที่สาม ตั้งสมมุติฐานให้ความแตกต่างกันระหว่าง HMM ที่สร้างจากการประมาณค่าในช่วง และแต่ละ HMM ที่สร้างมาจากเสียงพูดที่มีเสียงรบกวน สามารถวัดได้ด้วยวิธี Kullback-Leibler ระหว่าง p และ p_k ดังนั้นถ้าให้ $p_1, p_2, ..., p_N$ มีค่าถ่วงน้ำหนักเท่ากับ $a_1, a_2, ..., a_N$ ดังนั้น p ที่ดีที่สุด คือ p ที่ทำให้ค่า ε ในสมการที่ (5.6) มีค่าน้อยที่สุด

$$\varepsilon = \sum_{k=1}^{N} a_k I(p, p_k) \tag{5.6}$$

เมื่อ $I(b(O), b(O_k))$ คือ ฟังก์ชันของการวัดแบบ Kullback-Leibler เป็นดังสมการที่ (5.7)

$$I(p, p_k) = \int_{-\infty}^{\infty} N(O;\mu,\Sigma) \log \frac{N(O;\mu,\Sigma)}{N(O;\mu_k,\Sigma_k)} dO \tag{5.7}$$

$$I(p, p_k) = \frac{1}{2}\{\log \frac{|\Sigma_k|}{|\Sigma|} + tr[\Sigma_k^{-1}\{(\mu_k - \mu)(\mu_k - \mu)^T + \Sigma\}] + 1\} \tag{5.8}$$

เพื่อหาค่า μ และ Σ ของ p ที่ทำให้ค่า ε มีค่าน้อยที่สุด ด้วยการหาค่าอนุพันธ์ (Differentiation) สมการที่ (5.6) แล้วกำหนดให้เท่ากับ 0 นั้น คือ

$$\begin{aligned} \frac{\partial \varepsilon}{\partial \lambda} &= \sum_{k=1}^{N} a_k \frac{\partial I(p, p_k)}{\partial \mu} = 0 \\ &= \sum_{k=1}^{N} a_k \Sigma_k^{-1}(\mu_k - \mu) \end{aligned} \tag{5.9}$$

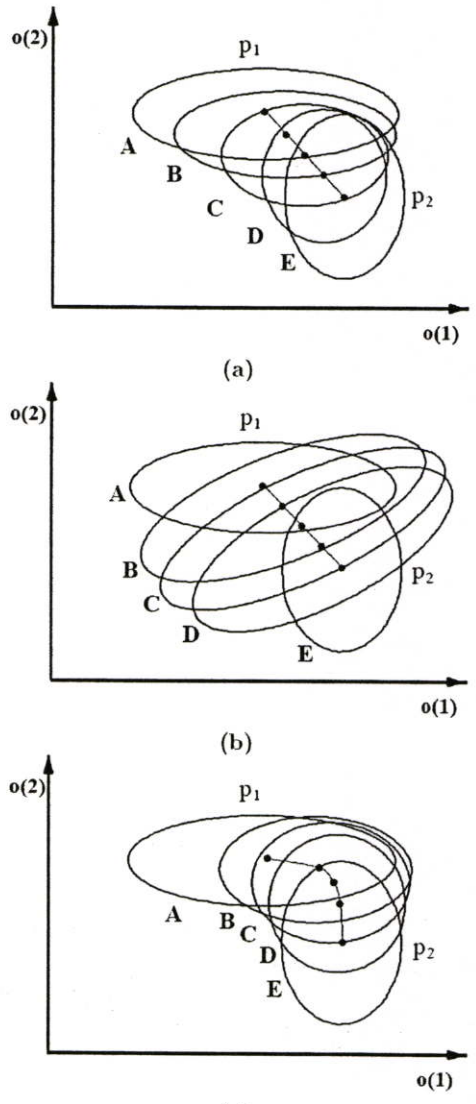
$$\begin{aligned} \frac{\partial \varepsilon}{\partial \Sigma} &= \sum_{k=1}^{N} a_k \frac{\partial I(p, p_k)}{\partial \Sigma} = 0 \\ &= \sum_{k=1}^{N} a_k \frac{1}{2}(\Sigma_k^{-1} - \Sigma^{-1}) \end{aligned} \tag{5.10}$$

เมื่อแก้สมการที่ (5.9) และ (5.10) จะได้ μ และ Σ ของ p เป็นสมการที่ (5.11) และ (5.12) ตามลำดับ

$$\mu = \left(\sum_{k=1}^{N} a_k \Sigma_k^{-1}\right)^{-1} \left(\sum_{k=1}^{N} a_k \Sigma_k^{-1} \mu_k\right) \tag{5.11}$$

$$\Sigma = \left(\sum_{k=1}^{N} a_k \Sigma_k^{-1} \right)^{-1} \tag{5.12}$$

ในการประมาณค่าในช่วงด้วยสมการทั้ง 3 แบบ ทำให้มีการกระจายตัวของเกาส์เซียนที่แตกต่างกันเป็นดังรูปที่ 5.3



รูปที่ 5.3 การเปรียบเทียบลักษณะการกระจายตัวของข้อมูลตามโมเดล ที่ได้จากการประมาณค่าในช่วงด้วยสมการแบบที่หนึ่ง (a), สมการแบบที่สอง (b) และสมการแบบที่สาม (c) [24]

o(1) และ o(2) คือ ค่าสังเกตของเสียงพูด, p_1 และ p_2 คือ โมเดลตั้งต้นของการประมาณค่าในช่วง และ A, B, C, D และ E คือ โมเดลที่ได้จากการประมาณค่าในช่วงของโมเดล p_1 และโมเดล p_2 ด้วยค่าถ่วงน้ำหนักที่แตกต่างกัน กำหนดให้ค่าถ่วงน้ำหนักระหว่างโมเดล p_1 และโมเดล p_2 เขียนแทนได้ด้วย $W(a_{p1},a_{p2})$ ค่าถ่วงน้ำหนักการประมาณค่าในช่วงของโมเดล A ถึงโมเดล E มีค่าเท่ากับ

W(1.0,0.0), W(0.75,0.25), W(0.5,0.5), W(0.25,0.75) และ W(0.0,1.0) ตามลำดับ จากรูปที่ 5.3 เห็นได้ว่าลักษณะการกระจายตัวของข้อมูลตามโมเดล A, B, C, D และ E ที่ได้จากการประมาณค่าในช่วงของโมเดล p_1 และโมเดล p_2 ด้วยสมการแบบที่สาม ให้ลักษณะการกระจายตัวของข้อมูลอยู่ในช่วงการกระจายตัวตามโมเดล p_1 และโมเดล p_2 มากกว่าการประมาณค่าในช่วงด้วยสมการแบบที่หนึ่งและแบบที่สอง นอกจากนี้ เมื่อสังเคราะห์เสียงพูดจากโมเดลที่ได้จากการประมาณค่าในช่วงทั้งสามสมการ และนำเสียงพูดที่สังเคราะห์ได้มาให้คนฟัง ก็พบว่าเสียงพูดที่สังเคราะห์ได้จากสมการแบบที่สามมีค่าระดับความพึงพอใจมากที่สุด [24],[25] ทำให้ระบบการสังเคราะห์เสียงพูดเลือกใช้สมการแบบที่สามในการประมาณค่าในช่วง แต่สำหรับในงานระบบรู้จำเสียงพูดยังไม่เคยมีการทดลองว่าสมการประมาณค่าในช่วงแบบใด ให้ผลการรู้จำเสียงพูดได้ดีที่สุด ดังนั้นในวิทยานิพนธ์นี้จึงมีการทดลองเปรียบเทียบประสิทธิภาพการรู้จำเสียงพูดด้วยการประมาณค่าในช่วงของ HMM จากทั้ง 3 สมการ ซึ่งได้มีการทดลองในบทถัดไป

5.2 การประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดลสำหรับระบบรู้จำเสียงพูด

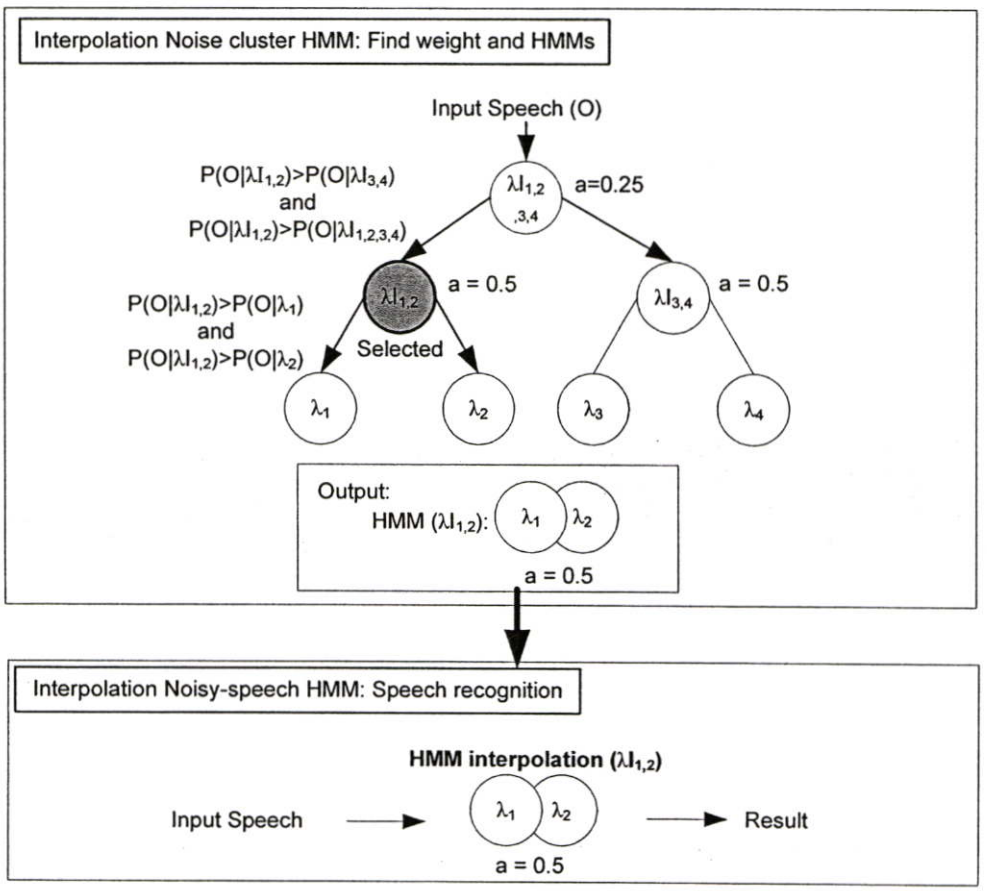
จากสมการทั้งสามสมการในการประมาณค่าในช่วงของโมเดล เห็นได้ว่าพารามิเตอร์ที่สำคัญของการประมาณค่าในช่วงมีอยู่ด้วยกัน 2 ตัว คือ 1) ค่าถ่วงน้ำหนัก และ 2) HMM ที่ใช้ในการประมาณค่าในช่วงของโมเดล ซึ่งเป็นการหาทั้งจำนวนและชนิดของโมเดล ในที่นี้เรียกว่า "จำนวนชนิดของโมเดล" ซึ่งการหาค่าของพารามิเตอร์ทั้งสองตัวนี้ ในระบบการสังเคราะห์เสียงพูด หาได้โดยอาศัยการปรับค่าถ่วงน้ำหนักและเลือกชนิดของโมเดลจากผู้ที่ใช้งานเอง เช่น ต้องการโมเดลเสียงพูดที่มีลักษณะผสมระหว่างเสียงพูดผู้ชายและเสียงพูดผู้หญิง ในขั้นตอนแรก คือ การเลือกโมเดลเสียงพูดของผู้ชาย แล้วเลือกโมเดลเสียงพูดของหญิง และขั้นตอนต่อไป คือ การหาค่าถ่วงน้ำหนักของโมเดลเสียงพูดของผู้ชาย และค่าถ่วงน้ำหนักของโมเดลเสียงพูดของผู้หญิง โดยการปรับด้วยมือ จากนั้นนำโมเดลที่ได้ไปสังเคราะห์เสียงพูด แล้วฟังว่าเสียงพูดที่สังเคราะห์จากโมเดลที่ค่าถ่วงน้ำหนักไหนให้เสียงพูดที่ต้องการ โดยไม่จำเป็นต้องหาค่าพารามิเตอร์ทั้งสองแบบอัตโนมัติ แต่ในการนำเทคนิคการประมาณค่าในช่วงมาใช้ในการผสมโมเดลเสียงพูดที่มีเสียงรบกวน ต้องมีการแก้ปัญหาในเรื่องการหาค่าถ่วงน้ำหนัก และจำนวนชนิดของโมเดลให้เป็นแบบอัตโนมัติ

วิทยานิพนธ์นี้นำเสนอวิธีการหาค่าถ่วงน้ำหนัก และจำนวนชนิดของโมเดล สำหรับการสร้างโมเดลใหม่ด้วย NCHI โดยตั้งสมมุติฐานว่าแบบจำเสียงพูดที่มีเสียงรบกวนซึ่งได้ถูกประมาณค่าในช่วงขึ้นมานั้น ควรถูกสร้างมาจากโมเดลเสียงพูดที่มีเสียงรบกวนซึ่งมีความใกล้เคียงกับเสียงพูดที่เข้ามา โดยโมเดลที่นำมาประมาณค่าในช่วงอาจมีมากกว่า 2 โมเดลก็ได้ ถ้ารวมกันแล้วให้โมเดลที่ดีกว่า หรืออาจไม่มีการประมาณค่าในช่วงเกิดขึ้นก็ได้ ถ้าโมเดลที่มีการประมาณค่าในช่วงมีความใกล้เคียงน้อยกว่าโมเดลเสียงพูดที่มีอยู่เดิม

วิทยานิพนธ์นี้จึงเสนอวิธีหาค่าถ่วงน้ำหนัก และจำนวนชนิดของโมเดล ที่ใช้ในการสร้างโมเดลด้วย NCHI จำนวน 2 วิธี คือ 1) วิธีการค้นหาแบบโครงสร้างต้นไม้ และ 2) วิธีการค้นหาแบบตรง ซึ่งอธิบายรายละเอียดในหัวข้อถัดไป

5.2.1 วิธีการค้นหาแบบโครงสร้างต้นไม้

วิธีการค้นหาแบบโครงสร้างต้นไม้มีโครงสร้างดังรูป 5.4 โดยแต่ละโหนดในโครงสร้างต้นไม้ได้จากการรวมของเสียงรบกวนที่มีความคล้ายกันเข้าด้วยกัน เช่น ในระดับ 2 มีโหนด 1,2 และ 3,4 เกิดจากการจัดกลุ่มในระดับ 1 โหนด 1,2,3,4 เป็น 2 กลุ่ม ซึ่งทำให้เสียงรบกวน 1 และ 2 ที่มีความคล้ายกันอยู่ด้วยกัน และเช่นเดียวกับเสียงรบกวน 3 และ 4 ที่มีความคล้ายกันอยู่ด้วยกัน โดยในการจัดกลุ่มเสียงรบกวนต้องมีการทำไว้ก่อน และการรวมกันของเสียงรบกวนจะไม่มีความสัมพันธ์กับเสียงพูดที่เข้ามา การจัดกลุ่มโมเดลของวิธีที่วิทยานิพนธ์นี้นำเสนอเหมือนกับ Noise Cluster HMM ของ MSTC [5] โดยจำนวนชนิดของโมเดลเป็นแบบเดียวกันกับโมเดลที่ใช้ในการสร้างโมเดลที่เลือกมาจาก MSTC และมีค่าถ่วงน้ำหนักแบบเฉลี่ยเท่ากัน เช่น ถ้ามีการรวมกันของ 4 โมเดล ค่าถ่วงน้ำหนักจะเป็น 0.25 เท่ากันทุกโมเดล



รูปที่ 5.4 โครงสร้างการค้นหาค่าถ่วงน้ำหนักและจำนวนชนิดของโมเดล ด้วยการวิธีค้นหาแบบโครงสร้างต้นไม้

วิธีการค้นหาแบบโครงสร้างต้นไม้มีขั้นตอนดังนี้

1) การค้นหาโมเดลแบบ Noise Cluster HMM ของ MSTC โดย MSTC ที่ใช้ในการค้นหาจะมีการเก็บเฉพาะโมเดลของโหนดที่ถูกสร้างมาจากเสียงพูดที่มีเสียงรบกวนหนึ่งชนิดและหนึ่ง SNR เท่านั้น กล่าวคือไม่เก็บโมเดลของโหนดกลางๆ ที่เป็นโมเดลแบบผสม แต่จะมีการสร้างโมเดลแบบผสมขึ้นมาใหม่ ด้วยการประมาณค่าในช่วงของโมเดลทุกครั้งที่มีการค้นหาผ่านโหนดประเภทนี้ ส่วนวิธีการตัดสินใจเลือกโมเดลเหมือนกับ MSTC
2) จากขั้นตอนที่หนึ่งทำให้รู้ว่าโหนดไหนมีความใกล้เคียงกับเสียงรบกวนในเสียงพูดที่เข้ามามากที่สุด ส่วนขั้นตอนที่สองเป็นขั้นตอนการสร้าง Noisy Speech HMM ด้วยการ ประมาณค่าในช่วงของโมเดลตามจำนวนชนิดของโมเดลที่ใช้ในการสร้างโหนดที่เลือกได้ และใช้ค่าถ่วงน้ำหนักเดียวกันหมด สำหรับทุกโมเดล ดังสมการที่ (5.13)

$$a = \frac{1}{\sum_{k=1}^{N} \delta(k)} \tag{5.13}$$

เมื่อ

O คือ สัญญาณเสียงพูดที่เข้ามา

k คือ ลำดับของชนิดของโมเดลเสียงรบกวน

N คือ จำนวนโมเดลเสียงรบกวน

λ_k คือ โมเดลเสียงรบกวนที่ k

$\lambda I_{\lambda_1,\lambda_2,.....}$ คือ โมเดลเสียงที่ถูกประมาณค่าในช่วงจาก $\lambda_1, \lambda_2, ...$

a คือ ค่าถ่วงน้ำหนัก

$\delta(k)$ = 1; เมื่อโมเดลลำดับที่ n อยู่ในโหนดที่ถูกเลือก

= 0; เมื่อโมเดลลำดับที่ n ไม่ได้อยู่ในโหนดที่ถูกเลือก

การประมาณค่าในช่วงของโมเดลด้วยการหาค่าพารามิเตอร์ที่ใช้วิธีค้นหาแบบโครงสร้างต้นไม้สามารถช่วยแก้ปัญหาของ MSTC เฉพาะเรื่องการลดจำนวนการจัดเก็บโมเดลแบบผสมเท่านั้น แต่ไม่ได้ช่วยในเรื่องรูปแบบการผสมโมเดลของ Noise Cluster HMM และ Noisy Speech HMM อย่างไรก็ดี วิธีการค้นหาแบบโครงสร้างต้นไม้สามารถช่วยลดจำนวนโมเดลที่ต้องจัดเก็บได้ทั้งหมดเท่ากับ 2(N-1) โมเดล

5.2.2 วิธีการค้นหาแบบตรง

วิธีการค้นหาแบบตรง มีโครงสร้างดังรูปที่ 5.5 วิธีนี้เป็นการประมาณค่าในช่วงของโมเดลเสียงรบกวนของ Noise Cluster HMM ที่มีความใกล้เคียงกับเสียงรบกวนในเสียงพูดที่เข้ามามากที่สุดสองอันดับแรกก่อน แล้วค่อยๆ เพิ่มจำนวนโมเดลเสียงรบกวนที่มีความใกล้เคียงถัดมาเข้าไปทีละหนึ่งโมเดล โดยหยุดการเพิ่ม เมื่อค่าความน่าจะเป็นของโมเดลที่สร้างขึ้นมาใหม่ มีค่าน้อยกว่าค่าความน่าจะเป็นของโมเดลที่สร้างมาก่อนหน้า ดังนั้นจำนวนชนิดของโมเดล และค่าถ่วงน้ำหนัก จะมีค่าเดียวกันกับของโมเดลก่อนหน้า ซึ่งมีขั้นตอนการหาค่าถ่วงน้ำหนัก และจำนวนชนิดของโมเดลดังต่อไปนี้

กำหนดให้

k คือ ลำดับของชนิดของโมเดลเสียงรบกวน

i คือ ลำดับของชนิดของเสียงรบกวนที่มีการเรียง $P(O|\lambda_k)$ จากมากไปหาน้อย

S คือ จำนวนชนิดของเสียงรบกวนตามการเรียงลำดับของ i ที่ใช้ในการประมาณค่าในช่วงของโมเดล

1) หาค่า $P(O|\lambda_k)$ (k = 1, 2, ……., N)

2) เรียงลำดับค่า $P(O|\lambda_i)$ (i = 1, 2, ……., N) จากมากไปน้อย

3) กำหนดให้ S = 2, $\lambda_{best} = \lambda_{i=1}$ และ $P_{best} = P(O|\lambda_{i=1})$

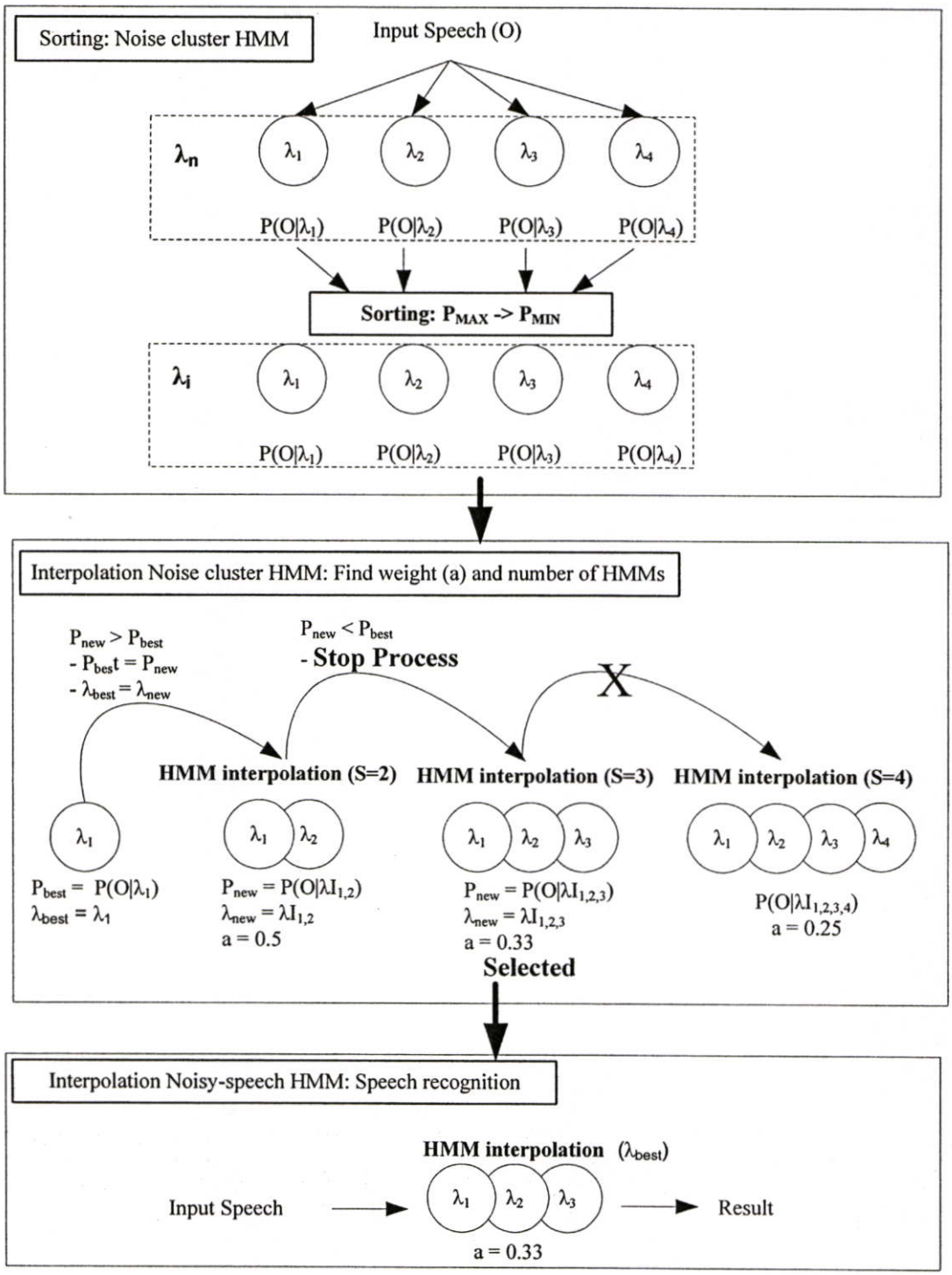
4) คำนวณหา $\lambda I_{new} = \lambda I_{1,\ldots,S}$ และ $P_{new} = P(O|\lambda I_{1,\ldots,S})$ โดยใช้ค่าถ่วงน้ำหนักเท่ากับ $\frac{1}{S}$

5) ตรวจสอบเงื่อนไข

กรณี $P_{new} < P_{best}$ ให้ "หยุด" การค้นหาแล้วใช้โมเดล λ_{best} สำหรับสร้าง Noisy speech HMM ในการรู้จำเสียงพูดให้มีโครงสร้างเดียวกับ λ_{best} ของการประมาณค่าในช่วงของ Noise cluster HMM

กรณี $P_{new} > P_{best}$ ให้ S = S + 1 แล้วกลับไปทำขั้นตอนที่สี่อีกครั้ง

การประมาณค่าในช่วงของโมเดลด้วยการหาค่าพารามิเตอร์ด้วยวิธีค้นหาแบบตรง ได้ช่วยแก้ปัญหาของ MSTC ทั้งสองข้อ คือ 1) เรื่องของจำนวนโมเดลแบบผสมที่ต้องมีการจัดเก็บเพิ่มขึ้น และ 2) รูปแบบการผสมโมเดลที่จำกัดตามโครงสร้างต้นไม้ของ MSTC วิธีการค้นหาแบบตรงสามารถแก้ไขปัญหาของ MSTC ได้มากกว่าการหาค่าพารามิเตอร์แบบโครงสร้างต้นไม้

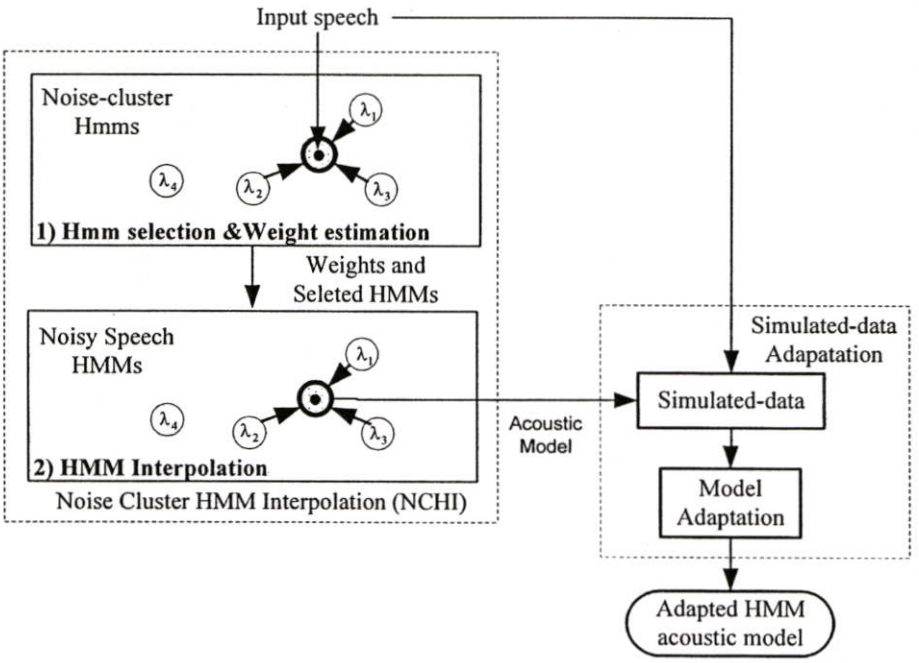


รูปที่ 5.5 โครงสร้างการค้นหาค่าถ่วงน้ำหนักและจำนวนชนิดของโมเดล ด้วยวิธีการค้นหาแบบตรง

สำหรับประสิทธิภาพการรู้จำเสียงพูดด้วยโมเดลที่ได้จากการประมาณค่าในช่วง โดยการค้นหาพารามิเตอร์จากวิธีค้นหาทั้งสองแบบที่นำเสนอ จะทดลองเปรียบเทียบผลในบทถัดไป

5.2.3 การประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดลที่มีการจัดกลุ่มเสียงรบกวนร่วมกับการปรับโมเดลด้วยข้อมูลจำลอง

การใช้ NCHI ร่วมกับการปรับโมเดลด้วยข้อมูลจำลอง เป็นวิธีในการปรับแบบออนไลน์วิธีใหม่ ซึ่งในที่นี้เรียกว่า "S-NCHI" โดยใช้ NCHI ในการสร้างโมเดลตั้งต้นให้มีความใกล้เคียงกับเสียงพูดที่เข้ามา และนำโมเดลที่ได้จาก NCHI มาปรับโมเดลด้วยข้อมูลจำลอง เพื่อให้โมเดลที่ได้มีความใกล้เคียงกับเสียงพูดที่เข้ามายิ่งขึ้น โดยมีโครงสร้างดังรูปที่ 5.6



รูปที่ 5.6 โครงสร้างของการปรับโมเดลแบบ S-NCHI

วิธีที่วิทยานิพนธ์นี้นำเสนอสามารถแก้ปัญหาข้อจำกัดของการปรับโมเดลด้วยวิธี PLT ได้ครบทั้ง 3 ปัญหา คือ 1) ปัญหาการไม่ทราบคำอ่านของเสียงพูดที่เข้ามา 2) ปัญหาเสียงพูดที่เข้ามามีปริมาณน้อย ซึ่งทั้งสองปัญหาสามารถแก้ไขได้ด้วยการปรับโมเดลด้วยข้อมูลจำลอง และ 3) ปัญหาที่เกิดจากการเลือกโมเดลแบบ MSTC ซึ่งทำให้มีโมเดลแบบผสมที่จำกัด และปัญหาพื้นที่ในการเก็บโมเดลของ Noise Cluster HMM และ Noisy Speech HMM ซึ่งสามารถแก้ได้ด้วย NCHI ส่วนเรื่องประสิทธิภาพการรู้จำเสียงพูดมีการทดลองในบทถัดไป

บทที่ 6
การทดลองและผลการทดลอง

วิทยานิพนธ์นี้นำเสนอเทคนิคใหม่ ในการปรับโมเดลแบบออนไลน์ สำหรับการรู้จำเสียงพูดแบบคงทน ด้วยการปรับโมเดลด้วยข้อมูลจำลองร่วมกับการประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดลที่มีการจัดกลุ่มเสียงรบกวน (NCHI) วิธีนี้สามารถเพิ่มประสิทธิภาพในการรู้จำเสียงพูด โดยการปรับโมเดลให้มีความใกล้เคียงกับเสียงพูดที่เข้ามามากขึ้น ด้วยการปรับโมเดลด้วยข้อมูลจำลอง [17] จากเสียงพูดที่เข้ามา เพื่อใช้ในการปรับโมเดลที่ได้จากการผสมขึ้นมาใหม่ด้วยการประมาณค่าในช่วงของ HMM [24],[25]

โดยประเด็นที่จะกล่าวถึงในบทนี้ประกอบด้วย หัวข้อ 6.1 ระบบรู้จำเสียงพูดและข้อมูลที่ใช้ในการทดลอง จากนั้นกล่าวถึงผลการทดลองซึ่งแบ่งออกเป็น 2 ส่วน คือ หัวข้อ 6.2 ผลการทดลองการปรับโมเดลด้วยข้อมูลจำลอง และหัวข้อ 6.3 ผลการทดลองประสิทธิภาพของโมเดลที่ได้จากการทำ NCHI เพียงอย่างเดียว และผลการปรับโมเดลด้วยข้อมูลจำลองร่วมกับ NCHI

6.1 ระบบรู้จำเสียงพูดและข้อมูลที่ใช้ในการทดลอง

ในหัวข้อนี้ประกอบด้วยหัวข้อย่อยจำนวน 4 หัวข้อ คือ หัวข้อ 6.1.1 ระบบรู้จำเสียงพูด ซึ่งได้กล่าวถึง โมเดล Noisy Speech HMM และ Noise Cluster HMM จากนั้นในหัวข้อที่ 6.1.2 – 6.1.4 เป็นการกล่าวถึงข้อมูลที่ใช้ทดลอง ซึ่งแบ่งออกเป็น 3 ชุด คือ หัวข้อ 6.1.2 ข้อมูลที่ใช้ในการฝึกสอน หัวข้อ 6.1.3 ข้อมูลที่ใช้ทดลอง และหัวข้อ 6.1.4 ข้อมูลของ Train-S

6.1.1 ระบบรู้จำเสียงพูด

โมเดลเสียงพูดของระบบรู้จำเสียงพูดที่ใช้ในการทดลองมี 3 แบบ คือ 1) โมเดลแบบฝึกสอนแบบหลากสภาวะ (MULTI) [14], 2) โมเดลแบบการเลือกโมเดลที่มีการจัดกลุ่มแบบโครงสร้างต้นไม้ (MSTC) [5] และ 3) โมเดลแบบ NCHI ในการนำไปใช้งานนั้น โมเดลแบบ MULTI มีเพียงโมเดล Noisy Speech HMM เท่านั้น ในขณะที่โมเดลแบบ MSTC และโมเดลแบบ NCHI ต่างประกอบด้วยโมเดล Noisy Speech HMM และ Noise Cluster HMM โดยโมเดลทั้งสองประเภทนี้มีโครงสร้างที่อธิบายได้ดังนี้

- โมเดล Noisy speech HMM ที่ใช้ในการทดลอง คือ HMM ที่มีการเปลี่ยนสถานะแบบซ้ายไปขวา ซึ่งมีจำนวนสเตทเท่ากับ 5 สเตท โดยแต่ละสเตทของ HMM มีเกาส์เซียนมิกเจอร์เท่ากับ 16 ส่วนประกอบ และแต่ละ HMM แทนด้วยหนึ่งหน่วยเสียง ซึ่งมีจำนวนโมเดลที่ใช้ในระบบรู้จำเสียงพูดทั้งสิ้น 76 โมเดล [26] ดังแสดงในตารางที่ 4.1 ส่วนลักษณะสำคัญ

ใช้ MFCC 39 พารามิเตอร์ (12 MFCC, 1 Log-energy, First Derivative และ Second Derivative) [27] การสร้าง Noisy Speech HMM ของโมเดลแบบ MULTI ใช้การปรับโมเดลด้วยวิธีการฝึกสอนซ้ำ ส่วนการสร้าง Noisy Speech HMM ของโมเดลแบบ MSTC และ NCHI ใช้การปรับโมเดลวิธี MLLR

- โมเดล Noise Cluster HMM ที่ใช้ในการทดลอง คือ HMM ที่มีการเปลี่ยนสถานะแบบซ้ายไปขวา ซึ่งมีจำนวนสเตทเท่ากับ 5 สเตท โดยแต่ละสเตทของ HMM มีเกาส์เซียนมิกเจอร์เท่ากับ 16 ส่วนประกอบ และแต่ละ HMM แทนด้วยหนึ่งเสียงรบกวน ส่วนลักษณะสำคัญใช้ MFCC 12 พารามิเตอร์ การสร้าง Noisy Cluster HMM ของโมเดลแบบ MSTC และ NCHI ใช้การปรับโมเดลวิธี MLLR

6.1.2 ข้อมูลฝึกสอน

ข้อมูลชุดฝึกสอนที่ใช้สร้างโมเดล Noisy speech HMM และ Noise cluster HMM คือ เสียงพูดที่มีเสียงรบกวนแบบบวก โดยข้อมูลที่ใช้สร้างชุดฝึกสอนประกอบด้วยเสียงพูดสะอาดจากคลังข้อมูลเสียงพูดภาษาไทย NECTEC-ATR [44] และเสียงรบกวนจากคลังข้อมูลเสียงรบกวนมาตรฐานของ Japan Electronic Industry Development Association (JEIDA) [45] และคลังข้อมูลเสียงรบกวนมาตรฐาน NOISEX-92 [46] ซึ่งข้อมูลเสียงพูดสะอาดและเสียงรบกวนมีการลดอัตราการสุ่มให้มีความถี่ของการสุ่มสัญญาณเท่ากับ 8 KHz และมีการบวกเสียงรบกวนกับเสียงพูดสะอาดที่ระดับ SNR เท่ากับ 15 dB, 10 dB และ 5 dB

เสียงพูดสะอาดที่ใช้ฝึกสอนเป็นเสียงพูดคำโดดจากผู้พูดผู้ชายทั้งหมด 16 คน โดยแต่ละคน พูดคนละ 1,000 คำ และมีคำที่อยู่ในชุดฝึกสอนทั้งหมด 5,000 คำ และส่วนเสียงรบกวนมีทั้งหมด 9 ชนิด ซึ่งมาจากคลังข้อมูลเสียงรบกวน JEIDA 8 ชนิด ได้แก่ Street, Factory, Station, Air Condition, Road, Elevator, Exhibition และ Train ส่วนเสียงรบกวนอีกหนึ่งชนิด ได้แก่ Car มาจากคลังข้อมูลเสียงรบกวน NOISEX-92 ดังนั้นจำนวนโมเดลของเสียงรบกวนพื้นฐาน ที่ใช้ทดลองในงานวิทยานิพนธ์นี้มีเท่ากับ 28 โมเดล (9 เสียงรบกวน x 3 SNR + 1 เสียงสะอาด)

6.1.3 ข้อมูลทดลอง

ข้อมูลทดลองเป็นเสียงพูดคำโดดที่มีเสียงรบกวน โดยแบ่งข้อมูลออกเป็น 2 ชุด ตามประเภทของเสียงรบกวน คือ ชุด Test-1 เป็นเสียงรบกวนแบบบวก และชุด Test-2 เป็นเสียงรบกวนจากสิ่งแวดล้อมจริง

ชุดทดลอง Test-1 คือ เสียงพูดที่มีเสียงรบกวนแบบบวก โดยเสียงพูดสะอาดมาจากคลังข้อมูลเสียงพูดภาษาไทย NECTEC-ATR ส่วนเสียงรบกวนมาจากคลังข้อมูลเสียงรบกวน JEIDA, NOISEX-92 และเสียงรบกวนที่บันทึกจากงานนิทรรศการ NSTDA Annual Conference (NAC) 2005 ซึ่งเสียงรบกวนทั้งสามเป็นเสียงรบกวนที่ไม่ได้อยู่ในชุดฝึกสอน ข้อมูลเสียงพูดสะอาดและ

เสียงรบกวนมีการลดอัตราการสุ่มให้มีความถี่ของการสุ่มสัญญาณเท่ากับ 8 KHz และมีการบวกเสียงรบกวนกับเสียงพูดสะอาดที่ระดับ SNR เท่ากับ 15 dB, 10 dB และ 0 dB

เสียงพูดสะอาดที่ใช้ทดลองเป็นเสียงพูดคำโดดจากผู้พูดผู้ชายทั้งหมด 5 คน โดยแต่ละคนจะพูดคนละ 640 คำ และมีคำที่อยู่ในชุดทดลอง Test-1 ทั้งหมด 640 คำ ส่วนเสียงรบกวนมีทั้งหมด 3 ชนิด คือ ชนิดที่หนึ่ง Computer Room มาจากคลังข้อมูลเสียงรบกวน JEIDA, ชนิดที่สอง White Noise มาจากคลังข้อมูลเสียงรบกวน NOISEX-92 และชนิดที่สาม เสียงรบกวนจากงาน NAC2005 ดังนั้นในการทดลองเสียงรบกวนหนึ่งชนิดและหนึ่ง SNR จะมีตัวอย่างที่ใช้ทดลองทั้งหมดเท่ากับ 3,200 ตัวอย่าง (5 คน x 640 ตัวอย่าง) สำหรับข้อมูลผู้พูดและเสียงรบกวนที่ใช้ในการทดลองเป็นข้อมูลที่ไม่ได้อยู่ในชุดฝึกสอน

ชุดทดลอง Test-2 คือ เสียงพูดที่มีเสียงรบกวนจากสิ่งแวดล้อมจริง ซึ่งได้บันทึกเสียงพูดที่ความถี่ของการสุ่มสัญญาณเท่ากับ 8 KHz จากงานนิทรรศการ Thailand ICTEXPO 2005 โดยเป็นเสียงพูดคำโดดจากผู้พูดผู้ชายจำนวนทั้งสิ้น 50 คน และมีคำที่อยู่ในชุดทดลอง Test-2 ทั้งหมด 76 คำ มีตัวอย่างที่ใช้ทดลองทั้งสิ้นเท่ากับ 760 ตัวอย่าง เสียงพูดที่ใช้ทดลองในชุดนี้มี SNR อยู่ที่ช่วง 5 – 0 dB เห็นได้ว่า SNR ที่เกิดขึ้นในสภาพแวดล้อมจริงนั้นไม่ได้มีแค่ SNR เพียงค่าเดียว ทั้งนี้เป็นเพราะเสียงรบกวนเป็นสัญญาณแบบไม่มีความคงที่

6.1.4 ข้อมูล Train-S

ข้อมูล Train-S คือ ข้อมูลเสียงพูดสะอาด ซึ่งใช้ในกรรมวิธีการปรับโมเดลด้วยข้อมูลจำลอง ขั้นตอนการคัดเลือกเสียงพูดในชุด Train-S แสดงดังหัวข้อที่ 4.1 และเนื่องจากต้องการลดผลกระทบของการปรับโมเดลในด้านอื่นที่ไม่ได้เกิดมาจากเสียงรบกวน ไม่ว่าจะเป็นจากเพศ หรือหน่วยเสียง จึงทำให้มีการเลือก Train-S จากเฉพาะข้อมูลเสียงพูดของผู้ชายที่พูดคำศัพท์เดียวกันเท่านั้น นอกจากนี้ คำศัพท์ที่ใช้ใน Train-S ต้องมีหน่วยเสียงครบทั้งหมดด้วย ด้วยเหตุผลดังกล่าวทำให้ Train-S มีจำนวนผู้ชายมากที่สุดเพียง 4 คน (M1, M2, M3 และ M4) ที่มีตัวอย่างเสียงพูดที่มีคำเหมือนกัน และเป็นตัวอย่างเสียงพูดที่นำไปรู้จำเสียงพูดแล้วมีความถูกต้อง โดยคำที่ใช้ทดลองใน Train-S คือ 1) เครื่องไมโครคอมพิวเตอร์ 2) รองนายกรัฐมนตรี 3) พฤหัสบดี 4) เจริญงอกงาม 5) ราชโองการ 6) ทฤษฎีบท 7) จุดหมายปลายทาง 8) ถือกำเนิด 9) ข้อได้เปรียบ 10) ข้อบกพร่อง 11) นครหลวง 12) ความเป็นมา 13) ข้อผิดพลาด 14) กลั่นกรอง 15) แผ่นดิสก์ 16) ครอบคลุม 17) แบบฟอร์ม 18) จนกว่า 19) เคเบิ้ล 20) ศึกษา 21) ผู้ว่า 22) ฟรอยด์ 23) สถาบันวิจัย 24) ห้างสรรพสินค้า 25) วัตถุนิยม 26) รถบรรทุก 27) พระไตรปิฎก 28) ใบอนุญาต 29) วิกฤตการณ์ 30) คำอธิบาย 31) ผู้ประกอบการ 32) เห็นอกเห็นใจ ในการทดลองเรื่องจำนวนคำของ Train-S มีการแบ่งกลุ่มคำข้างต้นในการทดลองเป็น 8 คำ (ลำดับคำที่ 1 – 8) , 22 คำ (ลำดับคำที่ 1 – 22) และ 32 คำ (ลำดับคำที่ 1 – 32) ซึ่งจำนวนคำที่น้อยที่สุดที่มีหน่วยเสียงครบ คือ 22 คำ

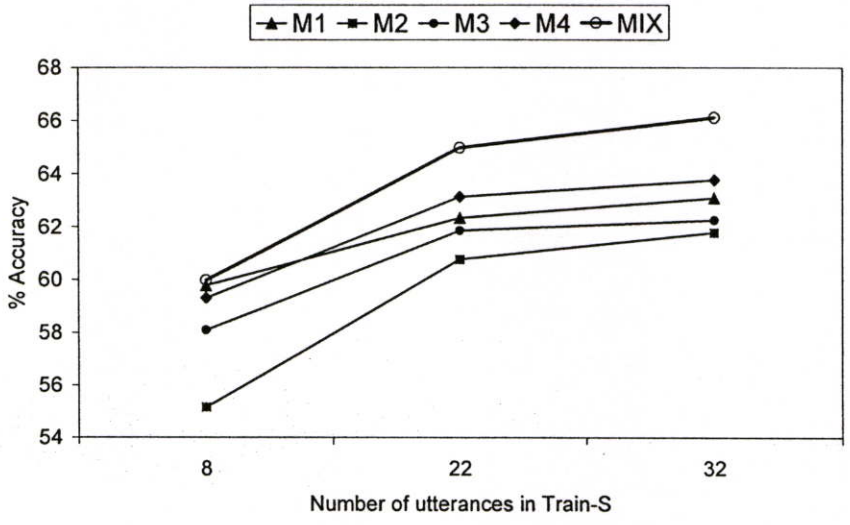
6.2 ผลการทดลองการปรับโมเดลด้วยข้อมูลจำลอง

ในหัวข้อนี้กล่าวถึง ผลการทดลองพารามิเตอร์ต่างๆ ที่ใช้ในการปรับโมเดลด้วยข้อมูลจำลอง แบ่งออกเป็น 4 ผลการทดลอง คือ 1) ผลการทดลองจำนวนคำและผู้พูดของ Train-S 2) ผลการทดลองกับวิธีการปรับโมเดลแบบต่างๆ 3) ผลการทดลองการดึงเสียงรบกวน และ 4) ผลการทดลองรูปแบบการปรับโมเดลด้วยข้อมูลจำลอง ซึ่งแสดงผลการทดลองในหัวข้อ 6.2.1, 6.2.2, 6.2.3 และ 6.2.4 ตามลำดับ นอกจากนี้ ในหัวข้อ 6.25 คือ การวิเคราะห์และเปรียบเทียบกับวิธีต่างๆ

6.2.1 ผลการทดลองจำนวนคำและผู้พูดของ Train-S

หัวข้อนี้เป็นการทดลองการเลือกผู้พูด และจำนวนคำของ Train-S ซึ่งมีผู้พูดที่ใช้ในการทดลองจำนวนทั้งสิ้น 4 คน คือ M1, M2, M3 และ M4 นอกจากนี้ ยังมีการเลือกผู้พูดแบบ MIX คือ การนำตัวอย่างเสียงจากผู้พูดทั้ง 4 คนมาใช้ จึงกล่าวได้ว่า มีจำนวนผู้พูดที่ใช้ทดลองทั้งสิ้น 5 คน คือ M1, M2, M3, M4 และ MIX ส่วนจำนวนคำที่ใช้ในการทดลองมีจำนวนเท่ากับ 8 คำ, 22 คำ และ 32 คำ

การทดลองการเลือกผู้พูดของ Train-S ด้วยการวัดผลการรู้จำเสียงพูดของ Test-1 ด้วยการปรับโมเดลด้วยข้อมูลจำลอง รูปแบบที่หนึ่ง ดังตารางที่ 4.2 ในหัวข้อ 4.5 และใช้โมเดลตั้งต้นแบบ MULTI ซึ่งเรียกว่า "S-MULTI" และใช้การปรับโมเดลวิธี MLLR ให้ผลการทดลอง ดังรูปที่ 6.1

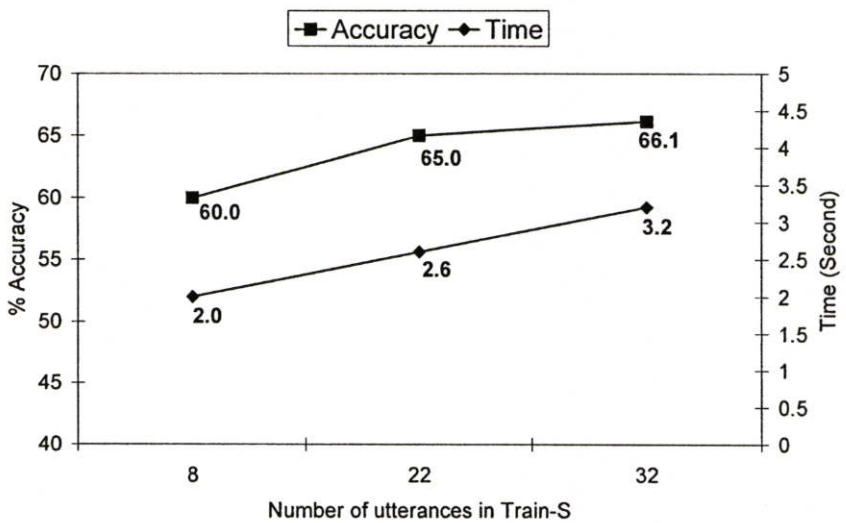


รูปที่ 6.1 ค่าเฉลี่ยของผลการรู้จำเสียงพูดของ Test-1 ด้วย S-MULTI ที่ใช้ผู้พูด และจำนวนคำใน Train-S ที่แตกต่างกัน

จากผลการทดลองในรูปที่ 6.1 แสดงให้เห็นว่า ทุกจำนวนคำของ Train-S ที่ใช้เสียงพูดของผู้พูดแบบ MIX ให้ผลการรู้จำเสียงพูดได้ดีที่สุด ทั้งนี้ปันผลจากการใช้เสียงพูดของผู้พูดคนใดคนหนึ่งอาจทำให้โมเดลที่ปรับได้ขึ้นอยู่กับผู้พูดของ Train-S มากกว่าผู้พูดที่ใช้งาน ดังนั้นถ้าผู้พูดของ Train-S มีเสียงพูดที่แตกต่างกับผู้พูดที่ใช้งานมาก ย่อมทำให้ผลการรู้จำเสียงพูดลดลงได้ แต่ถ้าใช้ผู้

พูดแบบ MIX ซึ่งมาจากผู้พูดหลายคนน่าจะช่วยให้โมเดลที่ปรับได้ขึ้นอยู่กับผู้พูดของ Train-S น้อยลง นอกจากนี้ ผลการรู้จำเสียงพูดยังเพิ่มขึ้นตามจำนวนคำของ Train-S อย่างไรก็ตาม ผลการรู้จำเสียงพูดในการทดลองนี้ยังคงเพิ่มขึ้นตามจำนวนคำของ Train-S และถึงแม้ว่าอัตราการเพิ่มขึ้นของผู้พูดแต่ละผู้พูดจะไม่เท่ากัน ผลการทดลองก็ยังแสดงให้เห็นได้ว่าจำนวนคำของ Train-S ที่ใช้ในการทดลองนี้ยังไม่มากพอที่จะทำให้โมเดลที่ปรับได้ขึ้นอยู่กับผู้พูดของ Train-S เพราะถ้าขึ้นอยู่กับผู้พูดของ Train-S ย่อมมีแนวโน้มของผลการรู้จำเสียงพูดลดลง

การทดลองผลของจำนวนคำของ Train-S ด้วยการวัดผลการรู้จำเสียงพูด และเวลาที่ใช้คำนวณของ Test-1 ด้วย S-MULTI ที่มีการใช้ผู้พูดของ Train-S แบบ MIX นั้น ในวิทยานิพนธ์นี้ได้วัดเวลาในการคำนวณของ S-MULTI บนเครื่องคอมพิวเตอร์ Intel Pentium IV 3.2 GHz CPU และ 2 GB RAM ซึ่งมีผลการทดลองดังรูปที่ 6.2



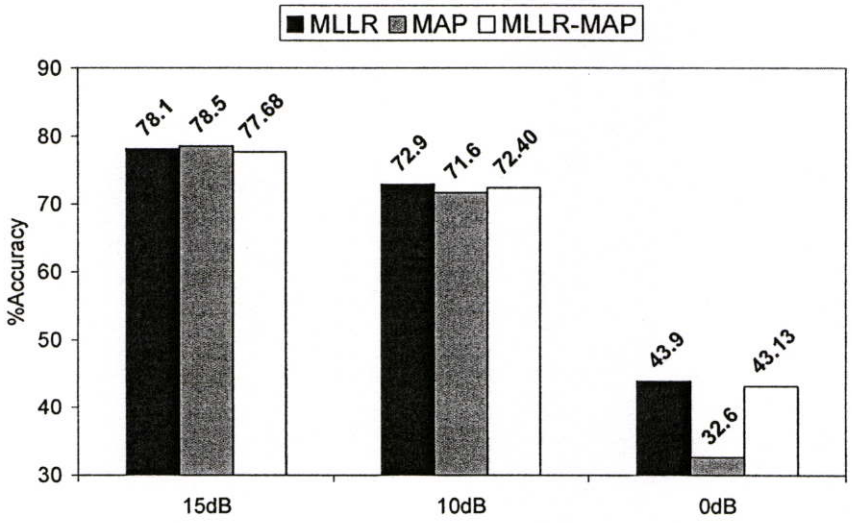
รูปที่ 6.2 ค่าเฉลี่ยของผลการรู้จำเสียงพูด และเวลาในการคำนวณของ Test-1 ด้วย S-MULTI ที่จำนวนคำใน Train-S แตกต่างกัน

จากผลการทดลองในรูปที่ 6.2 แสดงให้เห็นว่า S-MULTI ที่ใช้ผู้พูดของ Train-S แบบ MIX ที่จำนวนคำของ Train-S เท่ากับ 32 คำ ให้ผลการรู้จำเสียงพูดได้ดีที่สุด แต่ใช้เวลาคำนวณมากที่สุด

วิทยานิพนธ์นี้จึงเลือกใช้ Train-S ที่มาจากผู้พูดแบบ MIX เพราะให้ผลการรู้จำเสียงพูดได้ดีที่สุด และเลือกใช้จำนวนคำของ Train-S เท่ากับ 22 คำ เพราะเป็นจำนวนคำที่ใช้เวลาในการคำนวณไม่มากเกินไป นอกจากนี้ จำนวนคำของ Train-S เท่ากับ 22 คำ เป็นจำนวนคำที่น้อยที่สุด ซึ่งสามารถครอบคลุมหน่วยเสียงในภาษาไทยครบ 76 หน่วยเสียง

6.2.2 ผลการทดลองกับวิธีการปรับโมเดลแบบต่างๆ

หัวข้อนี้เป็นการทดลองการปรับโมเดลด้วย S-MULTI ที่ใช้วิธีการปรับโมเดลแบบต่างๆ ซึ่งวิธีการปรับโมเดล ที่ใช้ทดลองในวิทยานิพนธ์นี้ คือ MLLR [7], MAP [16] และ MLLR-MAP [39] โดยแสดงผลการทดลอง ดังรูปที่ 6.3

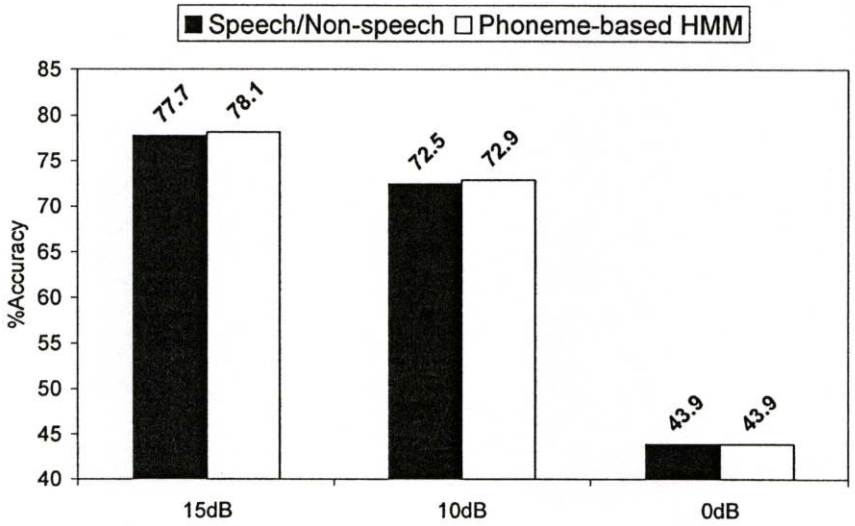


รูปที่ 6.3 ค่าเฉลี่ยของผลการรู้จำเสียงพูดของ Test-1 ด้วย S-MULTI แล้วนำไปปรับโมเดลด้วยวิธีที่แตกต่างกัน

จากผลการทดลองในรูป 6.3 เมื่อคำนวณเป็นค่าเฉลี่ยของผลการรู้จำเสียงพูดของ Test-1 ด้วย S-MULTI แล้วนำไปปรับโมเดลแบบวิธี MLLR, วิธี MAP และ วิธี MLLR-MAP ให้ผลการรู้จำเสียงพูด คือ 65.0%, 61.0% และ 64.4% ตามลำดับ ส่วนค่าเฉลี่ยของผลการรู้จำเสียงพูดของ Test-1 ด้วยการปรับโมเดลตั้งต้นแบบ MULTI ด้วยวิธี MLLR, วิธี MAP และวิธี MLLR-MAP ที่ไม่ได้ใช้การปรับโมเดลด้วยข้อมูลจำลอง ให้ผลการรู้จำเสียงพูด คือ 60.9%, 60.1% และ 60.1% ตามลำดับ แสดงให้เห็นว่าการปรับโมเดลด้วยวิธี MLLR ให้ผลการรู้จำเสียงพูดได้ดีที่สุด ในขณะที่การปรับโมเดลวิธี MAP และวิธี MLLR-MAP ให้ผลการการรู้จำเสียงพูดได้รองลงมา ทั้งนี้เป็นเพราะจำนวนคำของ Train-S ที่ใช้ปรับโมเดลมีจำนวนไม่มากพอ อย่างไรก็ตาม เห็นได้ว่า S-MULTI สามารถเพิ่มประสิทธิภาพวิธีการปรับโมเดลแบบอื่นๆ ที่ไม่ใช่เฉพาะวิธี MLLR ได้ด้วย วิทยานิพนธ์นี้จึงเลือกใช้การปรับโมเดลด้วยวิธี MLLR เพราะให้ผลการรู้จำเสียงพูดดีที่สุด

6.2.3 ผลการทดลองการดึงเสียงรบกวน

หัวข้อนี้เป็นการทดลองวิธีการดึงเสียงรบกวนแบบต่างๆ ของ S-MULTI ซึ่งวิธีการดึงเสียงรบกวนที่ใช้ทดลองในวิทยานิพนธ์นี้ คือ วิธีการหาส่วนที่เป็นเสียงพูดและส่วนที่ไม่เป็นเสียงพูด และ วิธีการอิงกับหน่วยเสียง โดยแสดงผลการทดลอง ดังรูปที่ 6.4



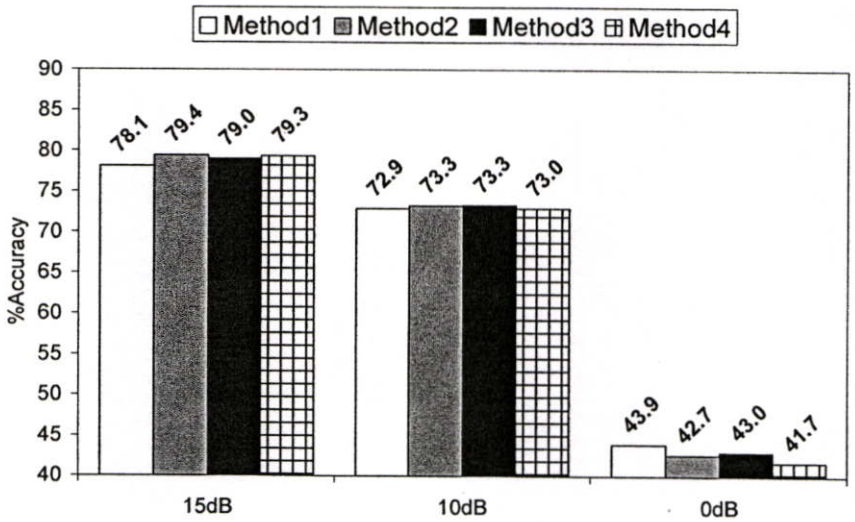
รูปที่ 6.4 ค่าเฉลี่ยของผลการรู้จำเสียงพูดและเวลาในการคำนวณของ Test-1 ด้วย S-MULTI ที่มีการดึงเสียงรบกวนที่แตกต่างกัน

จากผลการทดลองในรูป 6.4 เมื่อคำนวณเป็นค่าเฉลี่ยของผลการรู้จำเสียงพูดของ Test-1 ด้วย S-MLLR ที่ใช้วิธีการหาส่วนที่เป็นเสียงพูดและส่วนที่ไม่เป็นเสียงพูดกับที่ใช้วิธีการอิงกับหน่วยเสียงในการดึงเสียงรบกวน ให้ผลการรู้จำเสียงพูด คือ 64.7% และ 65.0% ตามลำดับ แสดงให้เห็นว่า S-MULTI ที่ดึงเสียงรบกวนด้วยวิธีการอิงกับหน่วยเสียง ให้ผลการรู้จำเสียงพูดที่ดีกว่าวิธีการหาส่วนที่เป็นเสียงพูดและส่วนที่ไม่เป็นเสียงพูด เป็นผลจากการหาตำแหน่งเสียงเงียบด้วยการดึงเสียงรบกวนแบบวิธีการอิงกับหน่วยเสียงหาตำแหน่งเสียงเงียบได้ดีกว่าที่ระดับ SNR สูง ส่วนที่ระดับ SNR ต่ำให้ผลการหาตำแหน่งเสียงเงียบที่ใกล้เคียงกัน ซึ่งการหาตำแหน่งเสียงเงียบที่ให้ผลดีกว่าทำให้การสเกลความยาวมีจำนวนครั้งในการนำเสียงเงียบมาต่อกันน้อยลง ทำให้สามารถลดเสียงรบกวนที่เกิดจากการต่อเสียงเงียบเข้าด้วยกันได้ เป็นผลให้ผลการรู้จำเสียงพูดดีขึ้นไปด้วย

ส่วนเวลาในการคำนวณด้วย S-MULTI ที่ใช้วิธีการหาส่วนที่เป็นเสียงพูดและส่วนที่ไม่เป็นเสียงพูด และวิธีการอิงกับหน่วยเสียงในการดึงเสียงรบกวน คือ 2.1 วินาที และ 2.6 วินาที ตามลำดับ เห็นได้ว่าการดึงเสียงรบกวนด้วยวิธีการหาส่วนที่เป็นเสียงพูดและส่วนที่ไม่เป็นเสียงพูดใช้เวลาในการคำนวณน้อยกว่า เพราะโมเดลที่ใช้ในการหาเสียงรบกวนมีน้อยกว่านั่นเอง อย่างไรก็ตาม วิธีการหาส่วนที่เป็นเสียงพูดและส่วนที่ไม่เป็นเสียงพูดต้องมีการเก็บโมเดลเพิ่มขึ้น แต่วิธีการอิงกับหน่วยเสียงไม่ต้องเก็บเพิ่ม เนื่องจากสามารถใช้โมเดลที่ใช้ในการรู้จำเสียงพูดในการหาส่วนที่เป็นเสียงรบกวน วิทยานิพนธ์นี้จึงเลือกใช้วิธีการอิงกับหน่วยเสียงในการดึงเสียงรบกวน เพราะไม่ต้องเก็บโมเดลของการดึงเสียงรบกวนเพิ่ม และให้ผลการรู้จำเสียงพูดที่ดีกว่า

6.2.4 ผลการทดลองรูปแบบการปรับโมเดลด้วยข้อมูลจำลอง

หัวข้อนี้เป็นการทดลองรูปแบบต่างๆ ของ S-MULTI ดังตารางที่ 4.2 ในหัวข้อ 4.5 ซึ่งเป็นการทดลองการใช้หรือไม่ใช้เสียงพูดที่เข้ามา ร่วมกับข้อมูลจำลอง และการทดลองการใช้หรือไม่ใช้การเลือกผลการรู้จำเสียงพูด จากคำตอบของโมเดลตั้งต้นหรือโมเดลที่ได้จากการปรับโมเดล โดย S-MULTI ที่ใช้ในวิทยานิพนธ์นี้มีรูปแบบการทดลองทั้งหมด 4 แบบ และเรียกแต่ละแบบว่า Method1, Method2, Method3 และ Method4 ตามลำดับ ซึ่งมีผลการทดลอง ดังรูปที่ 6.5



รูปที่ 6.5 ค่าเฉลี่ยของผลการรู้จำเสียงพูดของ Test-1 ด้วย S-MULTI ที่ใช้รูปแบบการปรับโมเดลด้วยข้อมูลจำลองที่แตกต่างกัน

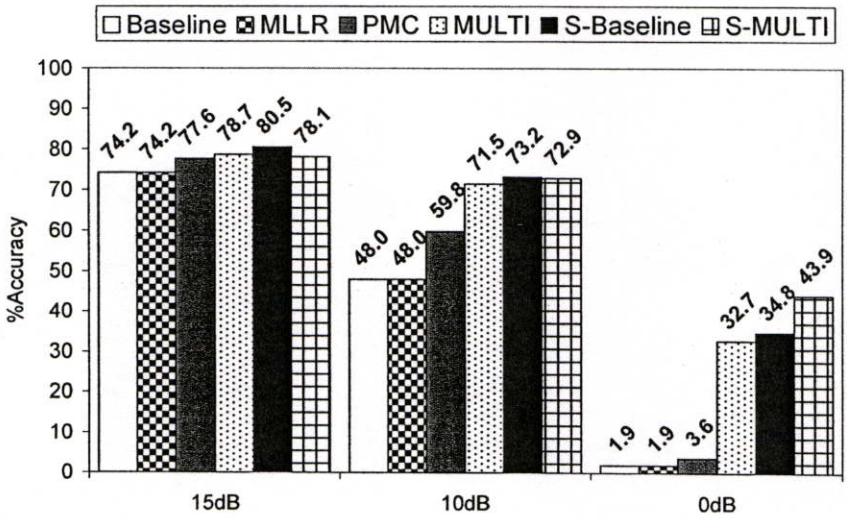
จากผลการทดลองในรูป 6.5 เมื่อคำนวณเป็นค่าเฉลี่ยของผลการรู้จำเสียงพูดของ Test-1 ด้วย S-MULTI ทั้ง 4 แบบให้ผลการรู้จำเสียงพูด คือ 65.0%, 65.1%, 65.1% และ 64.8% ตามลำดับ จากผลการทดลอง S-MULTI ทั้ง 4 แบบ ให้ผลการรู้จำเสียงพูดที่ใกล้เคียงกัน เป็นเพราะว่า 1) การไม่ทราบคำอ่านที่ถูกต้อง และเสียงพูดที่เข้ามามีขนาดเล็ก จึงทำให้โมเดลที่ปรับได้ มีการผลการรู้จำเสียงพูดที่ไม่เพิ่มขึ้น แต่ถ้าเสียงพูดที่เข้ามาเป็นเสียงพูดแบบต่อเนื่อง โมเดลที่ได้จากการปรับโมเดลด้วยเสียงพูดที่เข้ามาร่วมกับข้อมูลจำลองให้ผลการรู้จำเสียงพูดที่ดีกว่า [18] ทั้งนี้เป็นเพราะเสียงพูดที่เข้ามีขนาดที่มากเพียงพอต่อการปรับโมเดลนั่นเอง แต่อย่างไรก็ตาม การปรับโมเดลด้วยข้อมูลจำลอง ก็ยังให้ผลการรู้จำเสียงพูดที่ดีกว่าการใช้เสียงพูดที่เข้ามาเพียงอย่างเดียวในการปรับโมเดล และ 2) การเลือกผลการรู้จำเสียงพูด ไม่ทำให้ผลการรู้จำเสียงพูดเพิ่มขึ้น เป็นเพราะโมเดลส่วนใหญ่ที่ปรับโมเดลด้วยข้อมูลจำลอง ให้ผลการรู้จำเสียงพูดที่ดีกว่าโมเดลตั้งต้น วิทยานิพนธ์นี้จึงเลือกใช้ S-MULTI รูปแบบที่หนึ่ง สำหรับข้อมูลเสียงพูดคำโดด เพราะใช้เวลาในการคำนวณน้อยที่สุด เนื่องจากไม่ต้องใช้เสียงพูดที่เข้ามาในการปรับโมเดล และไม่ต้องมีการเลือกโมเดลอีกครั้ง

6.2.5 การวิเคราะห์และเปรียบเทียบกับวิธีต่างๆ

หัวข้อนี้เป็นการทดลองเปรียบเทียบผลการรู้จำเสียงพูด ระหว่างวิธีปรับโมเดลวิธีต่างๆ กับวิธีการปรับโมเดลด้วยข้อมูลจำลอง ซึ่งวิธีที่นำมาเปรียบเทียบ คือ

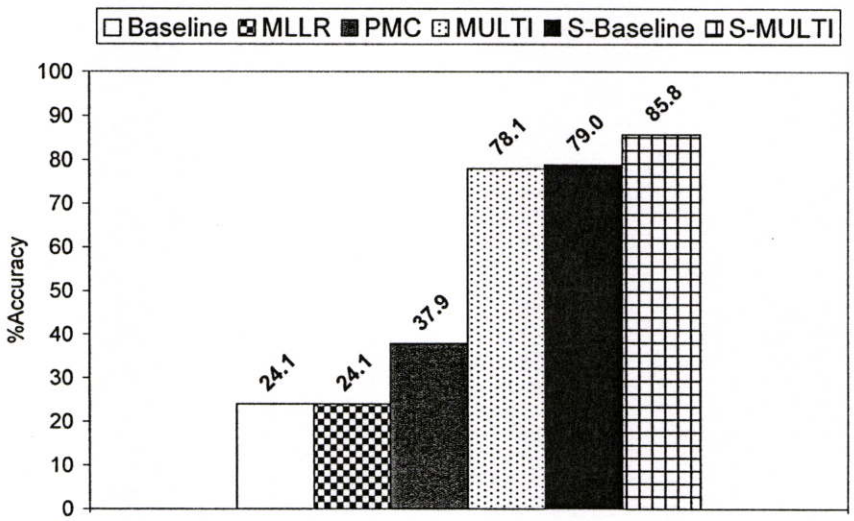
- โมเดลเสียงสะอาด เรียกว่า "Baseline"
- โมเดลแบบ Baseline ที่มีการปรับโมเดลด้วยวิธี MLLR เรียกว่า "MLLR"
- โมเดลแบบ Baseline ที่มีการปรับโมเดลด้วยวิธี PMC เรียกว่า "PMC"
- โมเดลแบบหลากสภาวะ เรียกว่า "MULTI"
- โมเดลแบบ Baseline ที่มีการปรับโมเดลด้วยวิธี MLLR ร่วมกับการปรับโมเดลด้วยข้อมูลจำลองแบบที่หนึ่ง เรียกว่า "S-Baseline"
- โมเดลแบบ MULTI ที่มีการปรับโมเดลด้วยวิธี MLLR ร่วมกับการปรับโมเดลด้วยข้อมูลจำลองแบบที่หนึ่ง เรียกว่า "S-MULTI"

ในการทดลองการปรับโมเดลด้วยข้อมูลจำลองของ S-Baseline และ S-MULTI ที่ใช้การดึงเสียงรบกวนแบบการอิงกับหน่วยเสียงและใช้ Train-S ด้วยผู้พูดแบบ MIX ที่มีจำนวนคำเท่ากับ 22 คำ ซึ่งมีผลการทดลองของ Test-1 และ Test-2 ดังรูปที่ 6.6 และ 6.7 ตามลำดับ สำหรับผลการทดลองของ Test-2 ไม่ได้แสดงผลแยกตามระดับ SNR เหมือน Test-1 เนื่องจาก Test-2 เป็นเสียงรบกวนในสภาพแวดล้อมจริง ทำให้มีค่าระดับ SNR หลายค่า จึงต้องแสดงผลเป็นค่าเฉลี่ยของผลการรู้จำเสียงพูดของ Test-2



รูปที่ 6.6 ผลการเปรียบเทียบผลการรู้จำเสียงพูดของ Test-1 ด้วย Baseline, MLLR, PMC, MULTI, S-Baseline และ S-MULTI

จากผลการทดลองในรูปที่ 6.6 เมื่อคำนวณเป็นค่าเฉลี่ยของผลการรู้จำเสียงพูดของ Test-1 ด้วย Baseline, MLLR, PMC, MULTI, S-Baseline และ S-MULTI ให้ผลการรู้จำเสียงพูด คือ 41.4%, 41.4%, 47.0%, 61.0%, 62.8% และ 65.0% ตามลำดับ



รูปที่ 6.7 ผลการเปรียบเทียบผลการรู้จำเสียงพูดของ Test-2 ด้วย Baseline, MLLR, PMC, MULTI, S-Baseline และ S-MULTI

จากผลการทดลองในรูปที่ 6.6 และรูปที่ 6.7 แสดงให้เห็นว่า

- S-MULTI ให้ผลการรู้จำเสียงพูดทั้งของ Test-1 และ Test-2 ได้ดีที่สุด
- S-Baseline ให้ผลการรู้จำเสียงพูดมากกว่า MLLR ซึ่งเป็นการยืนยันผลของการใช้ข้อมูลที่มีคำอ่านที่ถูกต้อง และขนาดของข้อมูลที่ใช้ในการปรับโมเดล มีผลต่อการปรับโมเดลแบบออนไลน์ด้วย MLLR ดังนั้นการที่สามารถเพิ่มข้อมูลด้วยการปรับโมเดลด้วยข้อมูลจำลองย่อมทำให้ได้ผลการรู้จำเสียงพูดที่ดีขึ้น
- PMC มีผลการรู้จำเสียงพูดมากกว่า MLLR ในการทดลองนี้ เป็นเพราะ PMC ไม่ต้องใช้คำอ่านในการปรับโมเดล และยังให้ผลการปรับโมเดลที่ดี แม้ว่าข้อมูลที่ใช้ในการปรับโมเดลมีขนาดเล็ก ส่งผลให้ PMC มีผลการรู้จำเสียงพูดที่ดีกว่า MLLR
- S-Baseline มีผลการรู้จำเสียงพูดมากกว่า PMC เป็นเพราะว่า โดยทั่วไปแล้ว ถ้า MLLR มีข้อมูลที่มากเพียงพอ แม้ว่าจะเป็นการปรับโมเดลแบบออนไลน์แบบ Unsupervised ก็ให้ผลการรู้จำเสียงพูดได้ดีกว่า PMC และยิ่งถ้าเป็นการทำ MLLR แบบ Supervised ดังเช่น การใช้การปรับโมเดลด้วยข้อมูลจำลอง ย่อมให้ผลการรู้จำเสียงพูดได้ดียิ่งขึ้นไปอีก ดังนั้น S-MULTI ซึ่งเป็นวิธีการที่สามารถเพิ่มจำนวนของข้อมูลและให้คำอ่านที่ถูกต้อง ย่อมส่งผลทำให้โมเดลที่ปรับได้มีผลการรู้จำเสียงพูดที่ดีขึ้นนั่นเอง

- S-Baseline และ S-MULTI มีผลการรู้จำเสียงพูดมากกว่า MULTI เป็นเพราะเสียงพูดที่มีเสียงรบกวนที่ใช้สร้าง MULTI ไม่ได้ครอบคลุมเสียงรบกวนที่เกิดขึ้นจริงทั้งหมด ทำให้โมเดลนี้ไม่สามารถรองรับเสียงรบกวนที่ไม่ได้อยู่ในชุดฝึกสอนได้ดีเท่าโมเดลที่มีการปรับโมเดลด้วยการจำลองข้อมูลจากเสียงพูดที่เข้ามา ส่วน S-MULTI ให้ผลการรู้จำเสียงพูดได้ดีกว่า S-Baseline นั้น เป็นเพราะโมเดลตั้งต้นของ S-MULTI มีผลการรู้จำเสียงพูดที่ดีกว่าโมเดลตั้งต้น S-Baseline นั่นเอง

จากผลการทดลองแสดงให้เห็นว่าการปรับโมเดลด้วยวิธี MLLR ที่ใช้ข้อมูลจำลอง สามารถแก้ปัญหาของเสียงพูดที่เข้ามามีขนาดเล็ก และไม่ทราบคำอ่านของเสียงพูดที่เข้ามาได้ เป็นผลให้โมเดลเสียงพูดที่ได้จากการปรับโมเดล ที่วิทยานิพนธ์นี้นำเสนอ มีผลการรู้จำเสียงพูดดีกว่าการปรับโมเดลด้วยวิธี MLLR และวิธี PMC ที่ใช้เฉพาะเสียงพูดที่เข้ามาเพียงอย่างเดียว

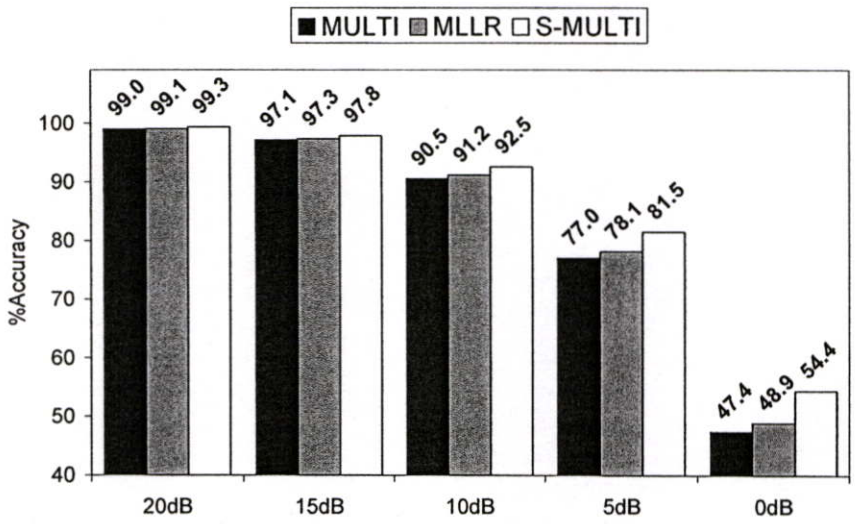
สำหรับเวลาที่ใช้ในการคำนวณการปรับโมเดลด้วย PMC, MLLR, S-Baseline และ S-MULTI คือ 12 วินาที, 1 วินาที, 2.6 วินาที และ 2.6 วินาที ตามลำดับ เห็นได้ว่าเวลาในการคำนวณของ PMC นั้น มากกว่าเวลาในการคำนวณของ MLLR, S-Baseline และ S-MULTI อยู่มาก ทำให้ PMC ไม่เหมาะสำหรับการปรับโมเดลแบบออนไลน์ สาเหตุที่ PMC ใช้เวลาในการคำนวณมาก เกิดจากการแปลงโดเมนไปกลับระหว่างโดเมนเซปสตรอลและโดเมนเสปคตรอลเชิงเส้น [4],[13] ส่วนเวลาที่ใช้ในการคำนวณของ S-Baseline และ S-MULTI เป็นเวลาที่ผู้ใช้งานสามารถยอมรับได้ ทำให้การปรับโมเดลด้วยข้อมูลจำลองเหมาะสม สำหรับการปรับโมเดลแบบออนไลน์มากกว่า PMC

นอกจากนี้ ยังได้ทดลองใช้การปรับโมเดลด้วยข้อมูลจำลองกับเสียงพูดภาษาอื่นๆ ในวิทยานิพนธ์นี้ทดลองกับเสียงพูดภาษาญี่ปุ่นจากคลังข้อมูลมาตรฐานออโรร่าทูเจ (AURORA-2J) [14] ซึ่งเป็นคลังข้อมูลเสียงพูดที่มีเสียงรบกวนแบบบวก และมีเสียงพูดแบบต่อเนื่องที่เป็นการพูดตัวเลข (0 – 9) เรียงต่อกัน โดยในแต่ละตัวอย่างมีการพูดตัวเลขเรียงต่อกันประมาณ 4 ตัว ข้อมูลชุดทดลองของออโรร่าทูเจแบ่งออกเป็น 3 ชุด คือ 1) ข้อมูลชุด TestsetA ประกอบด้วยเสียงรบกวน Subway, Babble, Car และ Exhibition ซึ่งเป็นเสียงรบกวนชนิดเดียวกันกับข้อมูลชุดฝึกสอน 2) ข้อมูลชุด TestsetB ประกอบด้วยเสียงรบกวน Restaurant, Street, Airport และ Station ซึ่งเป็นเสียงรบกวนคนละชนิดกับข้อมูลชุดฝึกสอน และ 3) ข้อมูลชุด TestsetC ประกอบด้วยเสียงรบกวน Subway และ Street ซึ่งเป็นเสียงรบกวนชนิดเดียวกันกับข้อมูลชุด TestsetA และ TestsetB แต่แตกต่างกันที่ช่องรับสัญญาณในการบันทึกเสียงรบกวน เนื่องจากวิทยานิพนธ์นี้เน้นการแก้ปัญหาเสียงรบกวนที่ไม่ได้อยู่ในชุดฝึกสอน ดังนั้นในการทดลองนี้จึงเลือกข้อมูลชุด TestsetB และ TestsetC เป็นข้อมูลในการทดลองการปรับโมเดลด้วยข้อมูลจำลอง

การทดลองเปรียบเทียบผลการรู้จำเสียงพูด ระหว่างวิธีปรับโมเดลต่างๆ กับการปรับโมเดลด้วยข้อมูลจำลองมีดังนี้

- โมเดลแบบหลากสภาวะ เรียกว่า "MULTI"
- โมเดล MULTI ที่มีการปรับโมเดลด้วยวิธี MLLR เรียกว่า "MLLR"
- โมเดล MULTI ที่มีการปรับโมเดลด้วยวิธี MLLR ร่วมกับการปรับโมเดลด้วยข้อมูลจำลอง เรียกว่า "S-MULTI" โดยในการทดลองนี้เลือกใช้การปรับโมเดลด้วยข้อมูลจำลองแบบที่สอง ดังตารางที่ 4.2 ในหัวข้อ 4.5 เนื่องจากเสียงพูดที่เข้ามาเป็นเสียงพูดแบบต่อเนื่อง ซึ่งจากงานวิจัยก่อนหน้า [18] พบว่าการใช้เสียงพูดที่เข้ามาร่วมกับเสียงพูดที่ได้จากการจำลองข้อมูล ให้ผลดีกว่าการใช้เสียงพูดที่ได้จากการจำลองข้อมูลเพียงอย่างเดียว

ในการทดลองการปรับโมเดลด้วยข้อมูลจำลองของ S-MULTI ใช้การดึงเสียงรบกวนแบบการอิงกับหน่วยเสียง และใช้ Train-S เป็นผู้พูดแบบ MIX จากผู้พูดชายและหญิงจำนวน 8 คน ที่มีจำนวนตัวอย่างเท่ากับ 8 ตัวอย่าง ซึ่งมีผลการทดลอง เป็นดังรูปที่ 6.8



รูปที่ 6.8 ผลการเปรียบเทียบผลการรู้จำเสียงพูดของออโรร่าทูเจด้วย MULTI, MLLR และ S-MULTI

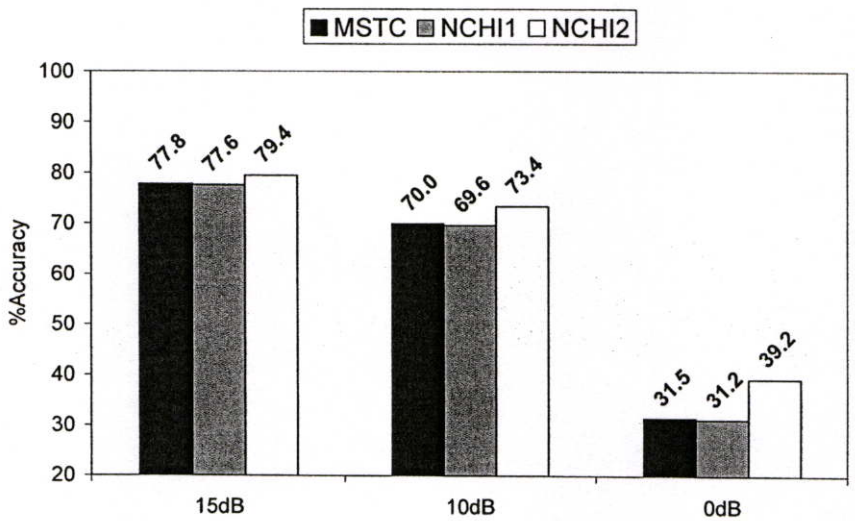
จากผลการทดลองในรูปที่ 6.8 เมื่อคำนวณเป็นค่าเฉลี่ยของผลการรู้จำเสียงพูดของ TestsetB และ TestsetC ด้วย MULTI, MLLR และ S-MLLR ให้ผลการรู้จำเสียงพูด คือ 82.2%, 82.9% และ 85.1% ตามลำดับ ผลการทดลองในรูปที่ 6.8 แสดงให้เห็นว่า S-MLLR ให้ผลการรู้จำเสียงพูดทั้งของ TestsetB และ TestsetC ได้ดีกว่า MULTI และ MLLR ซึ่งเป็นการยืนยันว่า S-MULTI สามารถนำไปใช้กับภาษาอื่นได้ และสามารถใช้กับเสียงพูดต่อเนื่องได้ โดยยังคงให้ผลการรู้จำเสียงพูดที่ดีขึ้น เหมือนกับการใช้กับเสียงพูดคำโดด

6.3 ผลการทดลองการประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดลที่มีการจัดกลุ่มเสียงรบกวน

ในหัวข้อนี้กล่าวถึง ผลการทดลองพารามิเตอร์ต่างๆ ของ NCHI แบ่งออกได้เป็น 2 การทดลอง คือ 1) ผลการทดลองรูปแบบการหาค่าถ่วงน้ำหนักและจำนวนชนิด HMM และผลการทดลองสมการในการประมาณค่าในช่วงของ HMM ซึ่งมีผลการทดลองในหัวข้อ 6.3.1 และ 6.3.2 ตามลำดับ ถัดจากนั้นเป็นผลการทดลองวิธี NCHI ร่วมกับการปรับโมเดลด้วยข้อมูลจำลองในหัวข้อ 6.3.3 และในหัวข้อสุดท้าย คือ หัวข้อ 6.3.4 เป็นการวิเคราะห์และเปรียบเทียบกับวิธีต่างๆ

6.3.1 ผลการทดลองรูปแบบการหาค่าถ่วงน้ำหนักและจำนวนชนิดของโมเดล

หัวข้อนี้เป็นการทดลองรูปแบบการค้นหาค่าถ่วงน้ำหนัก และจำนวนชนิดของโมเดลด้วยวิธีต่างๆ ของ NCHI ซึ่งวิธีที่ในวิทยานิพนธ์นี้ใช้ทดลอง คือ การค้นหาแบบโครงสร้างต้นไม้ และการค้นหาแบบตรง ดังแสดงในหัวข้อ 5.2 และ 5.3 ตามลำดับ โดยกำหนดให้ NCHI ที่ใช้การค้นหาแบบโครงสร้างต้นไม้ และ NCHI ที่ใช้การค้นหาแบบตรง เรียกว่า NCHI1 และ NCHI2 ตามลำดับ ในการทดลอง NCHI1 และ NCHI2 ใช้สมการแบบที่หนึ่ง ดังแสดงในหัวข้อ 5.1 ในการประมาณค่าในช่วงของ HMM นอกจากนี้ ในการทดลองยังมีการเปรียบเทียบผลการรู้จำเสียงพูดของโมเดลที่ได้จาก NCHI และโมเดลที่ได้จากการเลือกโมเดลที่มีการจัดกลุ่มแบบโครงสร้างต้นไม้ (MSTC) ซึ่งโครงสร้างต้นไม้ของ MSTC และ NCHI1 มีโครงสร้างแบบเดียวกัน ซึ่งผลการทดลอง ดังรูปที่ 6.9



รูปที่ 6.9 ค่าเฉลี่ยของผลการรู้จำเสียงพูดของ Test-1 ด้วย MSTC, NCHI1 และ NCHI2

จากผลการทดลองในรูปที่ 6.9 เมื่อคำนวณเป็นค่าเฉลี่ยของผลการรู้จำเสียงพูดของ Test-1 ด้วย MSTC, NCHI1 และ NCHI2 ให้ผลการรู้จำเสียงพูด คือ 59.8%, 59.5% และ 64.0% ตามลำดับ

ผลการทดลองในรูปที่ 6.9 แสดงให้เห็นว่า

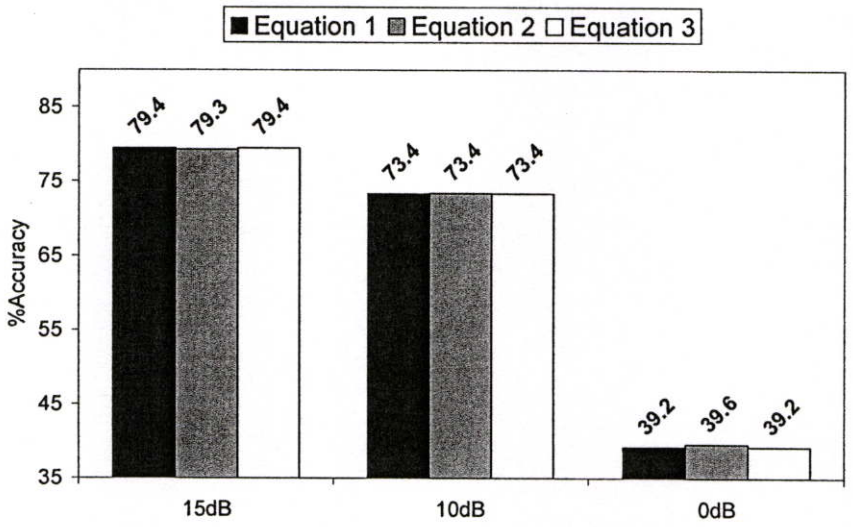
- NCHI2 ที่มีการค้นหาค่าถ่วงน้ำหนัก และจำนวนชนิดของเสียงรบกวน ด้วยวิธีการค้นหาแบบตรง ให้ผลการรู้จำเสียงพูดได้ดีที่สุด เป็นผลมาจากโมเดลแบบผสมที่ได้จาก NCHI2 มีความใกล้เคียงกับเสียงพูดที่เข้ามามากกว่าโมเดลแบบผสมที่มีอยู่ในโครงสร้างต้นไม้ เพราะ NCHI2 มีรูปแบบการผสมโมเดลได้มากกว่า NCHI1 และ MSTC ที่มีรูปแบบการผสมที่จำกัดตามโครงสร้างต้นไม้
- NCHI1 มีผลการรู้จำเสียงพูดที่น้อยกว่า MSTC ไม่มาก แสดงให้เห็นว่าโมเดลแบบผสมที่ได้จากการประมาณค่าในช่วงมีความใกล้เคียงกับโมเดลแบบผสมที่มีการสร้างไว้ล่วงหน้า
- NCHI1 และ NCHI2 มีการจัดเก็บโมเดล Noise Cluster HMM และ Noisy Speech HMM ที่น้อยกว่า MSTC เพราะ NCHI มีการจัดเก็บโมเดลทั้งสองแบบเท่ากับจำนวนเสียงรบกวนพื้นฐานเท่านั้น นั่นคือไม่จัดเก็บโมเดลที่เกิดจากการผสมโมเดลขึ้นมา เพราะ NCHI สามารถสร้างโมเดลแบบผสมขึ้นมาใหม่ได้ด้วยการประมาณค่าในช่วง
- เวลาเฉลี่ยในการคำนวณของ MSTC, NCHI1 และ NCHI2 คือ 0.8 วินาที, 1.1 วินาที และ 1.0 วินาที ตามลำดับ เห็นได้ว่า MSTC ใช้เวลาน้อยที่สุด เนื่องจากไม่ต้องเสียเวลาในการประมาณค่าในช่วงของโมเดลขึ้นมาใหม่ และมีจำนวนครั้งในการหาโมเดลที่น้อยกว่า โดย MSTC และ NCHI1 มีจำนวนครั้งในการค้นหามากที่สุดเท่ากัน คือ $2\log_2(N)$ ครั้ง ส่วน NCHI2 มีจำนวนครั้งในการค้นหามากที่สุด คือ 2N ครั้ง เมื่อ N คือ จำนวนเสียงรบกวนพื้นฐาน ถึงแม้ว่าโครงสร้างต้นไม้จะมีจำนวนครั้งในการค้นหาน้อยกว่าก็ตาม แต่เวลาในการคำนวณของ NCHI2 ก็ยังมีความใกล้เคียงกับเวลาในการคำนวณของ NCHI1 เป็นเพราะโหนดช่วงบนของโครงสร้างต้นไม้มีจำนวนโมเดล ที่จะต้องใช้ในการประมาณค่าในช่วงของโมเดลมากกว่าโหนดช่วงล่าง ซึ่งการค้นหาแบบโครงสร้างต้นไม้เป็นการค้นหาแบบบนลงล่าง ทำให้เวลาในการคำนวณโดยรวมมีความใกล้เคียงกับการค้นหาแบบตรง ซึ่งเป็นการประมาณค่าในช่วงของโมเดลจากจำนวนน้อยๆ แล้วค่อยๆ เพิ่มจำนวนขึ้นไปตามลำดับ แต่ถ้ามีการเพิ่มจำนวน N ที่มากขึ้น เวลาที่ใช้ในการคำนวณของ NCHI2 อาจใช้เวลามากกว่า NCHI1 เพราะ NCHI2 ต้องมีการหาค่าความน่าจะเป็นของแต่ละโมเดลเสียก่อน ถึงจะนำมาประมาณค่าในช่วง ต่างจาก NCHI1 ที่มีโครงสร้างต้นไม้ที่สามารถค้นหาได้เร็วกว่า

วิทยานิพนธ์นี้เลือกใช้ NCHI ที่มีการค้นหาค่าถ่วงน้ำหนัก และจำนวนชนิดของเสียงรบกวนด้วยวิธีการค้นหาแบบตรง เพราะให้ผลการรู้จำเสียงพูดได้ดีที่สุด

6.3.2 ผลการทดลองสมการในการประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดล

หัวข้อนี้เป็นการทดลองการรู้จำเสียงพูดของ NCHI ที่ใช้สมการในการประมาณค่าในช่วงของ HMM ที่แตกต่างกัน สำหรับสมการที่ใช้ทดลองในวิทยานิพนธ์นี้มี 3 สมการ [24],[25] ดังแสดงใน

หัวข้อ 5.1 สำหรับการทดลองในหัวข้อนี้ เกิดขึ้นเนื่องจากยังไม่เคยมีการวัดผลการรู้จำเสียงพูดของโมเดลที่ได้จากการประมาณค่าในช่วงของ HMM ด้วยสมการที่แตกต่างกันมาก่อน โดยผลการทดลองเป็นดังรูปที่ 6.10



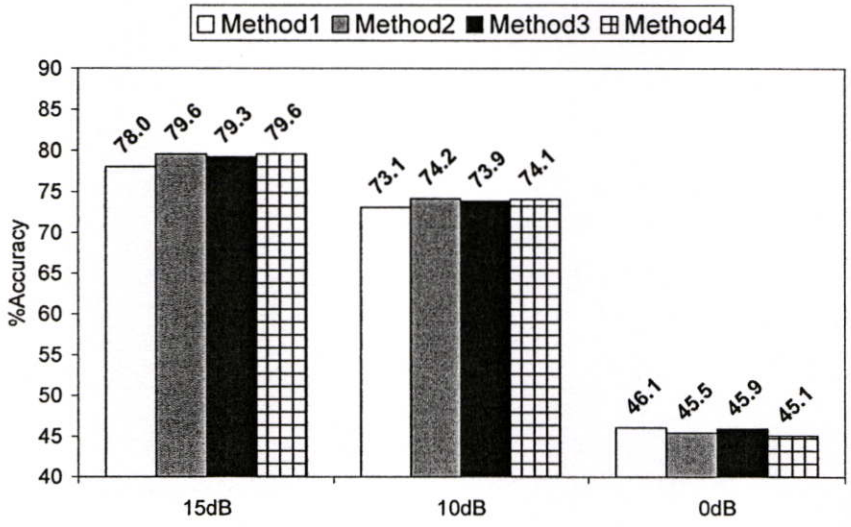
รูปที่ 6.10 ค่าเฉลี่ยของผลการรู้จำเสียงพูดของ Test-1 ด้วย NCHI ที่ใช้สมการแตกต่างกัน

จากผลการทดลองในรูปที่ 6.10 เมื่อคำนวณเป็นค่าเฉลี่ยของผลการรู้จำเสียงพูดของ Test-1 ด้วย NCHI ที่ใช้สมการแบบที่หนึ่ง, สมการแบบที่สอง และสมการแบบที่สาม ให้ผลการรู้จำเสียงพูด คือ 64.0%, 64.1% และ 64.0% ตามลำดับ ผลการทดลองในรูปที่ 6.10 แสดงให้เห็นว่า NCHI ที่ใช้สมการในการประมาณค่าในช่วงของ HMM ที่แตกต่างกัน สามารถสร้างโมเดลในการรู้จำเสียงพูดที่มีเสียงรบกวนขึ้นมาใหม่ได้ไม่แตกต่างกัน แต่เวลาที่ใช้ในการประมาณค่าในช่วงของ HMM ด้วยสมการแบบที่หนึ่ง ใช้เวลาน้อยที่สุด เนื่องจากสมการแบบที่หนึ่ง มีจำนวนการคำนวณที่น้อยกว่าสมการแบบที่สองและสาม วิทยานิพนธ์นี้จึงเลือก NCHI ที่ใช้สมการแบบที่หนึ่ง ในการประมาณค่าในช่วงของโมเดล เพราะใช้เวลาในการคำนวณน้อยที่สุด

6.3.3 ผลการทดลองวิธีการประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดลที่มีการจัดกลุ่มเสียงรบกวนร่วมกับการปรับโมเดลด้วยข้อมูลจำลอง

หัวข้อนี้เป็นการทดลองปรับโมเดล NCHI ด้วยการปรับโมเดลด้วยข้อมูลจำลอง ซึ่งเรียกว่า "S-NCHI" เพื่อทดลองผลของการใช้หรือไม่ใช้เสียงพูดที่เข้ามาร่วมกับข้อมูลจำลอง และผลของการเลือกคำตอบของโมเดลตั้งต้นหรือโมเดลที่ได้จากการปรับโมเดล ซึ่งมีรูปแบบการทดลองทั้งหมด 4 แบบ ดังตารางที่ 4.2 ในหัวข้อ 4.5 โดยเรียกแต่ละแบบว่า Method1, Method2, Method3 และ Method4 ตามลำดับ ในการทดลองการปรับโมเดลด้วยข้อมูลจำลองของ S-NCHI ใช้การดึงเสียง

รบกวนแบบการอิงกับหน่วยเสียง และใช้ Train-S เป็นผู้พูดแบบ MIX ที่มีจำนวนคำเท่ากับ 22 คำ ซึ่งมีผลการทดลอง ดังรูปที่ 6.11



รูปที่ 6.11 ค่าเฉลี่ยของผลการรู้จำเสียงพูดของ Test-1 ด้วย S-NCHI ที่ใช้รูปแบบของการปรับโมเดลด้วยข้อมูลจำลองที่แตกต่างกัน

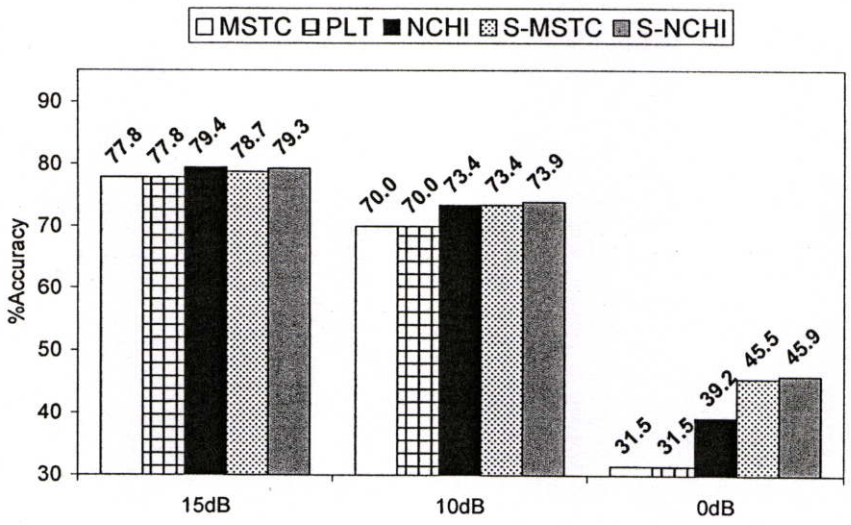
จากผลการทดลองในรูปที่ 6.11 เมื่อคำนวณเป็นค่าเฉลี่ยของผลการรู้จำเสียงพูดของ Test-1 ด้วย S-NCHI ทั้ง 4 รูปแบบ ให้ผลการรู้จำเสียงพูด คือ 65.7%, 66.4%, 66.4% และ 66.3% ตามลำดับ จากผลการทดลองเห็นได้ว่า S-NCHI ทั้ง 4 แบบ มีผลการรู้จำเสียงพูดที่มากกว่า NCHI ที่ไม่มีการปรับโมเดล แสดงให้เห็นว่าการปรับโมเดลด้วยข้อมูลจำลองทำให้โมเดลที่ได้จาก NCHI มีความใกล้เคียงกับเสียงพูดที่เข้ามามากยิ่งขึ้น นอกจากนี้ ยังพบว่าการใช้เสียงพูดที่เข้ามาร่วมกับข้อมูลจำลองให้ผลการรู้จำเสียงพูดที่ดีกว่าการใช้ข้อมูลจำลองเพียงอย่างเดียว ทั้งนี้เป็นผลมาจากโมเดลตั้งต้นจาก NCHI ให้ผลการรู้จำเสียงพูดที่สูงขึ้น ทำให้ได้คำอ่านที่มีความถูกต้องมากขึ้น จึงส่งผลให้โมเดลที่ปรับได้มีผลการรู้จำเสียงพูดที่สูงขึ้น โดยในบางครั้งมีโอกาสที่โมเดลที่ปรับด้วยข้อมูลจำลองเพียงอย่างเดียว ให้ผลการรู้จำเสียงพูดได้น้อยกว่าโมเดลตั้งต้น ดังนั้น S-NCHI แบบที่หนึ่งที่ไม่มีการเลือกคำตอบระหว่างโมเดลที่ปรับได้กับโมเดลตั้งต้น จึงมีผลการรู้จำเสียงพูดที่น้อยกว่าแบบที่สามที่มีการเลือกคำตอบ วิทยานิพนธ์นี้จึงเลือกจึงใช้ S-NCHI แบบที่สาม เพราะไม่ต้องใช้เสียงพูดที่เข้ามาในการปรับโมเดล ทำให้ใช้เวลาในการคำนวณน้อยกว่า และยังให้ผลการรู้จำเสียงพูดที่ใกล้เคียงกันกับ S-NCHI แบบที่สองและสี่ นอกจากนี้ ยังให้ผลการรู้จำเสียงพูดที่ดีกว่า S-NCHI แบบที่หนึ่งด้วย

6.3.4 การวิเคราะห์และเปรียบเทียบกับวิธีต่างๆ

หัวข้อนี้เป็นการทดลองเปรียบเทียบผลการรู้จำระหว่างระบบการปรับโมเดลแบบต่างๆ กับระบบการปรับโมเดลที่วิทยานิพนธ์นี้นำเสนอ โดยระบบที่นำมาเปรียบเทียบมีดังนี้

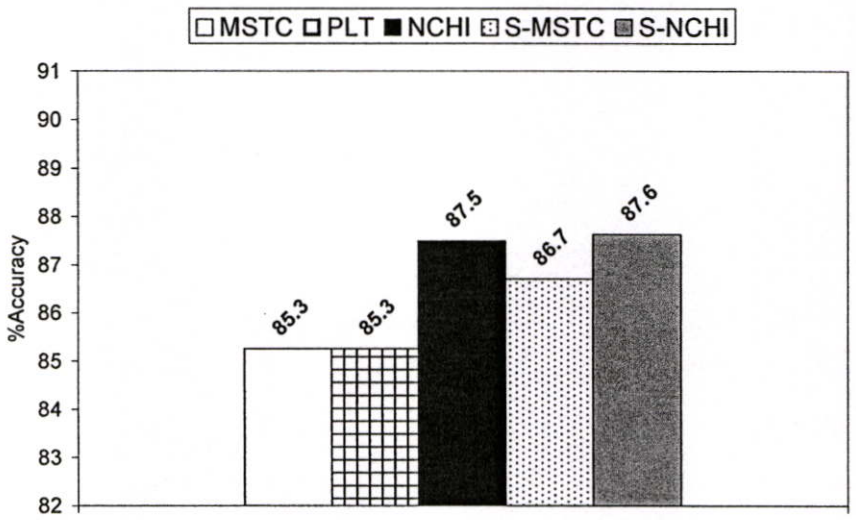
- การเลือกโมเดลที่มีการจัดกลุ่มแบบโครงสร้างต้นไม้ เรียกว่า "MSTC"
- การแปลงเชิงเส้นแบบแบ่งส่วน คือ โมเดล MSTC ที่มีการปรับโมเดลด้วยวิธี MLLR เรียกว่า "PLT"
- โมเดลเสียงพูดที่มีเสียงรบกวนหลายชนิดที่ได้จากการประมาณค่าในช่วงของ HMM เข้าด้วยกัน เรียกว่า "NCHI" และในการทดลองนี้ทำการประมาณค่าในช่วงของ HMM ด้วยสมการแบบที่หนึ่ง และใช้การค้นหาแบบตรงในการหาค่าถ่วงน้ำหนัก และจำนวนชนิดของโมเดล
- โมเดล MSTC ที่มีการปรับโมเดลด้วยวิธี MLLR ร่วมกับการปรับโมเดลด้วยข้อมูลจำลองแบบที่สาม เรียกว่า "S-MSTC"
- โมเดล NCHI ที่มีการปรับโมเดลด้วยวิธี MLLR ร่วมกับการปรับโมเดลด้วยข้อมูลจำลองแบบที่สาม เรียกว่า "S-HCHI"

ในการทดลองการปรับโมเดลด้วยข้อมูลจำลองของ S-MSTC และ S-NCHI ใช้การดึงเสียงรบกวนแบบการอิงกับหน่วยเสียง และใช้ Train-S เป็นผู้พูดแบบ MIX ที่มีจำนวนคำเท่ากับ 22 คำ ซึ่งได้ผลการทดลองของ Test-1 และ Test-2 เป็นดังรูปที่ 6.12 และ 6.13 ตามลำดับ



รูปที่ 6.12 ผลการเปรียบเทียบผลการรู้จำเสียงพูดของ Test-1 ด้วย MSTC, PLT, NCHI, S-MSTC และ S-NCHI

ผลการทดลองในรูปที่ 6.12 เมื่อคำนวณเป็นค่าเฉลี่ยของผลการรู้จำเสียงพูดของ Test-1 ด้วย MSTC, PLT, NCHI, S-MSTC และ S-NCHI ให้ผลการรู้จำเสียงพูด คือ 59.8%, 59.8%, 64.0%, 65.9% และ 66.4% ตามลำดับ



รูปที่ 6.13 ผลการเปรียบเทียบผลการรู้จำเสียงพูดของ Test-2 ด้วย MSTC, PLT, NCHI, S-MSTC และ S-NCHI

ผลการทดลองในรูปที่ 6.12 และ 6.13 แสดงให้เห็นว่า

- S-NCHI และ S-MSTC มีผลการรู้จำเสียงพูดของ Test-1 ดีกว่า NCHI ซึ่งแสดงให้เห็นว่าโมเดลที่ได้จาก NCHI ยังไม่ครอบคลุมเสียงรบกวนที่เกิดขึ้นจริงทุกชนิด เป็นผลทำให้ S-NCHI และ S-MSTC ที่มีการปรับโมเดลให้โมเดลที่ใกล้เคียงเสียงพูดที่เข้ามามากกว่า
- S-NCHI ให้ผลการรู้จำเสียงพูดของ Test-1 ได้ดีที่สุด ส่วนผลการรู้จำเสียงพูดของ Test-2 ด้วย S-NCHI ให้ผลการรู้จำเสียงพูดที่ใกล้เคียงกับ NCHI ทั้งนี้เป็นเพราะว่าโมเดลที่ได้จาก NCHI มีความใกล้เคียงกับเสียงพูดที่เข้ามาของ Test-2 อยู่แล้ว จึงทำให้โมเดลที่ได้จาก S-NCHI ไม่ได้ให้ผลการรู้จำเสียงพูดที่ดีขึ้น
- NCHI ให้ผลการรู้จำเสียงพูดของ Test-2 ได้ดีกว่า S-MSTC และ PLT แสดงให้เห็นว่าโมเดลที่ได้จากการประมาณค่าในช่วงอาจจะสร้างโมเดลขึ้นมาใหม่ ให้มีความใกล้เคียงกับเสียงพูดที่เข้ามาได้มากกว่า MSTC ที่มีการปรับโมเดลได้
- S-NCHI มีผลการรู้จำเสียงพูดมากกว่า PLT เป็นเพราะโมเดลตั้งต้นที่ได้จาก NCHI มีความใกล้เคียงกับเสียงพูดที่เข้ามามากกว่าโมเดลที่ได้จาก MSTC และโมเดลที่ได้จากการปรับโมเดลด้วยข้อมูลจำลองช่วยลดปัญหาของเสียงพูดที่เข้ามามีขนาดเล็ก และไม่ทราบคำอ่าน

ที่ถูกต้องได้ เป็นผลให้โมเดลจากการปรับโมเดลด้วย S-NCHI มีความใกล้เคียงเสียงพูดที่เข้ามามากกว่าโมเดลจากการปรับโมเดลด้วย PLT

- S-NCHI มีผลการรู้จำเสียงพูดมากกว่า S-MSTC ไม่มากนัก ทั้งนี้เป็นผลจากการปรับโมเดลด้วยข้อมูลการจำลอง ที่ทำให้โมเดลจากการปรับโมเดลด้วย NCHI และ MSTC มีผลการรู้จำเสียงพูดที่ใกล้เคียงกันมากขึ้น
- S-MSTC มีผลการรู้จำเสียงพูดมากกว่า MSTC ทั้งนี้ เป็นเพราะโมเดลเสียงพูดที่มีเสียงรบกวนที่สร้างเก็บไว้ล่วงหน้ามีไม่ครอบคลุมเสียงรบกวนที่เกิดขึ้นจริง ทำให้โมเดลที่ใช้ในการรู้จำเสียงพูดมีผลการรู้จำเสียงพูดได้ไม่ดีเท่า S-MSTC ที่มีการปรับโมเดลตามเสียงพูดที่เข้ามา
- PLT มีผลการรู้จำเสียงพูดเท่ากับ MSTC ทั้งนี้ เนื่องมาจากปัญหาเสียงพูดที่เข้ามามีขนาดเล็กและไม่ทราบคำอ่านที่ถูกต้อง จึงทำให้โมเดลที่ปรับได้ไม่ให้ผลการรู้จำเสียงพูดที่ดีขึ้น

จากผลการทดลองสรุปได้ว่า S-NCHI เป็นระบบปรับโมเดลแบบออนไลน์แบบใหม่ คือ การใช้โมเดลตั้งต้นจาก NCHI ที่ทำการประมาณค่าในช่วงของ HMM ด้วยสมการแบบที่หนึ่ง ซึ่งมีการค้นหาค่าถ่วงน้ำหนักและจำนวนชนิดของโมเดลด้วยการค้นหาแบบตรง แล้วนำโมเดลที่ได้ไปปรับโมเดลด้วยข้อมูลจำลองแบบที่สาม ที่มีการดึงเสียงรบกวนแบบการอิงกับหน่วยเสียงและใช้ Train-S เป็นผู้พูดแบบ MIX ที่มีจำนวนคำเท่ากับ 22 คำ ซึ่งวิธี S-NCHI ที่วิทยานิพนธ์นี้นำเสนอ ให้ผลการรู้จำเสียงพูดที่มากกว่าระบบปรับโมเดลแบบออนไลน์ที่นิยมใช้ในปัจจุบัน คือ PLT ซึ่งสาเหตุที่ทำให้ S-NCHI มีผลการรู้จำเสียงพูดที่ดีกว่า เป็นผลมาจาก S-NCHI สามารถแก้ปัญหาพื้นฐาน 3 ข้อของระบบปรับโมเดลแบบออนไลน์ได้ คือ 1) ปัญหาการไม่ทราบคำอ่านของเสียงพูดที่เข้ามา 2) ปัญหาเสียงพูดที่เข้ามามีปริมาณน้อย ซึ่งทั้งสองปัญหาแก้ไขได้ด้วยการจำลองข้อมูลที่สร้างมาจากการนำเสียงรบกวนจากเสียงพูดที่เข้ามา แล้วนำไปผสมกับเสียงพูดสะอาดที่เตรียมไว้ล่วงหน้า และ 3) ปัญหาการมีโมเดลที่มีรูปแบบการผสมได้จำนวนจำกัดของ MSTC ซึ่งแก้ไขด้วย NCHI ที่สามารถสร้างโมเดลแบบผสมได้ใหม่ตามเสียงพูดที่เข้ามา และการที่สามารถผสมโมเดลได้นี้เอง ยังช่วยลดจำนวนโมเดลแบบผสมที่ต้องจัดเก็บเพิ่มขึ้นได้อีกด้วย

บทที่ 7
สรุปผลการทดลอง และข้อเสนอแนะ

7.1 สรุปผลการทดลอง

วิทยานิพนธ์นี้นำเสนอ วิธีการปรับโมเดลแบบออนไลน์แบบใหม่ สำหรับการรู้จำเสียงพูดแบบคงทน (Robust Speech Recognition) ด้วยการปรับโมเดลด้วยข้อมูลจำลอง (Simulated-data Adaptation) ร่วมกับการประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดลที่มีการจัดกลุ่มเสียงรบกวน (Noise Cluster HMM Interpolation, NCHI) วิธีที่นำเสนอนี้เรียกว่า S-NCHI ซึ่งวิธีที่นำเสนอมีความใกล้เคียงกับวิธีการแปลงเชิงเส้นแบบแบ่งส่วน (Piecewise-linear Transformation, PLT)

วิธี PLT แบ่งเป็น 2 ขั้นตอน ขั้นตอนที่หนึ่ง คือ การเลือกโมเดลที่มีการจัดกลุ่มแบบโครงสร้างต้นไม้ (Model Selection based Tree-Structured Cluster, MSTC) โดยเลือกโมเดลเสียงรบกวนที่มีความใกล้เคียงกับเสียงพูดที่เข้ามา ขั้นตอนที่สอง คือ การปรับโมเดลแบบออนไลด์ ด้วยการถดถอยแบบเชิงเส้นตามความเป็นไปได้สูงสุด (Maximum Likelihood Linear Regression, MLLR) ซึ่งข้อมูลที่ใช้ในการปรับโมเดลในขั้นตอนที่สอง คือ เสียงพูดที่เข้ามา และใช้คำอ่านที่ได้จากการรู้จำเสียงพูดด้วยโมเดลที่ได้จากขั้นตอนที่หนึ่ง

วิธี S-NCHI แบ่งเป็น 2 ขั้นตอน ขั้นตอนที่หนึ่ง คือ การสร้างโมเดลเสียงพูดขึ้นใหม่แบบออนไลน์ ด้วยการประยุกต์ใช้การประมาณค่าในช่วงของโมเดลที่มีเสียงรบกวนหลายๆ ชนิดเข้าด้วยกัน ซึ่งการประมาณค่าในช่วงของโมเดลมีใช้งานอยู่แล้วในระบบสังเคราะห์เสียงพูด (Speech Synthesis System) แต่ยังไม่เคยมีการใช้งานในระบบรู้จำเสียงพูดแบบคงทนมาก่อน และสิ่งที่วิทยานิพนธ์นี้พัฒนาขึ้นมาใหม่ เพื่อให้การประมาณค่าในช่วงของโมเดลสามารถใช้กับระบบรู้จำเสียงพูดแบบคงทนได้ คือ วิธีการหาค่าถ่วงน้ำหนักและจำนวนชนิดของโมเดลจำนวน 2 วิธี คือ 1) วิธีการค้นหาแบบโครงสร้างต้นไม้ ซึ่งโครงสร้างต้นไม้มีวิธีการสร้างและใช้งานแบบเดียวกับ MSTC และ 2) วิธีการค้นหาแบบตรง ขั้นตอนที่สอง คือ การปรับโมเดลด้วยข้อมูลจำลองแบบออนไลน์ เป็นการปรับโมเดลที่ได้จากขั้นตอนที่หนึ่ง ด้วยข้อมูลจำลองที่สร้างจากการผสมเสียงรบกวนจากเสียงพูดที่เข้ามากับเสียงพูดสะอาดที่เตรียมไว้ล่วงหน้า ทำให้ทราบคำอ่านและสามารถเพิ่มข้อมูลให้เพียงพอ แล้วนำไปปรับโมเดลด้วยวิธีการที่มีอยู่แล้ว เช่น MLLR เป็นต้น สิ่งที่วิทยานิพนธ์พัฒนาขึ้นมาใหม่ในส่วนนี้ คือ วิธีการสร้างข้อมูลจำลอง เพื่อใช้เป็นข้อมูลในการปรับโมเดล

จากผลการทดลองสามารถสรุปผล ได้ดังนี้

1) การปรับโมเดลด้วยข้อมูลจำลอง เป็นการจำลองเสียงพูดให้มีเสียงรบกวนเหมือนกับเสียงรบกวนในเสียงพูดที่เข้ามา เพื่อใช้เป็นข้อมูลในการปรับโมเดลด้วยวิธี MLLR โดยข้อมูลที่จำลอง

ขึ้นมาเป็นเสียงพูดที่มีเสียงรบกวนแบบบวก ที่ได้จากการบวกเสียงรบกวนพื้นหลังที่ได้จากการดึงส่วนที่เป็นเสียงเงียบของเสียงพูดที่เข้ามาและเสียงพูดสะอาดที่เตรียมไว้ล่วงหน้า ซึ่งในที่นี้เรียกเสียงพูดสะอาดนี้ว่า "Train-S" ผลการทดลองสรุปได้ว่า

- การปรับโมเดลด้วยข้อมูลจำลองที่ใช้ Train-S ด้วยผู้พูดแบบ MIX คือ การนำตัวอย่างเสียงพูดจากผู้พูดทั้ง 4 คนมาใช้ ซึ่งให้ผลการรู้จำเสียงพูดที่ดีกว่าการใช้ Train-S ที่มาจากผู้พูดคนใดคนหนึ่ง เพราะการใช้ผู้พูดแบบ MIX ช่วยให้โมเดลที่ปรับได้ขึ้นอยู่กับผู้พูดของ Train-S น้อยลง นอกจากนี้ ผลการรู้จำเสียงพูดยังเพิ่มขึ้นตามจำนวนคำของ Train-S ด้วย เป็นผลให้ Train-S ที่มีจำนวนคำเท่ากับ 32 คำ ให้ผลการรู้จำเสียงพูดได้ดีที่สุด แต่ก็ใช้เวลาในการคำนวณการปรับโมเดลมากที่สุดด้วย จากสาเหตุดังกล่าว Train-S ที่มีจำนวนคำเท่ากับ 22 คำ จึงมีความเหมาะสมมากกว่า เพราะใช้เวลาในการคำนวณไม่มากเกินไป และยังคงให้ผลการรู้จำเสียงพูดที่ดี นอกจากนี้ จำนวนคำ 22 คำ เป็นจำนวนคำที่น้อยที่สุด ซึ่งสามารถครอบคลุมหน่วยเสียงในภาษาไทย และ 22 คำนี้เป็นคำในคลังข้อมูลเสียงพูดภาษาไทย NETEC-ATR
- โมเดลเสียงพูดที่ได้จากการปรับโมเดลวิธี MLLR ด้วยข้อมูลจำลอง ให้ผลการรู้จำเสียงพูดได้ดีกว่าวิธีการพิจารณาจากค่าประสบการณ์สูงสุด (Maximum a Posteriori, MAP) และวิธี MLLR-MAP ที่ใช้ข้อมูลในการปรับโมเดลชุดเดียวกันกับวิธี MLLR นั่นเป็นเพราะ ข้อมูลที่ใช้ในการปรับโมเดลของทั้งสองวิธียังไม่มากเพียงพอ ที่จะทำให้ได้ผลการรู้จำเสียงพูดที่ดีขึ้น อย่างไรก็ตาม จากผลการทดลองแสดงให้เห็นว่าการปรับโมเดลด้วยข้อมูลจำลองสามารถเพิ่มประสิทธิภาพให้กับวิธีปรับโมเดลแบบอื่นๆ นอกเหนือจากวิธี MLLR
- การปรับโมเดลด้วยข้อมูลจำลอง ที่ใช้การดึงเสียงรบกวนจากเสียงพูดที่เข้ามา ด้วยวิธีการอิงกับหน่วยเสียงให้ผลการรู้จำเสียงพูดที่ดีกว่าวิธีการหาส่วนที่เป็นเสียงพูดและส่วนที่ไม่เป็นเสียงพูด เพราะวิธีการอิงกับหน่วยเสียงหาส่วนที่เป็นเสียงเงียบได้ดีกว่า เป็นผลให้ได้เสียงเงียบที่ยาวกว่า ดังนั้นเมื่อนำส่วนเสียงเงียบที่ได้มาต่อกันก็จะมีจำนวนรอยต่อที่น้อยกว่า ทำให้ปัญหาของการเกิดเสียงรบกวนที่รอยต่อลดลง ข้อมูลจำลองที่ได้ก็จะมีเสียงรบกวนที่ใกล้กับเสียงรบกวนในเสียงพูดที่เข้ามามากขึ้น ซึ่งทำให้ผลการรู้จำเสียงพูดดีขึ้นตามไปด้วย แต่วิธีการอิงกับหน่วยเสียงก็ใช้เวลาในการคำนวณการปรับโมเดลที่มากกว่า นั่นเป็นเพราะโมเดลที่ใช้ในการหาเสียงรบกวนมีมากกว่านั่นเอง อย่างไรก็ตาม วิธีการหาส่วนที่เป็นเสียงพูดและส่วนที่ไม่เป็นเสียงพูดต้องมีการเก็บโมเดลเพิ่มขึ้น แต่วิธีการอิงกับหน่วยเสียงไม่ต้องเก็บโมเดลเพิ่ม เนื่องจากใช้โมเดลที่ใช้ในการรู้จำเสียงพูด เพื่อหาส่วนที่เป็นเสียงรบกวนในเสียงพูดที่เข้ามาได้นั่นเอง ดังนั้นการดึงเสียงรบกวนด้วยวิธีการอิงกับหน่วยเสียงจึงเหมาะสำหรับการปรับโมเดลด้วยข้อมูลจำลองมากกว่า เพราะไม่ต้องจัดเก็บโมเดลของการดึงเสียงรบกวนเพิ่ม และให้ผลการรู้จำเสียงพูดที่ดีกว่า

- ผลการรู้จำเสียงพูดของการปรับโมเดลด้วยข้อมูลจำลองทั้ง 4 รูปแบบ ดังตารางที่ 4.2 ในหัวข้อ 4.5 ขึ้นอยู่กับขนาดของข้อมูลที่ใช้ในการปรับโมเดล และโมเดลตั้งต้นของการปรับโมเดล จากผลการทดลองสรุปได้ดังนี้ ในกรณีเสียงพูดที่เข้ามาเป็นเสียงพูดแบบต่อเนื่อง การใช้เสียงพูดเข้ามาร่วมกับการปรับโมเดลด้วยข้อมูลจำลองให้ผลการรู้จำเสียงพูดได้ดีกว่า [18] ในกรณีโมเดลตั้งต้นมีความใกล้เคียงกับเสียงพูดที่เข้ามา ก็สามารถให้คำอ่านที่มีความถูกต้องมากขึ้น ส่งผลให้โมเดลที่ปรับได้ ให้ผลการรู้จำเสียงพูดที่สูงขึ้น อย่างไรก็ตาม ในบางครั้งโมเดลที่ปรับด้วยข้อมูลจำลองเพียงอย่างเดียว อาจให้ผลการรู้จำเสียงพูดได้น้อยกว่าโมเดลตั้งต้นได้ ดังนั้นรูปแบบที่หนึ่งที่ไม่มีการเลือกคำตอบระหว่างโมเดลที่ปรับได้กับโมเดลตั้งต้น จึงมีผลการรู้จำเสียงพูดที่น้อยกว่ารูปแบบที่สามที่มีการเลือกคำตอบ ส่วนในกรณีที่โมเดลที่ปรับได้ มีความใกล้เคียงกับเสียงพูดที่เข้ามามากกว่าโมเดลตั้งต้น การมีหรือไม่มีการเลือกคำตอบระหว่างโมเดลที่ปรับได้กับโมเดลตั้งต้น ก็ให้ผลการรู้จำเสียงพูดที่ใกล้เคียงกัน ดังนั้นการใช้งานการปรับโมเดลด้วยข้อมูลจำลอง จึงต้องมีการทดลองหารูปแบบให้เหมาะสมกับโมเดลตั้งต้น และขนาดเสียงพูดที่เข้ามาเสียก่อน จากผลการทดลองพบว่าการปรับโมเดลด้วยข้อมูลจำลองแบบที่หนึ่ง มีความเหมาะสมกับโมเดลตั้งต้นแบบการฝึกสอนแบบหลากสภาวะ (Multi-condition Training, MULTI) คือ โมเดลที่สร้างมาจากเสียงพูดที่มีเสียงรบกวนหลายชนิดรวมกัน และโมเดลเสียงพูดสะอาด (Baseline) คือ โมเดลที่สร้างมาจากเสียงพูดสะอาด ซึ่งมีเสียงพูดที่เข้ามาเป็นเสียงพูดคำโดด ในขณะที่การปรับโมเดลด้วยข้อมูลจำลองแบบที่สองเหมาะสมกับโมเดลตั้งต้นแบบ MULTI ซึ่งมีเสียงพูดที่เข้ามาเป็นเสียงพูดต่อเนื่อง [18] ส่วนการปรับโมเดลด้วยข้อมูลจำลองแบบที่สามมีความเหมาะสมกับโมเดลตั้งต้นแบบ MSTC คือ การเลือกโมเดลเสียงรบกวนที่ใกล้เคียงกับเสียงรบกวนในเสียงพูดที่เข้ามา และ NCHI ซึ่งมีเสียงพูดที่เข้ามาเป็นเสียงพูดคำโดด
- การปรับโมเดลแบบออนไลน์ ด้วยวิธีการปรับโมเดลด้วยข้อมูลจำลอง ให้ผลการรู้จำเสียงพูดได้ดีกว่าวิธี MLLR ที่ปรับโมเดลด้วยเสียงพูดที่เข้ามาเพียงอย่างเดียว นั่นเป็นเพราะการปรับโมเดลด้วยข้อมูลจำลองมีคำอ่านที่ถูกต้อง จึงสามารถปรับโมเดลตามเสียงพูดที่เข้ามาได้ นอกจากนี้ ยังมีจำนวนข้อมูลในการปรับโมเดลที่เพียงพอ จึงทำให้โมเดลที่ปรับได้มีความใกล้เคียงกับเสียงพูดที่เข้ามามากขึ้น
- การปรับโมเดลแบบออนไลน์ ด้วยการปรับโมเดลด้วยข้อมูลจำลอง ให้ผลการรู้จำเสียงพูดดีกว่าวิธีการรวมโมเดลขนาน (Parallel Model Combination, PMC) ที่ปรับโมเดลด้วยเสียงพูดที่เข้ามาเพียงอย่างเดียว นั่นเป็นเพราะ โดยทั่วไปแล้ว ถ้าวิธี MLLR มีข้อมูลที่มากเพียงพอ แม้ว่าจะเป็นการปรับโมเดลแบบ Unsupervised ก็สามารถให้ผลการรู้จำเสียงพูดที่ดีกว่าวิธี PMC และยิ่งถ้าเป็นการปรับโมเดลวิธี MLLR แบบ Supervised ย่อมให้ผลการรู้จำเสียงพูดได้ดียิ่งขึ้นไปอีก ดังนั้นการปรับโมเดลด้วยข้อมูลจำลอง ซึ่งเป็นวิธีการที่สามารถ

เพิ่มจำนวนของข้อมูล และให้คำอ่านที่ถูกต้อง ย่อมส่งผลให้โมเดลที่ปรับได้มีผลการรู้จำเสียงพูดที่ดีขึ้นนั่นเอง นอกจากนี้ การปรับโมเดลด้วยข้อมูลจำลองยังใช้เวลาคำนวณในการปรับโมเดลน้อยกว่าวิธี PMC ทั้งนี้เป็นเพราะวิธี PMC ต้องมีการแปลงโดเมนไปกลับระหว่างโดเมนเซปสตรอลและโดเมนเสปคตรอลเชิงเส้น [4],[13] จึงทำให้วิธีการปรับโมเดลด้วยข้อมูลจำลองเหมาะสมกับระบบการปรับโมเดลแบบออนไลน์ ที่ต้องการความเร็วในการปรับโมเดล

- การปรับโมเดลด้วยข้อมูลจำลองกับเสียงพูดภาษาอื่นๆ ในวิทยานิพนธ์นี้ได้ทดลองการปรับโมเดลด้วยข้อมูลจำลองกับเสียงพูดภาษาญี่ปุ่นจากคลังข้อมูลมาตรฐานออโรร่าทูเจ ซึ่งเป็นคลังข้อมูลเสียงพูดที่มีเสียงรบกวนแบบบวก และเป็นเสียงพูดแบบต่อเนื่อง โดยใช้การปรับโมเดลด้วยวิธี MLLR ซึ่งใช้โมเดลตั้งต้นแบบ MULTI และใช้การปรับโมเดลด้วยข้อมูลจำลองร่วมกับเสียงพูดที่เข้ามา ผลการทดลองปรากฏว่าได้ผลการรู้จำเสียงพูดที่ดีกว่าโมเดลแบบ MULTI ที่ไม่มีการปรับโมเดลและที่มีการปรับโมเดลด้วยวิธี MLLR ที่ใช้เฉพาะเสียงพูดที่เข้ามาในการปรับโมเดล ซึ่งเป็นการยืนยันว่าการปรับโมเดลด้วยข้อมูลจำลองสามารถนำไปใช้กับภาษาอื่นได้ และสามารถใช้ได้กับเสียงพูดต่อเนื่องได้ด้วย โดยยังคงให้ผลการรู้จำเสียงพูดที่ดีขึ้น เหมือนกับการใช้กับเสียงพูดคำโดด

2) การสร้างโมเดลเสียงพูดขึ้นใหม่แบบออนไลน์ จากการประมาณค่าในช่วงของโมเดลเสียงพูดที่มีเสียงรบกวนหลายๆ ชนิดเข้าด้วยกัน ซึ่งเรียกโมเดลนี้ว่า โมเดลแบบ NCHI กล่าวคือ การประมาณค่าในช่วงนั้นได้ใช้ข้อมูลเสียงพูดที่เข้ามาในการหาค่าถ่วงน้ำหนัก และจำนวนชนิดของโมเดล เพื่อใช้ในการสร้างโมเดลเสียงพูดขึ้นใหม่ โดยในวิทยานิพนธ์นี้นำเสนอวิธีการหาค่าถ่วงน้ำหนัก และจำนวนชนิดของโมเดล 2 วิธี คือ วิธีการค้นหาแบบโครงสร้างต้นไม้ และ วิธีการค้นหาแบบตรง จากผลการทดลองสรุปได้ว่า

- โมเดลแบบ NCHI ด้วยวิธีการค้นหาแบบโครงสร้างต้นไม้ ให้ผลการรู้จำเสียงพูดที่น้อยกว่า MSTC ไม่มาก แสดงให้เห็นว่าโมเดลที่ได้จากการประมาณค่าในช่วงมีความใกล้เคียงกับโมเดลแบบผสมที่มีการสร้างไว้ล่วงหน้า
- โมเดลแบบ NCHI ด้วยวิธีการค้นหาแบบตรง ให้ผลการรู้จำเสียงพูดได้ดีกว่าโมเดลแบบ MSTC และโมเดลแบบ NCHI ด้วยวิธีการค้นหาแบบโครงสร้างต้นไม้ นั่นเป็นเพราะโครงสร้างแบบต้นไม้ทำให้มีรูปแบบการผสมโมเดลได้จำกัด ดังนั้นโมเดลที่มีอาจไม่ครอบคลุมเสียงรบกวนที่เกิดขึ้นจริงได้ทั้งหมด เป็นผลให้โมเดลที่สร้างขึ้นใหม่ด้วย NCHI แบบวิธีการค้นหาแบบตรง มีความใกล้เคียงกับเสียงพูดที่เข้ามามากกว่า ส่วนเวลาในการคำนวณ MSTC ใช้เวลาน้อยที่สุด เนื่องจากไม่ต้องเสียเวลาในการประมาณค่าในช่วงของโมเดลขึ้นมาใหม่ และมีจำนวนครั้งในการค้นหาโมเดลที่น้อยกว่า

- โมเดลแบบ NCHI ใช้พื้นที่ในการจัดเก็บโมเดลน้อยกว่าโมเดลแบบ MSTC เพราะ NCHI สามารถสร้างโมเดลแบบผสมขึ้นมาได้ใหม่ ทำให้ไม่ต้องมีการจัดเก็บโมเดล Noise Cluster HMM และ Noisy Speech HMM ของโมเดลแบบผสม แต่จะจัดเก็บเฉพาะโมเดลเสียงรบกวนพื้นฐานเท่านั้น ถึงแม้ว่าโครงสร้างต้นไม้จะมีจำนวนครั้งในการค้นหาน้อยกว่าการค้นหาแบบตรงก็ตาม แต่เวลาในการคำนวณของ NCHI ของการค้นหาทั้งสองวิธีก็มีความใกล้เคียงกัน เป็นเพราะว่า โหนดบนๆ ของโครงสร้างต้นไม้มีจำนวนโมเดลในการประมาณค่าในช่วงมากกว่าโหนดล่างๆ ซึ่งการค้นหาแบบโครงสร้างต้นไม้เป็นการค้นหาแบบบนลงล่าง ทำให้เวลาในการคำนวณโดยรวมมีความใกล้เคียงกับการค้นหาแบบตรง ซึ่งเป็นการประมาณค่าในช่วงของโมเดลจากจำนวนน้อยๆ แล้วค่อยๆ เพิ่มจำนวนขึ้น
- โมเดลแบบ NCHI ด้วยวิธีการค้นหาแบบตรง ที่ใช้สมการในการประมาณค่าในช่วงที่แตกต่างกัน ให้ผลการรู้จำเสียงพูดที่ไม่แตกต่างกัน ดังนั้นโมเดลแบบ NCHI ที่ใช้สมการแบบที่หนึ่งมีความเหมาะสมที่สุด เพราะใช้เวลาในการสร้างโมเดลขึ้นมาใหม่น้อยที่สุด เนื่องจากมีขั้นตอนการคำนวณน้อยที่สุด

3) การปรับโมเดลด้วยข้อมูลจำลองร่วมกับ NCHI คือ การใช้ NCHI ในการสร้างโมเดลตั้งต้นให้มีความใกล้เคียงกับเสียงพูดที่เข้ามา แล้วนำโมเดลที่ได้จาก NCHI มาปรับโมเดลด้วยข้อมูลจำลอง เพื่อให้ได้โมเดลที่มีความใกล้เคียงเสียงพูดที่เข้ามายิ่งขึ้น จากผลการทดลองสรุปได้ว่า

- S-NCHI ให้ผลการรู้จำมากกว่า PLT ซึ่งมีการปรับโมเดลตั้งต้นแบบ MSTC ที่ใช้เฉพาะเสียงพูดที่เข้ามาในการปรับโมเดล นั่นเป็นเพราะว่าโมเดลตั้งต้นของ S-NCHI มีความใกล้เคียงกับเสียงพูดที่เข้ามามากกว่า และ S-NCHI ยังสามารถแก้ปัญหาเสียงพูดที่เข้ามามีขนาดเล็กและไม่ทราบคำอ่านที่ถูกต้องได้ ด้วยการปรับโมเดลด้วยข้อมูลจำลอง จึงทำให้โมเดลที่ได้มีความใกล้เสียงพูดที่เข้ามามากกว่า PLT
- S-NCHI ให้ผลการรู้จำเสียงพูดสูงกว่า การปรับโมเดลด้วยข้อมูลจำลองร่วมกับ MSTC ซึ่งในที่นี้เรียกว่า “S-MSTC” เพียงเล็กน้อย เป็นผลจากการปรับโมเดลด้วยข้อมูลการจำลอง ซึ่งทำให้โมเดลแบบ NCHI และแบบ MSTC ที่ผ่านการปรับโมเดลมีความใกล้เคียงกันมากขึ้น
- NCHI ให้ผลการรู้จำเสียงพูดของ Test-2 ซึ่งเป็นเสียงพูดคำโดดที่มีเสียงรบกวนจากสิ่งแวดล้อมจริง ได้ดีกว่า S-MSTC และ PLT ซึ่งแสดงให้เห็นว่าโมเดลที่ได้จากการประมาณค่าในช่วง อาจจะสร้างโมเดลขึ้นมาใหม่ ให้มีความใกล้เคียงกับเสียงพูดที่เข้ามาได้มากกว่า MSTC ที่มีการปรับโมเดลได้
- S-NCHI ให้ผลการรู้จำเสียงพูดของ Test-1 ซึ่งเป็นเสียงพูดคำโดดที่มีเสียงรบกวนแบบบวกได้ดีกว่า NCHI ซึ่งแสดงให้เห็นว่าโมเดลที่ได้จาก NCHI ยังไม่ครอบคลุมเสียงรบกวนที่เกิดขึ้นจริงทุกชนิด เป็นผลให้ S-NCHI ที่มีการปรับโมเดล ให้โมเดลที่ใกล้เคียงเสียงพูดที่เข้ามามากกว่า ส่วนผลการรู้จำเสียงพูดของ Test-2 ด้วย S-NCHI ให้ผลที่ใกล้เคียงกับ NCHI

ทั้งนี้เป็นเพราะว่าโมเดลที่ได้จาก NCHI มีความใกล้เคียงกับเสียงพูดที่เข้ามาของ Test-2 อยู่แล้ว จึงทำให้โมเดลที่ได้จาก S-NCHI ไม่ได้ให้ผลการรู้จำเสียงพูดที่ดีขึ้นนั่นเอง จากผลการทดลองพบว่า ในกรณีที่โมเดลแบบ NCHI ยังไม่ครอบคลุมเสียงรบกวนที่เกิดขึ้นจริง การใช้ S-NCHI ช่วยให้โมเดลที่ปรับได้มีความใกล้เคียงกับเสียงพูดที่เข้ามามากขึ้น ส่วนในกรณีที่โมเดลแบบ NCHI มีความใกล้เคียงเสียงพูดที่เข้ามาอยู่แล้ว การใช้ S-NCHI ไม่ช่วยให้โมเดลที่ปรับได้มีผลการรู้จำเสียงพูดที่เพิ่มขึ้น แต่ก็ไม่ทำให้ผลการรู้จำเสียงพูดของโมเดลแบบ NCHI ลดลง

วิทยานิพนธ์นี้สรุปได้ว่า S-NCHI ที่ใช้โมเดลตั้งต้นจาก NCHI ที่มีการประมาณค่าในช่วงของ HMM ด้วยสมการแบบที่หนึ่ง ซึ่งมีการค้นหาค่าถ่วงน้ำหนักและจำนวนชนิดของโมเดลด้วยการค้นหาแบบตรง แล้วนำโมเดลที่ได้ไปปรับโมเดลด้วยข้อมูลจำลองแบบที่สาม ที่มีการดึงเสียงรบกวนแบบการอิงกับหน่วยเสียงและใช้ Train-S เป็นผู้พูดแบบ MIX ที่มีจำนวนคำเท่ากับ 22 คำ ให้ผลการรู้จำเสียงพูดที่มากกว่าระบบปรับโมเดลแบบออนไลน์ ที่นิยมใช้ในปัจจุบัน คือ PLT สาเหตุที่ทำให้ S-NCHI ให้ผลการรู้จำเสียงพูดที่ดีกว่านั้น เป็นผลมาจาก S-NCHI สามารถแก้ปัญหาพื้นฐาน 3 ข้อ ของระบบปรับโมเดลแบบออนไลน์ได้ คือ 1) ปัญหาการไม่ทราบคำอ่านของเสียงพูดที่เข้ามา 2) ปัญหาเสียงพูดที่เข้ามามีปริมาณน้อย ซึ่งทั้งสองปัญหาแก้ไขได้ด้วยการปรับโมเดลด้วยข้อมูลจำลอง และ 3) ปัญหาการมีโมเดลที่มีรูปแบบการผสมได้จำนวนจำกัดของ MSTC ซึ่งแก้ไขด้วย NCHI นอกจากนี้ NCHI ยังช่วยลดจำนวนโมเดลแบบผสมที่ต้องจัดเก็บเพิ่มด้วย

ข้อจำกัดของ S-NCHI

- การปรับโมเดลด้วยข้อมูลจำลอง ไม่สามารถใช้กับเสียงพูดที่มีเสียงรบกวนแตกต่างกันตลอดทั้งเสียงพูด เพราะในขั้นตอนของการปรับโมเดลด้วยข้อมูลจำลอง ต้องมีการนำเสียงรบกวนจากส่วนที่เป็นเสียงเงียบส่วนหน้าและเสียงเงียบส่วนหลัง มาต่อกันจนกว่าจะได้ความยาวเท่ากับเสียงพูดสะอาด การต่อกันในลักษณะนี้ทำให้เสียงรบกวนที่จำลองได้ไม่เหมือนกับเสียงรบกวนที่เข้ามา
- การปรับโมเดลด้วยข้อมูลจำลอง ไม่สามารถใช้กับข้อมูลเสียงพูดที่ไม่มีส่วนที่เป็นเสียงเงียบได้ เพราะข้อมูลจำลองต้องใช้ส่วนที่เป็นเสียงเงียบ ในการสร้างเสียงรบกวนพื้นหลังเพื่อนำไปรวมกับเสียงพูดสะอาด
- NCHI ต้องการโมเดลเสียงรบกวนพื้นฐานที่มากเพียงพอ จึงจะสามารถรองรับเสียงรบกวนที่ไม่อยู่ในชุดฝึกสอนได้ เพราะการที่มีจำนวนโมเดลพื้นฐานที่น้อยจะทำให้มีรูปแบบการสร้างโมเดลแบบผสมได้น้อยตามไปด้วย ซึ่งทำให้มีโอกาสสร้างโมเดลให้สามารถรองรับเสียงรบกวนที่ไม่ได้อยู่ในชุดฝึกสอนได้น้อยลง

7.2 ข้อเสนอแนะ

ปัญหาที่พบในวิทยานิพนธ์นี้ แบ่งออกเป็น 2 ปัญหา คือ 1) ปัญหาที่เกิดจากการปรับโมเดลด้วยข้อมูลจำลอง และ 2) ปัญหาที่เกิดจากโมเดลแบบ NCHI ซึ่งแต่ละปัญหามีรายละเอียดดังนี้

1) ปัญหาของการปรับโมเดลด้วยข้อมูลจำลองที่ต้องการแก้ไข คือ

- การดึงเสียงรบกวนให้ได้ผลดีกว่านี้ เพราะการดึงเสียงรบกวนที่นำเสนอนี้ เป็นการใช้เฉพาะเสียงรบกวนในส่วนที่เป็นเสียงเงียบเท่านั้น อาจทำให้ขาดลักษณะสำคัญบางอย่างของเสียงรบกวนที่อยู่ในเสียงพูดได้ ซึ่งวิธีที่น่าจะนำมาใช้ในการแก้ปัญหานี้ คือ การแยกเสียงรบกวนและเสียงพูดสะอาดออกจากกัน เพื่อให้ได้เสียงรบกวนที่อยู่ในตำแหน่งที่มีเสียงพูดสะอาดด้วย เช่น การใช้เทคนิค Microphone Array [1] เป็นต้น
- การสร้างเสียงรบกวนพื้นหลังให้ได้ผลดีกว่านี้ เพราะการสร้างเสียงรบกวนพื้นหลังที่นำเสนอ เป็นการต่อส่วนที่เป็นเสียงรบกวนที่ได้มาโดยตรง ทำให้เกิดความไม่เรียบในช่วงรอยต่อ ส่งผลให้เกิดเป็นเสียงรบกวนขึ้นมา ทำให้เสียงรบกวนพื้นหลังที่สร้างได้มีความแตกต่างกับเสียงรบกวนที่เกิดขึ้นอยู่ ซึ่งวิธีที่ใช้ในการแก้ปัญหานี้ คือ การใช้เทคนิคการทำให้เรียบเข้ามาช่วย แต่ก็ทำให้ใช้เวลาในการคำนวณเพิ่มขึ้นตามไปด้วย
- การเลือกผู้พูดของ Train-S ให้ใกล้เคียงกับผู้พูดในเสียงพูดที่เข้ามา ย่อมทำให้ความแตกต่างเรื่องผู้พูดลดลง แต่วิธีการเลือกผู้พูดในสภาพแวดล้อมที่มีเสียงรบกวน ให้ได้ประสิทธิภาพดี ยังคงเป็นปัญหาอยู่ในปัจจุบัน
- พัฒนาวิธีการปรับโมเดลสำหรับการปรับโมเดลแบบออนไลน์ ที่ให้ผลการรู้จำเสียงพูดที่ดีขึ้น ใช้เวลาในการคำนวณน้อยลง
- วิธีที่วิทยานิพนธ์นี้นำเสนอ ไม่สามารถแก้ปัญหาของเสียงรบกวนที่เกิดขึ้นแบบทันทีทันใดแล้วหายไปได้ ซึ่งเสียงรบกวนประเภทนี้เรียกว่า เสียงรบกวนแบบไม่คงที่ ซึ่งโดยปกติแล้วเสียงรบกวนแบบไม่คงที่ จะใช้เทคนิคการหาส่วนที่เป็นเสียงพูดและส่วนที่ไม่เป็นเสียงพูดในการระบุช่วงเวลาที่เป็นเสียงรบกวนและช่วงเวลาที่เป็นเสียงพูด [47] เพื่อที่จะข้ามสัญญาณช่วงนี้ หรือตัดสัญญาณช่วงนี้ออก อย่างไรก็ตาม ปัญหาของการหาส่วนที่เป็นเสียงพูดและส่วนที่ไม่เป็นเสียงพูด ยังคงเป็นปัญหาที่มีการวิจัยอย่างต่อเนื่องอยู่ในปัจจุบัน

2) ปัญหาของโมเดลแบบ NCHI ที่ต้องแก้ไข คือ

- พัฒนาสมการในการประมาณค่าในช่วงของโมเดล รวมถึงการค้นหาค่าถ่วงน้ำหนัก และจำนวนชนิดของโมเดล เพื่อให้ได้โมเดลที่ใกล้เคียงกับเสียงพูดที่เข้ามามากยิ่งขึ้น ซึ่งทำให้ได้ผลการรู้จำเสียงพูดดีขึ้น นอกจากนี้ เวลาที่ใช้ในการคำนวณก็ต้องลดลงด้วย
- ปัญหาของ NCHI ที่ต้องมีโมเดลเสียงรบกวนพื้นฐานที่มีคุณภาพ และต้องมีจำนวนชนิดของเสียงรบกวนที่มาก ซึ่งปัญหานี้สามารถแก้ไขได้ ด้วยการสร้างคลังข้อมูลเสียงรบกวนขนาดใหญ่

เอกสารอ้างอิง

[1] Gang Y. "Speech recognition in noisy environments: A survey." **Speech Communication**, vol. 16, no. 3, Apr. 1995. pp. 261-291

[2] Gales M.J.F. "**Model-Based Techniques for Noise Robust Speech Recognition.**" PhD Thesis of University of Cambridge. 1995

[3] Zhang Z.P. "**A Study on Increasing Robustness against Speaker and Noise Variations in Speech Recognition.**" PhD thesis of Department of Computer Science Graduate School of Information Science and Engineering Tokyo Institute of Technology. 2002

[4] Zhang Z.P. and Furui S. "Piecewise-Linear Transformation-Based HMM Adaptation for Noisy Speech." **Speech Communication**, vol. 42, no. 1, Jan. 2004. pp. 43–58

[5] Zhang Z.P. and Furui S. "Tree-Structured Clustering Methods for Piecewise Linear transformation-Based Noise Adaptation." **IEICE Trans. Inf. & Syst.**, vol. E88–D, no. 9, Sep. 2005. pp. 2168-2176

[6] Leggetter C.J. and Woodland P.C. "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density HMMs." **Computer Speech Language**, vol.9, no. 2, Apr. 1995. pp 171–186

[7] Gales M.J.F. and Woodland P.C. "Mean and Variance Adaptation within the MLLR Framework." **Computer Speech Language**, vol. 10, no. 4, Oct. 1996. pp 249-264

[8] Kosaka T., Matsunaga S., and Sagayama S. "Speaker-Independent Speech Recognition Based on Tree-Structured Speaker Clustering." **Computer Speech Language**, vol. 10, no. 1, Jan. 1996. pp. 55–74

[9] Rabiner J.R. and Juang B.H. **Fundamentals of Speech Recognition**. New Jersey : Prentice-Hall, Inc. 1993

[10] Gales M.J.F. and Young S. "An Improved Approach to the Hidden Markov Model Decomposition of Speech and Noise." **Proc. IEEE-ICASSP**, San Francisco, USA, Mar. 1992. pp. 233-236

[11] Martin F., Shikano K. and Minami Y. "Recognition of Noisy Speech by Composition of Hidden Markov Models." **Proc. Eurospeech**, Belin, Germany, Sep. 1993. pp. 1031-1034.

[12] Minami Y. and Furui S. "A Maximum Likelihood Procedure for a Universal Adaptation Method Based on HMM Composition." **Proc. IEEE-ICASSP,** Michigan, USA, May 1995. pp. 129-132

[13] Furui S. and Itoh D. "Neural Networks-Based HMM Adaptation for Noisy Speech." **Proc. IEEE-ICASSP**, Salt Lake City, Utah, May 2001. pp. 365-368

[14] Nakamura S., Yamamoto K., Takeda K., Kuroiwa S., Kitaoka N., Yamada T., Mizumachi M., Nishiura T., Fujimoto M., Saso A. and T. Endo T. "Data Collection and Evaluation of Aurora-2 Japanese Corpus." **Proc. ASRU**, Virgin Islands, USA, Nov. 2003. pp. 619-623

[15] Gauvain J.L., and Lee C.H. "Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains." **IEEE Trans. Speech Audio Processing**, vol. 2, no. 2, Apr. 1994. pp. 291–298

[16] Chesta C., Siohan O., and Lee C.-H. "Maximum a Posteriori Linear Regression for Hidden Markov Model Adaptation." **Proc. EuroSpeech**, Budapest, Hungary, Sep. 1999. pp. 211-214

[17] Thatphithakkul N., Kruatrachue B., Wutiwiwatchai C., Marukatat S., and Boonpiam V. "A Simulated-data Adaptation Technique for Robust Speech Recognition." **Proc. Interspeech**, Pittsburgh, USA, Sep. 2006. pp. 777-780

[18] Thatphithakkul N., Kruatrachue B., Wutiwiwatchai C., Marukatat S., and Boonpiam V. "Combined Simulated Data Adaptation and Piecewise Linear Transformation for Robust Speech Recognition." **Proc. ECTI-CON**, Chiang Rai, Thailand, May 2007. pp. 1038-1041

[19] Thatphithakkul N., Kruatrachue B., Wutiwiwatchai C., Marukatat S., and Boonpiam V. "Tree-Structured Model Selection and Simulated-data Adaptation for Environmental and Speaker Robust Speech Recognition." **Proc. ISCIT**, Sydney, Australia, Oct. 2007. pp. 1570 – 1574

[20] Xu H., Tan Z-H., Dalsgaard P., and Lindberg B. "Robust Speech Recognition Based on Noise and SNR Classification - A Multiple-Model Framework." **Proc. Interspeech**, Lisbon, Portugal, Sep. 2005. pp. 977-980

[21] Thatphithakkul N., Kruatrachue B., Wutiwiwatchai C., Marukatat S., and Boonpiam V. "Robust Speech Recognition Using PCA-Based Noise Classification." **Proc. SPECOM**, Patras, Greece, Oct. 2005. pp. 345-348

[22] Thatphithakkul N., Kruatrachue B., Wutiwiwatchai C., Marukatat S., and Boonpiam V. "KPCA-Based Noise classification Model for Robust Speech Recognition system." **Proc. ECTI-CON**, Ubon Ratchathani, Thailand, May 2006. pp. 231-234

[23] Thatphithakkul N., Kruatrachue B., Wutiwiwatchai C., Marukatat S., and Boonpiam V. "Robust Speech Recognition Using KPCA-Based Noise Classification." **ECTI Trans. on Computer and Information Technology**, vol. 2, no. 1, May 2006. pp.60-68

[24] Yoshimura T. "**Simultaneous Modeling of Phonetic and Prosodic Parameters, and Characteristic Conversion for HMM-based Test-to-Speech Systems.**" PhD thesis of Department of Electrical and Computer Engineering Nagoya Institute of Technology. 2002

[25] Yoshimura T., Tokuda K., Masuko T., Kobayashi T. and Kitamura T. "Speaker Interpolation for HMM-Based Speech Synthesis System." **The Journal of the Acoustical Society of Japan (E)**, vol. 21, no. 4, Jul. 2000. pp. 199-206

[26] Kasuriya S., Kanokphara S., Thatphithakkul N., Cotsomrong P. and Sunpethniyom T. "Context-Independent Acoustic Models for Thai Speech Recognition." **Proc. ISCIT**, Sapporo, Japan, Oct. 2004. pp. 991-994

[27] Young S., Evermann G., Gales M., Hain T., Kershaw D., Liu X., Morre G., Odell J., Ollason D., Povey D., Valchev V. and Woodland P. "**The HTK book version 3.4.**" [Online]. Available : http://htk.eng.cam.ac.uk. 2006

[28] Mansour D. and Juang B.H. "The short-time modified coherence representation and noisy speech recognition." **IEEE Trans. ASSP**, vol. 37, no. 6, Jun. 1989. pp. 795-804

[29] Siohan O. "On the Robustness of Linear Discriminant Analysis as a Preprocessing Step for Noisy Speech Recognition." **Proc. IEEE-ICASSP**, Michigan, USA, May 1995. pp. 125-128

[30] Hermansky H. "Perceptual Linear Predictive (PLP) Analysis of Speech." **Journal of the Acoustical Society of America**, vol. 87, no. 4, Apr. 1990. pp. 1738-1752

[31] Koehler J., Morgan N., Hermansky H., Gunter-Hirsh H., and Tong G. "Integrating RASTA-PLP into Speech Recognition." **Proc. IEEE-ICASSP**, Adelaide, Australia, Apr. 1994. pp. 421-424

[32] Atal B. "Effectiveness of Linear Prediction Characteristics of the Speech Wave for Automatic Speaker Identification and Verification." **Journal of Acoustical Society of America**, vol. 55, no. 6, Jun. 1974. pp. 1304–1312

[33] Boll S. "Suppression of Acoustic Noise in Speech using Spectral Subtraction." **IEEE Trans. Speech Audio Processing**, vol. ASSP-27, no. 2, Apr. 1979. pp. 113–120

[34] Neumeyer L and Weintraub M. "Probabilistic Optimum Filtering for Robust Speech Recognition." **Proc. IEEE-ICASSP**, Adelaide, Australia, Apr. 1994. pp. 417-420

[35] Acero A. **Acoustical and Environmental Robustness in Automatic Speech Recognition**. Boston : Kluwer Academic Publishers. 1992

[36] Frazer R.H., Samsam S., Braida L.D. and Oppenheim A.V. "Enhancement of Speech by Adaptive Filtering." **Proc. IEEE-ICASSP**, Atlanta, Georgia, April 1976. pp. 251-253

[37] Kalman R.E. "A New Approach to Linear Filtering and Prediction Problems." **Transaction of the ASME-Journal of basic engineering**. vol. 82 (Series D), no. 1, Mar. 1960. pp. 35-45

[38] Huang C.S., Wang H.C., and Lee C.H. "An SNR-Incremental Stochastic Matching Algorithm for Noisy Speech Recognition." **IEEE Trans. Speech Audio Processing**, vol.9, no.8, Nov. 2001. pp. 866–873

[39] Zhao Y., Wang L., Chu M., Soong F.K., and Cao Z. "Refining Phoneme Segmentations using Speaker-Adaptive Context Dependent Boundary Models." **Proc. Interspeech**, Lisbon, Portugal, Sep. 2005. pp. 2557-2560

[40] Shao C. and Bouchard M. "Efficient Classification of Noisy Speech using Neural Networks." **Proc. ISSPA**, Paris, France, Jul. 2003. pp. 357-360

[41] Gaunard P., Mubikangiey C.G., Couvreur C., and Fontaine V. "Automatic Classification of Environmental Noise Events by Hidden Markov Models." **Proc. IEEE-ICASSP**, Washington, USA, May 1998. pp. 3609-3612

[42] Ma L., Smith D. and Milner B., "Context Awareness using Environmental Noise Classification." **Proc. Eurospeech**, Geneva, Switzerland, Sep. 2003. pp. 2237-2240

[43] The International Phonetic Association. Handbook of the International Phonetic Association : A Guide to the use of the International Phonetic Alphabet. Cambridge : Cambridge University Press. 1999

[44] Kasuriya S., Sornlertlamvanich V., Cotsomrong P., Jitsuhiro T., Kikui G. and Sagisaka Y. "NECTEC-ATR Thai Speech Corpus." **Proc. Oriental COCOSDA**, Singapore, Jun. 2003. pp 105-111

[45] Itahashi S. "Creating Speech Corpora for Speech Science and Technology." **IEICE Trans. on Fundamentals of Electronics, Communication and Computer Sciences,** vol. E74, no. 7, Jul. 1991. pp. 1906-1910

[46] Carnegie Mellon University. "**NOISEX-92 Database.**" [Online]. Available : http://www.speech.cs.cmu.edu/comp.speech/Section1/Data/noisex.html. 1992

[47] Martin A. and Mauuary L. "Robust Speech/Non-Speech Detection Based on LDA-Derived Parameter and Voicing Parameter for Speech Recognition in Noisy Environments." **Speech Communication**, vol. 48, no. 1, Jan. 2006. pp. 191-206

ภาคผนวก

ภาคผนวก ก
งานวิจัยที่ได้รับการตีพิมพ์

[1] Thatphithakkul N., Kruatrachue B., Wutiwiwatchai C., Marukatat S., and Boonpiam V. "Robust Speech Recognition Using PCA-Based Noise Classification." **Proc. SPECOM**, Patras, Greece, Oct. 2005. pp. 345-348

[2] Thatphithakkul N., Kruatrachue B., Wutiwiwatchai C., Marukatat S., and Boonpiam V. "Tree-Structured Model Selection and Simulated-data Adaptation for Environmental and Speaker Robust Speech Recognition." **Proc. ISCIT**, Sydney, Australia, Oct. 2007. pp. 1570 – 1574

[3] Thatphithakkul N., Kruatrachue B., Wutiwiwatchai C., Marukatat S., and Boonpiam V. "Simulated-data Adaptation Based Piecewise Linear Transformation for Robust Speech Recognition." **ASEAN Journal on Science & Technology for Development,** vol. 24, no. 4, 2007.

[4] Thatphithakkul N., Kruatrachue B., Wutiwiwatchai C., Marukatat S., and Boonpiam V. "KPCA-Based Noise classification Model for Robust Speech Recognition system." **Proc. ECTI-CON**, Ubon Ratchathani, Thailand, May 2006. pp. 231-234

[5] Thatphithakkul N., Kruatrachue B., Wutiwiwatchai C., Marukatat S., and Boonpiam V. "Robust Speech Recognition Using KPCA-Based Noise Classification." **ECTI Trans. on Computer and Information Technology**, vol. 2, no. 1, May 2006. pp.60-68

[6] Thatphithakkul N., Kruatrachue B., Wutiwiwatchai C., Marukatat S., and Boonpiam V. "A Simulated-data Adaptation Technique for Robust Speech Recognition." **Proc. Interspeech**, Pittsburgh, USA, Sep. 2006. pp. 777-780

[7] Thatphithakkul N., Kruatrachue B., Wutiwiwatchai C., Marukatat S., and Boonpiam V. "Combined Simulated Data Adaptation and Piecewise Linear Transformation for Robust Speech Recognition." **Proc. ECTI-CON**, Chiang Rai, Thailand, May 2007. pp. 1038-1041

SPECOM 2005

10th International Conference SPEECH and COMPUTER

17 – 19 October, 2005 Patras, Greece

PROCEEDINGS

Organizers:

University of Patras
Wire Communications Laboratory
Patras, Greece

Moscow State Linguistic University
Moscow, Russia

Robust Speech Recognition Using PCA-Based Noise Classification

Nattanun Thatphithakkul[1], Boontee Kruatrachue[1], Chai Wutiwiwatchai[2], Sanparith Marukatat[2] and Vataya Boonpiam[2]

[1]Department of Computer Engineering
King Mongkut's Institude of Technology Ladkrabang, Bangkok, 10520, Thailand
[2]Information Research and Development Division
National Electronics and Computer Technology Center, Bangkok, 12120, Thailand
S6060008@kmitl.ac.th, kkboontee@kmitl.ac.th, chai@nectec.or.th,
sanparith.marukatat@nectec.or.th, vataya.boonpiam@nectec.or.th

Abstract

This paper proposes a new environmental noise classification using principal component analysis (PCA) for robust speech recognition. Once the type of noise is identified, speech recognition performance can be enhanced by selecting the identified noise specific acoustic model. The proposed model applies PCA to a set of noise features, and results from PCA are used by a pattern classifier for noise classification. Instead of including both clean and noisy environments in a single classifier, two-step classification is introduced by separating the clean from noisy environments and then identifying the type of noisy environments. The proposed model is evaluated with four types of noise: white, pink, babble, and car from NOISEX-92 and shows a promising result regardless of signal-to-noise ratio (SNR).

1. Introduction

It is commonly known that a speech recognition system trained by speech in a clean or nearly clean environment cannot achieve good performance when used in noisy environment. Research on robust speech recognition is then necessary. Gales [7] has classified the techniques of robust speech recognition into 4 approaches: 1) extraction of robust speech feature, 2) estimation of clean speech, 3) construction of robust model, and 4) combination of 3 previous techniques. Each of these approaches has both advantages and disadvantages. This paper focuses on the third approach, the model-based approach, which has achieved good recognition results [7]. The model-based approach aims to create or adapt the acoustic model in specific environments. Several techniques of model adaptation have been proposed [13] such as linear regression adaptation, model-based stochastic matching, hypothesized wiener filtering, and parallel model combination. However, an acoustic model trained directly for specific noise is certainly superior to the adapted model, although multiple acoustic models are needed for various kinds of noise and an accurate automatic noise classification is required.

Many noise classification techniques have been studied previously. Classical technique is based on hidden Markov models (HMM) with mel-frequency cepstral coefficients (MFCC) [10], which have been proven to give better results than human listeners. Another successful technique is a neural network (NN) based system with combined features of line spectral frequencies (LSF), a zero-crossing (ZC) rate and energy [12]. In [11] several classifiers have been evaluated for noise classification. Experimentally, an optimal Bayes with LSF is the best technique for this task.

Since the LSF is complicated and implementing LSF in a real-time system is problematic, we aim to explore a simpler feature extraction method. This paper proposes a noise classification technique based on principal component analysis (PCA). PCA has been commonly used for pattern and image recognition [1]. In this paper, PCA is applied to extract environmental-noise features, which are used by a pattern classifier for noise classification. An advantage of PCA is to reduce the dimension of feature vectors while retaining as much significant information as possible. The computational requirement of PCA applied to normalized logarithmic spectrums (NLS) implemented in this paper is much lower than the MFCC or other effective features such as LSF [10, 12]. NN and Support Vector Machines (SVM) are evaluated for the noise classification. Moreover, instead of a single classifier, two-step classification is proposed by first separating the clean from noisy environments and then identifying the type of noisy environment. Our noise classification model is evaluated on 5 classes of environments: clean, white, pink, babble, and car. Our Thai isolated-word recognition [8] with noise-specific acoustic models is used in the evaluation. It is noted that although the task is isolated-word recognition, phonemes are used as basic recognition units. This facilitates new words addition.

The rest of paper is organized as follows: the next section describes an overall structure of our robust speech recognition system. In Sect. 3, the use of PCA in our system is presented. Sect. 4 describes our experiments, results and discussion. The last section concludes the paper and gives future works.

2. Robust Speech Recognition Using Noise Classification

As described in the previous section, our robust speech recognition system uses a model-based technique, in which acoustic models are trained from speech in specific environment. Given a speech signal, a set of features for noise classification is extracted from a short period of silence at the beginning of signal. It is noted that this short period is assumed to be a silence where the speaker has not yet uttered. This assumption can be realized for our push-to-talk interface. To apply our system with other user interfaces, we need an additional module of speech/non-speech classification or other strategies to capture a non-speech portion from an input signal. Features extracted from the silence portion are then used to identify the type of environment. Once knowing the

environment type, the recognizer selects a corresponding acoustic model for recognizing the rest of signal. An overall structure is illustrated in Fig. 1.

In our system, there are 2 particular difficulties:

- How to construct a robust acoustic model for a variation of signal-to-noise ratio (SNR)? In our system, a particular acoustic model is trained on noisy speech with various levels of SNR. Clean speech, whose SNR exceeds 30 dB is also combined in the training set of each noisy acoustic model.
- How to construct the environment or noise classification module? Time consuming by the noise classification module should be as low as possible, so that the overall system can achieve an acceptable processing time. The construction of such module is the main objective of this paper.

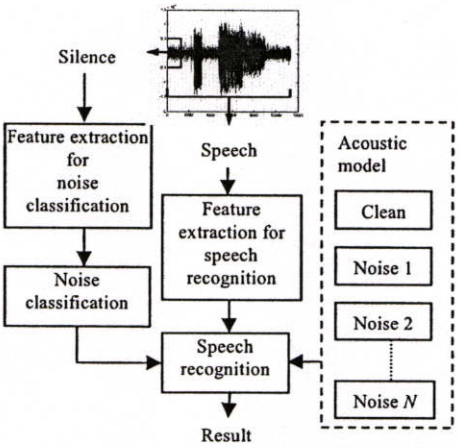


Figure 1: Overall structure of robust speech recognition

3. Noise Classification Based On PCA

The goal of noise classification is to identify the type of speech environments. To achieve this, we rely on NLS as basic feature vectors. Indeed, this feature is more robust to the change of SNR and less computational demanding when comparing to the conventional MFCC. In this paper, we are interested in applying the PCA technique [1] to these features in order to extract the most significant features. PCA has been widely used as feature extraction in pattern recognition. The main concept of PCA is to project the original feature vector onto principal component axes. These axes are orthogonal and correspond to the directions of greatest variance in the original feature space. Hence, projecting input vectors onto this principal subspace allows reducing the redundancy in the original feature space as well as the dimension of input vectors. For simplification, we will call the projected feature vector, the "weight vector" hereafter.

Fig. 2 shows weight vectors obtained from 108,800 feature vectors derived from all 5 types of environment (clean, white, pink, babble, and car) with various SNR, onto a two-dimensional space using PCA. There is still a lot of overlapping area among 5 environments. However, the type of noise is easily identified, if the clean environment is excluded as shown in Fig. 3, where weight vectors are obtained by applying PCA on only 4 environments. Therefore, to handle this set of environments, we propose a new design based on two-step classification. The first classifier separates a clean environment from noisy environments. Then the second classifier identifies the type of noisy environments.

Certainly, there are other ways to design multi-step classification; experimentally we have found that the proposed 2-step classification is the most appropriate for this set of environments. We are currently working on fully automatic determination of multi-stage noise classification technique. We believe that this idea can be further extended to cope with other noises.

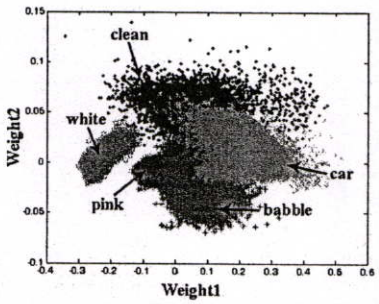


Figure 2: Distributions of PCA reduced features computed from 5 environments

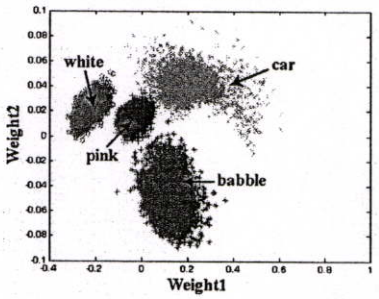


Figure 3: Distributions of PCA reduced features computed from 4 environments

For the classification algorithm, a fast and efficient technique is needed. In our experiment, two well-known classification algorithms; a neural network (NN) and support vector machines (SVM) are evaluated.

4. Experimental Results

4.1. Data preparation

Four types of noise from NOISEX-92 [3] including white, pink, babble, and car were preprocessed by reducing the sampling rate to 8 kHz. Clean speech at 16 bits and 16 kHz from NECTEC-ATR Thai speech corpus [2] was resampled down to 8 kHz and used for the speech in clean environment.

4.1.1. Data set for noise classification

A data set for noise classification contained noisy speech prepared by adding the noise from NOISEX-92 to the clean speech of NECTEC-ATR at various SNRs (0 – 15 dB). The data set was divided into 3 subsets: a PCA training set (5000 samples), a classifier training set (61,530 samples) and a classifier test set (108,800 samples). The first set was used for computing PCA weight vectors. The second and third sets were used for training and evaluating the noise classifier. A

small frame of 1,024 samples at the beginning of the speech signal, which was expected to be silence, was used for PCA and noise classification. As described in the Sect. 2, our speech recognizer is designed for a push-to-talk interface. With this interface, we can control the recorder to start record a silence signal before the beginning of speech. NLS used for noise classification were computed from this silence frame.

4.1.2. Data set for speech recognition

The speech recognition task in our experiment was phoneme-based isolated-word recognition. Speech files were taken from the NECTEC-ATR. 32000 speech utterances from 32 speakers (16 males and 16 females) were allocated for a training set. Another set of 6400 utterances from other 10 speakers (5 males and 5 females) are used for testing. The dictionary of 5,000 most frequently used Thai words is used in this task.

Four types of noise described in the Sect. 3 were added to the speech files at various SNRs of 15, 10, 5, and 0 dB. An acoustic model contained HMMs representing 35 Thai phones [8]. Each HMM consisted of 5 states and 16 Gaussian mixtures per state. 39 dimensional vectors (12 MFCC, 1 log-energy, and their first and second derivatives) were used as recognition features.

4.2. Noise classification results

Our proposed classification model described in the Sect. 3 was compared to the classical technique using HMM [10, 11], which served as a baseline system in our experiment. The following are details of each noise classification system.

4.2.1. Baseline HMM system

The HMM [6] based noise classification system contained 5-state HMMs with 16-Gaussian mixtures per state. MFCCs are used as classification features. This baseline system will be referred to as "HMM_MFCC".

4.2.2. PCA and NN based system

NN [4] used for our noise classification was a Multi-Layer Perceptron (MLP). There was 1 hidden layer with the number of hidden nodes empirically adjusted to 30. Two systems based on NN were constructed. The first system, called a "NN_1-step" model, classified 5 types of environment (4 types of noise plus a clean environment) using a single classifier, i.e. one NN. This classifier used the output of PCA applied to NLS as analyzed feature. In this work, only 2 principal axes have been proven to be sufficient for our noise classification. The other, called a "NN_2-step" model, was the two-step classifier described in Sect. 3.

4.2.3. PCA and SVM based system

SVM [5] used in our experiment was a multi-class SVM based on one-against-one algorithm. According to our preliminary experiment, radial basis function (RBF) was appropriate for the SVM kernel. Similar to the NN systems, two SVM-based systems, called "SVM_1-step" and "SVM_2-step" models, were constructed for 1-step and 2-step classification respectively.

Table 1 presents results obtained from each system described above. Table 2 also gives the distribution of samples misclassified by each system for every noise type. According to the results, PCA applied to the NLS is comparable to the baseline HMM systems. The two-step classification systems using either NN or SVM are superior to the conventional 1-step classification design and the baseline HMM systems. It should be noted that, in this experiment, the babble noise varies highly on time axis. From Table 2, the HMM seems to be effective for this noise. Neither the NN nor SVM can outperform the HMM for this case. In contrast, the rest of noise types are almost stationary, so that the NN and SVM can capture the characteristics of noises from only the short sample period.

Table 1: Error rate results of noise classification based on HMM, NN, and SVM.

System	% Error rate
HMM_MFCC	4.18
NN_1-step	4.33
SVM_1-step	3.86
NN_2-step	**1.20**
SVM_2-step	1.57

Table 2: Distribution of noise classification errors produced by HMM, NN, and SVM.

Type	Number of example error				
	HMM	NN_ 1-step	SVM_ 1-step	NN_ 2-step	SVM_ 2-step
Clean	176	118	77	172	89
White	112	2	0	1	1
Pink	263	99	114	1	5
Babble	143	2728	1494	28	30
Car	3854	1768	2425	1103	1587
total	4548	4715	4110	1305	1712

4.3. Speech recognition results

In this section, several robust speech recognition techniques including our proposed model are experimentally compared. The first system (S1) was a conventional system without any implementation for robust speech recognition. The second system (S2) used zero-mean static coefficient [6], a well-known technique for noise-robust speech features. The third system (S3) used a multi-condition acoustic model, which was trained by mixing data including both clean and noisy (white, pink, babble, and car) speech at various SNR levels (15dB, 10dB and 5dB) [9]. The forth system (S4) was our proposed model, where input speech environment was identified and the corresponding acoustic model was chosen for recognition. In the S4 system, an acoustic model for each environment was trained by mixing data including each noise at three SNR levels (15dB, 10dB and 5dB). The PCA and NN-based system (NN_2-step), which achieved the best result as shown in the previous section, was used in the S4 system. The last system (S5) is an ideal system, where perfect noise classification, i.e. 0% noise classification error, is used.

Evaluated by the test set described in the Sect. 4.1.2, comparative results are shown in Table 3. It is noted that the average values presented in the Table 3 are the means of average values of each environment type. According to the Table 3, it is obvious that our proposed model achieved the best recognition results in every case and the results are almost equal to the ideal case where noise is perfectly classified. It is noted that signals of the clean and car-noise environments at 15 dB are very similar and hence most of classification errors are made by these two environments.

However, our acoustic models are robust enough to handle this case.

Table 3: Comparative results of robust speech recognition.

Environments		Word accuracy (%)				
		S1	S2	S3	S4	S5
Clean		85.75	83.58	83.86	85.77	85.75
white	15 dB	51.00	60.44	61.67	72.52	72.52
	10 dB	26.59	42.30	46.59	67.44	67.44
	5 dB	5.12	19.36	33.70	57.80	57.80
	0 dB	0.44	1.16	21.92	40.86	40.86
pink	15 dB	74.72	70.09	71.59	78.02	78.03
	10 dB	58.02	56.28	59.55	74.38	74.38
	5 dB	27.84	29.50	45.69	65.88	65.88
	0 dB	2.38	2.47	28.73	46.50	46.50
babble	15 dB	71.42	73.34	70.31	79.58	79.58
	10 dB	38.28	58.92	58.50	75.63	75.63
	5 dB	9.06	24.66	41.61	67.66	67.67
	0 dB	2.12	3.42	21.09	43.58	43.61
car	15 dB	85.17	82.83	83.23	84.92	84.98
	10 dB	84.33	81.80	82.00	84.78	84.83
	5 dB	80.47	77.92	79.30	84.56	84.56
	0 dB	66.16	63.77	74.83	82.84	82.91
Average		51.31	54.13	60.79	**72.50**	72.51

4.4. Discussion

The major advantage of our proposed model is that the PCA can be applied to a simple speech feature such as NLS in order to reduce the analyzed feature vector without a drawback of recognition accuracy reduction. Our experiments were performed on an Intel IV, 1.5 GHz with 256 MB of RAM and using Windows operating system and all the routines were implemented in C. For the multi-condition recognizer (without noise classification module), the average running time for processing an utterance is 1.92 sec. The average processing times for noise classification with HMM, NN 2-step and SVM 2-step are 2.48, 2.25 and 2.27 sec respectively. These confirm that extending the PCA-based noise classification to two-step classification still maintains an acceptable processing time.

However, the size of PCA weight vector as well as the number of multiple steps for classification may be increased, if more classes of noise are considered. Extending our idea to the classification of more noise types is feasible with a careful design of these parameters.

It is necessary to note that by the 1.2% classification errors produced by the NN_2-step as shown in the Table 2, speech recognition accuracy achieves 72.5% as described in the Table 3 for the S4 system. In the case of HMM_MFCC model, where a 4.18% error rate is obtained, the speech recognizer produces 72.1% word-accuracy, which is almost equal to the case of NN_2-step. We believe that more effects of noise classification performance can be underlined when the size of test data as well as the number of noise classes is increased. This will be one of our main future works.

5. Conclusion and Future Works

This paper proposed a technique of robust speech recognition based on model selection. The recognizer selected a specific acoustic model from a pool of acoustic models that were trained by speech data in each noisy environment. A noise classification module was used to identify the type of environment. PCA applied to the NLS and the strategic design of two-step classification was proposed for noise classification. Experiments showed that the proposed model gave a promising result for noise. When combining the model to the speech recognizer, the proposed system produced almost equal recognition accuracy to the ideal system, where noise classification contained no error. The proposed system achieved 18.4% higher recognition accuracy over the robust system using zero-mean static coefficients, and 11.7% higher accuracy over the system using the multi-condition acoustic model.

Future works include increasing the types of noise, and improving the model so that it can handle new noise not previously trained by the system. Another task is to reduce the number of specific acoustic models by clustering noise and constructing one acoustic model for each noise cluster. Finally, a multi-step classification model is expected to be automatically and optimally constructed given a large data set of noisy speech.

6. References

[1] Turk, M. and Pentland, A., "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, Vol.3, No.1, 1991. pp. 71-86

[2] Kasuriya, S., Sornlertlamvanich, V., Cotsomrong, P., Jitsuhiro, T., Kikui G. and Sagisaka, Y., "Thai speech database for speech recognition", Proceedings of Oriental COCOSDA 2003, October 2003. pp 105-111

[3] NOISEX-92. http://www.speech.cs.cmu.edu/ comp. speech/ Section1/ Data/noisex.html

[4] SNNS – Stuttgart Neural Network Simulator. http://-ra.informatik.uni-tuebingen .de/SNNS/

[5] LIBSVM – A library for Support Vector Machines. http://csie.ntu.edu.tw/~cjlin/libsvm/

[6] The HTK book version 3.1, Cambridge University , December 2001, http://htk.eng.cam.ac.uk

[7] Gales, M. J. F., "Model-based techniques for noise robust speech recognition", PhD thesis University of Cambridge, September 1995

[8] Kasuriya, S., Kanokphara, S., Thatphithakkul, N., Cotsomrong, P. and Sunpethniyom, T., "Context-independent acoustic models for Thai speech recognition", Proceedings of ISCIT2004, 2004 pp.991-994

[9] Nakamura, S., Yamamoto, K., Takeda, K., Kuroiwa, S., Kitaoka, N., Yamada, T., Mizumachi, M., Nishiura, T., Fujimoto, M., Saso, A., Endo, T., "Data collection and evaluation of AURORA-2 JAPANESE corpus", Proceedings of ASRU2003, 2003. pp.619-623.

[10] Ma, L., Smith, D., and Milner, B., "Context awareness using environmental noise classification", Proceedings of Eurospeech2003, 2003. pp. 2237-2240.

[11] Maleh, K. E., Samouelian, A. and Kabal, P., "Frame-level noise classification in mobile environments", IEEE conf. Acoustics, Speech, Signal Processing, March 1999, pp 237-240.

[12] Shao, C. and Bouchard, M., "Efficient classification of noisy speech using neural networks", Proceedings of ISSPA2003, 2003. pp. 357-360.

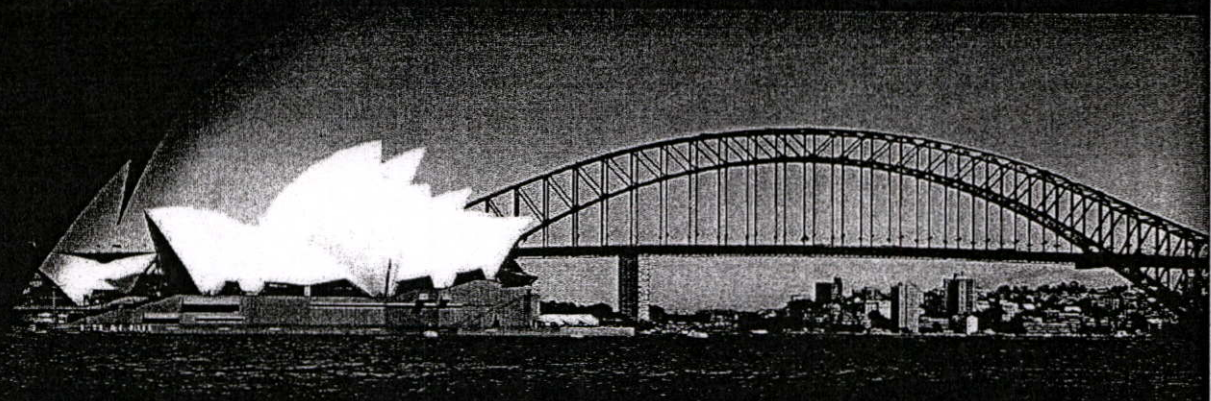
[13] Gang, Y., "Speech recognition in noisy environments: A survey", Speech Communication, Vol. 16, 1995. pp. 261-291

ISCIT 2007

2007 International Symposium on Communications and Information Technologies

Program & Abstracts

16 – 19 October 2007
Crowne Plaza Hotel,
Darling Harbour, Sydney, Australia



Patrons:

freescale semiconductor · CSIRO · NICTA · University of Wollongong

Technical Sponsorship:

IEEE

Endorsed by:

ECTI Association · EIC

In Co-operation with:

M2NM 2007

Tree-structured model selection and simulated-data adaptation for environmental and speaker robust speech recognition

Nattanun Thatphithakkul[1], Boontee Kruatrachue[1], Chai Wutiwiwatchai[2], Sanparith Marukatat[2], and Vataya Boonpiam[2]

[1]Department of Computer Engineering
King Mongkut's Institute of Technology Ladkrabang, Bangkok, 10520, Thailand
[2]Human Language Technology Laboratory
National Electronics and Computer Technology Center, Pathumthani, 12120, Thailand
s6060008@kmitl.ac.th, kkboontee@kmitl.ac.th, chai@nectec.or.th, sanparith.marukatat@nectec.or.th, vataya.boonpiam@nectec.or.th

Abstract—**This paper proposes the use of tree-structured model selection and simulated-data in maximum likelihood linear regression (MLLR) adaptation for environment and speaker robust speech recognition. The objective of this work is to solve major problems in robust speech recognition system, namely unknown speaker and unknown environmental noise. The proposed solution is composed of two components. The first one is based on a tree-structured model for selecting a speaker-dependent model that best matches to the input speech. The second component uses simulated-data to adapt the selected acoustic model to fit with the unknown noise. The proposed technique can thus alleviate both problems simultaneously. Experimental results show that the proposed system achieves a higher recognition rate than the system using only the input speech in adaptation and the system using a multi-conditioned acoustic model.**

I. Introduction

Acoustic variation in speech recognition system can be caused by several factors e.g. speaker, environmental noise, language dialect, channel, etc. [1]. In this paper, we are interested in two main causes, namely speaker variation and environment noise variation. The basic approach to deal with these problems is to train acoustic model from noise-added speech from various speakers [2]. More elaborate techniques include model adaptation with either maximum likelihood linear regression (MLLR) [3] or maximum a posteriori (MAP) [4], parallel model combination [1], piecewise linear transformation (PLT) [5].

The major problem for every robust speech recognition system is how to handle unknown environments. Two complementary techniques, tree-structured model selection and online adaptation, can be used to tackle this problem. Tree-structured model selection consists in constructing a tree in which each node represents a combination of some known environments. An acoustic model is built for each node. Using this tree structure, an unknown environment which is similar to a combination of known environments can be better handled. This approach has been applied to select a noise-specific acoustic model [5].

The online adaptation aims at adapting the available acoustic model to the current environment. An input speech is first phone labeled given an original acoustic model. The input speech with phone labels is then used to adapt the original acoustic model and the model after adaptation is exploited in the final recognition step. Both MAP and MLLR can be used in the adaptation process. However, this technique requires a large-enough set of adaptation data in order to achieve a good recognition result. Recently Thatphithakkul et al. [6] has proposed the simulated-data adaptation process which increases the size of adaptation data by combining pre-recorded clean speeches with noise portion extracted from the current input signal. This technique allows a high gain of online-adaptation performance.

It is noted that in previous works, speaker and noise variations have often been treated separately. The objective of this work is to explore how these two variations can be handled simultaneously and efficiently. The adopted solution is based on similar idea of tree-structured model selection but used for speaker modeling instead of noise modeling propose in [5]. The speaker tree determines a speaker-dependent acoustic model which best matches to the current input signal. This tree-structured model selection can handle unknown speakers. Then we apply the simulated-data adaptation, which can solve the problem of unknown environment noise.

The proposed system is evaluated by noisy speech in 3 sets of environment. The first set contained speech in a clean environment and 9 types of noisy environments that have been trained in the system. The second set contains speech in other 2 types of noisy environments not trained in the system. Noisy speech is prepared from noise signals taken from JEIDA [7] (Japan Electronic Industry Development Association), NOISEX-92 [8] and a real noise signal collected in an exhibition in Thailand. Noise signals are added to clean speech taken from NECTEC-ATR Thai speech

1-4244-0977-2/07/$25.00 © 2007 IEEE

corpus at various SNRs (0, 10, 15 dB). The third set contains speech signals recorded in a real environment of another exhibition in Thailand 2005. The estimated SNR of the last set is 0-5 dB.

. The next section explains our proposed model. Section III describes data sets used in experiments. Experimental results are reported in Section IV. Section V concludes this paper and discusses on the future work.

II. TREE-STRUCTURED MODEL SELECTION AND SIMULATED-DATA ADAPTATION

Our proposed method of using tree-structured model selection [5] and simulated-data adaptation [6] is illustrates in Fig 1. Two principal components in this method are:

- Model selection from speaker clusters (MSSC), which functions to select the closest speaker-dependent acoustic model from a tree-structured speaker model.
- Model adaptation using simulated-data adaptation.

Section II.A describes tree-structured HMM for speaker clustering method process. Section II.B describes simulated-data adaptation process.

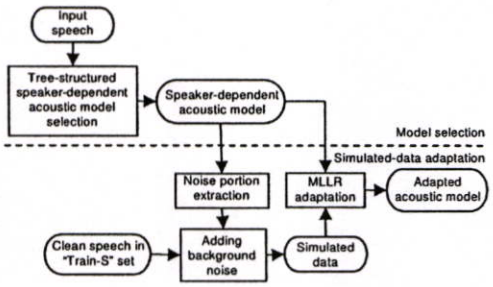


Fig. 1. Acoustic model selection using tree-structured speaker model and acoustic model adaptation using simulated-data.

A. *Tree-structured speaker modeling.*

The tree-structured clustering method has been successfully applied for speaker adaptation [5]. In this paper, we apply the tree-structured clustering method for speaker-dependent acoustic model selection. The tree structure used in our system contains two acoustic models in each node. The first one is used in for speaker-dependent acoustic model selection, so called a classification acoustic model. Once a node is selected, the other acoustic model, called a recognition acoustic model, is used for speech recognition.

The tree-structured speaker modeling method is illustrates in Fig 2. Speeches from various speakers in various environments are collected and classification acoustic model constructed using all data. Top-down clustering is applied on the obtained acoustic model to produce the tree structure. We retrain a classification acoustic model for each cluster. Finally, the root node of the tree includes all speakers and each leaf node consists of only one speaker. Intermediate nodes in this tree contain several speakers whose speech characteristics are similar. Model selection is performed based on these speaker-dependent classification acoustic models. For the recognition acoustic model, a phoneme-based acoustic model is constructed for each noise using noise-added speech from the particular speakers. In this paper, HMM is applied to both the classification and recognition acoustic models. Using this tree structure, an unknown speaker whose sound is similar to the combination of speech from known speakers can be effectively handled.

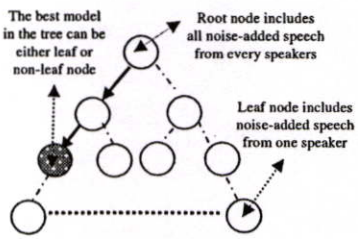


Fig. 2. Tree-structured noise speakers modeling for speaker-dependent acoustic model selection.

B. *Simulated-data adaptation*

Since the model selection step only chooses a recognition acoustic model that best matches to the input speaker, the obtained acoustic model is yet general for every environmental noise. To enhance the system performance, online-adaptation can be conducted to make the model closet to the input environment. While the conventional online-adaptation process employs only the input signal in adaptation, the simulated-data adaptation method extends the adaptation set by adding noise extracted from the input signal to an existing set of clean speech. In this work, MLLR is conducted for adaptation. We denote S-MLLR our process of MLLR-based simulated-data adaptation. Section II.A.a describes noise extraction process. The following subsections describe in brief the simulated-data adaptation process

(a) Noise portion extraction

The silence parts are supposed to be background noise of the current input signal. An HMM is first applied to segment the input signal into speech and silence portions. The noise extraction algorithm utilizes phone-based HMMs [6].

(b) Adding background noise

Given noise portions extracted from the input signal, several issues need to be considered in adding background noise to the pre-recorded clean speech. First we concatenate noise portions extracted from the input signal. There are two noise-only regions in the input signal, at the beginning and at the end of the signal. These noise portions are duplicated and concatenated so that the duration of noise signal is equal to the duration of clean-speech being added.

Second, simulated speech for adaptation should have a similar SNR to the input speech. Let "Train-S" be a set of pre-recorded clean speech. We denote by Tn and Ts the current input signal and a clean speech in the Train-S set. Tn_s and Ts_s is the speech portion of Tn and Ts. First, a scale_factor is calculated as follows:

$$scale\_factor = EngC/EngS \quad (1)$$

where EngC and EngS is the energy of Ts_s and Tn_s respectively. Next, the background noise, BN, is multiplied by the scale_factor and added to Ts, resulting a simulated noisy-speech Tsn as shown in (2).

$$Tsn = BN*scale\_factor + Ts \quad (2)$$

III. Experimental Setting

Our task domain is isolated-word recognition using monophone-based HMMs representing 75 Thai phones. Each monophone HMM consists of 5 states and 16 Gaussian mixtures per state. 39-dimensional vectors (12 MFCC, 1 log-energy, and their first and second derivatives) are used as recognition features.

The baseline recognition acoustic model is trained by phonetically-balanced clean-speech utterances read by 16-male and 16-female speakers. The total number of training utterances is 32,000. For comparison, a multi-conditioned acoustic model [2], denoted as "MULTI" hereafter, is prepared using speech data from both clean environment and noisy environments at various SNRs (5, 10, and 15 dB). In all experiments, clean-speech data are taken from NECTEC-ATR corpus.

A. *Noise data for training*

Eight kinds of noise from JEIDA [7], including crowded street, machinery factory, railway station, large air-condition, trunk road, elevator, exhibition in a booth, and ordinary train, and one large-size car noise from NOISEX-92 [8] are conducted. All noises from JEIDA and NOISEX-92 as well as the clean speech from NECTEC-ATR are preprocessed by reducing the sampling rate to 8 kHz. Noisy speech is prepared by adding the noise from JEIDA or NOISEX-92 to the clean speech of NECTEC-ATR at various SNRs (5, 10 and 15 dB).

B. *Noise data for testing*

Two test sets, "Test-1" and "Test-2", are used in evaluation. Test-1 contains 3,200 utterances from 640 words uttered by 5 male speakers. Two noises, a computer room from JEIDA and an exhibition (NSTDA Annual Conference S&T in Thailand) recorded over four days in March 2005, are added to clean-speech utterances at three SNR levels: 0, 10 and 15 dB. This test set represents speech with different noise from the training set.

Test-2 contains 760 utterances from 76 words uttered by 50 speakers over four days in another exhibition (ICT EXPO 2005 in Thailand). The environment is very noisy and consists of various kinds of noise. This set represents real noisy-speech with SNR ranged between 0 to 5 dB.

IV. Experimental Results

This section presents experimental results obtained from the proposed system including an evaluation of the MSSC technique with and without simulated-data adaptation. We also evaluate the MLLR adaptation process when the adaptation data include and exclude the input speech signal. The construction of Train-S with speakers selected by the MSSC technique is also considered. Section IV.A presents the evaluation of MSSC. The construction of Train-S is presented in Section IV.B. Section IV.C then compares our proposed system to conventional methods.

A. *Evaluation of the MSSC model*

First we evaluate the MSSC model selection technique for robust speech recognition by comparing to the baseline and the MULTI systems. Recognition accuracies obtained by test sets in three SNRs are shown in Fig 3. As expected, the MSSC, using tree-structured speaker modeling determines an acoustic model that best matches to the input speaker. Therefore, the selected model produces a higher recognition rate than the baseline and MULTI models.

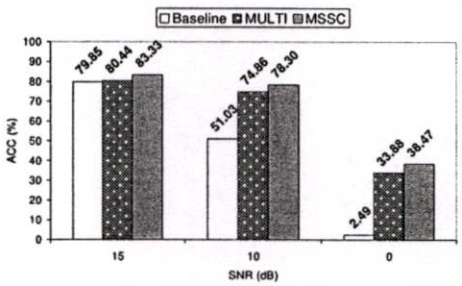


Fig. 3. Recognition accuracies of Baseline, MULTI, and MSSC evaluated by Test-1.

B. *Effects of different configurations in simulated-data adaptation*

In this subsection, we are interested in the efficient way to construct the Train-S set for simulated-data adaptation. Indeed, there are two intermediate choices in building the Train-S:

- Whether the input signal should be added into the Train-S set. Input signal actually contains useful information about the acoustic characteristic of the current speaker. However, adaptation can only rely on a transcription obtained automatically. Thus the uncertainty about transcription error is also presented if we include the input speech in the Train-S set. We would like to investigate if the adaptation can still take advantage of input signal with this uncertainty. Hereafter, "supervised" adaptation refers to the adaptation

without input speech and "semi-supervised" adaptation refers to the adaptation with input speech in the Train-S set.

- Another factor which affects adaptation performance is speakers in the Train-S set. Since the acoustic model selected by the MSSC method implies the closest speaker to the input speech, the model should be adapted with speech from speakers close to the selected model. However in noisy environments, especially in 3 low SNR, it is not always possible to correctly select speakers for the Train-S set. Therefore, we ask the question; should we adapt the selected model with all available data (limited to male-speakers 16 people, denoted as MIX) or should we use only the speech from the selected speaker cluster (denoted as "SELECT").

In this experiment, we consider both the MULTI model and the model selected by the MSSC technique. The MULTI model adapted with simulated-data in the supervised scheme will be denoted as S-MLLR1 and the model using semi-supervised adaptation will be denoted as S-MLLR2. In analogous manner, S-MSSC1 and S-MSSC2 denote the MSSC-based selected model adapted in the supervised and semi-supervised schemes using simulated-data.

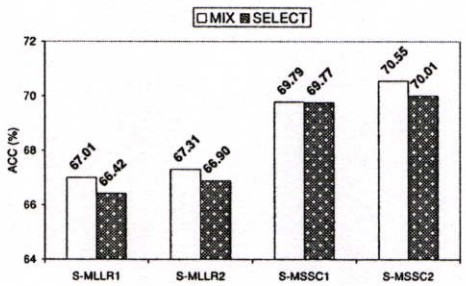


Fig. 4. Average recognition accuracies on Test-1 over all SNRs, produced by S-MLLR and S-MSSC in supervised (labeled as 1) and semi-supervised (labeled as 2) schemes, with different speakers prepared in the Train-S set.

Average recognition rates for every system over all SNRs (0, 10, and 15) are shown in Fig 4. Firstly, S-MSSC outperforms S-MLLR in every case. Secondly, the semi-supervised adaptation gives higher recognition rate than the supervised one. This reflects the fact that including the input speech in adaptation is preferable. Even if the transcription of input speech might be wrong due to the automatic transcription process, the speaker characteristic in the input speech is still useful for speaker-dependent model adaptation. Lastly, MIX speakers give better performance than using only speakers in the identified speaker cluster. We believe this is due to the fact that in noisy environment, the speaker cluster selection may not be done accurately. Thus the acoustic characteristics of speakers in SELECT may be different from current speaker. On the other hand, MIX data containing all speakers should also contain the speaker close to current one. As consequence, adapting the acoustic model with SELECT only give better performance than MIX in high SNR. On average, however, using SELECT data gives lower recognition rate than using MIX. In the following, our system will function with S-MSSC using semi-supervised adaptation with MIX speakers in the Train-S set.

C. *Comparison with conventional methods*

In this subsection, several robust speech recognition techniques including our proposed model are experimentally compared.

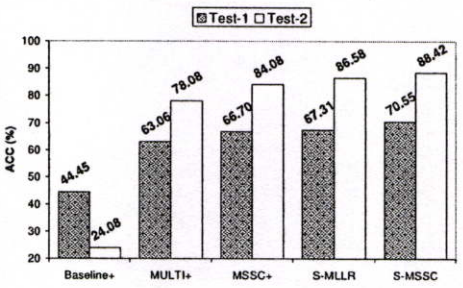


Fig. 5. Comparison of Baseline+, MULTI+, MSSC+, S-MLLR and S-MSSC evaluated by Test-1 and Test-2.

The first system, called "Baseline+", used the baseline system with online MLLR adaptation. The second system, called "MULTI+", was a multi-conditioned acoustic model with online MLLR adaptation. The third system, called "MSSC+", used the tree-structured speaker-dependent model selection with online MLLR adaptation. The fourth system, called "S-MLLR", applied simulated-data adaptation to the MULTI acoustic model. The fifth system "S-MSSC" adapts the MSSC-based selected model with simulated-data. The last system is the best configuration of our propose model

Figure 5 shows comparative results of five systems. According to results, it is obvious that our proposed method, S-MSSC, significantly outperform other conventional methods.

V. Conclusions

This paper proposed a robust speech recognition system that can deal with speaker and noise variations simultaneously. Unknown speakers and noise were handled by the tree-structured speaker-dependent model selection and online adaptation using simulated-data. Several configurations of the proposed system were investigated. Experiments showed that our proposed model achieved over 26% and 64% improvement of recognition accuracy on Test-1 (additive-noise speech) and Test-2 (real noisy speech), comparing to the conventional approach of online MLLR adaptation (Baseline+).

Future works include an evaluation of the proposed model by a larger set of speech from various environments. Further improvement of noise extraction and noise addition in simulated-data adaptation will be investigated. Moreover, the current tree structure is principally designed to handle the speaker variation. In the future work, we aim at constructing a

tree-structured model with fully support speaker as well as noise variation.

REFERENCES

[1] M.J.F. Gales, "Model-based techniques for noise robust speech recognition," PhD thesis University of Cambridge, 1995.

[2] S. Nakamura, K. Yamamoto, K. Takeda, S. Kuroiwa, N. Kitaoka, T. Yamada, M. Mizumachi, T. Nishiura, M. Fujimoto, A. Saso, and T. Endo, "Data collection and evaluation of AURORA-2 JAPANESE corpus", Proc. of ASRU 2003, pp.619-623, 2003.

[3] M.J.F. Gales, and P.C. Woodland, "Mean and variance adaptation within the MLLR framework", Comput. Speech Lang., 10(3):249–264, 1996.

[4] J.L. Gauvain, and C.H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains", IEEE Trans. Speech Audio Proc., 2:291–298, 1994.

[5] Z.P. Zhang, T. Sugimura, and S. Furui, "Tree-structured clustering methods for piecewise linear transformation-based noise adaptation," IEICE Trans. Inf. and Syst. vol. 9, pp. 2168-2176, 2005.

[6] N. Thatphithakkul, B. Kruatrachue, C. Wutiwiwatchai, S. Marukatat, and V. Boonpiam, "A simulated-data adaptation technique for robust speech recognition," Proc. of INTERSPEECH, pp. 777–780, 2006.

[7] http://www.milab.is.tsukuba.ac.jp/corpus/noise db.html

[8] http://www.speech.cs.cmu.edu/comp.speech/Section1/Data /noisex.html.

SIMULATED-DATA ADAPTATION BASED PIECEWISE LINEAR TRANSFORMATION FOR ROBUST SPEECH RECOGNITION

Nattanun Thatphithakkul[1], Boontee Kruatrachue[1], Chai Wutiwiwatchai[2], Sanparith Marukatat[2] and Vataya Boonpiam[2]

[1] Computer Engineering Department, Faculty of Engineering,
King Mongkut's Institute of Technology Ladkrabang, Bangkok, 10520, Thailand
S6060008@kmitl.ac.th and kkboontee@kmitl.ac.th,

[2]National Electronics and Computer Technology Center, Bangkok, 12120, Thailand
chai@nectec.or.th, sanparith.marukatat@nectec.or.th and vataya.boonpiam@nectec.or.th

---

ABSTRACT

This paper proposes an efficient method of simulated-data adaptation for robust speech recognition. The method is applied to tree-structured piecewise linear transformation (PLT). The original PLT selects an acoustic model using tree-structured HMMs and the acoustic model is adapted by input speech in an unsupervised scheme. This adaptation can degrade the acoustic model if the input speech is incorrectly transcribed during the adaptation process. Moreover, adaptation may not be effective if only the input speech is used. Our proposed method increases the size of adaptation data by adding noise portions from the input speech to a set of pre-recorded clean speech, of which correct transcriptions are known. We investigate various configurations of the proposed method. Evaluations are performed with both additive and real noisy speech. The experimental results show that the proposed system reaches higher recognition rate than MLLR, HMM-based model selection and PLT

Keywords: *Robust speech recognition, piecewise linear transformation, simulated-data adaptation.*

1. INTRODUCTION

It is commonly known that a speech recognition system trained by speech in a clean or nearly clean environment cannot achieve good performance when working in noisy environment. Research on robust speech recognition is then necessary. Gales[1] has classified the techniques of robust speech recognition into 4 approaches: 1) extraction of robust speech feature, 2) estimation of clean speech, 3) construction of robust model, and 4) combination of three previous techniques. Each of these approaches has both advantages and disadvantages. This paper focuses on the model-based approach, which has achieved good recognition results[1]. The model-based approach aims to create or to adapt the acoustic model in specific environments. Several techniques of model adaptation have been proposed such as maximum likelihood linear regression (MLLR)[2,3], maximum a posteriori (MAP) adaptation[4,5], parallel model combination (PMC) [6,7,8], and piecewise linear transformation (PLT) [9,10].

In this work, we are interested in the adaptation technique of piecewise linear transformation with tree-structured model selection[10,11], proposed by Zhang, Sugimura and Furui[10]. This technique is based on an unsupervised acoustic model adaptation using the incoming speech. It was proven to be efficient in both accuracy and computational cost compared to the PMC technique[9]. However, a problem of the PLT is that the acoustic model may not be well adapted if the incoming speech is very short as found in most of isolated-word recognition

tasks. Moreover, in the unsupervised adaptation, a wrong transcription of the input speech strongly degrades the adapted acoustic model.

Therefore, this paper presents a new adaptation scheme using simulated-data adaptation applied to piecewise linear transformation (PLT). Indeed, the simulated-data adaptation process aims to increase the data used in adaptation by adding the background noise extracted from the current input signal to existing clean speech. Since correct transcriptions of the clean speech are known, using the simulated-data is supervised adaptation. In this paper, both supervised adaptation and semi-supervised adaptation where the input speech is included in the adaptation set are investigated. Selection of a recognition result from the results provided by both the adapted and original acoustic models allows further improvement of the recognition accuracy. It should be noted that this idea of simulated-data adaptation is not limited to PLT adaptation. It can be used in other adaptation algorithms such as the general MLLR process. A comparison between normal MLLR and MLLR using simulated-data adaptation is also investigated.

The proposed system was evaluated with 3 groups of environments. The first group contained a clean environment and 9 types of noisy environments that have been trained in the system. The second group contained other 2 types of noises not trained in the system. Noisy speech was prepared by adding noise signals from JEIDA[12] , NOISEX-92[13] and an exhibition in Thailand (NAC 2005) to the clean speech taken from NECTEC-ATR Thai speech corpus[14] at various SNR (0, 10, 15 dB). The third group contained speech signal recorded in real environment of another exhibition in Thailand (ICT-EXPO 2005). The estimated SNR for this last group was 0-5 dB.
We will review the PLT algorithm in the next section, followed by an explanation of our proposed models in Section 3. Section 4 describes the data used in these works and experimental results are reported in Section 5. Section 6 concludes this paper with some future works

2. Piecewise Linear Transformation (PLT)

The PLT method[10] is composed of 2 main steps namely the tree-structure HMM construction and the MLLR adaptation for the current input signal. Figure 1 shows a flow diagram of the PLT method. In the first step, a wide variety of noise data were collected and classified into noise clusters using hierarchical clustering. The root node includes all noises and all SNR conditions and each leaf node consists of only one noise at one SNR condition. Intermediate nodes in this tree contain noises from different environments and from different SNR which are similar. A noise-added speech HMM is constructed for each node. Using this tree structure, an unknown noise environment which is similar to combination of known environments should be handled by non-leaf HMM. The resulting tree-structured HMM allows representing both known and some unknown noises. In the recognition phase, the noise-cluster HMM that best fitted the input speech was selected and further adapted to reduce mismatches with the input speech by the MLLR method. In both processes, HMM selection and adaptation using linear transformation are based on the likelihood maximization criterion.

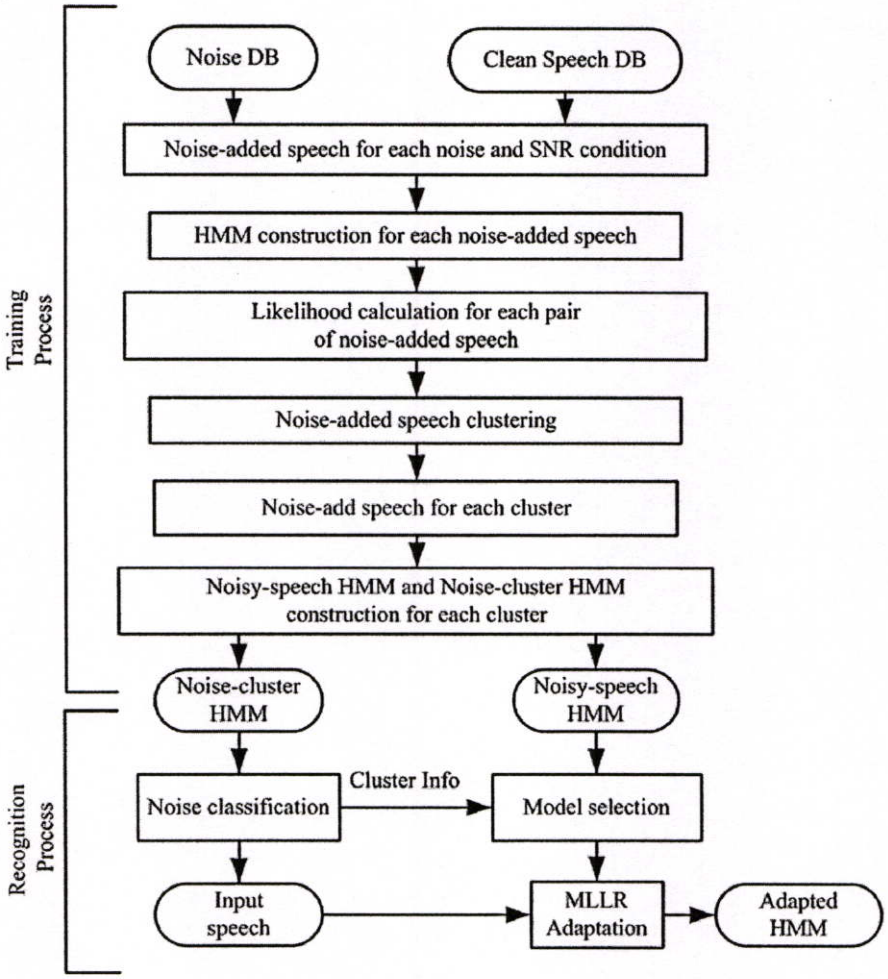


Figure 1: *Piecewise linear transformation for HMM noise adaptation.*

3. Piecewise Linear Transformation with Simulated-Data Adaptation

The piecewise linear transformation with simulated-data adaptation, called S-PLT hereafter, is similar to PLT in Figure 1 except that the recognition process is replaced by the procedure shown in Figure 2. The idea of the proposed method is to increase the number of adaptation data using a set of pre-recorded clean speech, of which exact transcriptions are known. A set of adaptation data is simulated by adding noise portions extracted from the input speech to a set of clean speech. Using the existing speech with known transcriptions can be called supervised adaptation. The next subsection explains in details the whole process.

There are variations of the proposed method. The first issue is whether the input speech is included in the adaptation set. An input speech with poor quality may degrade the efficiency of adapted acoustic model. Therefore, we evaluate both systems excluding (supervised adaptation) and including the input speech (semi-supervised adaptation). This issue is explained in Section 3.2. Furthermore, due to the variation of speech quality and noisy environment, unsupervised adaptation does not always improve the model quality. In some cases, the original acoustic model is even better than the adapted model. Thus, we also consider the selection between the result provided by the adapted model and that of the unadapted model. Section 3.3 describes this selection procedure.

It should be noted that the MLLR is used in all adaptation process, not only during the clustering but also in the recognition process. This allows sharing parameters among all distributions in the system, hence reducing the size of the overall system. Consequently, this also reduces the required size of data to be used in the adaptation process.

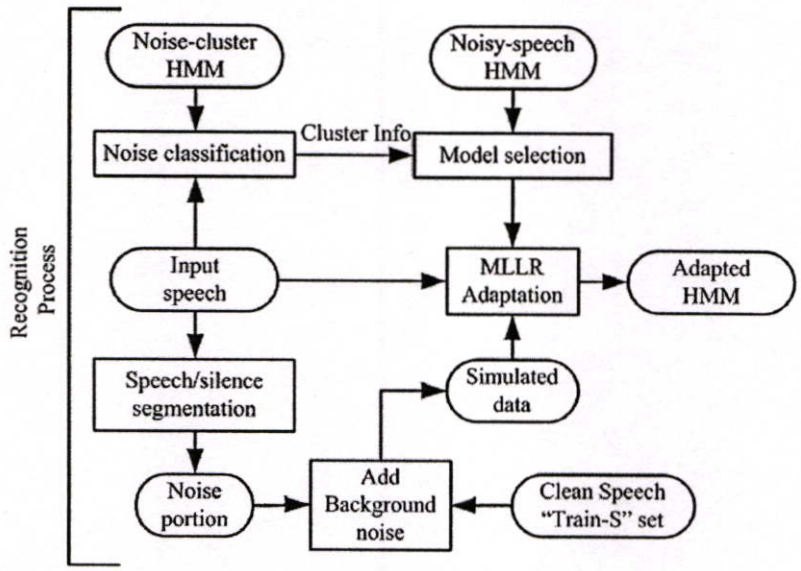


Figure 2: Recognition process in piecewise linear transformation with simulated-data adaptation for HMM noise adaptation.

3.1 Simulated-data adaptation

3.1.1 Speech/silence segmentation

Simulated-data adaptation begins with identifying silence parts in the input signal. The silence parts are supposed to be background noise of the current input signal. For our task of isolated-word recognition, we assume that there are short periods of silence at the beginning and the end of the input signal. A hidden Markov model (HMM) is used to segment the input signal into speech and silence portions. Two HMM architectures used for noise extraction. The first algorithm utilizes phone-based HMMs, where 64 HMMs of Thai phonemes including a special phoneme of silence "sil", as shown in Table 1, form an isolated-word recognizer. Figure 3(a) illustrates this HMM structure. The second noise extraction algorithm is based on speech/non-speech detection. Two states HMM, symbolized with speech and silence, are included in the module as shown in Figure 3(b). In both algorithms, noise portions are the signal regions labeled with silence "sil".

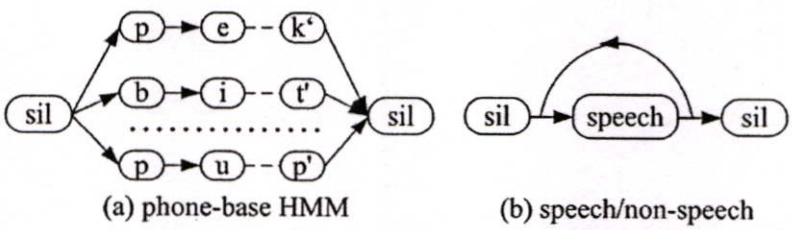


Figure 3: Two HMM architectures used for noise extraction.

Table 1: 64 Thai phonemes.

Type	IPA symbol
Initial consonant	p, t, c, k, ʔ, p^h, t^h, c^h, k^h, h, b, d, m, n, ŋ, l, r, f, s, h, w, j, pr, pl, p^hr, p^hl, tr, t^hr, kr, kl, kw, k^hr, k^hl, k^hw, fr
Vowel	i, i:, ɨ, ɨ:, u, u:, e, e:, ɜ, ɜ:, o, o:, æ, æ:, a, a:, ɔ:, i:a, ɨ:a, u:a
Final consonant	p', t', k', m', n', ŋ', s', w', j'
Silence	sil

In both algorithms, HMMs are composed of 16 Gaussian mixtures per state and were trained by the Baum–Welch algorithm. It is noted that the former algorithm gives better noise-region labeling performance with a drawback of computational demand comparing to the latter algorithm.

3.1.2 Adding background noise

Given noise portions extracted from the input signal, several issues need to be considered in adding background noise to the pre-recorded clean speech. First we concatenate noise portions extracted from the input signal. There are two noise-only regions in the input signal, at the beginning and at the end of the signal as shown in Figure 4. These noise portions are duplicated and concatenated so that the duration of noise signal is equal to the duration of clean-speech being added. It is noted that simply concatenating noise portions causes an unusual spectral change. However, in this paper, we discard spectral smoothing in order to save processing time.

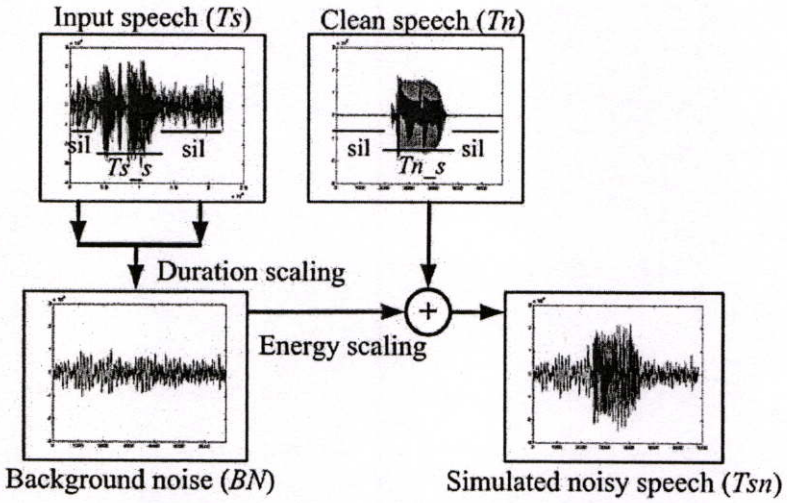


Figure 4: Adding background noise.

Second, simulated speech for adaptation should have a similar SNR to the input speech. However, estimation of SNR is not trivial and remains unsolved. In this work, we propose a simple way of signal-energy scaling. Let "Train-S" be a set of pre-recorded clean speech, of which correct transcriptions are known. We denote by *Tn* and *Ts* the current input signal and a clean speech in the Train-S set. Tn_s_i *(i=1,....,Ln)* and Ts_s_i *(i=1,....,Ls)* is the speech

portion of *Tn* and *Ts* and *Tn_sil* and *Ts_sil* is the silence portion of *Tn* and *Ts*. First, a *scale_factor* is calculated as follows:

$$EngC = \frac{\sum_{i=1}^{Ls} |Ts\_s_i|}{Ls} \quad (1)$$

$$EngS = \frac{\sum_{i=1}^{Ln} |Tn\_s_i|}{Ln} \quad (2)$$

$$scale\_factor = \frac{EngC}{EngS} \quad (3)$$

where *EngC* and *EngS* is the energy of *Ts_s* and *Tn_s* respectively. Next, the background noise, *BN*, is multiplied by the *scale_factor* and added to *Tn*, resulting a simulated noisy-speech *Tsn*.

$$Tsn = BN * scale\_factor + Tn \quad (4)$$

This signal *Tsn* is included in the adaptation set.

3.2 Supervised and semi-supervised adaptation

In the model adaptation process, the MLLR algorithm is applied to the selected HMM with data obtained from the input speech and/or the simulated-data. In order to use the input speech for adaptation, we need its phoneme label. However, the true label of the current input speech in unknown. Therefore, we relied on the label transcribed in the first pass by a selected HMM. The use of input speech with a label automatically transcribed is called an *unsupervised* adaptation process. In contrast to the use of simulated-data, the noise portion extracted in the process of speech/silence segmentation is added to pre-recorded clean-speech signals *Ts*, whose phoneme labels are known. Hence, adaptation using simulated-data is *supervised*. An adaptation process that uses both the input speech and a set of simulated-data is denoted as *semi-supervised* adaptation.

3.3 Recognition result selection

Theoretically, an adapted acoustic model should enhance the recognition accuracy. However, this assertion does not always hold in practice, due to many factors such as the recording condition and the speaking style. In the case where the quality of adaptation data is low, the recognition result given by the un-adapted HMM is useful.

In this work, we investigate 2 recognition schemes: one-step and two-step methods. The former method always relies on the result provided by the adapted model. The latter method chooses the final result from either the result of the un-adapted HMM or that of the adapted HMM. Indeed, the result whose likelihood, compared to the corresponding model, is higher is selected as final result. Figure 5 shows these 2 selection methods used in this work.

Table 2: Four configurations of the proposed S-PLT system which will be investigated in this paper.

Method	Description
Method1-1	One-Step method with supervised MLLR
Method1-2	One-Step method with semi-supervised MLLR
Method2-1	Two-Step method with supervised MLLR
Method2-2	Two-Step method with semi-supervised MLLR

Table 2 summarizes the different settings of the proposed S-PLT system investigated in this work. Certainly, other parameters like the number of words in Train-S or the number of clusters in MLLR adaptation also has influence on the accuracy of the system. These parameters will also be investigated experimentally in Section 5.

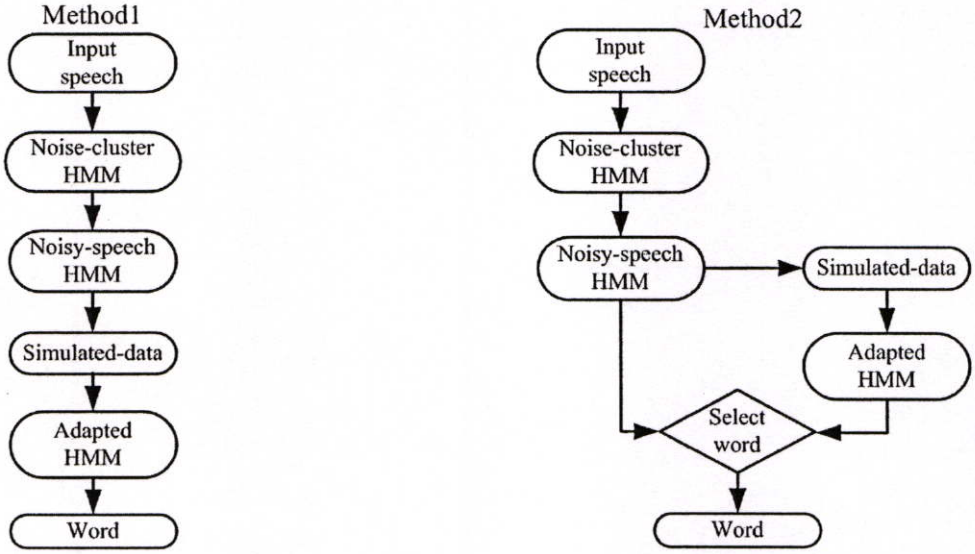


***Figure 5:** The two selection methods used in the S-PLT.*

3.4 Using simulated-data adaptation in the normal MLLR process

It is noted that the idea of simulated-data adaptation is to increase the number of adaptation data by adding the noise portion extracted from an incoming input speech to a set of pre-recorded clean speech. Therefore, it can be used in any adaptation algorithm such as the general MLLR process, where an original acoustic model is adapted with an incoming input speech. MLLR with simulated-data adaptation is named shortly as S-MLLR. S-MLLR can be performed in 4 variations, S-MLLR1-1, S-MLLR1-2, S-MLLR2-1, and S-MLLR2-2, similar to S-PLT described in the previous section.

4. Experimental Setting

4.1 Task

This work concerns the isolated-word recognition. Our system is a phoneme-based with HMMs representing 64 Thai phones. These allow cover up all phones from the NECTEC-ATR corpus[14] which is used throughout this work. Each monophone HMM consisted of 5

states and 16 Gaussian mixtures per state. 39 dimensional vectors (12 MFCC, 1 log-energy, and their first and second derivatives) were used as recognition features.

The baseline acoustic model was clean monophone HMM. It was trained by using phonetically-balanced read by 16 male and 16 female speakers. The total number of training utterances was 32,000.

In all experiments, the clean speech files were taken from the NECTEC-ATR.

4.2 Noise data for training

Eight kinds of noise from JEIDA (Japan Electronic Industry Development Association)[12], including crowded street, machinery factory, railway station, large air-condition, trunk road, elevator, exhibition in a booth, and ordinary train, 1 large-size car noise from NOISEX-92[13] were used for noise clustering. All noises from JEIDA and NOISEX-92 as well as the clean speech from NECTEC-ATR were preprocessed by reducing the sampling rate to 8 kHz. Noisy speech was prepared by adding the noise from JEIDA or NOISEX-92 to the clean speech of NECTEC-ATR at various SNR (0, 10 and 15 dB). A noise HMM with 16 mixtures was trained for each noise by the Baum–Welch algorithm.

4.3 Noise data for testing

Two test sets, Test-1 and Test-2, were used to evaluate the proposed method.

"Test-1" contained 3200 words uttered by 5 male speakers. Two noises, "computer room" (Noise1) from JEIDA and "exhibition" (Noise2) recorded over four days in March of 2005 at exhibition of NSTDA Annual Conference S&T in Thailand 2005, were digitally added to the utterances at three SNR levels: 0, 10 and 15 dB. These are new noises which differ from the 9 noise samples used to train the system.

"Test-2" contained 76 words utterances from 50 speakers collected over four days in August of 2005 in the actual environment of "exhibition" at ICTEXPO 2005 in Thailand. The noise power was estimated using the noise periods immediately before and after each sentence utterance. The power of noise-added speech was estimated as the mean value averaged over the utterance period. Based on these values, the estimated SNR was 5-0 dB. This task was difficult, since the noise was non-stationary.

4.4 Data for supervised adaptation

In order to constitute the Train-S set for model adaptation, several criteria are used to select speakers and lexical words from the NECTEC-ATR corpus. Speakers used in Train-S are selected from speakers in the training set, not from any test set.

For speaker selection, we limited to male speakers with clear speech. For this criterion, four speakers "M1", "M2", "M3", and "M4" were selected. These speakers were also used to test the effect of the number of speakers on the adaptation process. For this test we denote by "MIX" the set of data containing all of these 4 speakers.

Next, we choose the set of words to be used in Train-S. Two criteria were considered. First, these words should be correctly recognized by clean model and second these words should cover all 64 phones present in the system. According to these criteria, 22 words were selected for model adaptation.

Moreover, in order to investigate the effect of phones in Train-S on the adaptation procedure, we 2 additional subsets were constructed from these 22 words. The first subset contained 8 words which cover 41 phones and the second subset contained 16 words with 56 phones.

5. Experimental Results

5.1 Comparison of four methods of simulated-data adaptation

The first experiment aims at comparing four methods using simulated-data adaptation as described in Table 2. The four methods can be applied to both S-MLLR and S-PLT models. In this experiment, the set of pre-recorded clean speech Train-S contained speeches from one speaker (M1) and covers 22 distinct words. The number of clusters used in the MLLR process was set to 1.

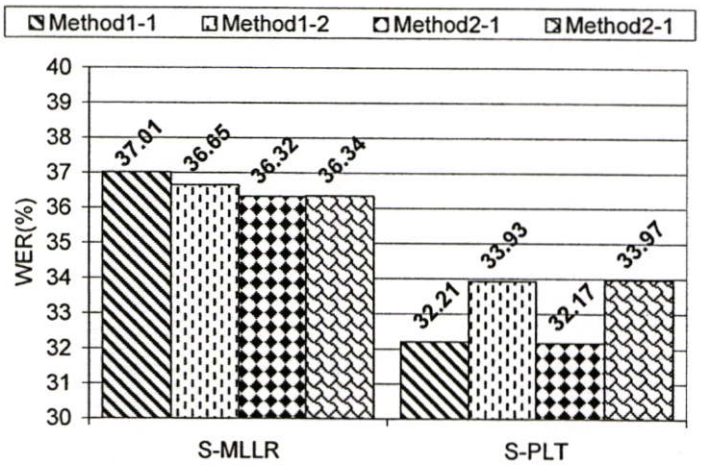


Figure 6: Comparison of S-MLLR and S-PLT on Test-1 data (Noise 1 noise-added speech, SNR: 15 dB, 10 dB and 0 dB)

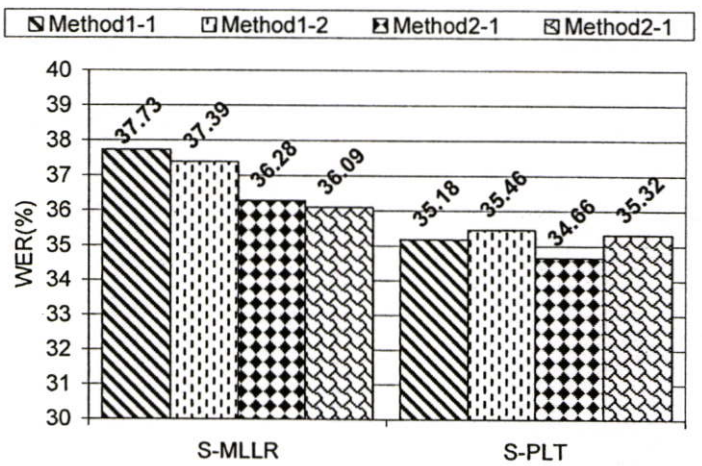


Figure7: Comparison of S-MLLR and S-PLT on Test-1 data (Noise 2 noise-added speech, SNR: 15 dB, 10 dB and 0 dB).

Figure 6 and 7show word error rate (WER) results of Test-1 for 2 types of noises, Noise1 and Noise2. The results consistently present the better performance of S-PLT over the simple S-MLLR. For S-PLT, the best configuration is the "Method2-1", where the adaptation set excludes the input speech and the recognition result is selected between results of the original

and adapted acoustic models. The most important reason that the adaptation set should exclude the input speech is that when the quality of input speech is poor, the original acoustic model will produce wrong transcription of the input speech. Using the wrong transcription in the adaptation process yields a distorted acoustic model. Though the input speech has a low quality, the noise-only portion in the speech signal is a good source for simulating adaptation data from pre-recorded clean speeches in which we know their correct transcriptions.

5.2 Experiments on the number of MLLR clusters and the size of adaptation data

In this subsection, the number of MLLR clusters and the size of adaptation data were optimized for the S-PLT model. Train-S was the same as that used in the previous subsection, but the number of MLLR clusters varied between 1, 8, and 16, while the number of words covered was set to 8, 16, and 22, spanning over 41, 56, and 64 phonemes respectively.

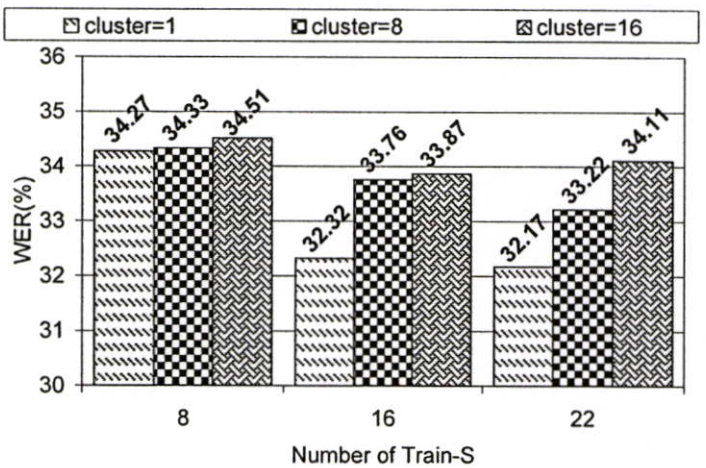


Figure 8: *Recognition results using S-PLT by "Method2-1" on Test-1 data. (Noise 1 noise-added speech, SNR: 15 dB, 10 dB and 0 dB).*

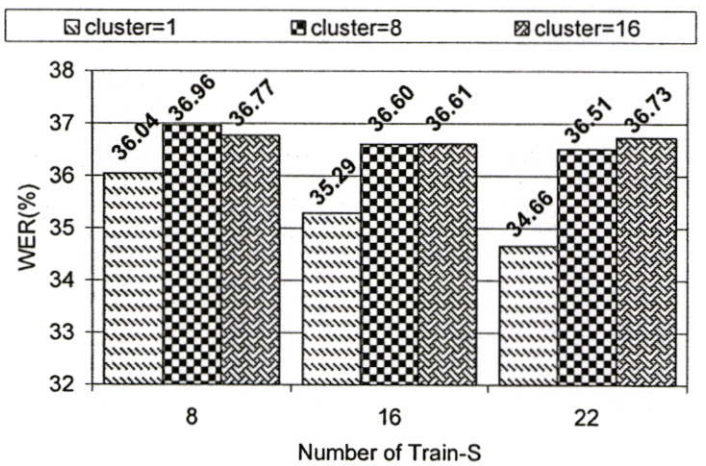


Figure 9: *Recognition results using S-PLT by "Method2-1" on Test-1 data (Noise 2 noise-added speech, SNR: 15 dB, 10 dB and 0 dB).*

Figure 8 and 9 show evaluation results of Test-1 for the noises Noise1 and Noise2. Both figures obviously show that one MLLR cluster achieves the best performance in every case of the size of Train-S and the type of noises. The more the number of words covered in Train-S

is used, the lower WER is achieved. However, we limit the largest number of words in Train-S to 22, which covers all phonemes appearing in the evaluation task. A larger set of Train-S may produce a higher accuracy, but increases computational time.

5.3 Experiments on different speakers used in simulated-data adaptation

In the case that speeches from only one speaker are included in the simulated adaptation data, increasing the size of adaptation data tends to produce a speaker-dependent acoustic model. Using the speaker-specific acoustic model may reduce the system accuracy when evaluating with speeches from various speakers. Therefore, in this subsection, five experiments on S-PLT were performed to explore this phenomenon. Each of the first four experiments uses speeches of only one speaker (M1 to M4). Randomly mixed speeches from M1 to M4 speakers were used in the last experiment, denoted as MIX. The number of MLLR cluster was set to 1.

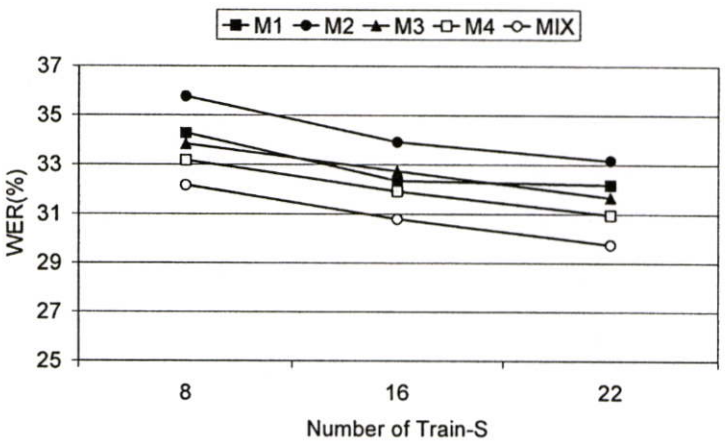


Figure 10: *Recognition results using S-PLT by "Method2-1" on Test-1 data (Noise 1 noise-added speech, SNR: 15 dB, 10 dB and 0 dB).*

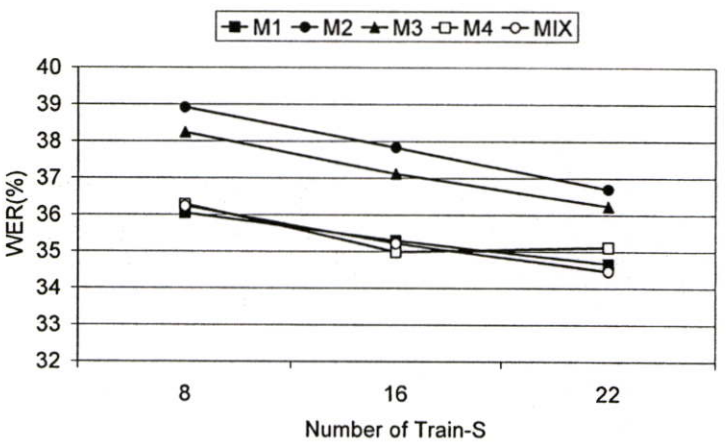


Figure 11: *Recognition results using S-PLT by "Method2-1" on Test-1 data (Noise 2 noise-added speech, SNR: 15 dB, 10 dB and 0 dB).*

Figure 10 and 11 plot results of Test-1 for Noise1 and Noise2 respectively. According to results, WER is reduced as the size of adaptation data increases. We conclude that the phenomenon of speaker-mismatching is not significant even when the largest set of adaptation data is conducted. The MIX case, where the adaptation set contains speeches from

various speakers is obviously better than the use of one specific speaker model. This indicates the original acoustic model should be adapted with its speaker independent property maintained.

5.4 Comparison with conventional methods

In this subsection, our methods of using simulated-data adaptation, S-MLLR and S-PLT, at their best configurations were compared to several conventional methods including the baseline system (without any adaptation process), MLLR, model selection, and PLT. "Train-S" consisted of speeches covering 22 words from mixed speakers and the number of MLLR cluster was 1. Evaluation results on "Test-1" with the noises "Noise-1" and "Noise-2" are expressed in Figure 12 and 13 respectively. Results obviously show high improvement of both the S-MLLR and S-PLT over the other methods.

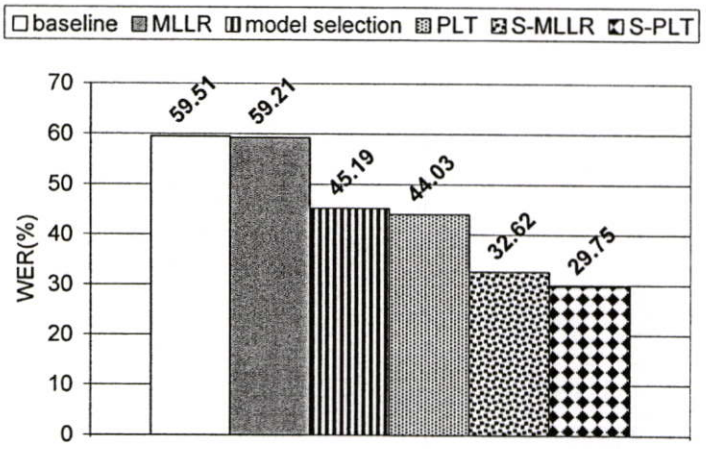


Figure 12: Comparison of baseline, MLLR, model selection and PLT using tree-structured clusters, S-MLLR and S-PLT by "Method2-1" on Test-1 data (Noise 1 noise-added speech, Train-S: Mix, SNR: 15, 10 and 0 dB).

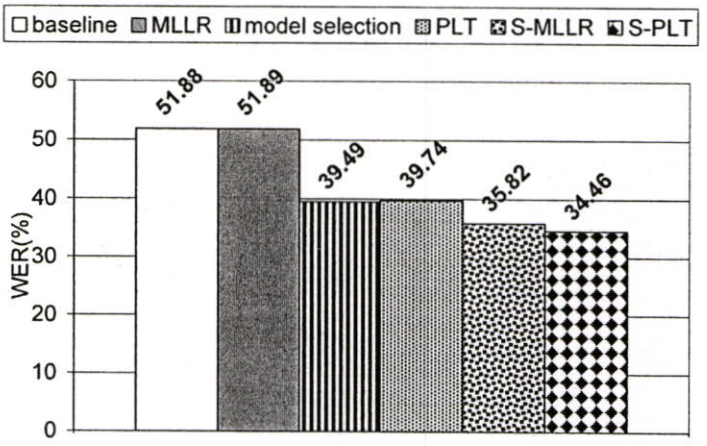


Figure 13: Comparison of baseline, MLLR, model selection and PLT using tree-structured clusters, S-MLLR and S-PLT by "Method2-1" on Test-1 data (Noise 2 noise-added speech, Train-S: Mix, SNR: 15, 10 and 0 dB).

5.5 Experiments on real noisy speech (Test-2)

The last experiment consisted in testing our proposed methods on the "Test-2" set. All parameters were the same as those set in the previous subsection. Results, shown in Figure 14, indicate the advantage of our methods on the real noisy speech. The best performance is by S-PLT, which achieves relatively 21.4% WER reduction from the normal PLT method.

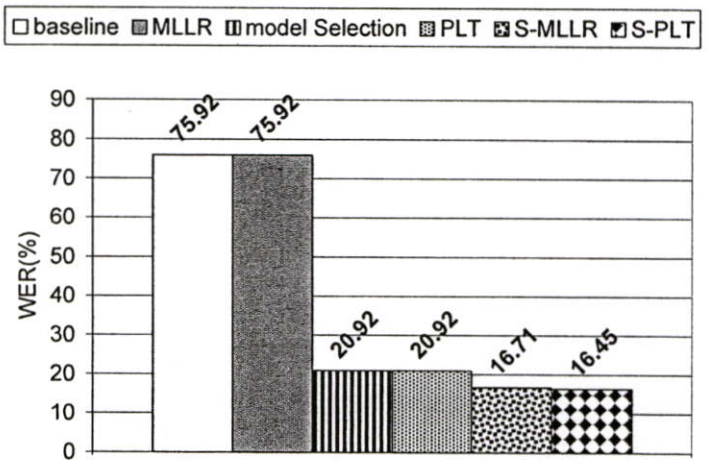


Figure 14: Comparison of baseline, MLLR, model selection using tree-structured clusters, PLT using tree-structured clusters, S-MLLR by "Method2-1" and S-PLT by "Method2-1" on Test-2 data. (Train-S: Mix).

6 Conclusion and Future Works

This paper proposed a new idea of using simulated-data adaptation in piecewise linear transformation (PLT). The idea is to increase the number of adaptation data used in PLT by adding a noise portion from an input speech to a set of clean speech. The experiment showed that the best performance was achieved when the set of clean speech contained as many distinctive phonemes and speaker as possible. The use of Method2-1, where a recognition results was selected from two results produced by the general acoustic model and the adapted acoustic model achieved better performance than using only the adapted acoustic model. The experiment also proved the advantage of using simulated-data adaptation even in the simple MLLR (S-MLLR).

Future works include an evaluation of the idea by a larger set of speech from real environments and in various situations. Further improvement of noise addition in the simulated-data adaptation will also be investigated. Another interesting issue is that the selection of clean speech from different speakers might affect the recognition result. Selecting a set of speakers that best match the input speech might give the better performance. This issue will be explored.

ACKNOWLEDGMENTS

A special thank is given to Dr. Zhipeng Zhang for his valuable advice.

REFERENCES

1. Gales M.J.F. (1995), Model-based techniques for noise robust speech recognition, PhD thesis University of Cambridge.

2. Leggetter C.J. and Woodland P.C. (1995), Maximum likelihood linear regression for speaker adaptation of continuous density HMMs, Computer Speech Language, vol.9, 171–186.
3. Gales M.J.F. and Woodland P.C. (1996), Mean and variance adaptation within the MLLR framework, Computer Speech Language, vol.10, 249–264.
4. Gauvain J.L., and Lee C.H. (1994), Maximuma posteriori estimation for multivariate Gaussian mixture observations of Markov chains, IEEE Trans. Speech Audio Processing, vol.2, 291–298.
5. Chesta C., Siohan O., and Lee C.-H. (1999), Maximum a posteriori linear regression for hidden Markov model adaptation, Proc. EuroSpeech, pp.211-214.
6. Gales M. J. F. and Young S. (1992), An improved approach to the hidden Markov model decomposition of speech and noise, Proc. ICASSP, pp. 233-236.
7. Martin F., Shikano K. and Minami Y. (1993), Recognition of noisy speech by composition of hidden Markov models, Proc. Eurospeech, pp. 1031-1034.
8. Minami Y. and Furui S. (1995), A maximum likelihood procedure for a universal adaptation method based on HMM composition, Proc. ICASSP, pp.129–132.
9. Zhang Z.P. and Furui S. (2004), Piecewise-linear transformation-based HMM adaptation for noisy speech, Speech Communication, vol.42, no.1, pp.43–58.
10. Zhang Z.P. and Furui S. (2005), Tree-Structured Clustering Methods for Piecewise Linear transformation-Based Noise Adaptation, IEICE TRANS. INF. & SYST., vol.E88–D, no.9, pp.2168-2176.
11. Kosaka T., Matsunaga S., and Sagayama S. (1996), Speaker-independent speech recognition based on tree-structured speaker clustering, Computer Speech Language, vol.10, pp.55–74.
12. http://www.milab.is.tsukuba.ac.jp/corpus/noise db.html
13. http://www.speech.cs.cmu.edu/comp.speech/ Section1/ Data/noisex.html
14. Kasuriya S., Sornlertlamvanich V., Cotsomrong P., Jitsuhiro T., Kikui G. and Sagisaka Y. (2003), NECTEC-ATR Thai speech corpus, Proc. of Oriental COCOSDA2003, pp 105-111.

ECTI-CON 2006

THE 2006 ECTI INTERNATIONAL CONFERENCE

Proceedings of the 2006 Electrical Engineering/ Electronics, Computer, Telecommunications and Information Technology (ECTI) International Conference

May 10-13, 2006

Ubonburi Hotel, Ubon Ratchathani, THAILAND

KPCA-Based Noise classification Module for Robust Speech Recognition system

Nattanun Thatphithakkul[1] , Boontee Kruatrachue[1], Chai Wutiwiwatchai[2], Sanparith Marukatat[2] and Vataya Boonpiam[2]

[1]King Mongkut's Institude of Technology Ladkrabang, Bangkok, 10520, Thailand
S6060008@kmitl.ac.th and kkboontee@kmitl.ac.th,
[2]National Electronics and Computer Technology Center, Bangkok, 12120, Thailand
chai@nectec.or.th, sanparith.marukatat@nectec.or.th and vataya.boonpiam@nectec.or.th

ABSTRACT

This paper proposes an environmental noise classification using kernel principal component analysis (KPCA) for robust speech recognition. Once the type of noise is identified, speech recognition performance can be enhanced by selecting the identified noise specific acoustic model. The proposed model applies KPCA to a set of noise features such as normalized logarithmic spectrums (NLS), and results from KPCA are used by a Support Vector Machines (SVM) classifier for noise classification. The proposed model is evaluated with 2 groups of environments. The first group contains a clean environment and 9 types of noisy environments that have been trained in the system. Another group contains other 6 types of noises not trained in the system. Noisy speech is prepared by adding noise signals from JEIDA and NOISEX-92 to the clean speech taken from NECTEC-ATR Thai speech corpus. The proposed model shows a promising result when evaluating on the task of phoneme-based 640 Thai isolated-word recognition.

Keywords: Speech recognition, Kernel PCA, SVM

1. INTRODUCTION

It is commonly known that a speech recognition system trained by speech in a clean or nearly clean environment cannot achieve good performance when working in noisy environment. Research on robust speech recognition is then necessary. This paper focuses on the construction of robust model approach which has achieved good recognition results [4]. Generally, this model-based approach aims to create an environment-specific acoustic model or to adapt the existing model to the specific environment. Several techniques of model adaptation have been proposed e.g. linear regression adaptation and parallel model combination. However, an acoustic model trained directly for specific noise is certainly superior to the adapted model, although multiple acoustic models are needed for various kinds of noise and an accurate automatic noise classification is required.

Many noise classification techniques have been studied previously. Classical technique is based on hidden Markov models (HMM) and Mel-frequency cepstral coefficients (MFCC) [5], which have been proven to give better results than human listeners [5]. Another successful technique is a neural network based system with combined features of line spectral frequencies (LSF), a zero-crossing rate and energy [8]. However, implementing LSF in a real-time system is problematic. Therefore, we aim to explore a simpler feature extraction method for noise classification.

In recent years, many kernel-based classification techniques, e.g. Support Vector Machine, Kernel PCA (KPCA) [9], Kernel FDA [10], have been proposed. These techniques have been successfully applied, not only for classification, but also for regression and feature extraction e.g. in speech recognition system [9].

This paper proposes another application of KPCA, which is noise classification. In this work, KPCA is applied to extract speech features, which are used by a pattern classifier for noise classification. An advantage of KPCA is that useful noise information can be extracted from the original feature. The computational requirement of KPCA applied to normalized logarithmic spectrums (NLS) implemented in this paper is similar to that of the MFCC or other effective features such as LSF, but with higher classification accuracy.

Our noise classification model is evaluated on 2 groups of environments. The first group contains 10 classes of environments that have been trained in the system. The second group is another set of 6 environments not trained in the system. Evaluating by the later group shows the speech recognition performance in unknown-noise environments. All noises are taken from Japan JEIDA [12] and NOISEX-92 [2]. Our Thai 640 isolated-word recognition with noise-specific acoustic models is used in the evaluation. It is noted that although the task is isolated-word recognition, phonemes are used as basic recognition units. This facilitates new word addition.

The rest of paper is organized as follows: the next section describes an overall structure of our robust speech recognition system. In Sect. 3, the KPCA algorithm is described. Sect. 4 describes our experiments, results and discussion. The last section concludes the paper and notices our future works.

2. ROBUST SPEECH RECOGNITON USING NOISE CLASSIFICATION

As described in the previous section, our robust speech recognition system uses the model-based technique, in which acoustic models are trained by speech in specific environment. An overall structure is illustrated in Fig. 1. Given a speech signal, a set of features for noise classification is extracted from a short period of silence at the beginning of signal. It is noted that this short period is assumed to be a silence where the speaker has not yet uttered. This assumption holds for our push-to-talk interface. To apply our system with other user interfaces, we need an additional module of speech/non-speech classification or other strategies to capture a non-speech portion from the input signal. Features extracted from the silence portion are then used to identify the type of environment. Once knowing the environment type, the recognizer selects a corresponding acoustic model for recognizing the rest of signal.

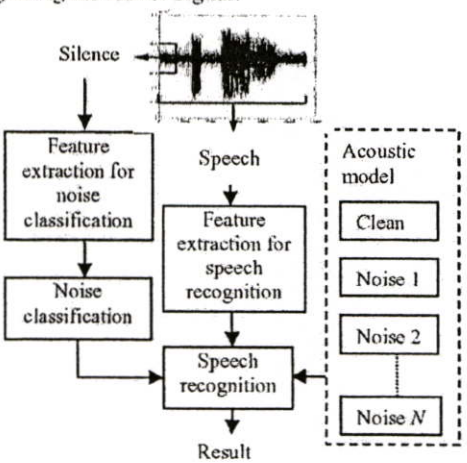


Fig. 1. Overall structure of robust speech recognition.

In this paper, speech features evaluated for noise classification include NLS, LSF, and MFCC. PCA and KPCA are applied to these basic features in order to extract meaningful features and enhance noise classification performance. For the noise classification algorithm, a fast and efficient technique is needed. In our experiment, a well-known SVM algorithm is evaluated. Speech recognition utilizes a state-of-the-art algorithm of HMM with MFCC as speech features.

3. KERNEL PRINCIPAL COMPONENT ANALYSIS

The idea of KPCA [9] is to extend the classical PCA for non-linear projection using the kernel trick. The classical PCA is based on eigenvectors of the covariance matrix of the data $X = [x_1;\ldots; x_M]$. This covariance matrix is defined by XX^T. The eigenvectors of XX^T form the principal subspace on which the data will be projected. To extend this approach using the kernel trick, we first notice that if v is an eigenvector of X^TX then $v'=Xv$ is an eigenvector of XX^T. The kernel trick is then applied by replacing the dot product in X^TX by a kernel function. It should be noted that the eigenvector v' produced by this procedure is not properly normalized and an additional normalization step is needed. The overall KPCA algorithm is as follow:

Compute the kernel matrix K. In this work we use a RBF kernel: $K_{ij} = \exp(-\|x_i - x_j\|^2 / g)$

Compute the eigen-couples of K. Let (λ_k, v_k), $k = 1, \ldots, M$ be these eigen-couples.

Normalize the k^{th} principal axis: $v_{ki} = v_{ki}\,\lambda_k^{-1/2}$. $(\lambda_k > 0)$

The projection of a vector $y \in R^n$ onto the k^{th} principal axis is done by computing $\sum_{i=1}^{M} v_{ki} k(x_i, y)$.

For simplification, we will call the feature vector projected on the principal subspace, the "weight vector" hereafter. While a basic speech feature such as NLS is effective, an optimal order of the NLS is considerably large. With limited training set, computing the eigen decomposition from a dot matrix, or kernel matrix, can be done more accurately [11].

4. EXPERIMENTS

4.1 Data preparation

Noises used in our experiments are from the JEIDA and NOISEX-92. They are clustered to 2 groups. The first group contains 8 kinds of noise from JEIDA, including crowded street, machinery factory, railway station, large air-condition, trunk road, elevator, exhibition in a booth, and ordinary train, 1 large-size car noise from NOISEX-92, and an additional clean environment. The second group contains other 6 kinds of noise from JEIDA, including exhibition in a passage, road crossing, medium-size car, computer room, telephone booth, and press factory. The former group of environments is reserved for training the noise classification and speech recognition models, and for testing the system for "known" noises (noises recognizable by the system). The later group is used for evaluating the system for "unknown" noises (noises not trained in the system).

Noisy speech was prepared by adding the noise from JEIDA or NOISEX-92 to the clean speech of NECTEC-ATR at various SNRs (0, 5, 10 and 15 dB). The preprocessed data were then clustered into several sets for noise classification and speech recognition experiments as summarized in Table 1.

4.1.1. Data set for noise classification

Three sets were prepared: a PCA and KPCA training set, a classifier training set and classifier test sets. The first set was used for computing PCA and KPCA weight vectors. The second set was used for training the noise classifier and the rest were used for evaluating the classifier.

A small frame of 1,024 samples at the beginning of the speech signal, which was expected to be silence, was used for PCA, KPCA and noise classification. As described in the Sect. 3, our speech recognizer is designed for a push-to-talk interface. With this interface, we can control the recorder to start record a silence signal

before the beginning of speech. NLS and LSF used for noise classification were computed from this silence frame.

4.1.2 Data set for speech recognition

The speech recognition task in our experiment was phoneme-based 640 isolated-word recognition. 32000 speech utterances from 32 speakers were allocated for a training set. Another set of 6400 utterances from other 10 speakers are used for testing in both known and unknown-noise modes.

An acoustic model contained HMMs representing 35 Thai phones. Each monophone HMM consisted of 5 states and 8 Gaussian mixtures per state. 39 dimensional vectors (12 MFCC, 1 log-energy, and their first and second derivatives) were used as recognition features.

Table 1. Number of utterances in experimental data sets.

TASK	DATA SET	NO. OF EXAMPLES
Noise classification	PCA/KPCA training	3,900
	Classifier training	24,000
	Known-noise test	256,000
Speech recognition	Recognizer training	32,000*
	Known-noise test	6,400*
	Unknown-noise test	6,400*

*Number of sample per noise per SNR

4.2 Noise classification results

Our proposed classification model using KPCA and SVM described in the Sect. III was compared to the classical technique using a HMM classifier [5, 6], which served as a baseline system in our experiment. The noise-classification data sets are used in this section. The followings are details of noise classification experiments.

4.2.1 Classification using a HMM system

For the HMM based noise classification system, the HTK tools are used with the same set of 39 MFCC features as in recognition. This baseline system will be referred to as "HMM_MFCC". Evaluated by the known-noise test set the lowest error rate is 8.93% with 5-state HMM and 16 mixtures/state.

4.2.2 Classification using SVM systems

A multi-class SVM classifier based on one-against-one algorithm is constructed by LIBSVM using RBF kernel. PCA and KPCA are applied to NLS (511 orders in total), denoted as "SVM_PCA" and "SVM_KPCA". The order of 24 was empirically selected for both SVM_PCA and SVM_KPCA. First we conducted an evaluation on known-noise environment. Fig. 2 shows the results obtained from different noise classification models using various kinds of features including our proposed KPCA-based feature. In this experiment, we also tested the SVM with on LSF with 10 orders as proposed in [7] and MFCC (12 orders without energy and derivative features). These 2 systems yield the error rate of 3.63% and 6.95% respectively. The same error rates were obtained when applying PCA to these two features. According to the results, the KPCA outperforms the other, except the NLS. The NLS, however, requires the largest order (511) to achieve the underlying result. Trading off between the accuracy and running time, we found the use of SVM_KPCA optimal our noise classification module.

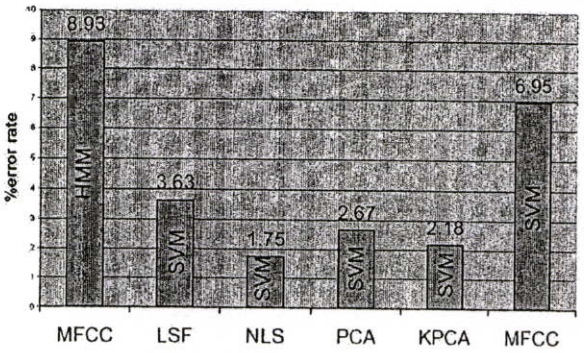


Fig. 2. Comparative results of known-noise classification error rates using various kinds of classification system.

4.3 Speech recognition results

In this section, several robust speech recognition techniques including our proposed model are experimentally compared. The first system (S1) was a conventional system without any implementation for robust speech recognition. The second system (S2) used zero-mean static coefficients [4], a well-known technique for noise-robust speech features. The third system (S3) was our proposed model, where input speech environment was identified and the corresponding acoustic model was chosen for recognition. In the S3 system, an acoustic model for each environment was trained by multi-SNR (5, 10, and 15 dB) data including each noise. The SVM_KPCA system, which achieved the best result, was used in the S3 system. The fourth system (S4) was as similar as the S3 system except that the noise classifier was replaced by the HMM_MFCC model. The last system (S5) was an ideal system, where noise is perfectly classified, i.e. 0% noise classification error. In the following experiments, the speech recognition data sets are used.

4.3.1 Speech recognition in known-noise

Evaluated by the known-noise test set, comparative results are shown in Table 2. It is obvious that our proposed model (S3) achieved the best recognition results in every case and the results are almost equal to the ideal case (S5).

4.3.2 Speech recognition in unknown-noise

Evaluated by the unknown-noise test set, comparative results are shown in Table 3. Although it is

not significant, the S4 system outperforms the S3 system. One possible reason is that the SVM classifier might over fit to the trained classes and hence underperformed the HMM classification handling unknown classes.

The results in Table 2 and 3 also underline the advantage of using noise classification module (S3 and S4) compared to conventional system (S2), even in unknown noise environments.

Table 2. Comparative results of robust speech recognition in known-noise environments.

WORD ACCURACY (%)				
S1	S2	S3	S4	S5
55.07	62.11	82.16	82.02	82.19

Table 3. Comparative results of robust speech recognition in unknown-noise environments.

WORD ACCURACY (%)			
S1	S2	S3	S4
49.52	60.88	78.18	78.58

4.4 Hybrid noise classification system

Although the SVM_KPCA classifier outperformed other classifiers, an intensive analysis showed that its errors can be recovered by selecting the noise model proposed by other classifier. Hence, we have also evaluated a hybrid architecture in which the SVM_KPCA is used in conjunction with the HMM_MFCC or the SVM_MFCC. Indeed, in this hybrid system, if both classifiers agree in noise classification, the corresponding noise model is used for recognition. Otherwise, we choose among the acoustic models proposed by both classifiers, the one which maximizes the acoustic probabilities. This combined system of SMV_KPCA and HMM_MFCC gives 82.20% accuracy on known-noise test set and 78.90% on unknown-noise test set. This combined system of HMM_MFCC and SVM_MFCC gives 82.21% on known-noise test set and 78.78% on unknown-noise test set. The overall running time is increased but still being faster than the NLS.

5. CONCLUSION AND FUTURE WORKS

This paper proposed a novel technique of robust speech recognition based on model selection. The recognizer selected a specific acoustic model from a pool of acoustic models that were trained by speech data in each type of noisy environment. A noise classification module was used to identify the type of environment. KPCA applied to the NLS was proposed for the noise classification features, and SVM was used as the noise classifier Experiments showed that the proposed model gave a promising result. When combining the model to the speech recognizer, the proposed system produced almost equal recognition accuracy to the ideal system, where the type of noisy environment was given. The proposed system working with known-noise environments achieved 20.05% higher recognition accuracy over the robust system using zero-mean static coefficients, and 0.14% higher accuracy over the baseline system using the HMM and MFCC for noise classification. A hybrid system that combined our proposed model and the baseline model was also investigated. Experimental results showed a small improvement over each individual model on both known and unknown noises.

For future works, a better way to treat unknown-noises will be intensively explored. Optimization of SVM training will be performed to avoid over-training if this is the case. Another interesting topic is to reduce the number of specific acoustic models by automatic noise clustering and constructing one acoustic model for each noise cluster.

6. REFERENCES

[1] S., Kasuriya, V. Sornlertlamvanich, P. Cotsomrong, T. Jitsuhiro, G. Kikui and Y. Sagisaka, "Thai speech database for speech recognition", Proceedings of Oriental COCOSDA2003, pp 105-111, 2003.

[2] NOISEX-92. http://www.speech.cs.cmu.edu/ comp. speech/ Section1/ Data/noisex.html

[3] A. Lima, H. Zen, Y. Nankaku, C. Miyajima, K. Tokuda and T. Kitamura, "On the use of kernel PCA for feature extraction in speech", IEICE Tran. INF.&SYST., Vol.E87-D, pp. 2802-2811, 2004.

[4] M.J.F. Gales, "Model-based techniques for noise robust speech recognition", PhD thesis University of Cambridge, 1995.

[5] L. Ma, D. Smith and B. Milner, "Context awareness using environmental noise classification", Proceedings of Eurospeech2003, pp. 2237-2240, 2003.

[6] P. Gaunard, C.G. Mubikangiey, C. Couvreur, and V. Fontaine, "Automatic classification of environmental noise events by hidden markov models", Proceedings of ICASSP1998, pp. 3609-3612, 1998.

[7] K.E. Maleh, A. Samouelian and P. Kabal, "Frame-level noise classification in mobile environments", IEEE conf. Acoustics, Speech, Signal Processing, pp 237-240, 1999.

[8] C. Shao and M. Bouchard, "Efficient classification of noisy speech using neural networks", Proceedings of ISSPA2003, pp. 357-360, 2003.

[9] B. Scholkopf, A. Amola and K.-R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem", Neural computation, 10:1299-1319, 1998.

[10] S. Mika, G. Ratsch, J. Weston, B. Scholkopf and K.-R. Muller, "Fisher Discriminant Analysis with Kernels", Neural Networks for Signal Processing IX, pp. 41-48, 1999.

[11] M. Turk and A. Pentland, "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, 3(1):71-86, 1991.

[12] www.milab.is.tsukuba.ac.jp/corpus/noise_db.html

ISSN 1905 - 050X

Transactions

on Computer and Information Technology

Volume 2, No. 1, May 2006

ECTI-CIT

Robust Speech Recognition Using KPCA-Based Noise Classification

Nattanun Thatphithakkul[1], Boontee Kruatrachue[1], Chai Wutiwiwatchai[2], Sanparith Marukatat[2], and Vataya Boonpiam[2], Non-members

ABSTRACT

This paper proposes an environmental noise classification method using kernel principal component analysis (KPCA) for robust speech recognition. Once the type of noise is identified, speech recognition performance can be enhanced by selecting the identified noise specific acoustic model. The proposed model applies KPCA to a set of noise features such as normalized logarithmic spectrums (NLS), and results from KPCA are used by a support vector machines (SVM) classifier for noise classification. The proposed model is evaluated with 2 groups of environments. The first group contains a clean environment and 9 types of noisy environments that have been trained in the system. Another group contains other 6 types of noises not trained in the system. Noisy speech is prepared by adding noise signals from JEIDA and NOISEX-92 to the clean speech taken from NECTEC-ATR Thai speech corpus. The proposed model shows a promising result when evaluating on the task of phoneme based 640 Thai isolated-word recognition.

Keywords: Speech recognition, Kernel PCA, SVM

1. INTRODUCTION

It is commonly known that a speech recognition system trained by speech in a clean or nearly clean environment cannot achieve good performance when working in noisy environment. Research on robust speech recognition is then necessary. This paper focuses on the construction of robust model approach which has achieved good recognition results [1]. Generally, this model-based approach aims to create an environment-specific acoustic model or to adapt the existing model to the specific environment. Several techniques of model adaptation have been proposed e.g. linear regression adaptation and parallel model combination [2]. However, an acoustic model trained directly for specific noise is certainly superior to the adapted model, although multiple acoustic models are needed for various kinds of noise and an accurate automatic noise classification is required.

Many noise classification techniques have been studied previously. Classical technique is based on hidden markov models (HMM), linear prediction coefficients (LPC) [3] and mel-frequency cepstral coefficients (MFCC) [4], which have been proven to give better results than human listeners [4]. Another successful technique is a neural network based system with combined features of line spectral frequencies (LSF) [5], a zero-crossing (ZC) rate and energy [6]. However, implementing LSF in a real-time system is problematic. Therefore, we aim to explore a simpler feature extraction method for noise classification.

In recent years, many kernel-based classification techniques, e.g. support vector machine (SVM) [7], kernel principal component analysis (KPCA) [8-12], kernel discriminate analysis (KDA) [13], kernel fisher discriminate analysis (FDA) [14], have been proposed. These techniques have been successfully applied, not only for classification, but also for regression and feature extraction e.g. in speech recognition [8] and image recognition system [12].

This paper proposes another application of KPCA, which is noise classification. In this work, KPCA is applied to extract speech features, which are used by a pattern classifier for noise classification. An advantage of KPCA is that useful noise information can be extracted from the original feature. The computational requirement of KPCA applied to normalized logarithmic spectrums (NLS) implemented in this paper is similar to that of the MFCC or other effective features such as LSF, but with higher classification accuracy.

Our noise classification model is evaluated on 2 groups of environments. The first group contains 10 classes of environments that have been trained in the system. The second group is another set of 6 environments not trained in the system. Evaluating by the later group shows the speech recognition performance in unknown-noise environments. All noises are taken from Japan JEIDA [15] and NOISEX-92 [16]. Our Thai 640 isolated-word recognition with noise-specific acoustic models is used in the evaluation. It is noted that although the task is isolated-word recognition, phonemes are used as basic recognition units. This facilitates new word addition.

Manuscript received on December 16, 2006; revised on March 16, 2007.

[1]King Mongkut's Institute of Technology Ladkrabang, Bangkok, 10520, Thailand; E-mail: S6060008@kmitl.ac.th and kkboontee@kmitl.ac.th,

[2]National Electronics and Computer Technology Center, Phathumthani, 12120, Thailand; E-mail: chai@nectec.or.th, sanparith.marukatat@nectec.or.th and vataya.boonpiam@nectec.or.th

The rest of paper is organized as follows: the next section describes an overall structure of our robust speech recognition system. In Sect. 3, the KPCA algorithm is described. Sect. 4 describes our experiments, results and discussion. The last section concludes the paper and notices our future works.

2. ROBUST SPEECH RECOGNITON USING NOISE CLASSIFICATION

As described in the previous section, our robust speech recognition system uses the model-based technique, in which acoustic models are trained by speech in specific environment. An overall structure is illustrated in Fig. 1. Given a speech signal, a set of features for noise classification is extracted from a short period of silence at the beginning of signal. It is noted that this short period is assumed to be a silence where the speaker has not yet uttered. This assumption holds for our push-to-talk interface. To apply our system with other user interfaces, we need an additional module of speech/non-speech classification or other strategies to capture a non-speech portion from the input signal. Features extracted from the silence portion are then used to identify the type of environment. Once knowing the environment type, the recognizer selects a corresponding acoustic model for recognizing the rest of signal.

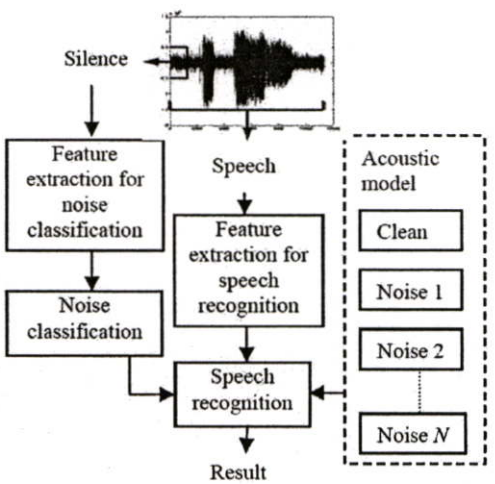


Fig.1: Overall structure of robust speech recognition.

With this model, there are 3 particular difficulties:

- How to construct a robust acoustic model for a variation of signal-to-noise ratios (SNR)? In our system, a particular acoustic model is trained on noisy speech with various levels of SNR. Clean speech, whose SNR exceeds 30 dB is also combined in the training set of each noisy acoustic model.
- How to construct the environment or noise classification module? Time consuming by the noise classification module should be as low as possible, so that the overall system can achieve an acceptable processing time. The construction of such module is the main objective of this paper.
- How can the robust speech recognition model deal with unknown noises, i.e. noises not trained in the model? Normally, several major noises are trained in the system and each of other noises is expected to be classified as one of the major noises. This paper also reports the effect of our model for unknown-noise classification.

In this paper, speech features evaluated for noise classification include NLS, LSF, LPCC and MFCC. PCA and KPCA are applied to these basic features in order to extract meaningful features and enhance noise classification performance. For the noise classification algorithm, a fast and efficient technique is needed. In our experiment, a well-known SVM algorithm is evaluated. Speech recognition utilizes a state-of-the-art algorithm of HMM with MFCC as speech features.

3. KERNEL PRINCIPAL COMPONENT ANALYSIS

3.1 Kernel functions

The use of nonlinear kernel functions is a strategy to raise the capability of simple algorithms such as PCA in dealing with more complicated data. Indeed, extending these algorithms for a non-linear case may be done by replacing the involved variables by their values on a new feature space. Transformation from the original space to a new space may be done by some mapping function ?. However, by choosing an appropriate mapping function, the dot product in the new feature space can be performed by a nonlinear function in the input space, the so-called kernel function. Hence, by replacing the dot product involving in a classical algorithm by some kernel function, we can extend this algorithm to the non-linear case. This is usually referred to as the kernel trick [10]. The commonly used kernels are shown in Table 1.

Table 1: Some useful kernel functions.

Kernel function	Equation
Polynomial	$k(x_i, x_j) = (x_i \cdot x_j + 1)^d$
RBF	$k(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / g)$

3.2 KPCA

The idea of KPCA [8-9] is to extend the classical PCA for non-linear projection using the kernel trick. Given a set of M samples x_i, $i = 1,2,...,M$ with $x_i \in R^n$. The classical PCA is done by computing eigenvectors and eigenvalues of the covariance matrix of these examples. Let $X = [x_1; x_2; ; x_M]$ be the matrix of these M examples, the covariance matrix is defined by $C = M^{-1} X X^T$. The normalized eigenvectors of C form the principal subspace on which the data will

be linearly projected. To extend this approach using the kernel trick, we first notice that if we dispose an eigen-couple $(\lambda, v)$of the dot product matrix$X^T X$ then we can also derive an eigen-couple $(\acute{\lambda}, \acute{v})$ of the covariance matrix C. Indeed, we have $\lambda\ v = X^T X\ v$, so by pre-multiplying both sides of the equation by $M^{-1}X$ we get $(\lambda M^{-1})(X\ v) = (M^{-1}X\ X^T)(X\ v) = C\ (X\ v)$. This means that $\acute{\lambda} = \lambda\ M^{-1}$ and $\acute{v} = X\ v$ forms an eigen-couple of the covariance matrix C. The kernel trick is then applied by replacing the dot product in $X^T X$ by a kernel function. It should be noted that the eigenvector produced by this procedure may not be properly normalized. Therefore an additional normalization step is needed. The overall KPCA algorithm is as follow:

- Compute the kernel matrix K with $K_{ij} = k(x_i,x_j)$ where k is a kernel function.
- Compute the eigen-couples of$K$.Let$(\lambda k, vk), k = 1, ..., M$ be these eigen-couples.
- Normalize the k^{th} principal axis by computing $v_{ki} = v_{ki}\ \lambda k_k^{-1/2}.(\lambda k > 0)$
- The projection of a vector $y \in R^n$ onto the k^{th} principal axis is done by computing $\sum_{i=1}^{M} v_{ki}\ k(x_i y)$.

For simplification, we will call the feature vector projected on the principal subspace, the "weight vector" hereafter.

For simplification, we will call the feature vector projected on the principal subspace, the "weight vector" hereafter. While a basic speech feature such as NLS is effective, an optimal order of the NLS is considerably large. With limited training set, computing the eigen decomposition from a dot matrix, or kernel matrix, can be done more accurately [11].

4. EXPERIMENTS

4.1 Data preparation

Noises used in our experiments are from the JEIDA and NOISEX-92. They are clustered to 2 groups. The first group contains 8 kinds of noise from JEIDA, including crowded street, machinery factory, railway station, large air-condition, trunk road, elevator, exhibition in a booth, and ordinary train, 1 large-size car noise from NOISEX-92, and an additional clean environment. The second group contains other 6 kinds of noise from JEIDA, including exhibition in a passage, road crossing, medium-size car, computer room, telephone booth, and press factory. The former group of environments is reserved for training the noise classification and speech recognition models, and for testing the system for "known" noises (noises recognizable by the system). The later group is used for evaluating the system for "unknown" noises (noises not trained in the system).

Noisy speech was prepared by adding the noise from JEIDA or NOISEX-92 to the clean speech of NECTEC-ATR [17] at various SNRs (0, 5, 10 and 15 dB). The pre-processed data were then clustered into several sets for noise classification and speech recognition experiments as summarized in Table 2.

4.1.1.Data set for noise classification

Three sets were prepared: a PCA and KPCA training set, a classifier training set and classifier test sets. The first set was used for computing PCA and KPCA weight vectors. The second set was used for training the noise classifier and the rest were used for evaluating the classifier.

A small frame of 1,024 samples at the beginning of the speech signal, which was expected to be silence, was used for PCA, KPCA and noise classification. As described in the Sect. 3, our speech recognizer is designed for a push-to-talk interface. With this interface, we can control the recorder to start record a silence signal before the beginning of speech. NLS and LSF used for noise classification were computed from this silence frame.

4.1.2 Data set for speech recognition

The speech recognition task in our experiment was phoneme-based 640 isolated-word recognition. 32000 speech utterances from 32 speakers were allocated for a training set. Another set of 6400 utterances from other 10 speakers are used for testing in both known and unknown-noise modes. The HMMs representing 35 Thai phones [18]. Each triphone HMM consisted of 5 states and 8 Gaussian mixtures per state. MFCC 39 dimensional vectors (12 MFCC, 1 log-energy, and their first and second derivatives) were used as recognition features.

Table 2: Number of utterances in experimental data sets.

TASK	DATA SET	AMOUNT
Noise classification	PCA/KPCA training	3,900
	Classifier training	24,000
	Known-noise test	256,000
Speech recognition	Recognizer training	32,000*
	Known-noise test	6,400*
	Unknown-noise test	6,400*

*Number of samples per noise per SNR

4.2 Noise classification results

Our proposed classification model using KPCA and SVM described in the Sect. 3 was compared to the classical technique using a HMM classifier [3-4], which served as a baseline system in our experiment. The noise-classification data sets are used in this section. The followings are details of noise classification experiments.

4.2.1 Classification using a HMM system

For the HMM [19] based noise classification system, we have varied the number of states as well as the number of Gaussian mixtures per state. The same set of MFCC and LPC features are used as classification features. This baseline system will be referred to as "HMM_ MFCC" and "HMM_ LPC". Fig. 2 and Fig. 3 present results of the evaluation of this system on the known-noise test set.

4.2.2 Classification using SVM systems

A multi-class SVM [20] classifier based on one-against-one algorithm. Two kinds of kernel functions, RBF and Polynomial, are evaluated. PCA and KPCA are applied to three types of speech features including NLS (511 orders), LSF (10 orders) and MFCC (10, 12, 16 and 20 orders without energy and derivative features). The order of PCA and KPCA weight vectors is empirically tuned for each comparison. The known-noise test set is also used for evaluation in this section. Results and discussions are as follows.

A preliminary experiment consists in comparing the three speech features namely NLS, LSF and MFCC as well as the kernel used in the SVM classifier. The Fig 4 and 5 show the results obtained from MLS and LSF features using polynomial and RBF kernel respectively. The results obtained from MFCC with various orders are shown in Fig 6 and 7 for polynomial and RBF kernel respectively.

From these 4 figures, we can see that the best result is obtained by the RBF-kernel SVM using NLS.

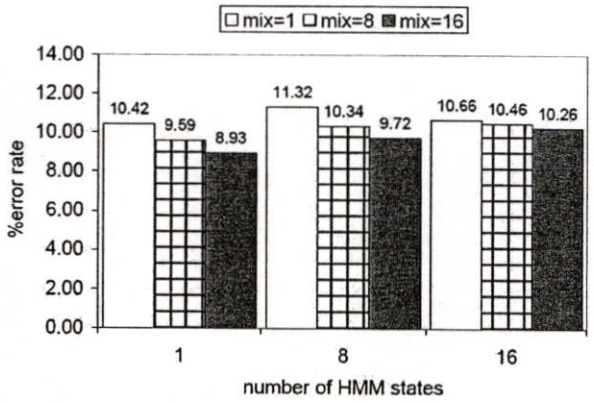


Fig.2: Error rate results (%) of known-noise classification based on HMM_ MFCC

However, a large order of NLS is needed to achieve such performance (511 orders in our case). The large number of features requires a longer time and larger storage to process. Reducing the order of NLS without a drawback of performance degradation is thus interesting.

Next, we investigate the effect of dimension reduction via PCA on the accuracy of our classi-

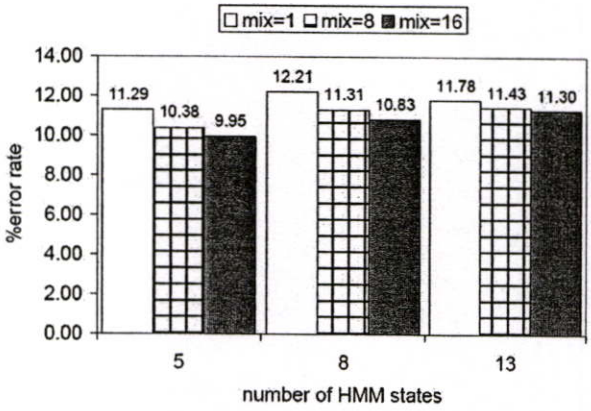


Fig.3: Error rate results (%) of known-noise classification based on HMM_ LPC.

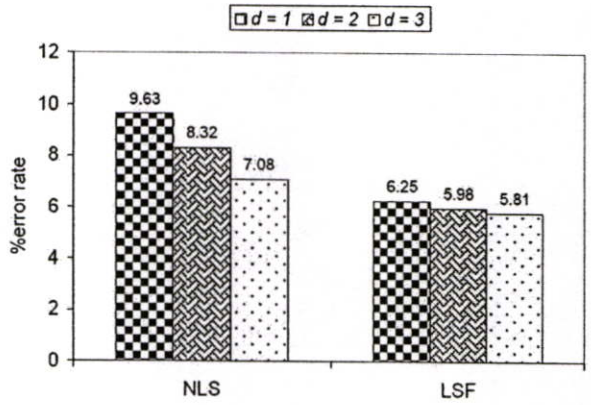


Fig.4: Error rate results (%) of known-noise classification based on SVM (10-order LSF and 511-order NLS, kernel functions of SVM: Polynomial).

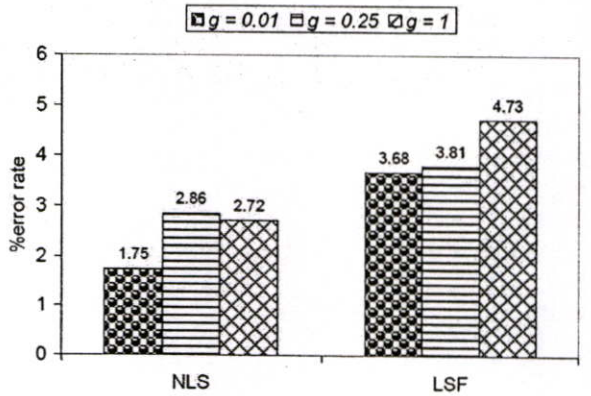


Fig.5: Error rate results (%) of known-noise classification based on SVM (10-order LSF and 511-order NLS, kernel functions of SVM: RBF).

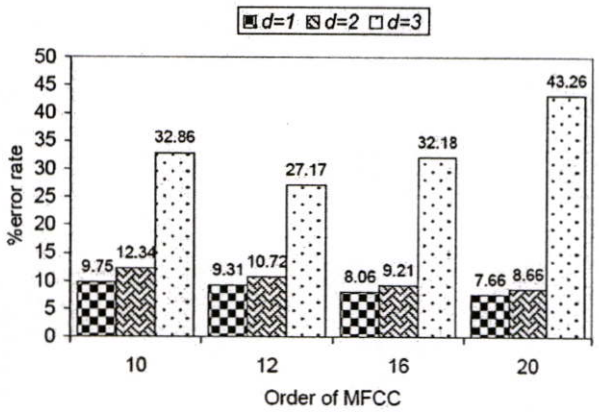


Fig.6: Error rate results (%) of known-noise classification based on SVM (MFCC with various orders, kernel functions of SVM: Polynomial).

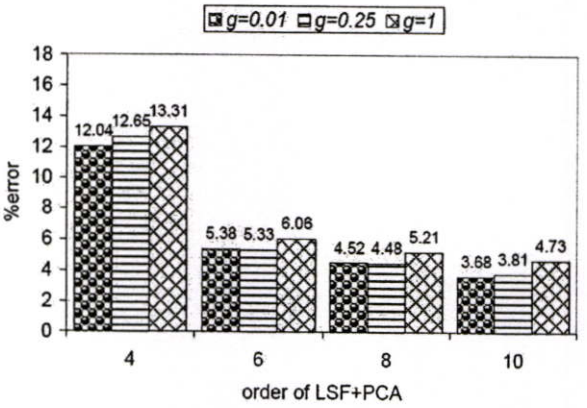


Fig.9: Error rate results (%) of known-noise classification based on SVM (LSF+PCA with various orders, kernel functions of SVM: RBF).

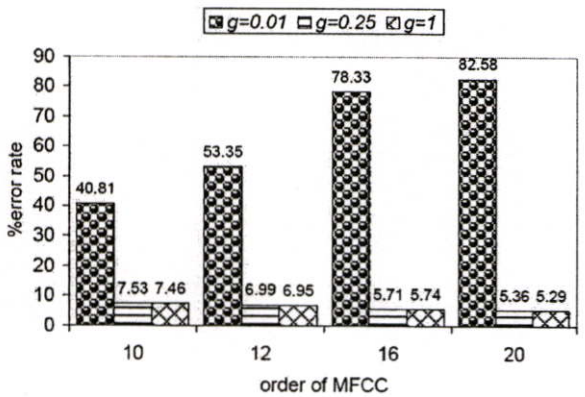


Fig.7: Error rate results (%) of known-noise classification based on SVM (MFCC with various orders, kernel functions of SVM: RBF).

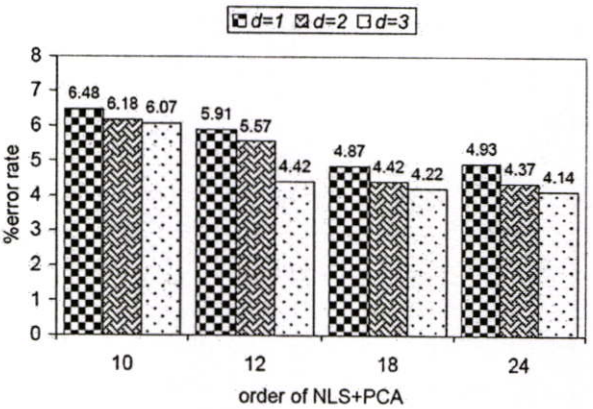


Fig.10: Error rate results (%) of known-noise classification based on SVM (NLS+PCA with various orders, kernel functions of SVM: Polynomial).

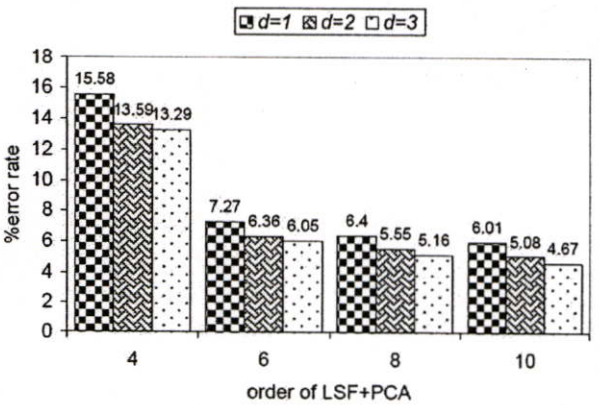


Fig.8: Error rate results (%) of known-noise classification based on SVM (LSF+PCA with various orders, kernel functions of SVM: Polynomial).

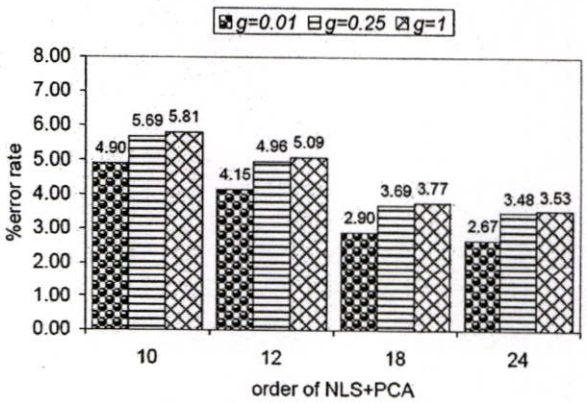


Fig.11: Error rate results (%) of known-noise classification based on SVM (NLS+PCA with various orders, kernel functions of SVM: RBF).

fier. Applications of PCA on the 10-order LSF (denoted as LSF+PCA) and 511-order NLS (denoted as NLS+PCA) are then performed and results are shown in Fig. 8-11. The Fig 8 and 9 show the results obtained from LSF+PCA feature when using polynomial and RBF kernel respectively. The Fig. 10 and 11 show the error rate obtained with NLS+PCA. From our preliminary experiments, the classification accuracy trends to be saturated when the order of PCA exceeds 24. Hence these 2 figures (10 and 11) show only the results obtained from NLS+PCA up to the order of 24.

From these 4 figures, it is clear that using the PCA-based feature of NLS and LSF does degrade the classification accuracy, with the advantage of faster processing time. For LSF+PCA, changing from 10 orders to 6 orders, we increase about 2% error rate while the gain in processing time is not significant. For NLS+PCA, reducing from full 511 orders to 24 orders allows us gaining a significant processing time, while increasing only a slight error rate. It should be noted that, even if the order of NLS+PCA is higher than that of the LSF, computing the LSF is much more complex than the NLS+PCA. From these results, the 24 first principal components of NLS with RBF kernel is a suitable choice for the noise classification module.

The objective of the next experiment is to see whether moving from the classical linear PCA to the non-linear analysis of KPCA allows further improvement KPCA has proved to be efficient for speech recognition [4]. In this experiment, RBF kernel is used for the KPCA (RBF at $g = 0.1$). Results of applying KPCA to the NLS (NLS+KPCA) are shown in Fig. 12 and Fig. 13 for polynomial and RBF kernel of the SVM classifier respectively. The lowest error rate achieved is 2.35% obtained from 24-order KPCA and RBF-kernel SVM, which is also the best case comparing to all previous experiments of PCA and KPCA. This also underlines the advantage of using non-linear analysis in extracting significant features by KPCA.

4.2.3 Comparison to other noise classification techniques

In this section, we evaluate the SVM classifier working on features extracted from 511 order NLS using PCA and KPCA against other approaches. The two systems are denoted as "SVM_PCA" and "SVM_KPCA" respectively. We use the order of 24 for the extracted feature from both PCA and KPCA. This order is selected empirically in previous experiments.

Fig. 14 shows the results obtained from different noise classification models using various kinds of features including our proposed KPCA-based feature. Other noise environment classifiers include the HMM with LPC and with MFCC features, the SVM with full 511-order NLS, 10-order LSF and 20-order MFCC (without energy and derivative features).

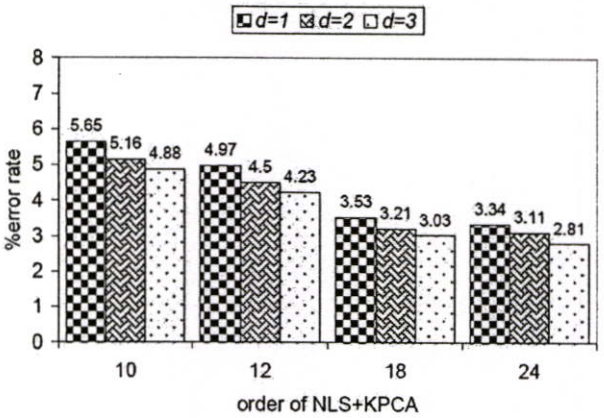


Fig.12: Error rate results (%) of known-noise classification based on SVM (NLS+KPCA (RBF at $g = 0.1$) with various orders, kernel functions of SVM: Polynomial).

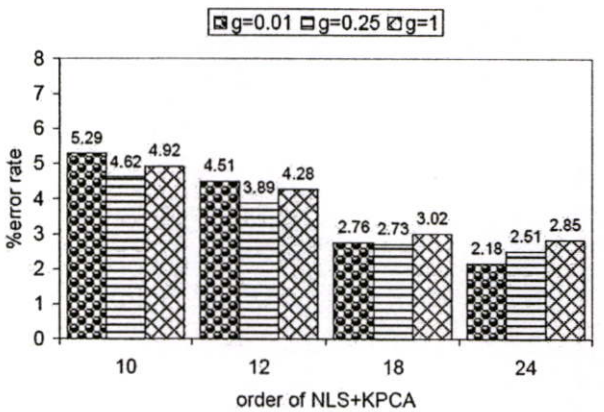


Fig.13: Error rate results (%) of known-noise classification based on SVM (NLS+KPCA (RBF at $g = 0.1$) with various orders, kernel functions of SVM: RBF).

From these results, the SVM classifiers outperform the HMM classifier in all case. Moreover, the SVM with LSF and MFCC give the error rate of 3.63% and 5.29% respectively. It should be noted that, the same error rate of 3.63% were obtained when applying PCA to the 10-order LSF. According to the results, the KPCA outperforms the other, except the NLS. The NLS, however, requires the largest order (511) to achieve the underlying result. Trading off between the accuracy and running time, we found the use of SVM_KPCA optimal our noise classification module.

4.3 Speech recognition results

In this section, several robust speech recognition techniques including our proposed model are experimentally compared. The first system (S1) was a conventional system without any implementation for robust speech recognition. The second system (S2) used

zero-mean static coefficients [19], a well-known technique for noise-robust speech features. The third system (S3) was our proposed model, where input speech environment was identified and the corresponding acoustic model was chosen for recognition. In the S3 system, an acoustic model for each environment was trained by multi-SNR (5, 10, and 15 dB) data including each noise. The SVM_ KPCA system (RBF at g = 0.1), which achieved the best result, was used in the S3 system. The fourth system (S4) was as similar as the S3 system except that the noise classifier was replaced by the HMM_ MFCC model. The next system (S5) was an ideal system, where noise is perfectly classified, i.e. 0% noise classification error. In order to underline the importance of the classification module, we also considered the last system (S6) which is equipped with random noise classification module. These two systems, S5 and S6, indicate the upper and the lower bounds of the recognition system using noise specific HMM. In the following experiments, the speech recognition data sets are used.

4.3.1 Speech recognition in known-noise

Evaluated by the known-noise test set, comparative results are shown in Table 3. It is obvious that our proposed model (S3) achieved the best recognition results in every case and the results are almost equal to the ideal case (S5).

4.3.2 Speech recognition in unknown-noise

Evaluated by the unknown-noise test set, comparative results are shown in Table 4. Although it is not significant, the S4 system outperforms the S3 system. One possible reason is that the SVM classifier might over fit to the trained classes and hence underperformed the HMM classification in handling unknown classes.

The results in Table 3 and 4 also underline the advantage of using noise classification module (S3 and S4) compared to conventional system (S2), even in unknown noise environments.

Table 3: Comparative results of robust speech recognition in known-noise environment.

Environments	Word accuracy (%)					
	S1	S2	S3	S4	S5	S6
Clean	93.02	92.45	93.02	92.91	93.02	86.22
Street	65.57	67.92	83.28	83.15	83.39	75.65
Factory	41.65	47.33	75.69	75.61	75.68	53.03
Station	45.12	52.79	77.62	77.44	77.67	63.69
Air condition	42.46	53.51	81.15	81.12	81.17	63.83
Road	53.90	56.99	77.30	76.35	77.39	64.68
Elevator	52.88	59.01	81.49	81.36	81.47	70.25
Exhibition	41.57	56.42	82.47	82.45	82.49	66.68
Train	25.43	45.72	79.66	79.63	79.70	49.55
Car	89.08	88.98	89.93	90.20	89.95	78.86
Average	55.07	62.11	82.16	82.02	82.19	67.24

Table 4: Comparative results of robust speech recognition in unknown-noise environments.

Environments	Word accuracy (%)			
	S1	S2	S3	S4
Exhibition	63.17	71.69	85.25	86.35
Road	45.61	60.54	77.09	76.83
Car	78.42	82.02	86.07	86.39
Computer room	40.20	56.71	77.69	78.13
Telephone booth	37.87	45.31	70.56	71.38
Factory	31.84	49.03	72.42	72.40
Average	49.52	60.88	78.18	78.58

4.4 Hybrid noise classification system

Although the SVM_ KPCA classifier outperformed other classifiers, an intensive analysis showed that its errors can be recovered by selecting the noise model proposed by other classifier. Hence, we have also evaluated a hybrid architecture in which the SVM_ KPCA is used in conjunction with the HMM_ MFCC or the SVM_ MFCC. Indeed, in this hybrid system, if both classifiers agree in noise classification, the corresponding noise model is used for recognition. Otherwise, we choose among the acoustic models proposed by both classifiers, the one which maximizes the acoustic probabilities. This combined system of SMV_ KPCA and HMM_ MFCC gives 82.20% accuracy on known-noise test set and 78.90% on unknown-noise test set. This combined system of HMM_ MFCC and SVM_ MFCC gives 82.21% on known-noise test set and 78.78% on unknown-noise test set. The overall running time is increased but still being faster than the NLS.

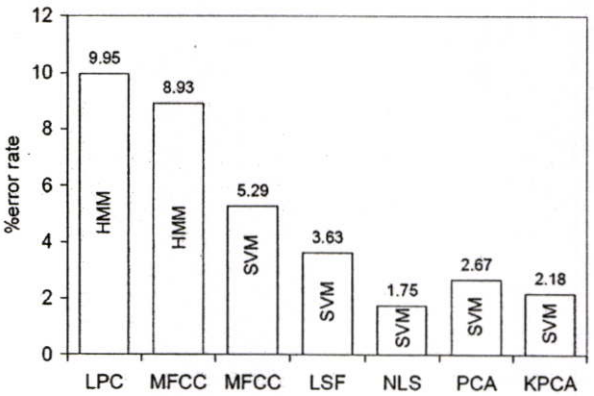


Fig.14: Comparative results of robust speech recognition in unknown-noise environments.

5. CONCLUSION AND FUTURE WORKS

This paper proposed a novel technique of robust speech recognition based on model selection. The recognizer selected a specific acoustic model from a pool of acoustic models that were trained by speech data

in each type of noisy environment. A noise classification module was used to identify the type of environment. KPCA applied to the NLS was proposed for the noise classification features, and SVM was used as the noise classifier Experiments showed that the proposed model gave a promising result. When combining the model to the speech recognizer, the proposed system produced almost equal recognition accuracy to the ideal system, where the type of noisy environment was given. The proposed system working with known-noise environments achieved 20.05% higher recognition accuracy over the robust system using zero-mean static coefficients, and 0.14% higher accuracy over the baseline system using the HMM and MFCC for noise classification. A hybrid system that combined our proposed model and the baseline model was also investigated. Experimental results showed a small improvement over each individual model on both known and unknown noises.

For future works, a better way to treat unknown-noises will be intensively explored. Optimization of SVM training will be performed to avoid over-training if this is the case. Other successful classifiers such as an optimal Bayes as well as applications of PCA and KPCA to other effective speech features such as MFCC will be investigated. Another interesting topic is to reduce the number of specific acoustic models by automatic clustering of noises and constructing one acoustic model for each noise cluster.

References

[1] M.J.F. Gales, "Model-based techniques for noise robust speech recognition", PhD thesis University of Cambridge, 1995.

[2] Y. Gang, "Speech recognition in noisy environments: A survey", Speech Communication, Vol. 16, pp. 261-291, 1995.

[3] P. Gaunard, C.G. Mubikangiey, C. Couvreur, and V. Fontaine, "Automatic classification of environmental noise events by hidden markov models", Proceedings of ICASSP1998, pp. 3609-3612, 1998.

[4] L. Ma, D. Smith and B. Milner, "Context awareness using environmental noise classification", Proceedings of Eurospeech2003, pp. 2237-2240, 2003.

[5] K.E. Maleh, A. Samouelian and P. Kabal, "Frame-level noise classification in mobile environments", IEEE conf. Acoustics, Speech, Signal Processing, pp 237-240, 1999.

[6] C. Shao and M. Bouchard, "Efficient classification of noisy speech using neural networks", Proceedings of ISSPA2003, pp. 357-360, 2003.

[7] N. Cristianini and J.S. Taylor. "An introduction to support vector machines and other kernel-based learning methods", Cambridge: Cambridge University Press, 2000.

[8] A. Lima, H. Zen, Y. Nankaku, C. Miyajima, K. Tokuda and T. Kitamura, "On the use of kernel PCA for feature extraction in speech", IEICE Tran. INF.SYST., Vol.E87-D, pp. 2802-2811, 2004.

[9] N. Thatphithakkul, B. Kruatrachue, C. Wutiwiwatchai, S. Marukatat and V. Boonpiam, "KPCA-Based Noise classification Module for Robust Speech Recognition system", Proceeding of ECTI-CON2006, pp. 231-234, 2006.

[10] B. Scholkopf, A. Amola and K.-R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem", Neural computation, 10:1299-1319, 1998.

[11] M. Turk and A. Pentland, "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, 3(1):71-86, 1991.

[12] K.I. Kim, K. Jung and H.J. Kim, "Face recognition using kernel principal component analysis", IEEE Signal Processing. Lett. vol.9, no.2, pp.40-42, 2002.

[13] V. Roth and V. Steinhage, "Nonlinear discriminate analysis using kernel function", Advances in neural information processing systems, pp. 568-574, 2000.

[14] S. Mika, G. Ratsch, J. Weston, B. Scholkopf and K.-R. Muller, "Fisher Discriminate Analysis with Kernels", Neural Networks for Signal Processing IX, pp. 41-48, 1999.

[15] www.milab.is.tsukuba.ac.jp/corpus/noise_db.html

[16] NOISEX-92. http://www.speech.cs.cmu.edu/comp. speech/ Section1/ Data/noisex.html

[17] S., Kasuriya, V. Sornlertlamvanich, P. Cotsomrong, T. Jitsuhiro, G. Kikui and Y. Sagisaka, "Thai speech database for speech recognition", Proceedings of Oriental COCOSDA2003, pp 105-111, 2003.

[18] S. Kasuriya, S. Kanokphara, N. Thatphithakkul, P. Cotsomrong and T. Sunpethniyom, "Context-independent acoustic models for Thai speech recognition", Proceedings of ISCIT2004, pp.991-994, 2004.

[19] The HTK book version 3.1, Cambridge University, http://htk.eng.cam.ac.uk, 2001.

[20] LIBSVM - A library for Support Vector Machines. http://csie.ntu.edu.tw/ cjlin/libsvm/

Nattanun Thatphithakkul received the B.Eng and M.Eng degree from Suranaree University, Thailand, in 2000 and in 2002, respectively. He is currently a Ph.D. student at King Mongkut's Institute of Technology Ladkrabang in Computer Engineering. His research activities are oriented toward robust speech recognition and noise model adaptation.

Boontee Kruatrachue received the BS. in Electrical Engineering from Kasetsart Univeristy, Thailand, in 1981, and M.S. and Ph.D degrees in Electrical Engineering from Oregon State Univeristy, USA., in 1984 and 1987, respectively. During 1988-1990, he was Software Engineer at Astronautics Corporation of America, Wisconsin, USA. He is now associate professor at computer engineering department, King Mongkut's Institute of Technology Ladkrabang, Thailand. His research interests include pattern recognition, data mining and machine learning.

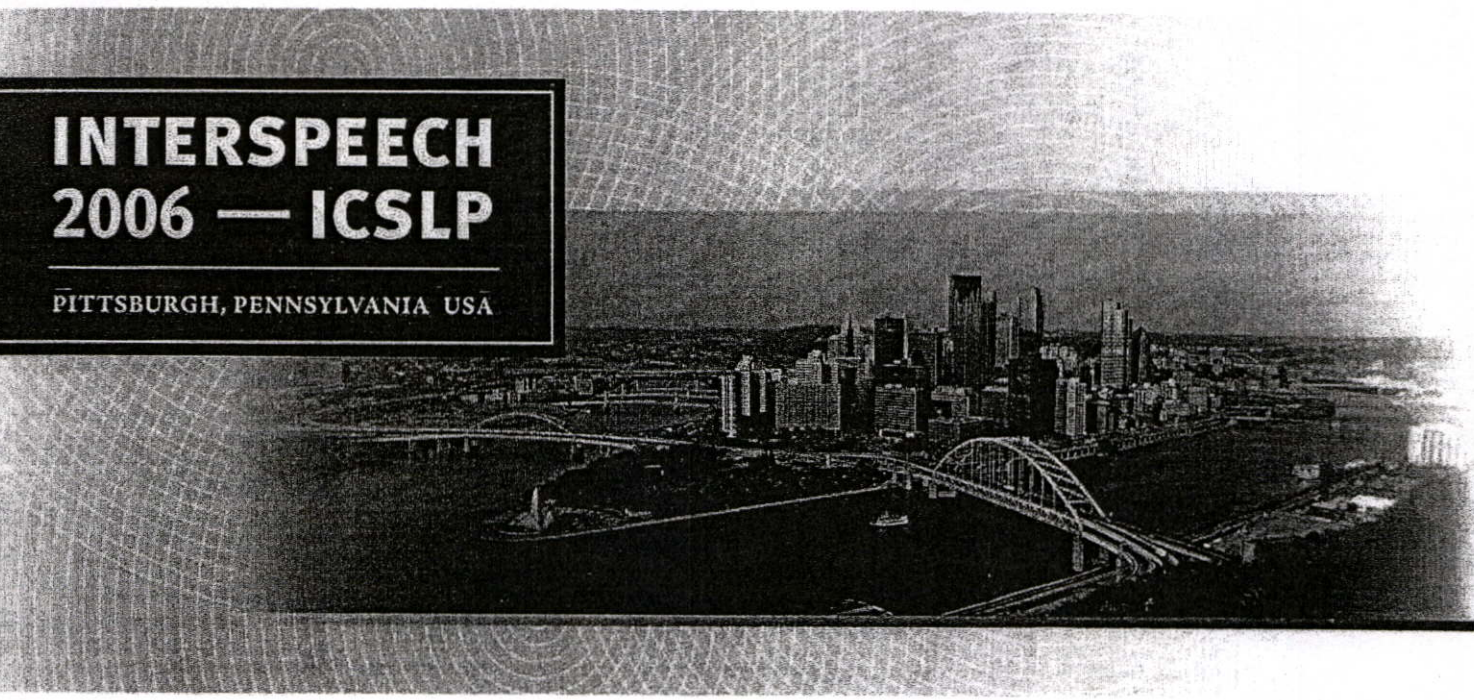
Chai Wutiwiwatchai received B.Eng. (the first honor) and M.Eng. degrees of electrical engineering from Thammasat and Chulalongkorn University, Thailand in 1994 and 1997 respectively. He received Ph.D. from Tokyo Institute of Technology in 2004 under a scholarship of Japanese government. He is now Chief of the Speech Technology Section of the National Electronics and Computer Technology Center (NECTEC), Thailand. His research interests include speech and speaker recognition, natural language processing, and human-machine interaction.

Sanparith Marukatat received the License and Ma?trise degree from University of Franche-Compte. He has finished his DEA (a kind of French one-year Master degree) and his doctoral degree at University of Paris 6 in 2000 and 2004 respectively. He is currently a researcher in the Information RD Division at National Electronics and Computer Technology Center (NECTEC), Thailand. His research interests include classification problem, subspace projection and sequence modelling.

Vataya Boonpiam received the B.Sc and M.Sc degree from King Mongkut's Institute of Technology North Bangkok, Thailand, in 2000 and in 2004, respectively. Her research interests include speech recognition. She is currently a researcher of Information Research and Development Division, National Electronics and Computer Technology Center (NECTEC).

CONFERENCE PROGRAM AND ABSTRACT BOOK

September 17 — 21, 2006 Ninth International Conference on Spoken Language Processing



INTERSPEECH 2006

A Simulated-Data Adaptation Technique for Robust Speech Recognition

Nattanun Thatphithakkul[1], Boontee Kruatrachue[1], Chai Wutiwiwatchai[2], Sanparith Marukatat[2], and Vataya Boonpiam[2]

[1]Department of Computer Engineering
King Mongkut's Institude of Technology Ladkrabang, Bangkok, 10520, Thailand
[2]Speech Technology Section, Information Research and Development Division
National Electronics and Computer Technology Center, Pathumthani, 12120, Thailand
S6060008@kmitl.ac.th, kkboontee@kmitl.ac.th, chai@nectec.or.th,
sanparith.marukatat@nectec.or.th, vataya.boonpiam@nectec.or.th

Abstract

This paper proposes an efficient acoustic model adaptation method based on the use of simulated-data in maximum likelihood linear regression (MLLR) adaptation for robust speech recognition. Online MLLR adaptation is an unsupervised process which requires an input speech with phone labels transcribed automatically. Instead of using only the input signal in adaptation, our proposed simulated data method increases the size of adaptation data by adding noise portions extracted from the input speech to a set of pre-recorded clean speech, whose correct transcriptions are known. Various configurations of the proposed method are explored. Evaluations are performed with both additive and real noisy speech. The experimental results show that the proposed system achieves higher recognition rate than the system using only the input speech in adaptation and the system using a multi-conditioned acoustic model.

Index Terms: robust speech recognition, MLLR, online-adaptation

1. Introduction

It is commonly known that a speech recognition system trained by speech in a clean or nearly clean environment cannot achieve good performance when working in noisy environment. Research on robust speech recognition is then necessary. This paper focuses on the model-based approach, which has achieved good recognition results [1]. The model-based approach aims to create or to adapt the acoustic model in specific environments. Research works on the model-based approach have been extensively carried out. Figure 1 illustrates a normal recognition process with online-adaptation. An input speech is first phone-labeled given an original acoustic model. The input speech with phone labels is then used to adapt the original acoustic model and the model after adaptation is exploited in the final recognition step. Both maximum a posteriori (MAP) adaptation [2] and maximum likelihood linear regression (MLLR) [3], and [4] are efficient adaptation algorithms.

The model presented in Figure 1, however, has two major limitations. First, the MLLR or MAP requires a large-enough set of adaptation data in order to achieve a good recognition result. In real world applications, users often input a very short sentence or, worst, only an isolated-word, which limits the improvement of adaptation. Second, online-adaptation is unsupervised adaptation, i.e. it uses phone-labels transcribed automatically by the original acoustic model. Given the original acoustic model, which may incorrectly transcribe the input speech, the adapted model cannot yield a satisfied result.

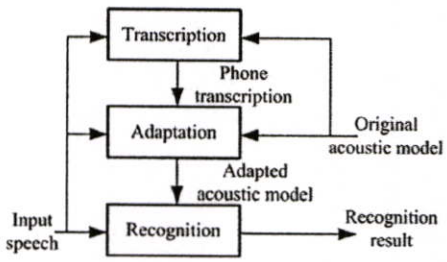


Figure 1 *A recognition process with online-adaptation.*

This paper proposes a novel approach of simulated-data adaptation, which resolves two limitations mentioned above. The simulated-data adaptation approach increases adaptation data by adding background noise extracted from the input signal to a pre-recorded set of clean speech, whose correct transcriptions are known. This process not only increases the size of adaptation set, but also reduces the problem of using incorrect transcriptions in adaptation. The MLLR algorithm performs faster and better than the MAP when the adaptation set is small, whereas the MAP becomes asymptotically more accurate than the MLLR when the size of adaptation set increases [5]. Since one of our concerns is real-time processing, the size of the adaptation data cannot be very large. In this condition, we choose only the MLLR adaptation in experiments.

The proposed system is evaluated by noisy speech in 3 sets of environment. The first set contained speech in a clean environment and 9 types of noisy environments that have been trained in the system. The second set contains speech in other 2 types of noisy environments not trained in the system. Noisy speech is prepared from noise signals taken from JEIDA (Japan Electronic Industry Development Association) [6] and a real noise signal collected in an exhibition in Thailand (NAC 2005). Noise signals are added to clean speech taken from NECTEC-ATR Thai speech corpus [7] at various SNRs (0, 10, 15 dB). The third set contains speech signals recorded in a real environment of another exhibition in Thailand (ICT-EXPO 2005). The estimated SNR of the last set is 0-5 dB.

The next section explains our proposed model. Section 3 describes data sets used in experiments. Experimental results are reported in Section 4. Section 5 concludes this paper and discusses on the future work.

2. Simulated-Data Adaptation

Our proposed method of using simulated-data in MLLR [4] adaptation, denoted as "S-MLLR", is illustrates in Figure 2. While the conventional process employs only an input signal in acoustic model adaptation, the S-MLLR method extends the adaptation set by using simulated-data created by adding noise extracted from the input signal to an existing set of clean speech. As described in the introduction, simulated-data adaptation overcomes problems of data sparseness in adaptation and unknown label of the input speech. Two issues are considered in the proposed method. The first issue is how to accurately extract noise portions from the input speech. Section 2.1 describes our noise extraction process. Once having extracted the noise portion, the second issue is how to add the noise signal to a given set of clean speech. We explain the process of adding noise in Section 2.2.

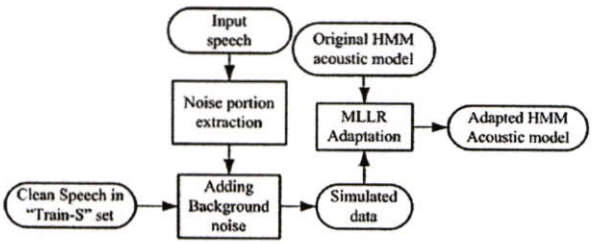


Figure 2 Simulated-data MLLR process (S-MLLR) for HMM adaptation.

2.1. Noise portion extraction

Simulated-data adaptation begins with identifying silence parts in the input signal. The silence parts are supposed to be background noise of the current input signal. For our task of isolated-word recognition, we assume that there are short periods of silence at the beginning and the end of the input signal. A hidden Markov model (HMM) is used to segment the input signal into speech and silence portions. Two noise extraction algorithms are evaluated in this paper. The first algorithm utilizes phone-based HMMs, where 64 HMMs of Thai phonemes including a special phoneme of silence "sil", as shown in Table 1, form an isolated-word recognizer. Figure 3(a) illustrates this HMM structure. The second noise extraction algorithm is based on speech/non-speech detection. Two states HMM, symbolized with speech and silence, are included in the module as shown in Figure 3(b). In both algorithms, noise portions are the signal regions labeled with silence "sil".

Table 1. *64 Thai phonemes.*

Type	IPA symbol
Initial consonant	p, t, c, k, ʔ, p^h, t^h, c^h, k^h, h, b, d, m, n, ŋ, l, r, f, s, h, w, j, pr, pl, p^hr, p^hl, tr, t^hr, kr, kl, kw, k^hr, k^hl, k^hw, fr
Vowel	i, i:, ɨ, ɨ:, u, u:, e, e:, ɜ, ɜ:, o, o:, æ, æ:, a, a:, ɔ:, i:a, ɨ:a, u:a
Final consonant	p', t', k', m', n', ŋ', s', w', j'
Silence	sil

In both algorithms, HMMs are composed of 16 Gaussian mixtures per state and were trained by the Baum–Welch algorithm. It is noted that the former algorithm gives better noise-region labeling performance with a drawback of computational demand comparing to the latter algorithm.

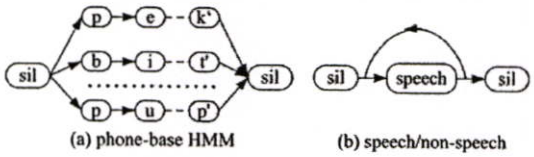


Figure 3 Two HMM architectures for noise extraction.

2.2. Adding background noise

Given noise portions extracted from the input signal, several issues need to be considered in adding background noise to the pre-recorded clean speech. First we concatenate noise portions extracted from the input signal. There are two noise-only regions in the input signal, at the beginning and at the end of the signal as shown in Figure 4. These noise portions are duplicated and concatenated so that the duration of noise signal is equal to the duration of clean-speech being added. It is noted that simply concatenating noise portions causes an unusual spectral change. However, in this paper, we discard spectral smoothing in order to save processing time.

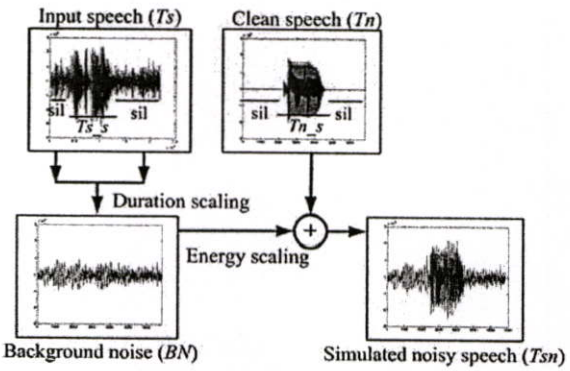


Figure 4 *Adding background noise.*

Second, simulated speech for adaptation should have a similar SNR to the input speech. However, estimation of SNR is not trivial and remains unsolved. In this work, we propose a simple way of signal-energy scaling. Let "Train-S" be a set of pre-recorded clean speech, of which correct transcriptions are known. We denote by *Tn* and *Ts* the current input signal and a clean speech in the Train-S set. *Tn_s* and *Ts_s* is the speech portion of *Tn* and *Ts* and *Tn_sil* and *Ts_sil* is the silence portion of *Tn* and *Ts*. First, a *scale_factor* is calculated as follows:

$$EngC = sum(abs(Ts\_s))/length(Ts\_s) \quad (1)$$

$$EngS = sum(abs(Tn\_s))/length(Tn\_s) \quad (2)$$

$$scale\_factor = EngC/EngS \quad (3)$$

where *EngC* and *EngS* is the energy of *Ts_s* and *Tn_s* respectively. Next, the background noise, *BN*, is multiplied by the *scale_factor* and added to *Ts*, resulting a simulated noisy-speech *Tsn* as shown in Equation 4.

$$Tsn = BN*scale\_factor + Ts \quad (4)$$

3. Experimental Setting

Our domain is isolated-word recognition using monophone-based HMMs representing 64 Thai phones. Each monophone HMM consists of 5 states and 16 Gaussian mixtures per state. 39-dimensional vectors (12 MFCC, 1 log-energy, and their first and second derivatives) are used as recognition features.

The baseline acoustic model (clean-speech model) is trained by phonetically-balanced utterances read by 16-male and 16-female speakers. The total number of training utterances is 32,000. For comparison, a multi-conditioned acoustic model [8], denoted as "MULTI" hereafter, is trained by speech data from both clean environment and noisy environments at various SNRs (5, 10, and 15 dB). In all experiments, clean-speech data are taken from NECTEC-ATR corpus [7].

3.1. Noise data for training

Eight kinds of noise from JEIDA [6], including crowded street, machinery factory, railway station, large air-condition, trunk road, elevator, exhibition in a booth, and ordinary train, and one large-size car noise from NOISEX-92 [9] are conducted. All noises from JEIDA and NOISEX-92 as well as the clean speech from NECTEC-ATR are preprocessed by reducing the sampling rate to 8 kHz. Noisy speech is prepared by adding the noise from JEIDA or NOISEX-92 to the clean speech of NECTEC-ATR at various SNRs (5, 10 and 15 dB).

3.2. Noise data for testing

Two test sets, "Test-1" and "Test-2", are used in evaluation. Test-1 contains 3,200 words uttered by 5 male speakers. Two noises, a computer room from JEIDA and an exhibition (NSTDA Annual Conference S&T in Thailand) recorded over four days in March 2005, are added to clean-speech utterances at three SNR levels: 0, 10 and 15 dB. This test set represents speech with different noise from the training set.

Test-2 contains utterances covering 76 Thai-province names recorded from 50 speakers over four days in another exhibition (ICT EXPO 2005 in Thailand). The environment is very noisy and consists of various kinds of noise. This set represents real noisy-speech with SNR ranged between 0 to 5 dB.

3.3. Simulated-data for adaptation (Train-S set)

In order to constitute the Train-S set for model adaptation, several criteria are used to select speakers and lexical words from the NECTEC-ATR corpus. For speaker selection, we limited to male-speakers with clear speech. Four speakers, denoted as "M1" to "M4", are selected.

For word selection, two criteria are considered. First, these words should be correctly recognized by our clean-speech model. Second, words should cover all 64 phones presented in the system. According to these criteria, 22 words out of 76 Thai-province names are chosen.

4. Experimental Results

Experiments are organized as follows. First, several parameters in the adaptation process are optimized. Among various parameters, we have found that the number of speakers and the size of adaptation data were influential. Section 4.1 gives the detail of optimization of these parameters. Given optimized parameters, Section 4.2 then compares our proposed system to conventional methods.

4.1. Effects of different speakers and data size in simulated-data adaptation

In the case that speech signals from only one speaker are included in the simulated-data set, increasing the size of adaptation data tends to produce a speaker-dependent acoustic model. Using the speaker-specific acoustic model may reduce recognition accuracy when evaluating with speech from various speakers. Therefore, in this subsection, five experiments on S-MLLR are performed to explore this phenomenon. Each of the first four experiments uses speech of only one speaker (M1 to M4). Randomly selected speech signals of M1 to M4 speakers are used in the last experiment, denoted as "MIX" case. Both noise extraction and MLLR adaptation in S-MLLR utilize phone-based HMMs trained by multi-conditioned data, i.e. speech data from clean and various noisy environments.

Figure 5 plots accuracies averaging on Test-1 and Test-2. According to results, the accuracy increases as a function of data size. We conclude that the phenomenon of speaker-mismatching is not significant even when a large number of speakers is evaluated (50 speakers in Test-2). The use of MIX case, where the adaptation set contains speech from various speakers, gives obviously better performance than the use of one specific-speaker model. This indicates that the acoustic model should be adapted with its speaker-independent property maintained.

Figure 6 plots processing time of the MIX case. All experiments are performed on an Intel Pentium IV 3.2 GHz CPU with 2 GB RAM. The graph shows that increasing the size of adaptation data yields higher processing time. An optimal number of words in the Train-S set we choose for the rest experiments is 22. We recall that these 22 words are the minimum set of words that covers all 64 phones.

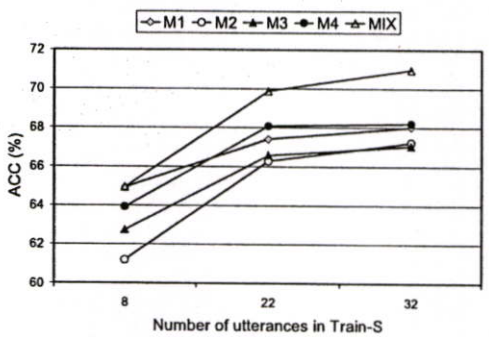


Figure 5 *Recognition accuracy of S-MLLR with different data sizes and selected speakers in Train-S.*

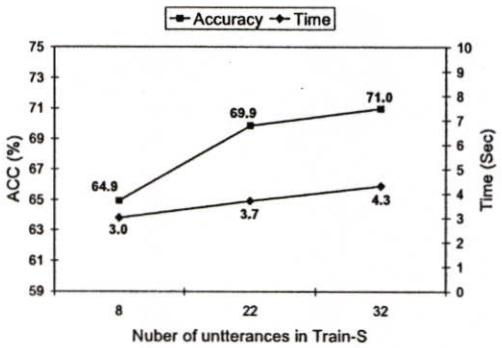


Figure 6 *Recognition accuracy and processing time of S-MLLR with different data sizes of Train-S.*

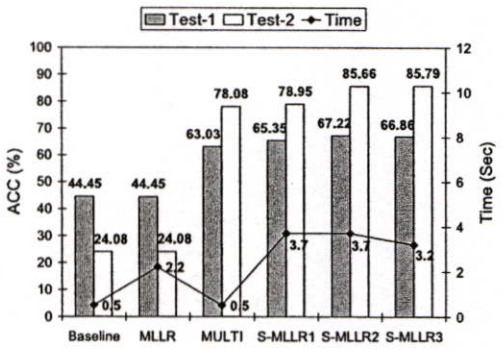


Figure 7 Comparison of Baseline, MLLR, MULTI, and S-MLLR systems evaluated by Test-1 and Test-2.

4.2. Comparison with conventional methods

In this subsection, several robust speech recognition techniques including our proposed model are experimentally compared. The first system is a baseline system without any implementation for robust speech recognition. The second system, denoted as "MLLR", exploits a conventional technique of online acoustic-model adaptation using MLLR as illustrated in Figure 1. The third system, called "MULTI", used a multi-conditioned acoustic model without any adaptation. The rest three systems are based on our proposed S-MLLR method. The forth system, called S-MLLR1, utilizes phone-based HMMs for noise extraction and online MLLR adaptation. The phone-based HMM is trained by clean-speech. The fifth system, S-MLLR2, is similar to the S-MLLR1 system except that the phone-based HMM was multi-conditioned. The last system, S-MLLR3, replaces the phone-based noise extraction module by a speech/non-speech detection module described in Section 2.1. The Train-S set contains 22-word utterances from M1 to M4 training speakers.

Figure 7 shows comparative results of five systems evaluated by Test-1 and Test-2. According to results, it is obvious that our proposed methods of S-MLLR outperform other conventional methods. Comparing among variations of S-MLLR, S-MLLR2 gives the highest accuracy but takes the longest processing time. S-MLLR3 is the fastest with the accuracy between the other two systems. We conclude that the phone-based HMM trained by multi-conditioned data in S-MLLR2 gives the best noise extraction result and hence causes the highest recognition accuracy. The speech/non-speech detection module in S-MLLR3 is much simpler than the phone-based HMM but achieves comparable performance.

5. Conclusions

This paper proposed a new approach of using simulated-data in MLLR acoustic-model adaptation. The approach solved limitations of the conventional online MLLR adaptation. The adaptation data was increased by conducting simulated-data created by adding a noise signal extracted from input signal to a pre-recorded set of clean speech. Since correct transcriptions of simulated-data are given, adaptation is more effective than using only the input speech with unknown transcription. Experiments showed that our proposed model achieved over 20% improvement of recognition accuracy comparing to the conventional approach of online MLLR adaptation.

Future works include an evaluation of the proposed model by a larger set of speech from various real environments. Further improvement of noise extraction and noise addition in simulated-data adaptation will be investigated. Another interesting issue is that selection of clean speech from different speakers should affect the recognition result. Even if we have found that maintaining speaker-independency in adaptation gives good recognition result, selecting a set of speakers that best matches to the input speech might give better performance. This issue will also be explored.

6. References

[1] Gales, M.J.F., "Model-based techniques for noise robust speech recognition", PhD. Thesis, University of Cambridge, 1995.

[2] Gauvain, J.L., and Lee, C.H., "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains", IEEE Trans. Speech Audio Proc., 2:291–298, 1994.

[3] Leggetter, C.J. and Woodland, P.C., "Maximum likelihood linear regression for speaker adaptation of continuous density HMMs", Comput. Speech Lang., 9:171–186, 1995.

[4] Gales, M.J.F. and Woodland, P.C., "Mean and variance adaptation within the MLLR framework", Comput. Speech Lang., 10(3):249–264, 1996.

[5] Zhao, Y., Wang, L., Chu, M., Soong, F.K. and Cao, Z., "Refining phoneme segmentations using speaker-adaptive context dependent boundary models", Proc. of INTERSPEECH 2005, pp.2557-2560, 2005.

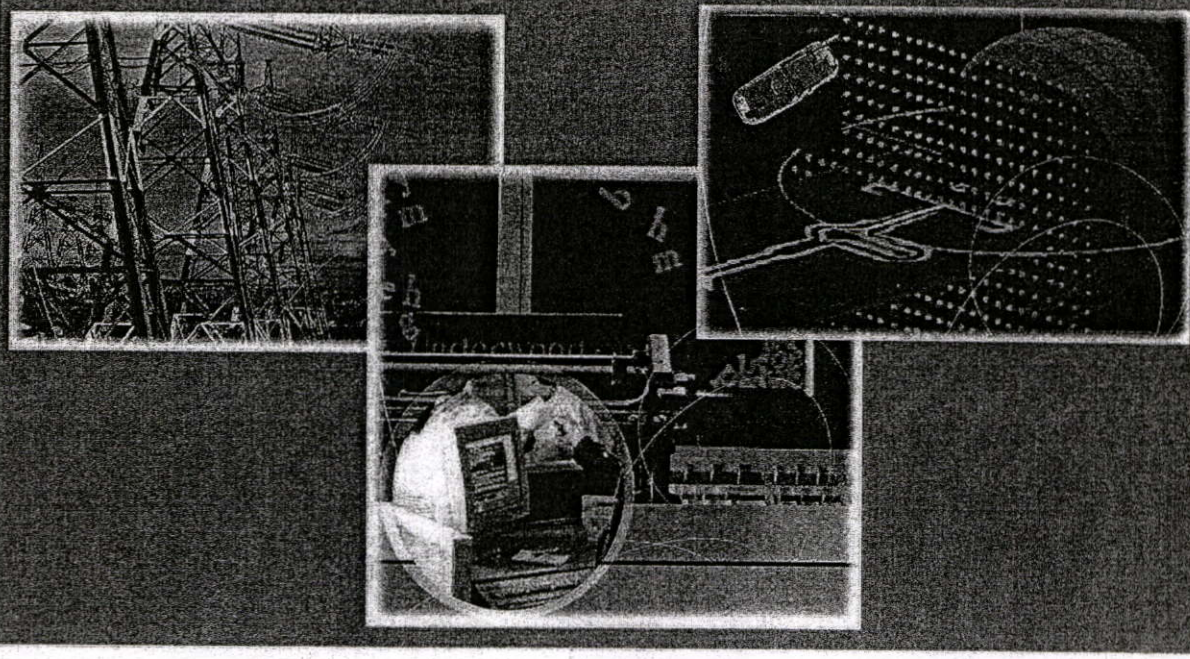
[6] http://www.milab.is.tsukuba.ac.jp/corpus/noise db.html

[7] Kasuriya, S., Somlertlamvanich, V., Cotsomrong, P., Jitsuhiro, T., Kikui, G. and Sagisaka, Y., "NECTEC-ATR Thai speech corpus", Proc. of Oriental COCOSDA 2003, pp.105-111, 2003.

[8] Nakamura, S., Yamamoto, K., Takeda, K., Kuroiwa, S., Kitaoka, N., Yamada, T., Mizumachi, M., Nishiura, T., Fujimoto, M., Saso, A., Endo, T., "Data collection and evaluation of AURORA-2 JAPANESE corpus", Proc. of ASRU 2003, pp.619-623, 2003.

[9] http://www.speech.cs.cmu.edu/ comp. speech/ Section1/ Data/noisex.html

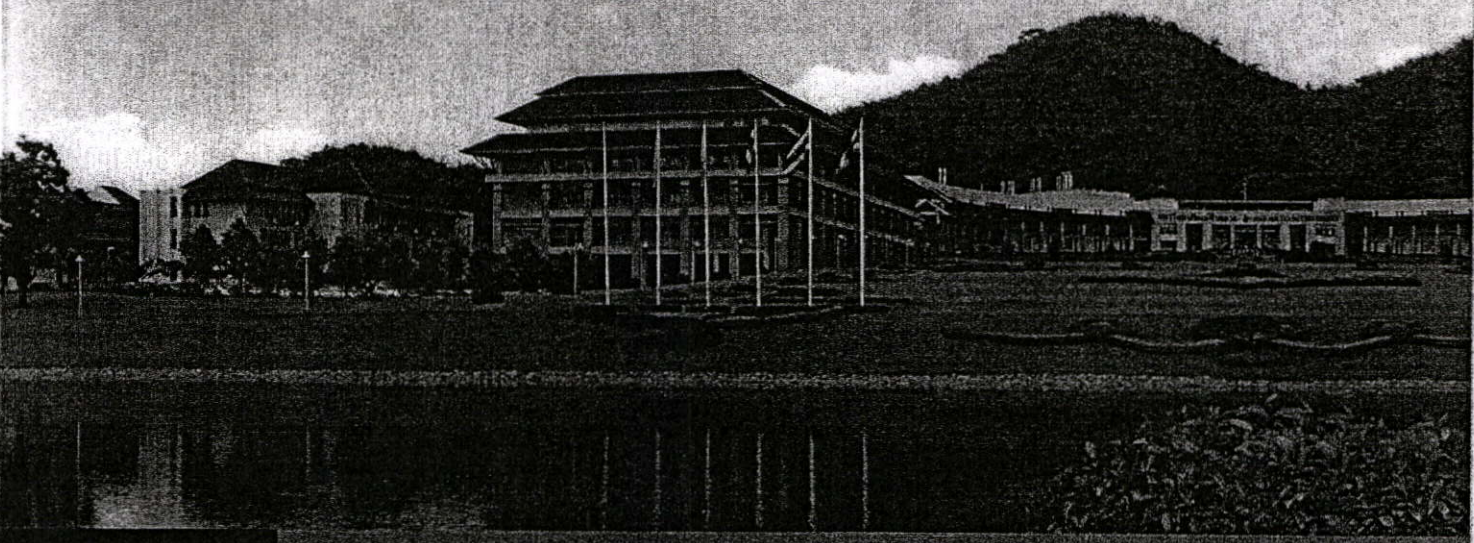
VOLUME 2



ECTI-CON 2007

Mae Fah Luang University, Chiang Rai, Thailand
May 9-12, 2007

VOLUME 2
- *Communication Systems*
- *Signal Processing*
- *Computer and Information*



ECTI Association | IEEE THAILAND SECTION | NECTEC a member of NSTDA | WD Western Digital

Combined simulated data adaptation and piecewise linear transformation for robust speech recognition

Nattanun Thatphithakkul[1], Boontee Kruatrachue[1], Chai Wutiwiwatchai[2], Sanparith Marukatat[2] and Vataya Boonpiam[2]
[1]King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand
[2]National Electronics and Computer Technology Center, Pathumthani, 12120, Thailand

Abstract- **This paper proposes a combination of simulated data adaptation and piecewise linear transformation (PLT) for robust continuous speech recognition. The original PLT selects an appropriate acoustic model using tree-structured HMMs and the acoustic model is adapted by the input speech in an unsupervised scheme. This adaptation can improve the acoustic model if the input speech is long enough and is correctly transcribed in the adaptation process. Indeed, an incorrect transcription can drastically degrade the acoustic model. Our proposed method increases the size of adaptation data by adding noise portions from the input speech to a set of pre-recorded clean speech, of which correct transcriptions are known. We investigate various configurations of the proposed method. Evaluations are performed with additive noisy continuous speech. The experimental results show that the proposed system reaches higher recognition rates than MLLR and PLT.**

I. Introduction

It is commonly known that a speech recognition system trained by speech in a clean or nearly clean environment cannot achieve good performance when working in noisy environment. Research on robust speech recognition is then necessary. Gales [1] has classified the techniques of robust speech recognition into 4 approaches: 1) robust feature approach, 2) clean speech estimation approach, 3) robust model approach, and 4) combination of three previous approaches. This paper focuses on the model-based approach, which has achieved good recognition results [1]. The model-based approach aims to create or to adapt the acoustic model in specific environments. Several techniques of model adaptation have been proposed such as maximum likelihood linear regression (MLLR) [2], maximum a posteriori (MAP) [3], parallel model combination (PMC) [4], and piecewise linear transformation (PLT) [5].

In this work, we are interested in the adaptation technique of piecewise linear transformation with model selection based tree-structured cluster (MSTC) [5, 6], proposed by Zhang, Sugimura and Furui [5]. This technique is based on unsupervised acoustic model adaptation using the incoming speech. It was proven to be efficient in both the recognition accuracy and computational cost. However, a problem of the PLT is that the acoustic model may not be well adapted if the incoming speech is very short. Moreover, in the unsupervised adaptation, a wrong transcription strongly degrades the adapted model. These 2 problems can be solved by a recently proposed technique called simulated-data adaptation [7]. Indeed, the simulated-data adaptation process aims to increase the data used in adaptation by adding the background noise extracted from the current input signal to existing clean speech. Since correct transcriptions of the clean speech are known, using the simulated-data is supervised adaptation.

This paper presents a new adaptation scheme which applies simulated-data adaptation to piecewise linear transformation (called S-PLT hereafter). S-PLT combines the strength of both techniques; PLT allows selecting a good initial acoustic model while simulated-data adaptation enlarges the size of the adaptation data. S-PLT uses the MLLR adaptation technique. The proposed method is compared with other model adaptation techniques.

The proposed system was evaluated in 3 groups of environments. The first group contained a clean environment and 9 types of noisy environments that have been trained in the system. The second group contained isolated-words. The third group contained continuous-speech. Noisy speech of second and third group were prepared by adding noise signals from an exhibition in Thailand (NAC 2005) to the clean speech taken from NECTEC-ATR Thai speech corpus [8] at various SNR (0, 10, 15 dB).

We will review the PLT in the next section. Section 3 reviews the simulated-data adaptation. Section 4 describes our proposed models. Section 5 describes the data used in these works and experimental results are reported in Section 6. Section 7 concludes this paper with future works

II. Piecewise Linear Transformation (PLT)

The PLT method is composed of 2 main steps namely MSTC [5, 6] model selection and MLLR adaptation [2]. In the training step, a wide variety of noise data were collected and classified into noise clusters using hierarchical clustering. The root node includes all noises and all SNR conditions and each leaf node consists of only one noise at one SNR condition. Intermediate nodes in this tree contain similar noises from different environments and from different SNR. A noise-added speech HMM is constructed for each node. Using this tree structure, an unknown noise environment which is similar to combination of known environments should be handled by

non-leaf HMM. The resulting tree-structured HMM allows representing both known and some unknown noises. In the recognition phase, the noise-cluster HMM which is best fitted to the input speech was selected and adapted by the input speech itself using the MLLR method.

III. SIMULATED DATA ADAPTATION

A. *Noise portion extraction*

Simulated-data adaptation begins with identifying silence parts which are supposed to be background noise of the current input signal. We assume that there are short periods of silence at the beginning and the end of the input signal. A phone-based HMM is used to segment the input signal into speech and silence portions. In this work, we use 64 HMMs of Thai phonemes including a special phoneme of silence "sil", form a speech recognizer. Fig. 1 illustrates this HMM structure. Noise portions are the signal regions labeled with silence "sil".

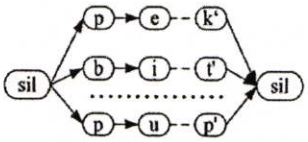


Figure 1 HMM architecture for noise extraction.

B. *Background- noise addition*

Given noise portions extracted from the input signal, several issues need to be considered in adding background noise to the pre-recorded clean speech. First we concatenate noise portions extracted from the input signal. There are two noise-only regions in the input signal, at the beginning and at the end of the signal as shown in Fig. 2. These noise portions are duplicated and concatenated so that the duration of noise signal is equal to the duration of clean-speech being added.

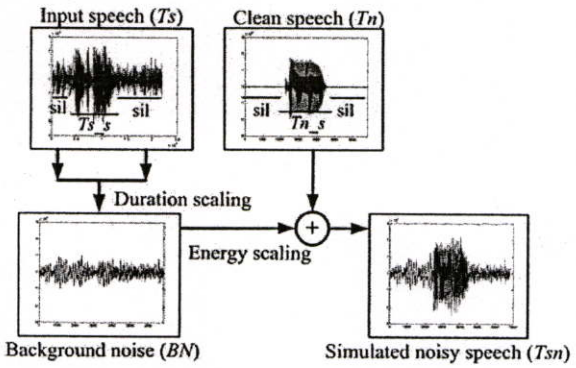


Figure 2 Background-noise addition.

Second, simulated speech for adaptation should have a similar SNR to the input speech. However, estimation of SNR is not trivial and remains unsolved. The simulated-data adaptation relies on a simple signal-energy scaling. Let "Train-S" be a set of pre-recorded clean speech, of which correct transcriptions are known. We denote by Tn and Ts the current input signal and a clean speech in the Train-S set. Tn_s and Ts_s is the speech portion of Tn and Ts and Tn_sil and Ts_sil is the silence portion of Tn and Ts. First, a scale_factor is calculated as follows:

$$EngS = sum(abs(Ts\_s))/length(Ts\_s) \quad (1)$$

$$EngC = sum(abs(Tn\_s))/length(Tn\_s) \quad (2)$$

$$scale\_factor = EngC/EngS \quad (3)$$

where |T| denotes the energy of a signal T. EngS and EngC is respectively the energy averaged over all Ts_si and Tn_si extracted from Ts and Tn. Next, the background noise, BN, given by the noise-portion extraction step, is multiplied by the scale_factor and added to Tn, resulting in a simulated noisy-speech Tsn as shown in Equation (4). The signal Tsn is then included in the adaptation set.

$$Tsn = BN*scale\_factor + Ts \quad (4)$$

C. *Adaptation-data preparation*

Adaptation data set is a key element in our robust speech recognition system. Two variations of the proposed method are considered. The first one uses only the pre-recorded clean speech with known transcription. This is called supervised adaptation. The second variation includes the input speech in the adaptation set with its label transcribed automatically. This second variation is called semi-supervised adaptation.

Beside these two variations, other important parameters including the selection of speakers and lexicon in the Train-S data are required to adjust. Basically, we selected speakers and words that can be correctly recognized by our clean speech model. Moreover, selected words should cover all phones presented in the system.

IV. COMBINED SIMULATED DATA ADAPTATION AND PLT

Combined simulated-data adaptation [7] and PLT [5], called S-PLT is shown in Fig. 3.

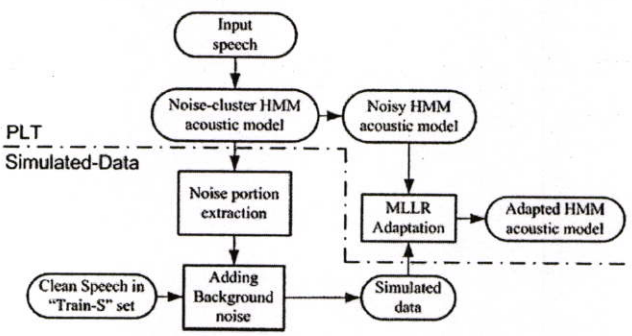


Figure 3 Combine simulated-data adaptation and PLT process (S-PLT) for HMM adaptation.

The basic idea is to adapt the model selected from the tree-structured cluster using the adaptation data prepared by the simulated-data process either in supervised or semi-supervised

mode. The adapted HMM is then used to recognize the incoming signal.

V. Experimental Setting

Our domain is continuous speech recognition using monophone-based HMMs representing 64 Thai phones. The aim of using context-independent phone model is the flexibility to add any word in the recognition lexicon. Each monophone HMM consists of 5 states and 16 Gaussian mixture per state. 12 MFCC, 1 log-energy, and their first and second derivatives are used as recognition features. The clean model is trained by phonetically-balanced utterances read by 32 speakers. The total number of training utterances is 32,000. In all experiments, clean-speech data are taken from the NECTEC-ATR corpus [8].

D. Noise data for training

Eight kinds of noise from JEIDA, including crowded street, machinery factory, railway station, large air-condition, trunk road, elevator, exhibition in a booth, and ordinary train, and one large-size car noise from NOISEX-92 are considered. All noises and the clean speech are preprocessed by resampling to 8 kHz. Noisy speech is prepared by adding the noise from JEIDA or NOISEX-92 to the clean speech of the NECTEC-ATR corpus at various SNRs (5, 10 and 15 dB).

E. Noise data for testing

Two test sets, "Test-1" and "Test-2", are used in evaluation. Test-1 contains 3,200 utterances from 640 words uttered by 5 male speakers. Test-2 contains 1,950 utterances from 390 sentences uttered by 5 male speakers.

An exhibition (NSTDA Annual Conference S&T in Thailand) recorded over four days in March 2005, are added to clean-speech utterances at three SNR levels: 0, 10 and 15 dB. This test set represents speech with different noise from the training set.

F. Simulated-data for adaptation (Train-S set)

In order to constitute the Train-S set for model adaptation, speakers and lexicon are selected from the NECTEC-ATR corpus. For speaker selection, we limited to male-speakers with clear speech and randomly selected speech signals of four speakers are used in the experiment, denoted as "MIX" speaker.

For word selection, the selected lexicon should cover all 64 phones presented in the system. According to these criteria, two sets of train-s are prepared. The first set (Train-S1) contains 22 words and the second set (Train-S2) contains 10 sentences, both are the minimum set of isolated-words and continuous speech that cover 64 phones used in the recognizer

VI. Experimental Results

In this section we investigate various settings of S-PLT and compare them to other robust speech recognition techniques. Subsection A evaluates different settings of our system. Subsection B compares our proposed system to conventional methods.

A. Effects of type MLLR adaptation and type Train-S in simulated-data adaptation

In our experiments the acoustic model before adaptation can be either a clean-speech model or the model selected by the MSTC algorithm. For each model, we test adaptation in both supervised and semi-supervised modes. Semi-supervision means that the input speech is also included in the adaptation data. Table I defines four systems varying on the supervision made of adaptation and the initial acoustic model

Table I Definition of comparative systems

	Initial AM	
	Clean	MSTC
supervised	S-MLLR1	S-PLT1
Semi supervised	S-MLLR2	S-PLT2

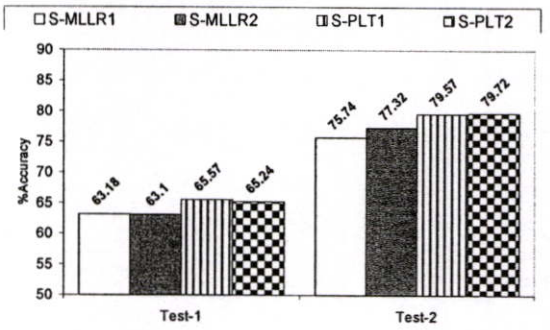


Figure 4 Recognition accuracies of S-MLLR and S-PLT evaluated by Test-1 and Test-2.

Fig. 4 shows the experimental results from various settings using the Train-S1 set in adaptation. According to these results, using the MSTC algorithm to initialize the acoustic model gives better accuracy than using the clean model. As expected, since the MSTC-selected model is closer to the real environment than the clean model and hence give the higher performance. For semi-supervised adaptation, results show that it rather suit for continuous speech (Test-2) than for isolated word (Test-1). This may be due to the fact that for the isolated word case, the signal is too short and no useful information can be automatically and reliably extracted. It is then better to rely only on assured transcription as in supervised adaptation. On the other hand, for continuous speech the signal is long enough to compensate the incertitude concerning the transcription thus make it useful for model adaptation.

Next, we investigate the use of continuous speech as adaptation data. For this propose, we ran the experiment on the Test-2 set with the Train-S2 adaptation data. Fig. 5 summarizes the results from this experiment. We can see that using the Train-S2 outperforms the use of Train-S1 in every case. This is due to the increasing in the size of the adaptation data. However, the computational time unavoidably increases as a drawback.

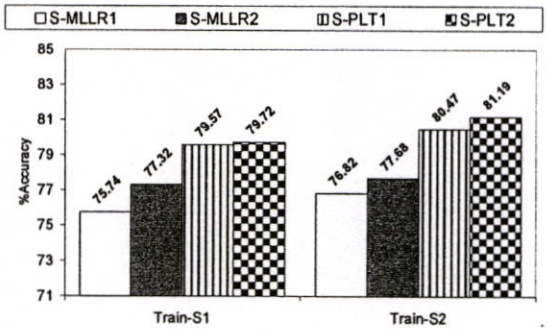


Figure 5 Recognition accuracies of S-MLLR and S-PLT with different Train-S evaluated by Test-1.

B. Comparison with conventional methods

In this subsection, several robust speech recognition techniques including ours are compared. The first one is a baseline system without any implementation for robust speech recognition. The second system, denoted as "MLLR", uses online acoustic-model adaptation based on the well-known MLLR approach. The third system, called "MSTC", uses model selection based on the tree-structured cluster model without any adaptation. The forth system called "PLT" uses the PLT method (see section III). The fifth and the sixth systems are S-MLLR2 and S-PLT2. The Train-S2 is used in adaptation.

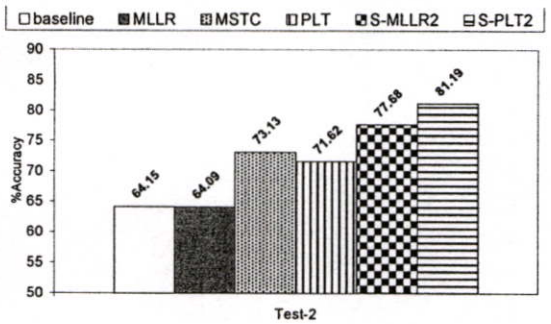


Figure 6 Comparison of baseline, MLLR, MSTC, PLT, S-MLLR2 and S-PLT2 systems evaluated by Test-2.

Fig. 6 shows comparative results. According to the results, it is obvious that our proposed methods, S-MLLR2 and S-PLT2, outperform other conventional methods. In the case of S-MLLR2 we gain approximately 14% improvement of recognition accuracies over the conventional MLLR model. S-PLT2 achieves approximately 10% improvement of recognition accuracies over the PLT-based method. Moreover, the PLT's accuracy is lower than that of the MSTC but both systems perform worse than the S-PLT2. We believe that the continuous speech signal is not long enough, so that purely unsupervised adaptation can always improve the initial acoustic model. This underlines the benefit of using clean speech with known transcription in simulated-data adaptation.

VII. Conclusions and Future Work

This paper proposed combined simulated-data adaptation and PLT in acoustic-model adaptation. The approach solved limitations of the conventional unsupervised MLLR adaptation in the PLT method. The adaptation data were increased by adding a noise-signal extracted from the input signal to a pre-recorded set of clean speech. Since correct transcriptions of simulated-data are given, adaptation is more effective than using only the input speech with unknown transcription. Experiments showed that our technique achieved over 10% improvement of recognition accuracy comparing to the original PLT.

Future works include an evaluation of the proposed model by a larger set of speech from various real environments. Further improvement of noise extraction and noise addition in simulated-data adaptation will also be investigated.

References

[1] M.J.F. Gales, "Model-based techniques for noise robust speech recognition," PhD thesis University of Cambridge, 1995.

[2] M.J.F. Gales and P.C. Woodland, "Mean and variance adaptation with in the MLLR framework," Comput. Speech Lang. vol.10, pp. 249–264, 1996.

[3] J.L Gauvain and C.H. Lee, "Maximuma posteriori estimation for multivariate Gaussian mixture observations of Markov chains," IEEE Trans. Speech Audio Process. vol. 2, pp. 291–298, 1994.

[4] M.J.F Gales and S. Young, "An improved approach to the hidden Markov model decomposition of speech and noise," Proc. of ICASSP, pp. 233–236, 1992.

[5] Z.P. Zhang, T. Sugimura and S. Furui, "Tree-structured clustering methods for piecewise linear transformation-based noise adaptation," IEICE Trans. Inf. and Syst. vol. 9, pp. 2168-2176, 2005.

[6] T. Kosaka, S. Matsunaga, and S. Sagayama, "Speaker-independent speech recognition based on tree-structured speaker clustering," Computer Speech Language, vol. 10, pp. 55–74, 1996.

[7] T. Nattanun, K. Boontee, W. Chai, M. Sanparith and B. Vataya, "A simulated-data adaptation technique for robust speech recognition," Proc. of INTERSPEECH, pp. 777–780, 2006.

[8] S. Kasuriya, V. Sornlertlamvanich, P. Cotsomrong, T. Jitsuhiro, G. Kikui and Y. Sagisaka, "NECTEC-ATR Thai speech corpus," Proc. of COCOSDA, pp. 105-111, 2003.

ภาคผนวก ข

อภิธานศัพท์ อังกฤษ-ไทย

Acoustic Model	โมเดลเสียงพูด
Additive Noise	เสียงรบกวนแบบบวก
Amplitude	แอมปลิจูด
AURORA-2J	ออโรราทูเจ
Background Noise	เสียงรบกวนพื้นหลัง
Background Noise Addition	การบวกเสียงรบกวนพื้นหลัง
Backtracking	เส้นทางเดินย้อนกลับ
Band Pass Filter	วงจรกรองแบบผ่านแถบความถี่
Bias	ไบแอส
Binary Tree Structure	โครงสร้างต้นไม้แบบไบนารี
Cepstral Domain	โดเมนเซปสตรอล
Channel Noise	เสียงรบกวนจากช่องสัญญาณ
Clean Speech Model	โมเดลเสียงพูดสะอาด
Clean Speech	เสียงพูดสะอาด
Cofactors	โคแฟคเตอร์
Continuous Distribution	การกระจายแบบต่อเนื่อง
Differentiation	การหาค่าอนุพันธ์
Dimension	มิติ
Direct Search	การค้นหาแบบตรง
Discrete Cosine Transform	การแปลงดิสครีตโคไซต์
Duration Scaling	การสเกลความยาว
Dynamic Programming	โปรแกรมแบบพลวัต
Dynamic Range	ช่วงพิสัยพลวัต
Energy Scaling	การสเกลพลังงาน
Energy	พลังงาน
Fast Fourier Transform	การแปลงฟูริเยร์แบบเร็ว
Feature	ลักษณะสำคัญ
Feature Extraction	การดึงลักษณะสำคัญ

First Order Digital Filter	ตัวกรองเชิงเลขอันดับหนึ่ง
Frame	เฟรม
Gaussian Mixture	เกาส์เซียนมิกเจอร์
Hamming Window	หน้าต่างแบบแฮมมิ่ง
Hidden Markov Model Interpolation	การประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดล
Hidden Markov Model	ฮิดเดนมาร์คอฟโมเดล
Induction	การเหนี่ยวนำ
International Phonetic Alphabet	สัทอักษร
Interpolation	การประมาณค่าในช่วง
Isolated Word	คำโดด
Kernel Principal Component Analysis	การวิเคราะห์องค์ประกอบหลักแบบเคอร์เนล
Language Model	โมเดลทางภาษา
Linear Spectral Domain	โดเมนเสปคตรอลเชิงเส้น
Linear Transformation	การแปลงเชิงเส้น
Logarithm	ลอการิทึม
Maximum a Posteriori	การพิจารณาจากค่าประสบณ์การสูงสุด
Maximum Likelihood Linear Regression	การถดถอยแบบเชิงเส้นตามความเป็นไปได้สูงสุด
Mel Filter Bank	วงจรกรองความถี่เมล
Mel Frequency Cepstral Coefficients	สัมประสิทธิ์เซปสตรอลบนความถี่เมล
Model Adaptation	การปรับโมเดล
Model Composition and Decomposition	การแยกและการรวมโมเดล
Model Parameter Adaptation	การปรับพารามิเตอร์โมเดล
Model Selection based Tree-Structured Cluster	การเลือกโมเดลที่มีการจัดกลุ่มแบบโครงสร้างต้นไม้
Model Selection	การเลือกโมเดล
Multi-condition Training	การฝึกสอนแบบหลากสภาวะ
Neural Network	โครงข่ายประสาทเทียม
Node	โหนด
Noise Classification	การจำแนกชนิดของเสียงรบกวน

Noise Cluster HMM Interpolation	การประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดลที่มีการจัดกลุ่มเสียงรบกวน
Noise Portion Extraction	การดึงส่วนเสียงรบกวน
Noise	เสียงรบกวน
Noisy Speech	เสียงพูดที่มีเสียงรบกวน
Normalized Logarithm Spectrums	นอร์มอไลซ์ลอการิทึมสเปกตรัม
Observation	ค่าสังเกต
Offline Model Adaptation	การปรับโมเดลแบบออฟไลน์
Online Model Adaptation	การปรับโมเดลแบบออนไลน์
Parallel Model Combination	การรวมโมเดลขนาน
Parameter	พารามิเตอร์
Phoneme-based	การอิงกับหน่วยเสียง
Phoneme	หน่วยเสียง
Piecewise-linear Transformation	การแปลงเชิงเส้นแบบแบ่งส่วน
Pre-emphasis	การเน้นล่วงหน้า
Principal Component Analysis	การวิเคราะห์องค์ประกอบหลัก
Priori Knowledge	ความรู้เบื้องต้น
Pronunciation Lexicon	การออกเสียงในพจนานุกรม
Real Noise	เสียงรบกวนจากสภาพแวดล้อมจริง
Recognition Result Selection	การเลือกผลการรู้จำเสียงพูด
Retrain	การฝึกสอนซ้ำ
Robust Speech Feature	ลักษณะสำคัญของเสียงพูดแบบคงทน
Robust Speech Recognition System	ระบบการรู้จำเสียงพูดแบบคงทน
Separation	การแยก
Signal to Noise Ratio	อัตราส่วนสัญญาณต่อสัญญาณรบกวน
Silence	เสียงเงียบ
Simulated-data	ข้อมูลจำลอง
Simulated-data Adaptation	การปรับโมเดลด้วยข้อมูลจำลอง
Speaker Adaptation	การปรับผู้พูด
Spectral Smoothing	การทำให้เสปคตรอลเรียบ
Speech Enhancement	การปรับปรุงเสียงพูด
Speech Recognition System	ระบบการรู้จำเสียงพูด

Speech Synthesis System	ระบบการสังเคราะห์เสียงพูด
Speech/Non-speech Detection	การหาส่วนที่เป็นเสียงพูดและส่วนที่ไม่เป็นเสียงพูด
State Transition Probability	ความน่าจะเป็นในการเปลี่ยนแปลงสเตท
State	สเตท
Stationary	ความคงที่
Support Vector Machines	ซัพพอร์ทเวกเตอร์แมชชีน
Text to Speech System	ระบบการแปลงข้อความเป็นเสียงพูด
Tone	วรรณยุกต์
Training Set	ชุดฝึกสอน
Transcript	คำอ่าน
Tree Structure Search	การค้นหาแบบโครงสร้างต้นไม้
Weight	ค่าถ่วงน้ำหนัก
Window Function	ฟังก์ชันหน้าต่าง

ภาคผนวก ค
อภิธานศัพท์ไทย-อังกฤษ

การกระจายแบบต่อเนื่อง	Continuous Distribution
การค้นหาแบบโครงสร้างต้นไม้	Tree Structure Search
การค้นหาแบบตรง	Direct Search
การจำแนกชนิดของเสียงรบกวน	Noise Classification
การดึงลักษณะสำคัญ	Feature Extraction
การดึงส่วนเสียงรบกวน	Noise Portion Extraction
การถดถอยแบบเชิงเส้นตามความเป็นไปได้สูงสุด	Maximum Likelihood Linear Regression
การทำให้เสปคตรอลเรียบ	Spectral Smoothing
การเน้นล่วงหน้า	Pre-emphasis
การบวกเสียงรบกวนพื้นหลัง	Background Noise Addition
การประมาณค่าในช่วง	Interpolation
การประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดล	Hidden Markov Model Interpolation
การประมาณค่าในช่วงของฮิดเดนมาร์คอฟโมเดลที่มีการจัดกลุ่มเสียงรบกวน	Noise Cluster HMM Interpolation
การปรับปรุงเสียงพูด	Speech Enhancement
การปรับผู้พูด	Speaker Adaptation
การปรับพารามิเตอร์โมเดล	Model Parameter Adaptation
การปรับโมเดล	Model Adaptation
การปรับโมเดลด้วยข้อมูลจำลอง	Simulated-data Adaptation
การปรับโมเดลแบบออนไลน์	Online Model Adaptation
การปรับโมเดลแบบออฟไลน์	Offline Model Adaptation
การแปลงเชิงเส้น	Linear Transformation
การแปลงเชิงเส้นแบบแบ่งส่วน	Piecewise-linear Transformation
การแปลงดิสครีตโคไซต์	Discrete Cosine Transform
การแปลงฟูริเยร์แบบเร็ว	Fast Fourier Transform
การฝึกสอนซ้ำ	Retrain
การฝึกสอนแบบหลากสภาวะ	Multi-condition Training
การพิจารณาจากค่าประสบณ์การสูงสุด	Maximum a Posteriori

การแยก	Separation
การแยกและการรวมโมเดล	Model Composition and Decomposition
การรวมโมเดลขนาน	Parallel Model Combination
การเลือกผลการรู้จำเสียงพูด	Recognition Result Selection
การเลือกโมเดล	Model Selection
การเลือกโมเดลที่มีการจัดกลุ่มแบบโครงสร้างต้นไม้	Model Selection based Tree-Structured Cluster
การวิเคราะห์องค์ประกอบหลัก	Principal Component Analysis
การวิเคราะห์องค์ประกอบหลักแบบเคอร์เนล	Kernel Principal Component Analysis
การสเกลความยาว	Duration Scaling
การสเกลพลังงาน	Energy Scaling
การหาค่าอนุพันธ์	Differentiation
การหาส่วนที่เป็นเสียงพูดและส่วนที่ไม่เป็นเสียงพูด	Speech/Non-speech Detection
การเหนี่ยวนำ	Induction
การออกเสียงในพจนานุกรม	Pronunciation Lexicon
การอิงกับหน่วยเสียง	Phoneme-based
เกาส์เซียนมิกเจอร์	Gaussian Mixture
ข้อมูลจำลอง	Simulated-data
ความคงที่	Stationary
ความน่าจะเป็นในการเปลี่ยนแปลงสเตท	State Transition Probability
ความรู้เบื้องต้น	Priori Knowledge
ค่าถ่วงน้ำหนัก	Weight
ค่าสังเกต	Observation
คำโดด	Isolated Word
คำอ่าน	Transcript
โคแฟคเตอร์	Cofactors
โครงข่ายประสาทเทียม	Neural Network
โครงสร้างต้นไม้แบบไบนารี	Binary Tree Structure
ช่วงพิสัยพลวัต	Dynamic Range
ชุดฝึกสอน	Training Set
ซัพพอร์ทเวกเตอร์แมชชีน	Support Vector Machines
โดเมนเซปสตรอล	Cepstral Domain

โดเมนเสปคตรอลเชิงเส้น	Linear Spectral Domain
ตัวกรองเชิงเลขอันดับหนึ่ง	First Order Digital Filter
นอร์มอไลซ์ลอการิทึมสเปกตรัม	Normalized Logarithm Spectrums
ไบแอส	Bias
โปรแกรมแบบพลวัต	Dynamic Programming
พลังงาน	Energy
พารามิเตอร์	Parameter
ฟังก์ชันหน้าต่าง	Window Function
เฟรม	Frame
มิติ	Dimension
โมเดลทางภาษา	Language Model
โมเดลเสียงพูด	Acoustic Model
โมเดลเสียงพูดสะอาด	Clean Speech Model
ระบบการแปลงข้อความเป็นเสียงพูด	Text to Speech System
ระบบการรู้จำเสียงพูด	Speech Recognition System
ระบบการรู้จำเสียงพูดแบบคงทน	Robust Speech Recognition System
ระบบการสังเคราะห์เสียงพูด	Speech Synthesis System
ลอการิทึม	Logarithm
ลักษณะสำคัญ	Feature
ลักษณะสำคัญของเสียงพูดแบบคงทน	Robust Speech Feature
วงจรกรองความถี่เมล	Mel Filter Bank
วงจรกรองแบบผ่านแถบความถี่	Band Pass Filter
วรรณยุกต์	Tone
สเตท	State
สัทอักษร	International Phonetic Alphabet
สัมประสิทธิ์เซปสตรอลบนความถี่เมล	Mel Frequency Cepstral Coefficients
เส้นทางเดินย้อนกลับ	Backtracking
เสียงเงียบ	Silence
เสียงพูดที่มีเสียงรบกวน	Noisy Speech
เสียงพูดสะอาด	Clean Speech
เสียงรบกวน	Noise
เสียงรบกวนจากช่องสัญญาณ	Channel Noise

เสียงรบกวนจากสภาพแวดล้อมจริง	Real Noise
เสียงรบกวนแบบบวก	Additive Noise
เสียงรบกวนพื้นหลัง	Background Noise
หน่วยเสียง	Phoneme
หน้าต่างแบบแฮมมิ่ง	Hamming Window
โหนด	Node
ออโรราทูเจ	AURORA-2J
อัตราส่วนสัญญาณต่อสัญญาณรบกวน	Signal to Noise Ratio
แอมปลิจูด	Amplitude
ฮิดเดนมาร์คอฟโมเดล	Hidden Markov Model

ประวัติผู้เขียน

ชื่อ-นามสกุล นายณัฐนันท์ ทัดพิทักษ์กุล

วัน เดือน ปีเกิด 20 พฤศจิกายน 2521 ที่กรุงเทพฯ

ที่อยู่ 4/844 หมู่บ้านสหกรณ์ ถนนเสรีไทย
แขวงคลองกุ่ม เขตบึงกุ่ม กรุงเทพฯ 10240 โทร.0-2379-7194

ประวัติการศึกษา 2543 วิศวกรรมศาสตรบัณฑิต สาขาวิชาโทรคมนาคม
มหาวิทยาลัยเทคโนโลยีสุรนารี
2545 วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาไฟฟ้า
มหาวิทยาลัยเทคโนโลยีสุรนารี

ความชำนาญเฉพาะด้าน 1.) การรู้จำเสียงพูด
2.) การสังเคราะห์เสียงพูด

ประสบการณ์การทำงานและผลงานวิจัย

พ.ศ.2545-ปัจจุบัน ตำแหน่งนักวิจัยสังกัดห้องปฏิบัติการมนุษยภาษา
หัวหน้ากลุ่มเทคโนโลยีเสียงพูด
ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ
- โปรแกรมรู้จำเสียงพูด i-Speech-W เวอร์ชั่น1.0
- โปรแกรมรู้จำเสียงพูด i-Speech-W เวอร์ชั่น 1.5
- โปรแกรมสังเคราะห์เสียงพูดภาษาไทย VAJA เวอร์ชั่น 5.0