อัลกอริทึมสำหรับการแบ่งนับเวกเตอร์โดยใช้การประมาณความหนาแน่น

นายอรรคพล นงนุช

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

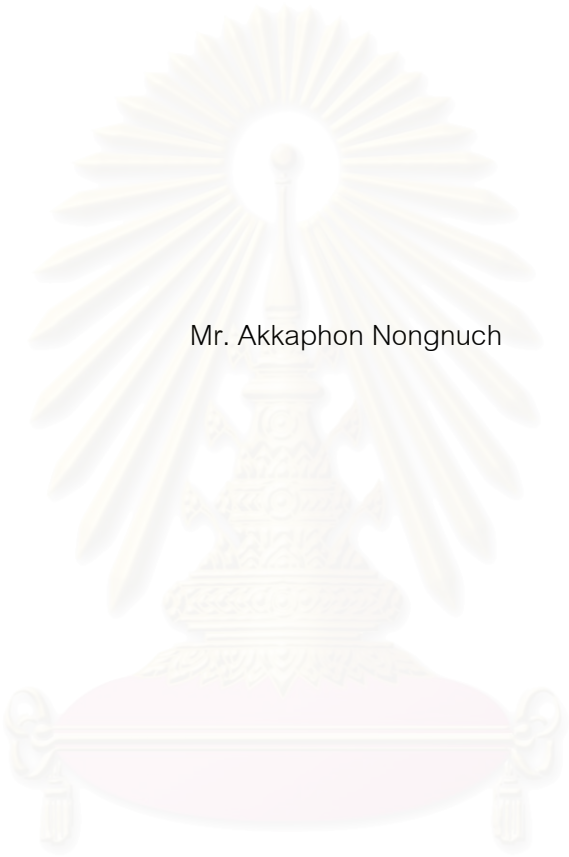
วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2548

ISBN 974-17-4283-5

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

AN ALGORITHM FOR VECTOR QUANTIZATION USING DENSITY ESTIMATION

Mr. Akkaphon Nongnuch



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2005

ISBN 974-17-4283-5

หัวข้อวิทยานิพนธ์	อัลกอริทึมสำหรับการแบ่งนับเวกเตอร์โดยใช้การประมาณความหนาแน่น
โดย	นายอรรคพล นงนุช
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา	อาจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร.ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ

(รองศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา)

..... อาจารย์ที่ปรึกษา

(อาจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์)

..... กรรมการ

(อาจารย์ ดร.พิษณุ คนองชัยยศ)

..... กรรมการ

(อาจารย์ ดร.เศรษฐา ปานงาม)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

อรรคพล นงนุช : อัลกอริทึมสำหรับแบ่งนับเวกเตอร์โดยใช้การประมาณความหนาแน่น. (AN ALGORITHM FOR VECTOR QUANTIZATION USING DENSITY ESTIMATION) อ. ที่ปรึกษา : อ.ดร.อรรถสิทธิ์ สุรฤกษ์, 37 หน้า. ISBN 974-17-4283-5.

การแบ่งนับเวกเตอร์ได้ถูกนำมาประยุกต์ใช้งานอย่างหลากหลาย งานวิจัยส่วนมากจะมุ่งเน้นไปที่การสร้างหนังสือรหัสให้สอดคล้องกับระยะทางเชิงพื้นที่ สมรรถนะของการแบ่งนับจะถูกวัดจากค่าความผิดเพี้ยน ในวิทยานิพนธ์นี้ได้นำเสนอแนวทางใหม่สำหรับการแบ่งนับเวกเตอร์โดยใช้เทคนิคการประมาณความหนาแน่น หนังสือรหัสที่ได้จากอัลกอริทึมที่เสนอในวิทยานิพนธ์นี้สามารถแสดงให้เห็นถึงการกระจายความหนาแน่นของเวกเตอร์ต้นฉบับได้ และเพื่อทำให้อัลกอริทึมยืดหยุ่นขึ้น ส่วนขยายของอัลกอริทึมได้ถูกเสนอในวิทยานิพนธ์นี้ด้วย ผลการทดลองแสดงให้เห็นว่าค่าความผิดเพี้ยนลดลงเมื่อใช้กับส่วนขยายของอัลกอริทึมที่เสนอนี้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา..... วิศวกรรมคอมพิวเตอร์.....ลายมือชื่อนิสิต.....อรรคพล นงนุช.....
สาขาวิชา.....วิทยาศาสตร์คอมพิวเตอร์....ลายมือชื่ออาจารย์ที่ปรึกษา.....
ปีการศึกษา ...2548.....ลายมือชื่ออาจารย์ที่ปรึกษาร่วม.....

4670597521 : MAJOR COMPUTER SCIENCE

KEY WORD: VECTOR QUANTIZATION / LBG ALGORITHM / COMPETITIVE SPLITTING TECHNIQUE / DISTORTION

AKKAPHON NONGNUCH : AN ALGORITHM FOR VECTOR QUANTIZATION USING DENSITY ESTIMATION. THESIS ADVISOR : ATHASIT SURARERKS, Ph.D., 37 pp. ISBN: 974-17-4283-5.

Vector quantization is widely used in various applications. Most researches focus on creating a codebook with respect to spatial distance. The quantization performance is usually measured by distortion. In our work, we introduce a novel approach for vector quantization that is a density estimation technique. Codebooks obtained from our algorithm can express the density distribution of the original vectors. To make the algorithm more flexible, the extend procedure for the algorithm is also proposed in this thesis. Experimental results show that the distortion can be reduced using a flexible version of our algorithm.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Department..... Computer Engineering..... Student's signature.....

Field of study.....Computer Science..... Advisor's signature.....

Academic year ...2005.....Co-advisor's signature.....

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงออกมาได้ด้วยความกรุณาของอาจารย์ที่ปรึกษา อ.ดร.อรรถสิทธิ์ สุรฤกษ์ ซึ่งได้ให้คำแนะนำและแนวทางที่เป็นประโยชน์ต่องานวิจัย ตลอดจนคอยติดตามให้งานวิจัยนี้สำเร็จออกมาด้วยดี

ขอขอบคุณคณะกรรมการสอบวิทยานิพนธ์ รศ.ดร.ประภาส จงสถิตย์วัฒนา อ.ดร.พิษณุ คนองชัยยศ และอ.ดร.เศรษฐา ปานงาม ที่กรุณาให้คำแนะนำและช่วยตรวจสอบแก้ไขวิทยานิพนธ์ฉบับนี้

สุดท้ายนี้ ขอขอบคุณ บิดา มารดา ญาติสนิท และมิตรสหายทุกคน ที่คอยดูแลห่วงใย เป็นกำลังใจ และให้การสนับสนุนในทุกๆ ด้านตลอดมา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง
บทคัดย่อภาษาอังกฤษ ..... จ
กิตติกรรมประกาศ ..... ฉ
สารบัญ ..... ช
สารบัญภาพ ..... ฌ
สารบัญตาราง ..... ญ
บทที่
1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ 2
1.3 ขอบเขตการดำเนินงาน 2
1.4 การดำเนินการวิจัย 3
1.5 ประโยชน์ที่คาดว่าจะได้รับ 3
1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์ 3
2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 4
2.1 งานวิจัยที่เกี่ยวข้อง 4
2.2 การแบ่งนับเวกเตอร์ 5
2.3 อัลกอริทึมแอลบีจี 7
2.4 ซอม 8
3 อัลกอริทึมสำหรับการแบ่งนับเวกเตอร์โดยใช้การประมาณความหนาแน่น 12
3.1 บทนำ 12
3.2 อัลกอริทึม 13
3.3 ผลการทดลอง 17
3.4 การวิเคราะห์ผลการทดลอง 23
3.5 การทดสอบลักษณะการกระจายของหนังสือรหัส 24
4 ส่วนขยายของอัลกอริทึม 26
4.1 บทนำ 26
4.2 ส่วนขยายของอัลกอริทึม 26

หน้า

4.3 ผลการทดลอง 28
5 สรุปผลการวิจัยและข้อเสนอแนะ 31
5.1 สรุปผลการวิจัย 31
5.2 ข้อจำกัดของงานวิจัย 32
5.2 ข้อเสนอแนะ 33
รายการอ้างอิง 34
ประวัติผู้เขียนวิทยานิพนธ์ 37

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

หน้า

รูปที่ 1 ตัวอย่างการแบ่งนับเวกเตอร์ 6
รูปที่ 2 สถาปัตยกรรมของซอมที่ใช้หน่วยกลุ่ม 3 หน่วยแบบ 1 มิติสำหรับเวกเตอร์ 3 มิติ 8
รูปที่ 3 สถาปัตยกรรมของซอมที่ใช้หน่วยกลุ่ม 9 หน่วยแบบ 2 มิติสำหรับเวกเตอร์ 3 มิติ 9
รูปที่ 4 ตัวอย่างการหาย่านใกล้เคียงทางทอพอโลยีของหน่วยกลุ่มที่มีทอพอโลยีแบบ 1 มิติ..... 10
รูปที่ 5 ตัวอย่างการหาย่านใกล้เคียงทางทอพอโลยีของหน่วยกลุ่มที่มีทอพอโลยีแบบ 2 มิติ..... 11
รูปที่ 6 ปัญหา 2 ปัญหาที่มีหนังสือรหัสเหมือนกัน 12
รูปที่ 7 หนังสือรหัสแสดงให้เห็นถึงความหนาแน่นของปัญหา 13
รูปที่ 8 ตัวอย่างการแบ่งกั้นโดยใช้ระยะทางคลัสเตอร์ 15
รูปที่ 9 หนังสือรหัสที่ได้จากเซตที่ 1..... 21
รูปที่ 10 หนังสือรหัสที่ได้จากเซตที่ 2..... 21
รูปที่ 11 หนังสือรหัสที่ได้จากเซตที่ 3..... 22
รูปที่ 12 หนังสือรหัสที่ได้จากเซตที่ 4..... 22
รูปที่ 13 หนังสือรหัสที่ได้จากส่วนขยายของอัลกอริทึมที่ค่ายืดหยุ่นต่างๆ 29
รูปที่ 14 คำรหัส 4 คำรหัสที่ได้จากเวกเตอร์ 5 กลุ่ม..... 32
รูปที่ 15 คำรหัส 8 คำรหัสที่ได้จากเวกเตอร์ 5 กลุ่ม..... 33

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

หน้า

ตารางที่ 1 ค่าความผิดเพี้ยน 19
ตารางที่ 2 ค่าความคลาดเคลื่อนความหนาแน่น 20
ตารางที่ 3 ค่า χ^2 ของหนังสือรหัสขนาด 16 คำรหัส 25
ตารางที่ 4 ค่า χ^2 ของหนังสือรหัสขนาด 32 คำรหัส 25
ตารางที่ 5 ค่าความคลาดเคลื่อนความหนาแน่นและค่าความผิดเพี้ยน 28

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การแบ่งนับเวกเตอร์ (vector quantization) [1-3] เป็นเทคนิคที่ใช้ในการประมาณค่าเวกเตอร์ซึ่งแทนเซตของเวกเตอร์จำนวนมากด้วยเซตของเวกเตอร์ซึ่งมีขนาดน้อยกว่าที่เรียกว่าหนังสือรหัส (codebook) และ เรียกสมาชิกของหนังสือรหัสว่าคำรหัส (codeword) ในปัจจุบันการแบ่งนับเวกเตอร์ได้ถูกนำไปประยุกต์ใช้อย่างหลากหลาย เช่น การบีบอัดคำพูด (speech compression) การบีบอัดรูปภาพ (image compression) การรู้จำรูปแบบ (pattern recognition) และ ทัศนศาสตร์คอมพิวเตอร์ (computer vision) เป็นต้น

เป็นที่ทราบกันดีว่าสมรรถนะของการแบ่งนับเวกเตอร์จะขึ้นกับหนังสือรหัส ในการออกแบบหนังสือรหัสมีอัลกอริทึมที่ใช้ในการออกแบบหนังสือรหัสที่ได้รับความนิยมมากที่ชื่อว่าอัลกอริทึมแอลบีจี [4] ซึ่งเป็นกระบวนงานแบบวนซ้ำ (iterative procedure) จากหนังสือรหัสเริ่มต้น (initial codebook) จะเปลี่ยนเป็นหนังสือรหัสที่ผ่านการหาค่าที่เหมาะสม (optimized codebook) แต่คุณภาพของหนังสือรหัสที่ได้จากอัลกอริทึมนี้จะขึ้นกับการกำหนดค่าเริ่มต้นของหนังสือรหัส ดังนั้นจึงได้มีผู้ริเริ่มวิจัยและพัฒนาอัลกอริทึมที่ใช้ในการกำหนดค่าเริ่มต้นของหนังสือรหัสที่มีวัตถุประสงค์เพื่อลดความผิดเพี้ยน (distortion) ของการแบ่งนับลง เช่นงานวิจัยของ [4-6] เป็นต้น แต่อย่างไรก็ตามทั้งอัลกอริทึมแอลบีจีและอัลกอริทึมในการกำหนดค่าเริ่มต้นของหนังสือรหัสที่ผ่านมาไม่ได้คำนึงถึงลักษณะการกระจายของเวกเตอร์ ด้วยเหตุผลดังกล่าวข้างต้นอัลกอริทึมแอลบีจีจึงไม่รับประกันว่าหนังสือรหัสที่ได้สามารถแสดงให้ถึงการกระจายตัวของเวกเตอร์ การที่หนังสือรหัสสามารถอธิบายถึงการกระจายตัวของเวกเตอร์ได้จะทำให้รู้คุณสมบัติของปัญหาที่อยู่ในรูปของเวกเตอร์ได้มากขึ้น จึงเรียกได้ว่าลักษณะการกระจายของเวกเตอร์มีความสำคัญต่อการแบ่งนับเวกเตอร์ ในปี ค.ศ. 2002 มีงานวิจัยที่ให้ความสำคัญกับลักษณะการกระจายตัวของเวกเตอร์โดย [7] ได้ทำการพัฒนาอัลกอริทึมในการแบ่งนับเวกเตอร์ที่ได้นำซอม (SOM : self-organizing map) [8] ซึ่งเป็นโครงข่ายประสาทเทียม (artificial neural network) แบบที่ใช้การเรียนรู้เชิงแข่งขัน (competitive learning) มาช่วยในการค้นหาลักษณะการกระจายของเวกเตอร์ และต่อมาในปี ค.ศ.2004 งานวิจัย [6] ได้พัฒนาอัลกอริทึมกำหนดค่าเริ่มต้นของ

หนังสือรหัสโดยใช้หลักของการเรียนรู้เชิงแข่งขัน การแบ่งแยก (splitting) และการวัดเชิงเรขาคณิต (geometrical measurement) มาช่วยในการหาลักษณะการกระจายของเวกเตอร์

แต่อย่างไรก็ตามการใช้เทคนิคของการเรียนรู้เชิงแข่งขันของงานวิจัยทั้งสองงานที่กล่าวมาเพื่อค้นหาลักษณะการกระจายตัวของเวกเตอร์นั้นใช้ระยะทางเชิงพื้นที่ (spatial distance) ในการค้นหาลักษณะการกระจายตัวของเวกเตอร์นั้นไม่ได้คำนึงถึงความหนาแน่นของเวกเตอร์ในปริภูมิ (space) ซึ่งความหนาแน่นของเวกเตอร์ก็จัดเป็นสิ่งที่สามารถบอกได้ถึงลักษณะการกระจายของเวกเตอร์ได้เช่นกัน และยังไม่เคยมีการทำวิจัยทำมาก่อน

จากปัญหาและความจำเป็นที่กล่าวมาแล้วจึงทำให้เกิดวิทยานิพนธ์นี้ ซึ่งมีวัตถุประสงค์เพื่อพัฒนาอัลกอริทึมสำหรับการแบ่งนับเวกเตอร์เพื่อให้ได้หนังสือรหัสที่แสดงให้เห็นถึงการกระจายของเวกเตอร์ โดยในงานวิจัยนี้มีแนวความคิดที่ว่า การศึกษาลักษณะการกระจายตัวของเวกเตอร์สามารถตรวจจับได้ที่ความหนาแน่นของเวกเตอร์ ดังนั้นการสร้างหนังสือรหัสที่คงคุณสมบัติตามลักษณะการกระจายของเวกเตอร์ก็สามารถหาได้จากการพิจารณาความหนาแน่นและสร้างรหัสหนังสือที่คงคุณสมบัติความหนาแน่นของเวกเตอร์ทั้งหมดได้เช่นเดียวกัน โดยวิธีนี้ทำให้ได้หนังสือรหัสที่มีคุณสมบัติทั้งสองประการ คือ ลักษณะการกระจายและความหนาแน่น โดยอัลกอริทึมจะใช้หลักในการพัฒนาจากเลือกที่จะทำการแบ่งกั้น (partition) เวกเตอร์ทีละมิติให้ได้ผลแบ่งกั้นที่มีความหนาแน่นใกล้เคียงกันโดยพิจารณาเลือกมิติจากการวัดความต่อเนื่อง (continuity) ของค่าในมิติของเวกเตอร์

1.2 วัตถุประสงค์

เพื่อพัฒนาอัลกอริทึมสำหรับการแบ่งนับเวกเตอร์เพื่อให้ได้หนังสือรหัสที่แสดงให้เห็นถึงลักษณะการกระจายตัวของเวกเตอร์โดยใช้แนวคิดของความหนาแน่น และกำหนดให้ค่าความแปรปรวนความหนาแน่นเป็นตัววัดสมรรถนะของอัลกอริทึมอย่างหนึ่ง

1.3 ขอบเขตการดำเนินงาน

1. พัฒนาอัลกอริทึมสำหรับการแบ่งนับเวกเตอร์เพื่อให้ได้หนังสือรหัสที่แสดงให้เห็นถึงลักษณะการกระจายของเวกเตอร์

2. เปรียบเทียบค่าความคลาดเคลื่อนความหนาแน่นของหนังสือรหัสซึ่งได้หนังสือรหัสจากอัลกอริทึมที่ได้พัฒนาในงานวิจัยนี้กับหนังสือรหัสที่ได้จากอัลกอริทึมแอลบีจีซึ่งได้หนังสือรหัสเริ่มต้นจากการสุ่ม (random) และวิธีของซีออง [6]

3. เปรียบเทียบค่าความผิดเพี้ยนของหนังสือรหัสซึ่งได้จากอัลกอริทึมที่ได้พัฒนาในงานวิจัยนี้กับหนังสือรหัสที่ได้จากอัลกอริทึมแอลบีจีซึ่งได้หนังสือรหัสเริ่มต้นจากการสุ่มและวิธีของซีออง [6]

4. เปรียบเทียบเวลาที่ใช้ในการทำงานของอัลกอริทึมที่ได้พัฒนาในงานวิจัยนี้กับอัลกอริทึมแอลบีจีซึ่งได้หนังสือรหัสเริ่มต้นจากการสุ่มและวิธีของซีออง [6]

1.4 การดำเนินการวิจัย

1. ศึกษาการแบ่งนับเวกเตอร์และอัลกอริทึมแอลบีจี
2. พัฒนาอัลกอริทึม
3. ทดสอบอัลกอริทึมที่นำเสนอ
4. วิเคราะห์ผลการทดลอง
5. สรุปผลและเรียบเรียงวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ได้อัลกอริทึมสำหรับการแบ่งนับเวกเตอร์ที่หนังสือรหัสแสดงให้เห็นถึงการกระจายความหนาแน่นของปัญหาที่อยู่ในรูปของเวกเตอร์

1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นบทความทางวิชาการในหัวข้อเรื่อง “An Algorithm for Vector Quantization Using Density Estimation” โดย Akkaphon Nongnuch และ Athasit Surarerks ในงานประชุมวิชาการ “The 5th WSEAS International Conference on Signal, Speech and Image Processing (SSIP’05)” ณ Corfu Island Greece ในระหว่างวันที่ 17-19 สิงหาคม 2548

นอกจากนี้ยังได้รับการตีพิมพ์ในวารสาร WSEAS TRANSACTIONS ON COMPUTERS ฉบับเดือนกันยายน ปี ค.ศ. 2005 ในหัวข้อเรื่อง “A Novel Approach of Density Estimation for Vector Quantization” โดย Akkaphon Nongnuch และ Athasit Surarerks

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 งานวิจัยที่เกี่ยวข้อง

การแบ่งนับเวกเตอร์ถูกวิจัยเพื่อนำไปใช้งานด้านการประมวลผลสัญญาณโดยในช่วงแรกถูกนำไปใช้ในงานด้านการประมวลผลคำพูด [1-3] ต่อมาจึงได้ถูกนำไปใช้ในงานด้านอื่นที่หลากหลายขึ้น เช่น การบีบอัดคำพูด [9] การบีบอัดรูปภาพ [10] การรู้จำรูปแบบ [11] และทัศนศาสตร์คอมพิวเตอร์ [12] เป็นต้น การแบ่งนับเวกเตอร์สามารถแบ่งได้เป็นสองประเภท คือ แบบหนัก (hard) [4,8] และแบบคลุมเครือ (fuzzy) [13-14] กล่าวกันว่าแบบคลุมเครือจะอ่อนไหวต่อหนังสือรหัสเริ่มต้นน้อยกว่าแบบหนัก [14-15] สิ่งที่บ่งบอกสมรรถนะของการแบ่งนับเวกเตอร์ก็คือหนังสือรหัส ในงานวิจัย [4] ได้เสนออัลกอริทึมสำหรับการแบ่งนับเวกเตอร์ที่เรียกว่าอัลกอริทึมแอลบีจีโดยจะให้เอาทพุตออกมาเป็นหนังสือรหัสที่ให้ความผิดเพี้ยนของการแบ่งนับเวกเตอร์น้อยลง อัลกอริทึมแอลบีจีพัฒนาจากแนวคิดในงานวิจัย [16] ที่เป็นวิธีสำหรับออกแบบตัวแบ่งนับสำหรับการแบ่งนับสเกลาร์ อัลกอริทึมแอลบีจีจะสร้างหนังสือรหัสเริ่มต้นขึ้นมาต่อจากนั้นจะทำการหาค่าที่เหมาะสมโดยการทำการแบ่งกั้นโวโรนอยที่จะทำให้เวกเตอร์ไปอยู่ในผลแบ่งกั้นที่มีคำรหัสที่ใกล้ที่สุดและและจากนั้นจึงทำการหาคำรหัสใหม่จากจุดเซนทรอยด์ของแต่ละผลแบ่งกั้นและทำการวนซ้ำจนกระทั้งความผิดเพี้ยนมีการเปลี่ยนแปลงค่าน้อยว่าค่าที่กำหนดไว้ เพื่อที่จะทำให้อัลกอริทึมแอลบีจีทำงานได้อย่างอัตโนมัติในงานวิจัย [17] ได้เสนอการปรับปรุงอัลกอริทึมแอลบีจีที่ต้องไม่ใช้เงื่อนไขการหยุดการทำงานเป็นอินพุตของอัลกอริทึม ในงานวิจัย [4] ได้เสนอวิธีในการกำหนดค่าหนังสือรหัสเริ่มต้นเรียกว่า การแบ่งแยก (splitting) ซึ่งจะเริ่มต้นจากจุดเซนทรอยด์ของเวกเตอร์ทั้งหมดจากนั้นจึงแบ่งแยกจุดเซนทรอยด์ออกเป็นคำรหัส 2 จุด และทำการแบ่งแยกคำรหัสไปจนกระทั่งได้จำนวนคำรหัสเท่าที่ต้องการ วิธีนี้จะให้หนังสือรหัสเริ่มต้นที่มีกระจายตัวไม่ดีนักแต่ใช้เวลาน้อยมากในการหาหนังสือรหัสเริ่มต้น ต่อมาในงานวิจัย [5] ได้เสนอการกำหนดค่าเริ่มต้นเรียกว่าวิธีระยะทางไกลสุด (maximum distance) ขั้นแรกจะหาจุดเซนทรอยด์ของเวกเตอร์ทั้งหมดมาเป็นคำรหัสแรกของหนังสือรหัส จากนั้นจึงหาเวกเตอร์ที่ห่างจากคำรหัสในหนังสือรหัสมากที่สุดมาเป็นคำรหัสทำไปจนกระทั่งได้ขนาดของหนังสือรหัสที่ต้องการ วิธีนี้มีข้อดีคือสามารถให้หนังสือรหัสที่มีขนาดเป็นจำนวนใดก็ได้ต่างจากการแบ่งแยกที่ให้หนังสือรหัสเป็นเลขยกกำลังของ 2 เท่านั้น แต่ข้อเสียก็คือใช้เวลาค่อนข้างมากเนื่องจากต้องหาเวกเตอร์ที่ห่างจากหนังสือรหัส

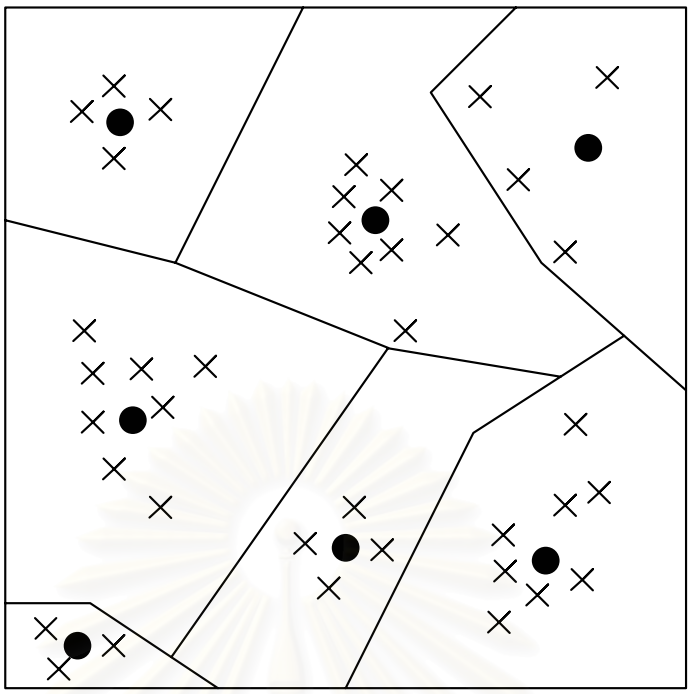
ที่สุดในแต่ละรอบและงานวิจัยที่น่าจะเกี่ยวข้องกับวิทยานิพนธ์นี้มากที่สุดก็คือ งานวิจัยของซีออง [6] ได้ใช้แนวคิดของการเรียนรู้เชิงแข่งขันมาพัฒนาเทคนิคในการกำหนดค่าเริ่มต้นของหนังสือรหัส โดยจะเริ่มต้นจากการหาจุดเซนทรอยด์ของเวกเตอร์จากนั้นวนซ้ำสุ่มรับเวกเตอร์มาทีละตัวแล้วแล้วทำการปรับค่าและให้คะแนนคำรหัสที่ใกล้กับเวกเตอร์นั้นมากที่สุด และลดคะแนนเวกเตอร์ที่อยู่ไกลกว่า เมื่อคะแนนมากกว่าที่กำหนดไว้จึงทำการแบ่งแยกคำรหัสนั้นออกเป็น 2 คำรหัส และวนรอบซ้ำไปจนกระทั่งได้หนังสือรหัสขนาดเท่าที่ต้องการ ซึ่งเทคนิคนี้สามารถให้หนังสือรหัสที่เป็นไปตามการกระจายเชิงพื้นที่ของเวกเตอร์ได้ทำให้เมื่อนำไปใช้เป็นตัวกำหนดค่าเริ่มต้นให้กับอัลกอริทึมแอลบีจีทำให้ความผิดเพี้ยนน้อยลง และงานวิจัย [7] ได้เสนออัลกอริทึมที่สนใจนำการกระจายของเวกเตอร์มาใช้กับการแบ่งนับเวกเตอร์เพื่อให้ได้ความผิดเพี้ยนน้อยลง โดยดัดแปลงซอมและนำมาใช้ค้นหาลักษณะการกระจายของเวกเตอร์ และทำการแบ่งกลุ่มคำรหัส และใช้อัลกอริทึมแอลบีจีกับคำรหัสในแต่ละกลุ่มแยกจากกัน แต่อย่างไรก็ตามการวิจัยในเรื่องการแบ่งนับเวกเตอร์จะเน้นไปทางด้านลดความผิดเพี้ยน [18-22] มากกว่าเรื่องอื่นๆ เช่น งานวิจัย [23-24] ซึ่งนำปรับปรุงการแบ่งนับเวกเตอร์มาใช้กับเวกเตอร์ฐานสองเพื่อให้ทำการแบ่งนับได้เร็วขึ้น

2.2 การแบ่งนับเวกเตอร์

การแบ่งนับเวกเตอร์ (vector quantization: VQ) [1-3] เป็นการประมาณเซตของเวกเตอร์ใดๆด้วยเซตของเวกเตอร์ตัวแทน (representative vector) หรือนิยมเรียกอีกชื่อหนึ่งว่าหนังสือรหัส (codebook) อาจกล่าวอีกนัยหนึ่งได้ว่าการแบ่งนับเวกเตอร์เป็นการส่ง (mapping) จากเซตของเวกเตอร์ใดๆ ไปเป็นหนังสือรหัส จะขอกล่าวถึงนิยามของการแบ่งนับเวกเตอร์ดังนี้

นิยามที่ 1 *กำหนดให้* X *เป็นเซตของเวกเตอร์* k *มิติ โดยที่* $X = \{\mathbf{x} \in \Re^k\}$ *และ* $|X| = P$ *และ* หนังสือรหัส (codebook) Y *เป็นเซตของเวกเตอร์* k *มิติ โดยที่* $Y = \{\mathbf{y} \in \Re^k\}$ *และ* $|Y| = N$ *โดย* $P >> N$ *และเรียกเวกเตอร์ที่เป็นสมาชิกของหนังสือรหัสว่าคำรหัส* (codeword)

นิยามที่ 2 *การแบ่งนับเวกเตอร์เป็นการส่งจากเซตของเวกเตอร์ใดๆไปเป็นหนังสือรหัสสามารถเขียนแทนได้ด้วยฟังก์ชัน* $q: X \to Y$ *หรือ* $q(X) = Y$ *โดย* $\mathbf{y}_i$ *จะเป็นตัวแทนของผลแบ่งกั้น* (partition) S_i *โดยที่* $S_i = q^{-1}(\mathbf{y}_i) = \{\mathbf{x} \in X : q(\mathbf{x}) = \mathbf{y}_i\}$, $i = 1,2,3,\ldots,N$ *ซึ่งแต่ละผลแบ่งกั้น* S_i *มีสมบัติดังนี้คือ* $\bigcup_{i=1}^{N} S_i = X$ *และ* $S_i \cap S_j = \varnothing$, $i \neq j$



รูปที่ 1 ตัวอย่างการแบ่งนับเวกเตอร์

ในรูปที่ 1 แสดงให้เห็นตัวอย่างของการแบ่งนับเวกเตอร์ 2 มิติ ที่มีหนังสือรหัสขนาด 8 คำรหัสโดยจุดสีดำแทน คำรหัส และกากบาทแทนเวกเตอร์ จากรูปที่ 1 สามารถอธิบายได้ดังนี้ในแต่ละผลแบ่งกั้นที่สามารถแยกจากกันได้ด้วยกรอบเส้นตรงสีดำดังในรูป และในแต่ละผลแบ่งกั้นจะประกอบด้วยคำรหัสซึ่งทำหน้าเป็นตัวแทนของผลแบ่งกั้นนั้น

โดยทั่วไปหลักในการวัดสมรรถนะ (performance) ของตัวแบ่งนับจากความผิดพลาดแบ่งนับ (quantization error) หรือเรียกอีกอย่างหนึ่งว่าความผิดเพี้ยน D จากทุกเวกเตอร์มาหาค่าเฉลี่ย ซึ่งความผิดเพี้ยนของการแบ่งนับเวกเตอร์มีนิยามดังนี้

$D = d(\mathbf{x}, q(\mathbf{x}))$ โดยที่ d คือ ตัวดำเนินการระยะทาง (distance operator)

มีฟังก์ชันมากมายที่สามารถใช้เป็นตัวดำเนินการระยะทางได้แต่ที่ได้รับความนิยมมากที่สุด คือ การใช้ระยะทางยูคลิเดียน (Euclidean distance) ซึ่งมีนิยามดังนี้

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$$

2.3 อัลกอริทึมแอลบีจี

ในการออกแบบตัวแบ่งนับนั้นสิ่งที่จัดว่ามีความสำคัญอย่างหนึ่งคือ หนังสือรหัส เนื่องจากหนังสือรหัสจะมีผลโดยตรงต่อสมรรถนะของตัวแบ่งนับ และอัลกอริทึมแอลบีจี (LBG algorithm: Linde-Buzo-Gray algorithm) หรือเรียกอีกอย่างหนึ่งว่าจีแอลเอ (GLA: generalized Lloyd algorithm) ก็เป็นหนึ่งในอัลกอริทึมที่ได้รับความนิยมที่สุดในการออกแบบหนังสือรหัส อัลกอริทึมนี้เสนอโดย [4] ในปี ค.ศ. 1980 อัลกอริทึมแอลบีจีมีลักษณะเป็นกระบวนงานแบบวนซ้ำ โดยมีเซตของเวกเตอร์ X จำนวน P เวกเตอร์ ขนาดของหนังสือรหัสที่ต้องการ N และขีดเริ่มเปลี่ยน (threshold) γ เป็นอินพุตของอัลกอริทึม และจะให้หนังสือรหัส Y เป็นเอาทพุตของอัลกอริทึม โดยที่ $P >> N$

ขั้นตอนของอัลกอริทึมแอลบีจีมีดังต่อไปนี้

1. ทำการกำหนดค่าเริ่มต้น (initialization) ของคำรหัสจำนวน N คำ
2. ทำการแบ่งกั้นโวโรนอย (Voronoi partition) โดยเมื่อได้รับหนังสือรหัสมาจะทำการจัดเวกเตอร์แต่ละตัวไปอยู่กับคำรหัสที่อยู่ใกล้ที่สุด ในแต่ละผลแบ่งกั้นจะเขียนอยู่ในรูปของเซตได้ดังนี้

$$S_i = \{\mathbf{x} \in X : d(\mathbf{x}, \mathbf{y}_i) \le d(\mathbf{x}, \mathbf{y}_j), j = 1,2,3,..., N, j \ne i\}, i = 1,2,3,..N$$

3. ตรวจสอบเงื่อนไขการหยุด ซึ่งเปรียบเทียบความผิดเพี้ยนในรอบปัจจุบันกับในก่อนหน้า โดยพิจารณาจากอัตราส่วนของ $\left|D_{prev} - D_{curr}\right| / D_{curr}$ ว่าน้อยกว่าขีดเริ่มเปลี่ยนหรือไม่ ถ้าน้อยกว่าจึงทำการเลิก มิฉะนั้นจะไปยังขั้นตอนต่อไป
4. ทำการคำนวณค่าเซนทรอยด์ (centroid calculation) โดยเมื่อได้ผลแบ่งกั้นมาจะทำการหาหนังสือรหัสที่เหมาะที่สุดจากค่าเซนทรอยด์ของผลแบ่งกั้นดังสมการ

$$C_i = \frac{1}{n_i} \sum_{\mathbf{x} \in S_i} \mathbf{x}, \; i = 1, \ldots, N$$ โดย n_i คือจำนวนสมาชิกของผลแบ่งกั้น S_i

5. วนซ้ำไปยังข้อ 2.-4.

เมื่อเริ่มต้นการทำงานของอัลกอริทึมเวกเตอร์แต่ละตัวในเซต X จะถูกนำมาหาคำรหัสเริ่มต้นที่ใกล้ที่สุด โดยจะใช้ตัวดำเนินการระยะทางในการวัดระยะทาง และจากนั้นเวกเตอร์ที่มีคำรหัสที่ใกล้ที่สุดเป็นคำเดียวกันก็จะถูกทำการแบ่งกั้นไปยังผลแบ่งกั้นเดียวกัน ต่อจากนั้นก็ทำการตรวจสอบเงื่อนไขว่าความผิดเพี้ยนในรอบปัจจุบันกับในรอบก่อนหน้าว่าเปลี่ยนแปลงไปมากน้อยแค่ไหนซึ่งการเปลี่ยนแปลงของความผิดเพี้ยนจะสามารถบอกได้ว่าหนังสือรหัสที่ได้มีเสถียรภาพเพียงพอหรือไม่ ถ้าเสถียรภาพพอก็จะหยุดการทำงานของอัลกอริทึม แต่ถ้าไม่ก็จะทำงานต่อไปโดยการคำนวณค่าเซนทรอยด์ของแต่ละผลแบ่งกั้นซึ่ง จะทำการหาผมรวมของเวกเตอร์ในแต่ละผลแบ่งกั้นแล้วหารด้วยจำนวนเวกเตอร์ในผลแบ่งกั้นนั้น จากนั้นจึงให้ค่าเซนทรอยด์ที่คำนวณได้เป็นคำรหัสในเริ่มต้นของอัลกอริทึมในรอบถัดไป

2.4 ซอม

ซอม (SOM: self-organizing map) เป็นโครงข่ายประสาทเทียมแบบการเรียนรู้โดยไม่ต้องสอน (unsupervised learning) เสนอโดย [8] ที่ใช้สำหรับการแบ่งกลุ่มเวกเตอร์โดยใช้หลักของการเรียนรู้เชิงแข่งขัน สถาปัตยกรรมของซอมเป็นโครงข่ายแบบสองชั้น (2-layer network) ซึ่งประกอบด้วยหน่วยเข้า (input unit) และหน่วยกลุ่ม (cluster unit) หน่วยเข้าจะมีจำนวนเท่ากับมิติของเวกเตอร์ K หน่วยกลุ่มจะมีจำนวนเท่ากับจำนวนของกลุ่ม J ซึ่งหน่วยกลุ่มจะมีโครงสร้างทางทอพอโลยี (topological structure) แบบ 1 มิติหรือ 2 มิติก็ได้ขึ้นการกำหนดฟังก์ชันย่านใกล้เคียง (neighborhood function) (จะกล่าวถึงฟังก์ชันย่านใกล้เคียงต่อไปภายหลัง) และหน่วยเข้าทุกหน่วยจะเชื่อมต่อไปยังหน่วยกลุ่มโดยมีค่าน้ำหนัก (weight) W_{kj} กำหนดไว้ โดยหน่วยเข้าจะมีไว้สำหรับรับเวกเตอร์ตัวอย่าง $\mathbf{x}$ และหน่วยกลุ่มไว้สำหรับบอกว่าเวกเตอร์นั้นอยู่ในกลุ่มใด โดยในรูปที่ 2 และ 3 แสดงตัวอย่างสถาปัตยกรรมของซอมแบบ 1 มิติและ 2 มิติตามลำดับ

การทำงานของซอมจะใช้การเรียนรู้เชิงแข่งขันมาช่วยในการปรับค่าน้ำหนักของโครงข่าย สามารถอธิบายได้ดังนี้ เมื่อโครงข่ายรับเวกเตอร์เข้ามาจะหาว่าเวกเตอร์ตัวนั้นมีระยะใกล้กับหน่วยกลุ่มหน่วยไหนมากที่สุดโดยพิจารณาจากค่าน้ำหนักของหน่วยกลุ่มกับเวกเตอร์ และจะให้หน่วยกลุ่มตัวนั้นเป็นผู้ชนะ (winner) จากนั้นจะหาหน่วยกลุ่มอื่นที่เป็นย่านใกล้เคียงทางทอพอโลยี (topological neighborhood) ของผู้ชนะโดยพิจารณาจากฟังก์ชันย่านใกล้เคียง ซึ่งในรูปที่ 4 และ 5 ได้แสดงตัวอย่างการหาย่านใกล้เคียงทางทอพอโลยีที่มีฟังก์ชันย่านใกล้เคียงเป็นฟังก์ชันลดเชิงเส้นของรัศมีกับรอบวนซ้ำแบบ 1 มิติและ 2 มิติตามลำดับ โดยสมมุติว่าแต่ละรอบวนซ้ำมีผู้ชนะตัวเดียวกัน จากนั้นจะทำการปรับค่าน้ำหนักของทั้งผู้ชนะและหน่วยกลุ่มย่านใกล้เคียง และลดอัตราการเรียนรู้ (learning rate) ลง โดยจะทำการเรียนรู้ไปจนกระทั่งไม่สามารถเห็นถึงการเปลี่ยนแปลงของค่าน้ำหนักได้ โดยฟังก์ชันย่านใกล้เคียงและอัตราการเรียนรู้เป็นฟังก์ชันของ รอบวนซ้ำ m แทนโดย $h(m)$ และ $\eta(m)$ ตามลำดับซึ่ง $\eta(m)$ เป็นฟังก์ชันลด

ขั้นตอนของอัลกอริทึมเรียนรู้ของซอมมีดังต่อไปนี้

1. ทำการกำหนดค่าเริ่มต้นของค่าน้ำหนักและกำหนดรอบวนซ้ำ $m = 0$ กำหนดฟังก์ชันย่านใกล้เคียง อัตราการเรียนรู้
2. รับเวกเตอร์เข้ามาในโครงข่ายแล้วคำนวณหาค่าระยะทางยูคลิเดียนของเวกเตอร์กับค่าน้ำหนักของทุกหน่วยออก ดังสมการนี้

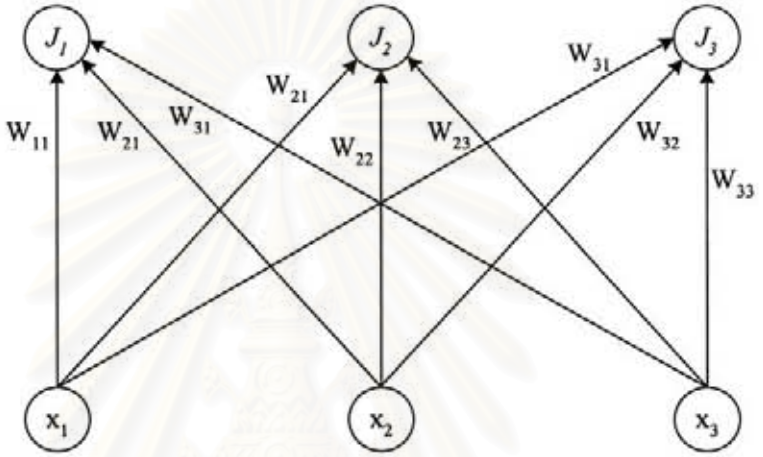
$$d(\mathbf{x}, \mathbf{w}_j) = \sqrt{\sum_{i=1}^{k} (x_i - w_{ij})^2} \ , j = 1,2,3,\ldots,J$$

จากนั้นหาผู้ชนะโดยดูว่าหน่วยกลุ่มไหนที่อยู่ใกล้เวกเตอร์ที่สุด

3. หาย่านใกล้เคียงทางทอพอโลยีของผู้ชนะจากฟังก์ชันย่านใกล้เคียงและทำการปรับค่าน้ำหนักของทั้งผู้ชนะและย่านใกล้เคียงทางทอพอโลยีดังสมการนี้

$$\mathbf{w}_j(m+1) = \mathbf{w}_j(m) + \eta(m)(\mathbf{x}(m) - \mathbf{w}_j(m))$$

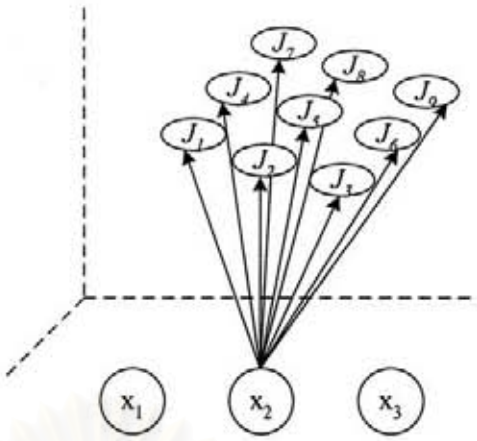
4. วนซ้ำไปยังข้อ 2. และ 3. จนกระทั่งไม่สามารถเห็นถึงการเปลี่ยนแปลงของค่าน้ำหนักได้



รูปที่ 2 สถาปัตยกรรมของซอมที่ใช้หน่วยกลุ่ม 3 หน่วยแบบ 1 มิติสำหรับเวกเตอร์ 3 มิติ

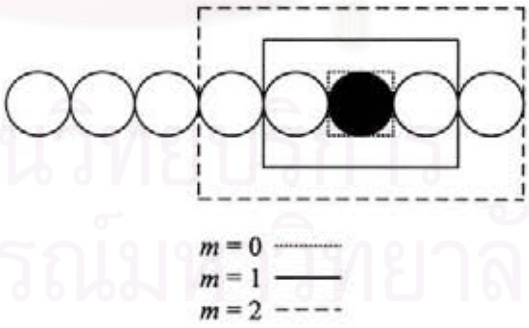
ในรูปที่ 2 แสดงให้เห็นตัวอย่างของสถาปัตยกรรมของซอมที่ใช้หน่วยกลุ่ม 3 หน่วยแบบ 1 มิติสำหรับเวกเตอร์ 3 มิติ เมื่อรับค่าเวกเตอร์ตัวอย่างในแต่ละมิติเข้ามายังหน่วยเข้า $\mathbf{x}_1$, $\mathbf{x}_2$ และ $\mathbf{x}_3$ แล้ว จากนั้นจึงส่งของค่าเวกเตอร์ตัวอย่างที่ผ่านการถ่วงน้ำหนักของแต่ละหน่วยกลุ่ม J_i เพื่อเปรียบเทียบว่าหน่วยใดมีค่าของเวกเตอร์ตัวอย่างที่ผ่านการถ่วงน้ำหนักที่มากที่สุดแล้วจึงให้เวกเตอร์ตัวอย่างอยู่ในกลุ่มนั้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3 สถาปัตยกรรมของซอมที่ใช้หน่วยกลุ่ม 9 หน่วยแบบ 2 มิติสำหรับเวกเตอร์ 3 มิติ

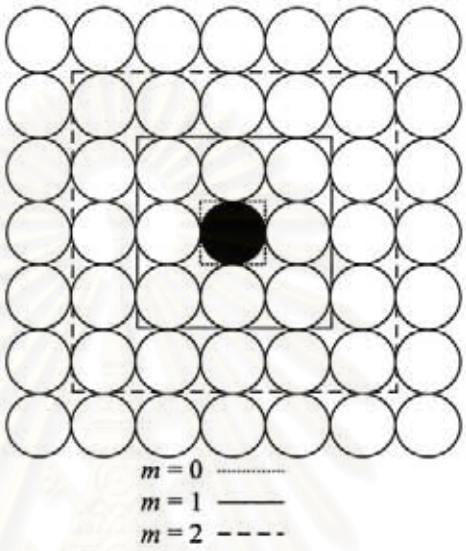
ในรูปที่ 3 แสดงให้เห็นถึงสถาปัตยกรรมของซอมที่ใช้หน่วยกลุ่ม 9 หน่วยแบบ 2 มิติ สำหรับเวกเตอร์ 3 มิติ โดยเมื่อรับค่าของเวกเตอร์ตัวอย่างในแต่ละมิติเข้ามายังหน่วยเข้า $\mathbf{x}_1$, $\mathbf{x}_2$ และ $\mathbf{x}_3$ จากนั้นจึงส่งของค่าของเวกเตอร์ตัวอย่างที่ผ่านการถ่วงน้ำหนักของแต่ละหน่วยกลุ่ม J_i แต่ละหน่วยเพื่อเปรียบเทียบว่าหน่วยไหนมีค่าของเวกเตอร์ตัวอย่างที่ผ่านการถ่วงน้ำหนักที่มากที่สุดแล้วจึงให้เวกเตอร์ตัวอย่างอยู่ในกลุ่มนั้นซึ่งในรูปที่ 3 แสดงเฉพาะการเชื่อมต่อของหน่วยเข้า $\mathbf{x}_2$ โดยในรูปที่ 2 มีทอโพโลยีของหน่วยกลุ่มต่างจากในรูปที่ 2 ซึ่งในรูปที่ 2 เป็นแบบ 1 มิติ ส่วนในรูปที่ 3 เป็นแบบ 2 มิติ



รูปที่ 4 ตัวอย่างการหาย่านใกล้เคียงทางทอพอโลยีของหน่วยกลุ่มที่มีทอพอโลยีแบบ 1 มิติ

ในรูปที่ 4 แสดงตัวอย่างการหาย่านใกล้เคียงทางทอพอโลยีของหน่วยกลุ่มที่มีทอพอโลยีแบบ 1 มิติ โดยให้วงกลมสีดำเป็นผู้ชนะในรอบวนซ้ำแรกมีย่านใกล้เคียงทางทอพอโลยีเพียงแค่วงกลมสีดำ 1 วงเท่านั้น และถ้าในรอบวนซ้ำถัดไปวงกลมสีดำยังคงเป็นผู้ชนะอยู่ย่านใกล้เคียงทาง

ทอพอโลยีจะกลายเป็นวงกลมสีดำรวมถึงวงกลมสีขาว 2 วงที่อยู่ในทั้งสองข้างของวงกลมสีดำด้วย รวมทั้งสิ้น 3 วง และถ้าในรอบวนซ้ำถัดไปวงกลมสีดำยังคงเป็นผู้ชนะอยู่ย่านใกล้เคียงทางทอพอโลยีจะกลายเป็นวงกลมสีดำรวมถึงวงกลมสีขาว 4 วงที่อยู่ในทั้งสองข้างของวงกลมสีดำด้วย รวมทั้งสิ้น 5 วง



รูปที่ 5 ตัวอย่างการหาย่านใกล้เคียงทางทอพอโลยีของหน่วยกลุ่มที่มีทอพอโลยีแบบ 2 มิติ

ในรูปที่ 5 แสดงตัวอย่างการหาย่านใกล้เคียงทางทอพอโลยีของหน่วยกลุ่มที่มีทอพอโลยีแบบ 2 มิติ โดยให้วงกลมสีดำเป็นผู้ชนะในรอบวนซ้ำแรกมีย่านใกล้เคียงทางทอพอโลยีเพียงแค่วงกลมสีดำ 1 วงเท่านั้น และถ้าในรอบวนซ้ำถัดไปวงกลมสีดำยังคงเป็นผู้ชนะอยู่ย่านใกล้เคียงทางทอพอโลยีจะกลายเป็นวงกลมสีดำรวมถึงวงกลมสีขาว 8 วงที่ล้อมรอบของวงกลมสีดำด้วย รวมทั้งสิ้น 9 วง และถ้าในรอบวนซ้ำถัดไปวงกลมสีดำยังคงเป็นผู้ชนะอยู่ย่านใกล้เคียงทางทอพอโลยีจะกลายเป็นวงกลมสีดำรวมถึงวงกลมสีขาว 24 วงที่ล้อมรอบวงกลมสีดำด้วย รวมทั้งสิ้น 25 วง

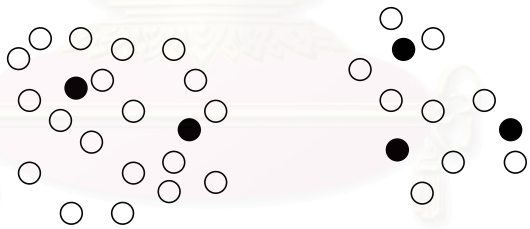
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

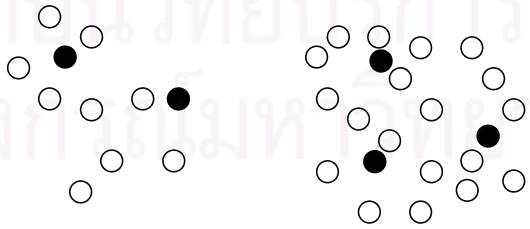
อัลกอริทึมสำหรับการแบ่งนับเวกเตอร์โดยใช้การประมาณความหนาแน่น

3.1 บทนำ

บทนี้ได้เสนออัลกอริทึมที่มาจากสมมุติฐานที่ว่าลักษณะการกระจายของปัญหาสามารถหาได้จากความหนาแน่นของปัญหาดังนั้นอัลกอริทึมนี้จะให้หนังสือรหัสที่แทนทั้งการกระจายและความหนาแน่นของปัญหาด้วย ซึ่งความหนาแน่นในงานวิจัยนี้หมายถึงจำนวนเวกเตอร์ในผลแบ่งกั้น ก่อนอื่นจะขอกล่าวถึงความสำคัญของความหนาแน่นก่อน ในรูปที่ 6(1) และ 6(2) แสดงให้เห็นถึง 2 ปัญหาที่มีหนังสือรหัสเหมือนกัน แต่หนังสือรหัสไม่ได้แสดงถึงความแตกต่างของทั้ง 2 ปัญหาที่มีความหนาแน่นต่างกัน เนื่องจากพิจารณาเพียงแค่ระยะทางเชิงพื้นที่เท่านั้น ในวิทยานิพนธ์นี้จะพัฒนาอัลกอริทึมที่สร้างหนังสือรหัสที่แสดงให้เห็นถึงความหนาแน่นของปัญหา ดังในรูปที่ 7 ที่หนังสือรหัสแสดงให้เห็นถึงความหนาแน่นของปัญหาได้ โดยบริเวณที่มีเวกเตอร์หนาแน่นมากก็จะมีคำรหัสอยู่ในบริเวณนั้นมากตามไปด้วย



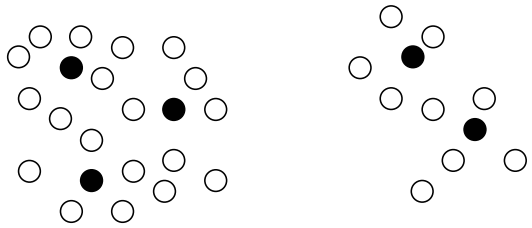
(1)



(2)

รูปที่ 6 ปัญหา 2 ปัญหาที่มีหนังสือรหัสเหมือนกัน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 7 หนังสือรหัสแสดงให้เห็นถึงความหนาแน่นของปัญหา

นอกจากนี้ยังได้มีการเปรียบเทียบสมรรถนะของอัลกอริทึมที่เสนอกับอัลกอริทึมอื่นใน 3 ด้านคือ ความผิดเพี้ยน ค่าคลาดเคลื่อนความหนาแน่น และความซับซ้อนเชิงเวลา อีกทั้งยังได้ทำการทดสอบว่าหนังสือที่ได้จากอัลกอริทึมสามารถแสดงให้เห็นถึงการกระจายของเวกเตอร์โดยใช้การทดสอบความเหมาะสมของการแจกแจง

3.2 **อัลกอริทึม**

อัลกอริทึมที่จะขอเสนอนี้เป็นอัลกอริทึมที่สามารถนำมาใช้เป็นตัวกำหนดค่าเริ่มต้น (initializer) ของอัลกอริทึมแอลบีจี หรืออัลกอริทึมสำหรับการแบ่งนับเวกเตอร์ได้ อัลกอริทึมนี้จะให้เอาทพุตเป็นหนังสือรหัสที่สามารถอธิบายถึงลักษณะการกระจายของปัญหาได้ แนวคิดของอัลกอริทึมในการทำให้หนังสือรหัสสามารถแสดงให้เห็นถึงความหนาแน่นของปัญหาได้ก็คือการทำให้แต่ละผลแบ่งกั้นมีจำนวนเวกเตอร์ใกล้เคียงกันเพื่อที่จะรักษาความหนาแน่นของแต่ละผลแบ่งกั้นให้ใกล้เคียงตามไปด้วยและเมื่อหาคำรหัสที่แทนแต่ละผลแบ่งกั้นก็จะได้หนังสือรหัสที่แสดงให้เป็นถึงความหนาแน่นตามไปด้วย อัลกอริทึมพัฒนาขึ้นโดยใช้เทคนิคการแบ่งแยกและเอาชนะ (divide and conquer) โดยจะใช้ความสัมพันธ์เวียนเกิดทำการแบ่งกั้นแต่ละผลแบ่งกั้นออกเป็นสองผลแบ่งกั้นที่มีความหนาแน่นใกล้เคียงกันไปจนกระทั่งได้จำนวนผลแบ่งกั้นเท่ากับขนาดของหนังสือรหัสที่ต้องการ จากนั้นจึงทำการหาคำรหัสจากค่าเซนทรอยด์ของแต่ละผลแบ่งกั้น อาจเรียกแนวคิดนี้ได้ว่าเป็นการแบ่งแยกที่พยายามจะให้แต่ละผลแบ่งกั้นมีความหนาแน่นใกล้เคียงกัน

การพิจารณาว่าจะแบ่งกั้นเวกเตอร์ในแต่ละมิติอย่างไรจะใช้แนวคิดดังนี้

1. พิจารณาแต่ละมิติแยกจากกัน
2. แต่ละมิติพิจารณาความต่อเนื่องของค่าในมิติของเวกเตอร์
3. ทำการแบ่งกั้นเวกเตอร์ โดยพิจารณาจากมิติที่มีความต่อเนื่องน้อยที่สุด

4. ในแต่ละผลแบ่งกั้นของเวกเตอร์ ให้ทำกระบวนงานในข้อ 1 ถึง 3 จนครบทุกมิติหรือหรือได้จำนวนผลแบ่งกั้น(ขนาดของหนังสือรหัส)เท่าที่ต้องการ

สำหรับเหตุผลที่พิจารณาแต่ละมิติแยกจากกัน เนื่องมาจากถ้าพิจารณาทุกมิติพร้อมกันจะทำให้อัลกอริทึมทำงานช้า การพิจารณาทุกมิติจะช่วยให้อัลกอริทึมทำงานเร็วขึ้นได้ และเหตุผลที่แต่ละมิติพิจารณาความต่อเนื่องของค่าในมิติของเวกเตอร์ เนื่องจากต้องการให้อัลกอริทึมรู้ถึงระยะห่างของค่าในมิติของเวกเตอร์แต่ละตัวเพื่อที่จะสามารถทำการแบ่งกั้นอย่างเหมาะสมได้

จากแนวคิดดังกล่าวสามารถพัฒนาเป็นอัลกอริทึมได้ดังนี้ กระบวนงานแบ่งกั้นจะเลือกมิติของเวกเตอร์มา 1 มิติเพื่อทำการแบ่งกั้นซึ่งมิตินี้จะสามารถสร้างความแตกต่างระหว่าง 2 ผลแบ่งกั้นได้ดีที่สุดเนื่องจากมีความต่อเนื่องน้อยที่สุด และมีการหาตัวหลัก (pivot) เพื่อนำไปใช้สำหรับการแบ่งกั้นด้วย โดยในมิติที่ถูกเลือกว่าจะทำการแบ่งกั้น ค่าของเวกเตอร์ที่มากกว่าตัวหลักและน้อยกว่าตัวหลักจะอยู่ในคนละผลแบ่งกั้น ถ้าเวกเตอร์ถูกแบ่งกั้นออกเป็นผลแบ่งกั้นซ้ายและขวา ระยะทางระหว่างค่าขวาสุดของเวกเตอร์ในผลแบ่งกั้นซ้ายกับค่าซ้ายสุดของเวกเตอร์ในผลแบ่งกั้นขวาจะถูกนำมาพิจารณาหาค่ามากสุดในการแบ่งกั้น ซึ่งขอเรียกระยะทางนั้นว่า *ระยะทางคลัสเตอร์* (*cluster distance*) ในรูปที่ 8 แสดงตัวอย่างการเลือกมิติเพื่อทำการแบ่งกั้น เมื่อพิจารณาในแกนนอนรูปที่ 8(1) จะพบว่ามีความต่อเนื่องน้อยกว่าในแกนตั้งรูปที่ 8(2) ดังนั้นจึงเลือกที่จะทำการแบ่งกั้นในแนวนอนดังรูปที่ 8(3) ตัวหลักจะเป็นค่าเฉลี่ยของค่าขวาสุดของเวกเตอร์ในผลแบ่งกั้นซ้ายกับค่าซ้ายสุดของเวกเตอร์ในผลแบ่งกั้นขวา

เพื่อที่จะประเมินคุณสมบัติด้านความหนาแน่น เราได้ทำการ*นิยาม ค่าความคลาดเคลื่อนความหนาแน่น* (*variance of density*) ε ดังนี้

$$\varepsilon = \sqrt{\frac{\sum_{i=1}^{N} (n_{avg} - n_i)^2}{N}}$$

โดย n_{avg} คืออัตราส่วนของจำนวนเวกเตอร์ทั้งหมดต่อขนาดของหนังสือรหัส n_i คือจำนวนเวกเตอร์ในผลแบ่งกั้นที่ i และ N คือขนาดของหนังสือรหัส

จากแนวคิดที่กล่าวมาข้างต้นสามารถเขียนสรุปเป็นขั้นตอนของอัลกอริทึมได้ดังนี้

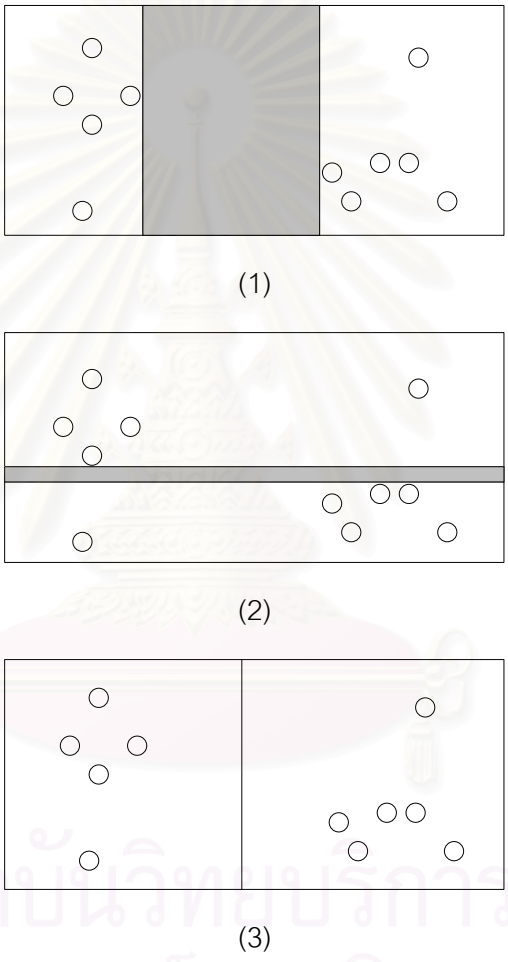
อัลกอริทึมสำหรับการแบ่งนับเวกเตอร์โดยใช้การประมาณความหนาแน่น

อินพุต เซตของเวกเตอร์ A ขนาด d มิติ และขนาดของหนังสือรหัส m

เอาทพุต หนังสือรหัส C

1. ให้เซตของเวกเตอร์เป็นหนึ่งผลแบ่งกั้น

2. หาระยะทางแบ่งกั้นในทุกมิติของเวกเตอร์ในแต่ละผลแบ่งกั้น
3. แบ่งกั้นเวกเตอร์ออกเป็นสองผลแบ่งกั้นที่มีขนาดใกล้เคียงกันที่เป็นลำดับจากน้อยไปมาก ในมิติที่มีระยะทางแบ่งกั้นมากที่สุดและใช้ตัวหลักของมิตินั้นด้วย
4. ทำการแบ่งกั้นต่อไปเรื่อยๆจนกระทั่งมีจำนวนผลแบ่งกั้นเท่ากับขนาดหนังสือรหัส
5. หาค่าเซนทรอยด์ของแต่ละผลแบ่งกั้น และส่งเซตของค่าเหล่านี้เป็นหนังสือรหัส



รูปที่ 8 ตัวอย่างการแบ่งกั้นโดยใช้ระยะทางคลัสเตอร์

จากขั้นตอนของอัลกอริทึมข้างต้นสามารถแสดงเป็นรหัสเทียมของอัลกอริทึมได้ดังนี้

Input : a set of vector A size *n*, dimension *d* and codebook size *m*

Output : A *m*-size codebook C

VQ(A,*m*,*d*)

01: **begin**

```
02: m_c ← 0
03: DensityPartititon(A,m,d)
04:     begin
05:       if m_c ≠ m then
06:         for i ← 1 to d do
07:           D[i] ← FindClusterDistance(A^i)
08:         end
09:         PartitionDimension ← argmax(D[1…d])
10:         Pivot ← FindPivot(PartitionDimension)
11:         (P1,P2) ← Partition(A,PartitionDimension,Pivot)
12:         Update(m_c)
13:         DensityPartition(P1,m,d)
14:         DensityPartition(P2,m,d)
15:       endif
16:     end
17:  for i ← 1 to m do
18:    C[i] ← Centriod(A_i)
19:  end
20:  return(C)
21: end
```

จากรหัสเทียมข้างต้นสามารถอธิบายได้ดังนี้ อัลกอริทึมจะรับอินพุตเป็น เซตของเวกเตอร์ A ที่มี d จำนวน n เวกเตอร์ และขนาดของหนังสือรหัส m และให้เอาทพุตเป็นหนังสือรหัสขนาด m ในตอนแรกจะมีการตั้งค่าให้ตัวแปร m_c ซึ่งทำหน้าที่นับจำนวนผลแบ่งกั้นที่มีอยู่ให้มีค่าเท่ากับศูนย์ดังในบรรทัดที่ 2 หลังจากนั้นฟังก์ชัน DensityPartititon ก็จะเริ่มทำการแบ่งกั้นเซตของเวกเตอร์ A โดยใช้เซตของเวกเตอร์ A ขนาดของหนังสือรหัส m และมิติของเวกเตอร์ d เป็นอินพุต ดังในบรรทัดที่ 3 ในขั้นแรกจะตรวจสอบว่าจำนวนผลแบ่งกั้นที่มีอยู่เท่ากับขนาดของหนังสือรหัสหรือไม่ดังในบรรทัดที่ 5 ถ้าจำนวนของผลแบ่งกั้นยังเท่ากับจำนวนของหนังสือรหัสแล้วก็จะทำการหาค่าเซนทรอยด์ของแต่ละผลแบ่งกั้นแล้วส่งค่าทั้งหมดไปเป็นเอาทพุตดังในบรรทัดที่ 17-20 แต่ถ้าจำนวนของผลแบ่งกั้นยังไม่เท่ากับจำนวนของหนังสือรหัสก็จะทำการแบ่งกั้นต่อไป โดยจะทำการหาระยะทางคลัสเตอร์ในแต่ละมิติดังบรรทัดที่ 6-8 จากนั้นจึงเลือกมิติที่จำทำการแบ่งกั้นจากมิติที่มีค่าของระยะทางคลัสเตอร์สูงสุดดังในบรรทัดที่ 9 จากนั้นจึงหาค่าตัวหลักและทำการแบ่งกั้นดังในบรรทัดที่ 10-11 แล้วจึงปรับค่า m_c ตามจำนวนผลแบ่งกั้นที่มีอยู่ดังในบรรทัดที่ จากนั้นจึงเรียก

ฟังก์ชัน DensityPartititon มาทำการแบ่งกั้นต่อไปโดยใช้ผลแบ่งกั้นสองผลแบ่งกั้นก่อนหน้าเป็นอินพุตดังบรรทัดที่ 13-14

เมื่อพิจารณาอัลกอริทึมดูแล้วพบว่าอัลกอริทึมที่เสนอทำงานโดยใช้การแบ่งแยกและเอาชนะ โดยแบ่งข้อมูลออกเป็นสองส่วนที่เท่ากันในแต่ละรอบการทำงาน ดังนั้นถ้ากำหนดให้ $T(n,m)$ เป็นเวลาในการแบ่งนับเวกเตอร์ด้วยเวกเตอร์จำนวน n และหนังสือรหัสขนาด m ความซับซ้อนเชิงเวลาของอัลกอริทึมที่เสนอนี้เท่ากับ

$$T(n,m) = 2T(\frac{n}{2},\frac{m}{2}) + \Theta(dn)$$

ในแต่ละรอบเวียนเกิดอัลกอริทึมจะใช้เวลาทำงานโดยรวมเท่ากับ $\Theta(dn)$ และมีจำนวนรอบเวียนเกิดทั้งหมดเท่ากับ $\log m$ นั้นคือเวลาที่ใช้ในการทำงานทั้งหมดเป็น $\Theta(dn\log m)$ เมื่อ d คือจำนวนมิติของเวกเตอร์

ส่วนอัลกอริทึมแอลบีจีนั้นเวลาที่ใช้ในการทำงานนั้นจะขึ้นกับทั้งจำนวนเวกเตอร์และจำนวนคำรหัสเท่ากับ ดังนั้นความซับซ้อนเชิงเวลาของอัลกอริทึมแอลบีจีจึงเท่ากับ $\Theta(mn)$ ส่วนอัลกอริทึมแอลบีจีที่ใช้อัลกอริทึมที่เสนอในงานวิจัยนี้เป็นตัวเริ่มต้นจึงมีความซับซ้อนเชิงเวลาเท่ากับ $\Theta(dn\log m)$ เนื่องจากในส่วนของการกำหนดค่าเริ่มต้นของหนังสือรหัสใช้เวลามากกว่า และสำหรับวิธีของซีอองนั้นการทำงานนั้นขึ้นกับทั้งจำนวนเวกเตอร์และจำนวนคำรหัสเช่นเดียวกับอัลกอริทึมแอลบีจีจึงมีค่าความซับซ้อนชิงเวลาเท่ากับ $\Theta(mn)$ และอัลกอริทึมแอลบีจีที่ใช้อัลกอริทึมของซีอองเป็นตัวกำหนดค่าเริ่มต้นจึงมีความซับซ้อนเชิงเวลาเท่ากับ $\Theta(mn)$ เช่นเดียวกัน

3.3 ผลการทดลอง

ในหัวข้อนี้จะทำการประเมินสมรรถนะของอัลกอริทึมในสามด้านด้วยกันคือ ความผิดเพี้ยน ความคลาดเคลื่อนความหนาแน่น และความซับซ้อนเชิงเวลา โดยจะทำการประเมินสมรรถนะของอัลกอริทึมในงานวิจัยนี้ทั้งในฐานะของตัวกำหนดค่าเริ่มต้นของอัลกอริทึมแอลบีจีและอัลกอริทึมสำหรับการแบ่งนับเวกเตอร์ เปรียบเทียบกับอัลกอริทึมแอลบีจีที่ได้หนังสือรหัสเริ่มต้นจากการสุ่มและจากงานวิจัยของซีออง [6] สำหรับความผิดเพี้ยนในงานวิจัยนี้จะใช้ค่าความผิดพลาดยกกำลังสองเฉลี่ย (mean square error)

การเปรียบเทียบความผิดเพี้ยนและความคลาดเคลื่อนความหนาแน่นทำการประเมินสมรรถนะโดยทำการทดลองที่ได้สร้างเซตของเวกเตอร์ 3 เซตดังนี้

1. เซตที่ 1 ได้จากเวกเตอร์สองมิติที่มีการกระจายอย่างสม่ำเสมอ (uniform distribution) แบบต่อเนื่องที่มีค่าในช่วง [0,1] จำนวนทั้งสิ้น 160 เวกเตอร์
2. เซตที่ 2 ได้จากคู่ลำดับ(x',y') ที่ได้เก็บตัวอย่างจากเส้นโค้งไซน์ (sine curve) y = sin(x) โดยที่ $x \in [0,4\pi]$ จำนวนทั้งสิ้น 257 เวกเตอร์
3. เซตที่ 3 ได้จากกลุ่มเวกเตอร์ 2 มิติ 4 กลุ่มที่มีขนาดแตกต่างกัน และแต่ละกลุ่มมีการกระจายแบบเกาส์ (Gaussian distribution) โดยที่มีส่วนเบี่ยงเบนมาตรฐาน 1 ทุกกลุ่ม และมีค่าเฉลี่ยในสองแกนและขนาดของแต่ละกลุ่มดังนี้ (1,1) 20 เวกเตอร์ (1,10) 40 เวกเตอร์ (10,1) 60 เวกเตอร์ และ (10,10) 40 เวกเตอร์ จำนวนทั้งสิ้น 160 เวกเตอร์
4. เซตที่ 4 ได้จากกลุ่มเวกเตอร์ 2 มิติ 4 กลุ่มที่มีขนาดเท่ากัน และแต่ละกลุ่มมีการกระจายแบบเกาส์ โดยที่มีส่วนเบี่ยงเบนมาตรฐาน 1 ทุกกลุ่ม และมีค่าเฉลี่ยในสองแกนและขนาดของแต่ละกลุ่มดังนี้ (1,1) 40 เวกเตอร์ (1,10) 40 เวกเตอร์ (10,1) 40 เวกเตอร์ และ (10,10) 40 เวกเตอร์ จำนวนทั้งสิ้น 160 เวกเตอร์

สำหรับเหตุผลที่ใช้เซตของเวกเตอร์ทั้ง 4 เซต มาทำการทดลองก็คือ เวกเตอร์เซตที่ 1 ไว้สำหรับทดสอบกรณีที่เซตของเวกเตอร์มีโครงสร้างไม่ชัดเจน เวกเตอร์เซตที่ 2 ไว้สำหรับทดสอบกรณีที่เซตของเวกเตอร์มีโครงสร้างชัดเจน และสุดท้ายเวกเตอร์เซตที่ 3 และ 4 ไว้สำหรับทดสอบในกรณีที่เซตของเวกเตอร์มีความหนาแน่นแต่ละบริเวณไม่เท่ากันและเท่ากัน โดยได้ทำการทดลองเพื่อสร้างหนังสือรหัสที่มีขนาด 4 เวกเตอร์และ 8 เวกเตอร์ อัลกอริทึมแอลบีจีตั้งให้มีค่าเริ่มเปลี่ยนเท่ากับ 0.005 และตั้งค่าตัวแปรในวิธีของซีอองให้เท่ากับ การทดลองใน [6] โดยผลการทดลองแสดงในตารางที่ 1 และ 2

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 1 ค่าความผิดเพี้ยน

อัลกอริทึมสำหรับการแบ่งนับเวกเตอร์	เซตที่ 1		เซตที่ 2		เซตที่ 3		เซตที่ 4	
	ขนาดหนังสือรหัส		ขนาดหนังสือรหัส		ขนาดหนังสือรหัส		ขนาดหนังสือรหัส	
	4	8	4	8	4	8	4	8
(a) อัลกริทึมแอลบีจีที่ใช้หนังสือรหัสเริ่มต้นจากการสุ่ม	0.0404	0.0203	0.9249	0.3072	11.5735	1.6788	11.2291	1.3909
(b) อัลกอริทึมแอลบีจีที่ใช้หนังสือรหัสเริ่มต้นจากวิธีของซีออง	0.0404	0.0223	0.9249	0.3036	1.7430	1.1782	1.8368	1.1373
(c) อัลกอริทึมที่เสนอในงานวิจัยนี้	0.0407	0.0225	0.9249	0.3030	4.9854	1.2612	1.8368	1.1222
(d) อัลกริทึมแอลบีจีที่ใช้หนังสือรหัสเริ่มต้นจากอัลกอริทึมที่เสนอในงานวิจัยนี้	0.0404	0.0214	0.9249	0.3030	1.7430	1.0261	1.8368	1.1077

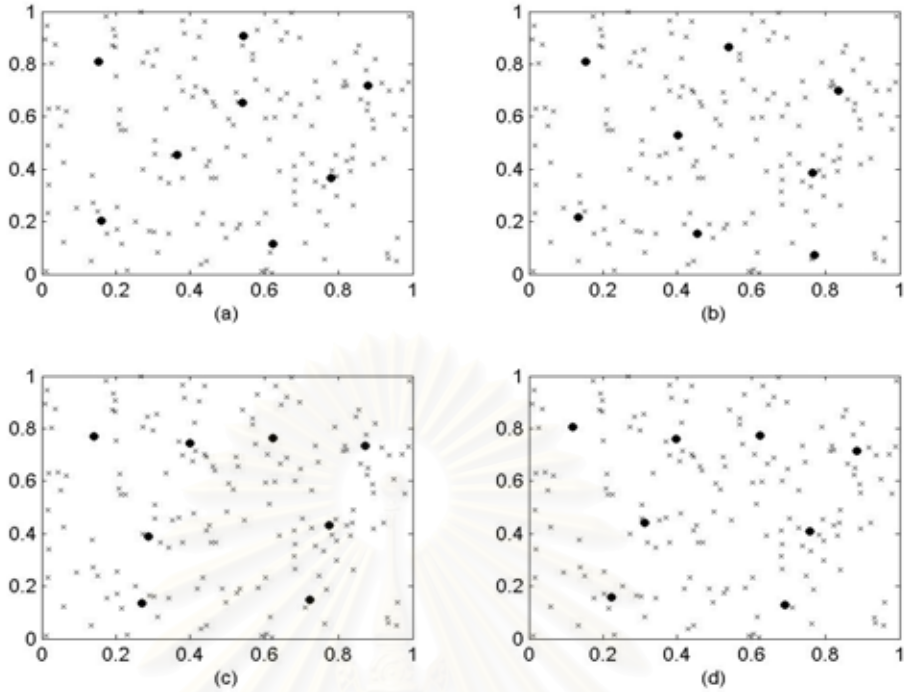
ในตารางที่ 1 แสดงค่าความผิดเพี้ยนของหนังสือรหัสที่มีขนาด 4 และ 8 คำรหัสที่ได้จากอัลกอริทึมแอลบีจีที่ใช้หนังสือรหัสเริ่มต้นจากการสุ่ม อัลกอริทึมแอลบีจีที่ใช้หนังสือรหัสเริ่มต้นจากวิธีของซีออง อัลกอริทึมสำหรับการแบ่งนับเวกเตอร์โดยการประมาณความหนาแน่นและอัลกอริทึมแอลบีจีที่ใช้หนังสือรหัสเริ่มต้นจากอัลกอริทึมสำหรับการแบ่งนับเวกเตอร์โดยการประมาณความหนาแน่น โดยทำการทดลองกับเวกเตอร์ทั้ง 3 เซตที่ได้กล่าวไว้ข้างต้น ในตารางที่ 2 แสดงค่าความคลาดเคลื่อนความหนาแน่นของจากอัลกอริทึมสำหรับการแบ่งนับเวกเตอร์และเซตของเวกเตอร์เดียวกับในตารางที่ 1 โดยในรูปที่ 9, 10, 11 และ 12 แสดงให้เห็นถึงหนังสือรหัสขนาด 8 เวกเตอร์ที่ได้จากการทดลองจากเซตที่ 1, 2, 3 และ 4 ตามลำดับ โดยใช้อัลกอริทึมแอลบีจีที่ได้หนังสือรหัสจากการสุ่ม อัลกอริทึมแอลบีจีที่ได้หนังสือรหัสจากงานของซีออง อัลกอริทึมสำหรับการแบ่งนับเวกเตอร์โดยการประมาณความหนาแน่นและอัลกอริทึมแอลบีจีที่ได้หนังสือรหัสจากอัลกอริทึม

สำหรับการแบ่งนับเวกเตอร์โดยการประมาณความหนาแน่นซึ่งแสดงในรูปย่อย (a) (b) (c) และ (d) ของแต่ละรูปตามลำดับ โดยกากบาทแทนเวกเตอร์ จุดสีดำแทนคำรหัส

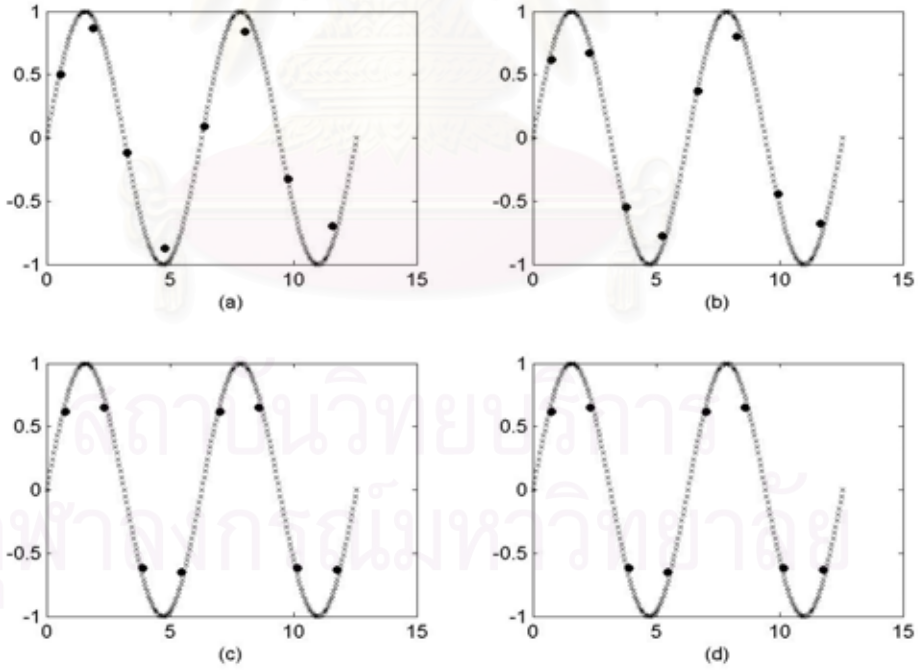
ตารางที่ 2 ค่าความคลาดเคลื่อนความหนาแน่น

อัลกอริทึมสำหรับการแบ่งนับเวกเตอร์	เซตที่ 1		เซตที่ 2		เซตที่ 3		เซตที่ 4	
	ขนาดหนังสือรหัส		ขนาดหนังสือรหัส		ขนาดหนังสือรหัส		ขนาดหนังสือรหัส	
	4	8	4	8	4	8	4	8
(a) อัลกริทึมแอลบีจีที่ใช้หนังสือรหัสเริ่มต้นจากการสุ่ม	5.3852	7.0887	0.4330	5.6885	38.9230	19.1442	25.1396	13.3629
(b) อัลกอริทึมแอลบีจีที่ใช้หนังสือรหัสเริ่มต้นจากวิธีของชีออง	4.3012	5.5902	0.4330	3.1400	22.3607	9.4868	0.0000	9.3941
(c) อัลกอริทึมที่เสนอในงานวิจัยนี้	1.8708	1.5811	0.4330	0.3307	17.7341	3.6742	0.0000	1.8028
(d) อัลกริทึมแอลบีจีที่ใช้หนังสือรหัสเริ่มต้นจากอัลกอริทึมที่เสนอในงานวิจัยนี้	4.3012	4.3012	0.4330	0.3307	22.3607	3.3166	0.0000	3.0414

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

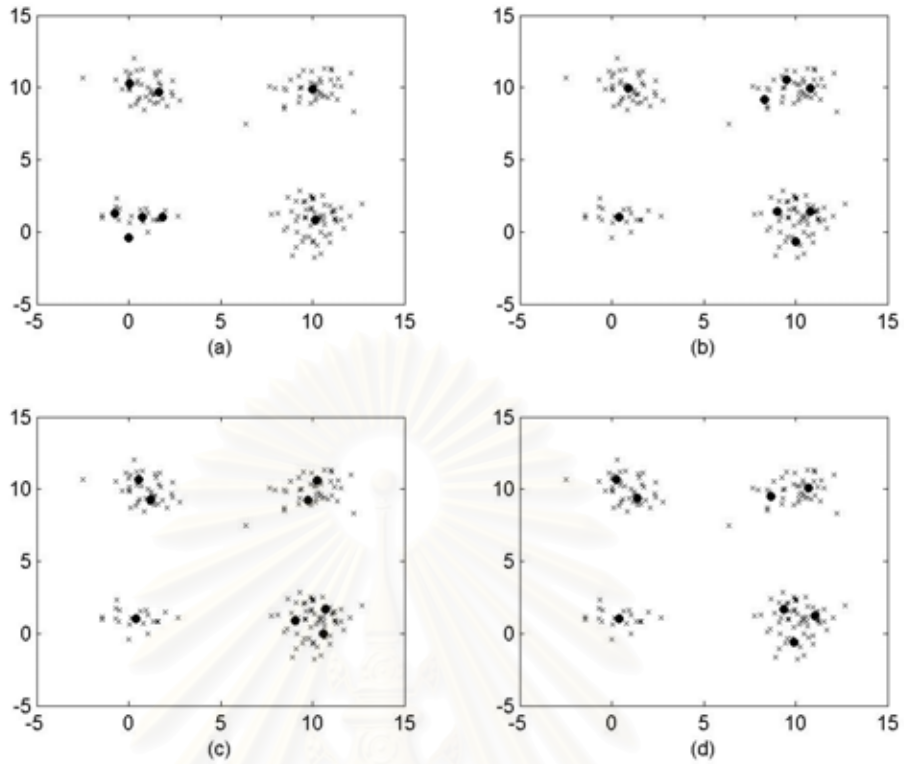


รูปที่ 9 หนังสือรหัสที่ได้จากเซตที่ 1

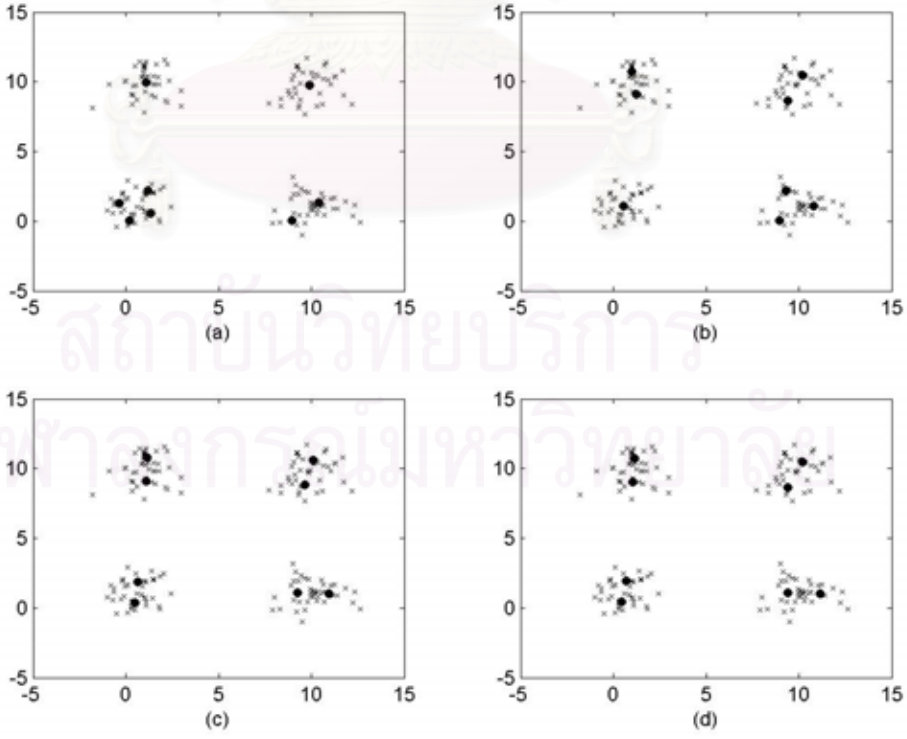


รูปที่ 10 หนังสือรหัสที่ได้จากเซตที่ 2

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 11 หนังสือรหัสที่ได้จากเซตที่ 3



รูปที่ 12 หนังสือรหัสที่ได้จากเซตที่ 4

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

3.4 การวิเคราะห์ผลการทดลอง

จากการทดลองพบว่าอัลกอริทึมแอลบีจีที่ได้หนังสือรหัสจากการสุ่มไม่สามารถให้หนังสือรหัสที่อธิบายการกระจายความหนาแน่นของปัญหาได้ เนื่องจากอัลกอริทึมแอลบีจีได้รับหนังสือรหัสเริ่มต้นที่ไม่สอดคล้องกับลักษณะการกระจายของปัญหา และอัลกอริทึมแอลบีจีพยายามที่จะทำการแบ่งกั้นเพื่อให้เวกเตอร์ในผลแบ่งกั้นมีค่าใกล้เคียงกับคำรหัสในแต่ละรอบมากกว่าพยายามให้ในแต่ละผลแบ่งกั้นมีขนาดใกล้เคียงกัน เมื่อพิจารณาค่าความผิดเพี้ยนและค่าคลาดเคลื่อนความหนาแน่นอัลกอริทึมนี้ก็ยังไม่น่าพอใจนัก

เมื่อพิจารณาอัลกอริทึมแอลบีจีที่ได้หนังสือรหัสเริ่มต้นจากงานของซีออง จะพบว่าอัลกอริทึมนี้มีค่าความผิดเพี้ยนค่อนข้างน้อยเมื่อเทียบกับอัลกอริทึมอื่น แต่ก็ให้ค่าความคลาดเคลื่อนการกระจายที่มากเมื่อเทียบกับวิธีอื่น เนื่องจากวิธีของซีอองจะให้หนังสือรหัสเริ่มต้นที่สอดคล้องกับการกระจายเชิงพื้นที่ของปัญหา และเมื่อนำมาใช้เป็นค่าเริ่มต้นให้กับอัลกอริทึมแอลบีจีจึงปรับค่าความผิดเพี้ยนให้ลดลงไปได้ค่อนข้างมาก อัลกอริทึมนี้จะให้ผลดียิ่งขึ้นกับปัญหาที่มีการกระจายเชิงพื้นที่ชัดเจนดังเช่นเวกเตอร์เซตที่ 3 และ 4 แต่อัลกอริทึมนี้ก็ยังไม่คำนึงถึงความหนาแน่นเช่นกัน โดยจะสังเกตได้ว่าจำนวนคำรหัสในแต่ละกลุ่มจะไม่เป็นไปตามสัดส่วน

สำหรับอัลกอริทึมสำหรับการแบ่งนับเวกเตอร์โดยการประมาณความหนาแน่นสามารถให้หนังสือรหัสที่แสดงให้เห็นถึงความหนาแน่นของปัญหาได้ เนื่องจากกลไกของอัลกอริทึมพยามยามที่จะทำการแบ่งกั้นให้ได้ผลแบ่งกั้นที่มีขนาดเท่ากัน แต่ค่าความผิดเพี้ยนที่ได้จากอัลกอริทึมนี้จัดว่ามากเมื่อเทียบกับปัญหาที่บริเวณของปัญหามีความหนาแน่นต่างกันอย่างเห็นได้ชัด เช่น กรณีของเวกเตอร์เซตที่ 3 อัลกอริทึมนี้จะทำการแบ่งกั้นโดยไม่พิจารณาว่าในแต่ละผลแบ่งกั้นมีเวกเตอร์ที่มีความห่างกันของเวกเตอร์เพียงใด

สุดท้ายอัลกอริทึมแอลบีจีที่ใช้หนังสือรหัสเริ่มต้นจากอัลกอริทึมสำหรับการแบ่งนับเวกเตอร์โดยการประมาณความหนาแน่น วิธีนี้จะเป็นการผสมกันระหว่างสองจุดมุ่งหมายคือการกระจายเชิงพื้นที่และการกระจายความหนาแน่น จึงให้ค่าความผิดเพี้ยนและความคลาดเคลื่อนการกระจายค่อนข้างน้อยทั้งสองค่า เนื่องจากเมื่อใช้อัลกอริทึมสำหรับการแบ่งนับเวกเตอร์โดยการประมาณความหนาแน่นจะให้หนังสือรหัสเริ่มต้นที่มาจากความหนาแน่นของปัญหา จากนั้นอัลกอริทึมแอลบีจีจึงทำการปรับแต่งการแบ่งกั้นเพื่อให้สอดคล้องกับการกระจายเชิงพื้นที่ให้มากขึ้น

3.5 การทดสอบลักษณะการกระจายของหนังสือรหัส

เพื่อที่จะทดสอบดูว่าหนังสือรหัสที่ได้จากอัลกอริทึมที่เสนอจะสามารถใช้เป็นตัวแทนของเวกเตอร์ต้นฉบับทางด้านการกระจายตัวของเวกเตอร์ได้หรือไม่ โดยได้ใช้การทดสอบความเหมาะสมของการแจกแจง (test of goodness of fit) [25] การทดสอบนี้เป็นการทดสอบว่าความถี่ของตัวแปรสุ่ม Z ที่ได้จากการทดลองเป็นไปตามความถี่ที่ได้จากสมมติฐานเกี่ยวกับแจกแจงความน่าจะเป็นที่ได้ตั้งไว้หรือไม่ ความถี่เหล่านี้มีค่าเท่ากับ $N \cdot P(Z = z)$ โดยจะทดสอบว่าความแตกต่างของความถี่จากการทดลอง และความถี่ที่คาดว่าจะได้ตามทฤษฎีมีนัยสำคัญหรือไม่ โดยใช้การทดสอบไคสแควร์ (chi-square test) ดังขั้นตอนต่อไปนี้

1. หาค่า χ^2 เป็นค่าของตัวแปรสุ่ม Z^2 ซึ่งมีการแจกแจงใกล้เคียงกับการแจกแจงไคสแควร์มากที่สุดจากสมการ $\chi^2 = \sum_{i=1}^{k} \frac{(o_i - e_i)^2}{e_i}$ โดยที่ o_i e_i และ k เขียนแทนความถี่ที่ได้จากการทดลอง ความถี่ทีได้ตามทฤษฎีตามลำดับ และจำนวนเซลล์
2. ถ้า $\chi^2 < \chi^2_{\alpha,v}$ จะยอมรับสมมุติฐานว่าความถี่การทดลองเป็นไปตามความถี่ที่ได้จากทฤษฎี โดย $\chi^2_{\alpha,v}$ เป็นค่าที่ได้จากตารางการแจกแจงไคสแควร์ α เป็นระดับนัยสำคัญ (significant) v เป็นระดับขั้นความเสรี (degree of freedom) ซึ่ง $v = k - m - 1$ โดย k = จำนวนเซลล์ m = จำนวนตัวแปรที่ต้องประมาณค่าเพื่อหาค่า e_i

ถ้าความถี่ที่ได้จากการทดลองมีค่าใกล้เคียงกับความถี่ที่คาดว่าจะได้ตามทฤษฎี χ^2 จะมีค่าน้อย ถ้าความถี่ต่างกันมากค่าของ χ^2 ก็จะมากด้วย ฉะนั้นจึงได้กำหนดขอบเขตที่ยอมรับสมมุติฐานได้ เมื่อ $\chi^2 < \chi^2_{\alpha,v}$ และโดยทั่วจะกล่าวว่าการทดสอบมีนัยสำคัญ เมื่อ $\alpha = 0.05$ และกล่าวว่าการทดสอบมีนัยสำคัญสูงเมื่อ $\alpha = 0.01$

การนำการทดสอบความเหมาะสมของการแจกแจงมาใช้ทดสอบว่าหนังสือรหัสที่ได้จากอัลกอริทึมที่เสนอจะสามารถใช้เป็นตัวแทนของเวกเตอร์ต้นฉบับได้หรือไม่สามารถอธิบายได้ดังนี้ โดยจะให้ความถี่ที่ได้จากทฤษฎีในแต่ละเซลล์มีค่าเท่ากับจำนวนของเวกเตอร์ในแต่ละผลแบ่งกั้น และให้ความถี่ที่ได้จากการทดลองในแต่ละเซลล์เท่ากับ จำนวนเวกเตอร์ทั้งหมดต่อจำนวนหนังสือรหัส โดยในตารางที่ 4 และ 5 แสดงให้เห็นค่าของ χ^2 ที่ได้จากหนังสือรหัสขนาด 16 และ 32 คำรหัส ที่ใช้เซตทดลองที่ 1, 2, 3 และ 4

การทดสอบในตารางที่ 3 จะใช้ค่าจากค่า $\chi^2_{0.01,15}$ เพื่อทดสอบการยอมรับสมมุติฐาน เนื่องมี $\alpha = 0.01$ และ $v = 16 - 0 - 1 = 15$ และเมื่อดูค่าจากในตารางค่าวิกฤตของการแจกแจงไคสแควร์พบว่า $\chi^2_{0.01,15}$ มีค่าเท่ากับ 30.578 จึงยอมรับสมมุติฐานด้วยนัยสำคัญสูงว่าหนังสือรหัสในตารางที่ 3 สามารถเป็นตัวแทนของเวกเตอร์ได้ และเมื่อพิจารณาค่าในตารางที่ 4 จะใช้ค่าจากค่า

$\chi^2_{0.01,31}$ เพื่อทดสอบการยอมรับสมมุติฐาน เนื่องมี $\alpha = 0.01$ และ $v = 32 - 0 - 1 = 31$ และเมื่อดูค่าจากในตารางค่าวิกฤตของการแจกแจงไคสแควร์พบว่า $\chi^2_{0.01,32}$ มีค่าเท่ากับ 52.191 จึงยอมรับสมมุติฐานด้วยนัยสำคัญสูงว่าหนังสือรหัสในตารางที่ 4 สามารถเป็นตัวแทนของเวกเตอร์ได้

ตารางที่ 3 ค่า χ^2 ของหนังสือรหัสขนาด 16 คำรหัส

เซตทดลอง	χ^2
1	16.8000
2	0.1206
3	20.4000
4	16.4000

ตารางที่ 4 ค่า χ^2 ของหนังสือรหัสขนาด 32 คำรหัส

เซตทดลอง	χ^2
1	16.8000
2	0.1206
3	20.4000
4	16.4000

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4

ส่วนขยายของอัลกอริทึม

4.1 บทนำ

จากการวิเคราะห์ผลการทดลองจะเห็นได้ว่าอัลกอริทึมที่เสนอสามารถให้หนังสือรหัสที่สามารถแสดงให้เห็นถึงความหนาแน่นของเวกเตอร์ได้ดี แต่อย่างไรก็ตามจะสังเกตเห็นได้ว่าเมื่อค่าความคลาดเคลื่อนความหนาแน่นมีค่าสูงขึ้นค่าความผิดเพี้ยนจะมีค่าลดลง หรืออาจเรียกได้ว่าอัลกอริทึมนั้นแข็งแรงเกินไป (too strong) สำหรับปัญหานี้ เพื่อที่จะทำให้อัลกอริทึมอ่อนแอลงเราได้เสนอส่วนขยายของอัลกอริทึมที่สามารถจะควบคุมการให้ความสำคัญต่อความหนาแน่นได้สำหรับงานประยุกต์ใช้งานบางประเภทที่ต้องการทั้งความหนาแน่นและความผิดเพี้ยน

4.2 ส่วนขยายของอัลกอริทึม

ให้ μ เป็น *ค่ายืดหยุ่น (flexible value)* โดยที่ μ เป็นจำนวนจริงใน [0,1) ส่วนขยายของอัลกอริทึมจะทำการสร้างคำรหัสให้มีจำนวนเวกเตอร์ในแต่ละผลแบ่งกั้นให้อยู่ในช่วง $[\lceil n_{avg}(1-\mu) \rceil, \lceil n_{avg}(1+\mu) \rceil]$ โดยที่ n_{avg} คือ ค่าเฉลี่ยของจำนวนเวกเตอร์ในแต่ละผลแบ่งกั้น กระบวนงานที่เพิ่มเติมจะทำการแบ่งกั้นใหม่โดยให้เวกเตอร์ที่ห่างจากจุดเซนทรอยด์มากที่สุดออกไปจากผลแบ่งกั้นจนกระทั่งขนาดของผลแบ่งกั้นเท่ากับ $\lceil n_{avg}(1-\mu) \rceil$ จากนั้นจึงการปรับค่าเซนทรอยด์ของผลแบ่งกั้นอีกครั้ง และเวกเตอร์ที่ถูกย้ายออกมาจะจัดเข้าไปอยู่กับผลแบ่งกั้นที่มีเซนทรอยด์ที่ใกล้ที่สุด แต่ถ้าผลแบ่งกั้นนั้นมีขนาดเท่ากับ $\lceil n_{avg}(1+\mu) \rceil$ แล้วถูกจัดให้ไปอยู่กับผลแบ่งกั้นที่มีจุดเซนทรอยด์ที่ใกล้รองลงมาแทน จากนั้นจะทำการหาค่าจุดเซนทรอยด์ใหม่อีกครั้งและใช้เป็นหนังสือรหัส โดยกระบวนงานเพิ่มเติมนั้นเขียนสรุปเป็นขั้นตอนได้ดังนี้

ส่วนขยายของอัลกอริทึมสำหรับการแบ่งนับเวกเตอร์โดยใช้การประมาณความหนาแน่น

อินพุต: เซตของผลแบ่งกั้น $A_1 \ldots A_m$ หนังสือรหัส C และค่าความยืดหยุ่น μ

เอาทพุต: หนังสือรหัสใหม่ C_{NEW}

1. แต่ละผลแบ่งกั้นลดขนาดของผลแย่งกั้นลงโดยให้เวกเตอร์ที่ห่างจากจุดเซนทรอยด์มากที่สุดออกจากผลแบ่งกั้นจนกระทั่งขนาดของผลแบ่งกั้นเท่ากับ $\lceil n_{ang}(1-\mu) \rceil$

2. คำนวณจุดศูนย์ในแต่ละผลแบ่งกั้นอีกครั้ง
3. จัดเวกเตอร์ที่ถูกย้ายออกไปยังผลแบ่งกั้นที่อยู่ใกล้ที่สุดที่ขนาดยังไม่เท่ากับ $\lceil n_{avg}(1+\mu) \rceil$
4. คำนวณจุดเซนทรอยด์ใหม่อีกครั้งและ ใช้เป็นหนังสือรหัส

และแสดงเป็นรหัสเทียมได้ดังนี้

Input : Partitions $A_1 \ldots A_m$, a codebook C and a flexible value μ
Output : A new codebook C_{NEW}

```
ExtendedProcedure(A1…Am,C,μ)
01: begin
02:     for i ← 1 to m do
03:       while |Ai| ≥ ⌈ n_ang (1 − μ) ⌉ do
04:         RemovedVector ← RemoveTheMostDistanceVector(Ai)
05:       end
06:     end
07:     for i ← 1 to m do
08:       CTEMP[i] ← Centriod(Ai)
09:     end
10:     for i ← 1 to |RemovedVector| do
11:       k ← FindNotFullNerestPartition(RemovedVector[i],CTEMP[1…N])
12:       Update(Ak)
14:     end
15:     for i ← 1 to m do
13:     CNEW[i] ← Centriod(Ai)
14:     end
15:     return(CNEW)
16: end
```

โดยส่วนขยายของอัลกอริทึมจะมีความซับซ้อนเชิงเวลาเท่ากับ $\Theta(n - \lceil n_{avg}(1-\mu) \rceil)$ เนื่องจากคำสั่งมาตรเวลาคือ คำสั่งวนรอบในบรรทัดที่ 2-6 ที่จะวนรอบทั้งหมดเท่ากับ $n - \lceil n_{avg}(1-\mu) \rceil$ รอบ

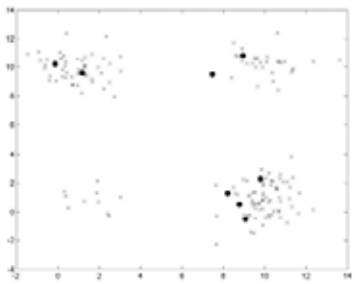
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4.3 ผลการทดลอง

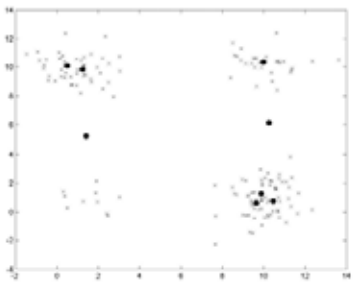
ได้ทำการทดลองโดยใช้ส่วนขยายของอัลกอริทึมต่อจากการทดลองในบทที่ 3 โดยใช้กลุ่มเวกเตอร์ 2 มิติ 4 กลุ่มที่มีขนาดแตกต่างกัน และแต่ละกลุ่มมีการกระจายแบบเกาส์ (Gaussian distribution) โดยที่มีส่วนเบี่ยงเบนมาตรฐาน 1 ทุกกลุ่ม และมีค่าเฉลี่ยในสองแกนและขนาดของแต่ละกลุ่มดังนี้ (1,1) 10 เวกเตอร์ (1,10) 50 เวกเตอร์ (10,1) 70 เวกเตอร์ และ (10,10) 30 เวกเตอร์ จำนวนทั้งสิ้น 160 เวกเตอร์ มาทำการทดลองเนื่องจากเวกเตอร์เซตนี้มีความหนาแน่นของแต่ละกลุ่มแตกต่างกันจึงทำให้เห็นถึงการจัดสรรคำรหัสของอัลกรอทึมได้อย่างชัดเจน ทำการทดลองโดยใช้ค่าความยืดหยุ่น μ เท่ากับ 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 และ 0.9 โดยใช้หนังสือรหัสขนาด 8 คำรหัส ในตารางที่ 5 แสดงค่าความคลาดเคลื่อนความหนาแน่นและค่าความผิดเพี้ยนของหนังสือรหัสที่ค่าความยืดหยุ่นค่าต่างๆ และในรูปที่ 13 แสดงหนังสือรหัสที่ได้จากการทดลอง

ตารางที่ 5 ค่าความคลาดเคลื่อนความหนาแน่นและค่าความผิดเพี้ยน

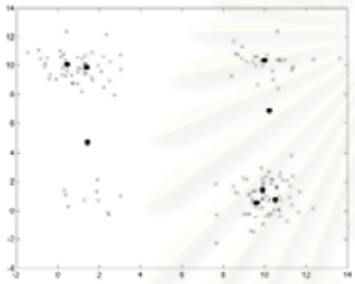
μ	ค่าความผิดเพี้ยน	ค่าคลาดเคลื่อนความหนาแน่น
0.0	4.5765	10.0374
0.1	1.2498	9.4340
0.2	1.2498	9.4604
0.3	1.2633	8.9303
0.4	1.2318	6.9821
0.5	1.2264	6.6895
0.6	1.2603	10.4043
0.7	1.2662	8.3666
0.8	1.2902	10.5712
0.9	1.3941	11.2250



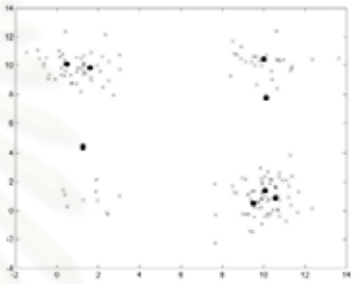
(1) μ = 0.0



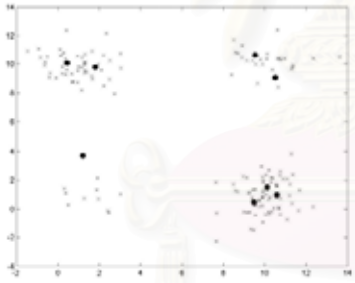
(2) μ = 0.1



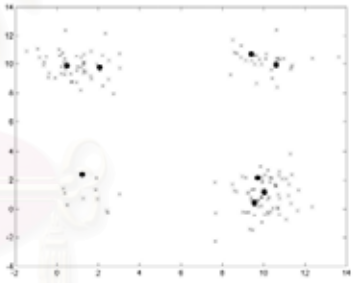
(3) μ = 0.2



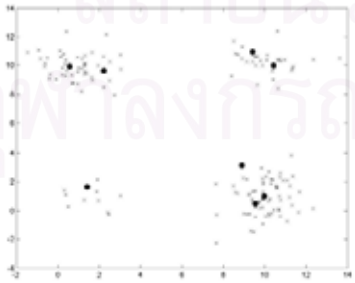
(4) μ = 0.3



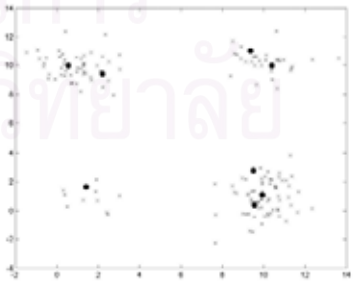
(5) μ = 0.4



(6) μ = 0.5



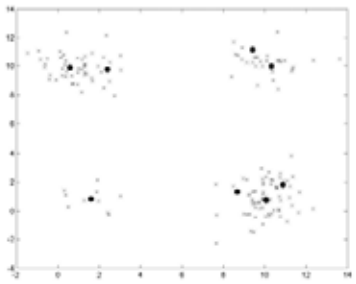
(7) μ = 0.6



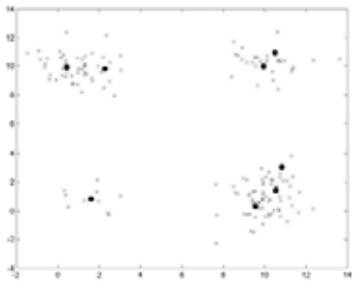
(8) μ = 0.7

รูปที่ 13 หนังสือรหัสที่ได้จากส่วนขยายของอัลกอริทึมที่ค่ายืดหยุ่นต่างๆ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



(9) μ = 0.8



(10) μ = 0.9

รูปที่ 13(ต่อ) หนังสือรหัสที่ได้จากส่วนขยายของอัลกอริทึมที่ค่ายืดหยุ่นต่างๆ

จากผลการทดลองพบว่าในช่วงแรกการเพิ่มค่าความยืดหยุ่นจะทำให้ค่าความคลาดเคลื่อนความหนาแน่นและค่าความผิดเพี้ยนมีค่าลดลงไป จนกระทั่งถึงที่ค่าความยืดหยุ่นเท่ากับ 0.5 แล้วในช่วงหลังจากนั้นเมื่อค่าความยืดหยุ่นเพิ่มขึ้นไปอีกทั้งค่าความคลาดเคลื่อนความหนาแน่นและความผิดเพี้ยนจะเพิ่มขึ้น ทั้งนี้คาดว่าเนื่องจากในช่วงแรกส่วนขยายของอัลกอริทึมยอมให้ขนาดของผลแบ่งกั้นเปลี่ยนแปลงได้ไม่มากนัก จึงทำให้มีเวกเตอร์ย้ายไปอยู่ในผลแบ่งกั้นที่มีคำรหัสอยู่ใกล้กว่าได้จึงทำให้ค่าความผิดเพี้ยนลดลง นอกจากนั้นการที่มีการเปลี่ยนแปลงจำนวนเวกเตอร์น้อยในบางผลแบ่งกั้นจึงมีโอกาสทำให้ความหนาแน่นของแต่ละผลแบ่งกั้นมีขนาดใกล้เคียงกันมากขึ้นค่าความคลาดเคลื่อนความหนาแน่นจึงลดลงตามไปด้วย และต่อมาในช่วงหลังอัลกอริทึมยอมให้ขนาดของผลแบ่งกั้นเปลี่ยนแปลงได้ค่อนข้างมากจึงทำให้มีเวกเตอร์ที่ถูกย้ายมีมากขึ้นจนมีจำนวนมากเกินกว่าขอบเขตที่ตั้งไว้ เวกเตอร์บางเวกเตอร์จึงย้ายไปยังผลแบ่งกั้นที่มีหนังสือรหัสที่ใกล้ที่สุดไม่ได้เนื่องจากผลแบ่งกั้นนั้นเต็มแล้วจึงทำให้ค่าความผิดเพี้ยนลดลง รวมทั้งการที่มีความหนาแน่นของผลแบ่งกั้นต่างกันมากขึ้นจึงทำให้ค่าความคลาดเคลื่อนความหนาแน่นสูงขึ้น จากรูปที่ 12 จะสังเกตได้ว่าเมื่อเพิ่มค่าความยืดหยุ่นขึ้นคำรหัสจากกลุ่ม (10,1) จะค่อยๆเคลื่อนไปยังกลุ่ม (1,1) ทีละน้อยจนในที่สุดก็กลายเป็นตัวแทนของกลุ่ม (1,1) ไป ทั้งนี้เนื่องมาจากส่วนขยายของอัลกอริทึมจะรักษาความหนาแน่นไว้บางส่วนแล้วยอมให้มีการโยกย้ายเวกเตอร์ในผลแบ่งกั้นเพื่อที่จะพยายามลดค่าความผิดเพี้ยน จากที่กล่าวมาจึงสรุปได้ว่าส่วนขยายของอัลกอริทึมสามารถทำให้ทั้งค่าความผิดเพี้ยนและค่าความคลาดเคลื่อนปรับลดลงมาได้เมื่อปรับค่าความยืดหยุ่นขึ้นถึงค่าหนึ่ง

จากการทดลองอาจสรุปเป็นเกณฑ์ในการปรับค่ายืดหยุ่นได้ว่าควรทำการปรับค่าให้สูงขึ้นไปจนกระทั่งค่าของความผิดเพี้ยนลดลงก็จึงหยุด เนื่องจากวัตถุประสงค์ของส่วนขยายของอัลกอริทึมมีไว้เพื่อการปรับให้ได้หนังสือรหัสที่ให้ความสำคัญของความผิดเพี้ยนมากขึ้น

บทที่ 5

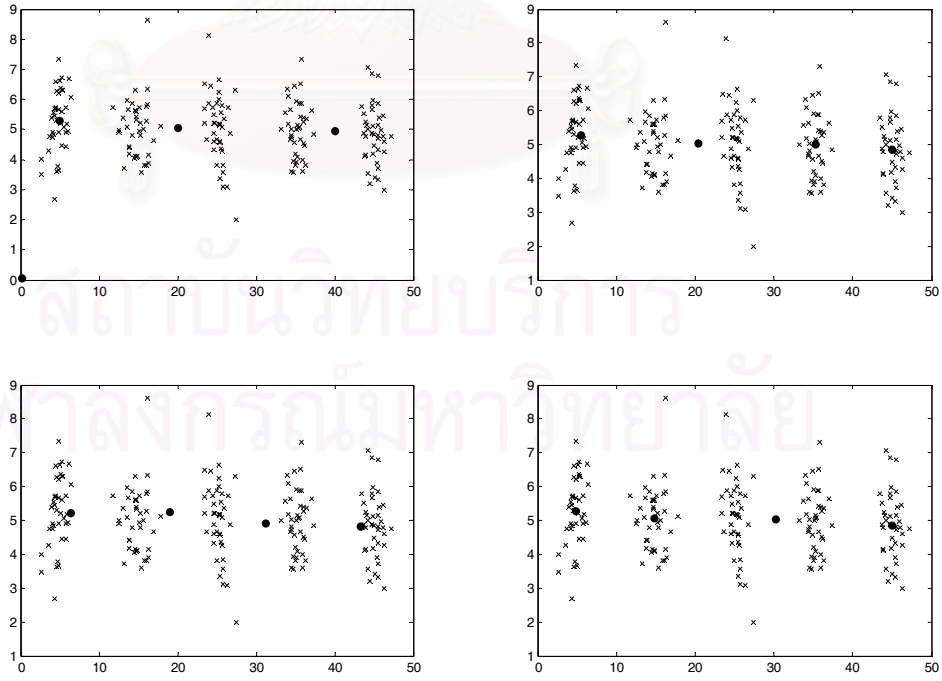
สรุปการวิจัยและข้อเสนอแนะ

5.1 สรุปการวิจัย

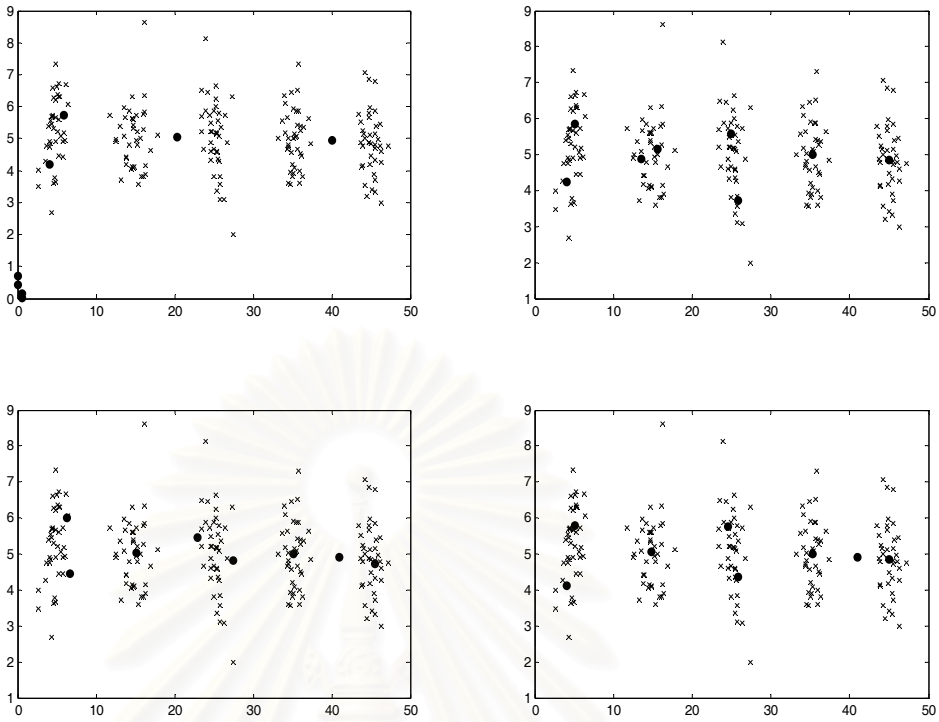
การแบ่งนับเวกเตอร์มีวัตถุประสงค์เพื่อการประมาณเวกเตอร์ด้วยหนังสือรหัสให้มีความใกล้เคียงกับเวกเตอร์ในแต่ละผลแบ่งกั้น ความผิดเพี้ยนได้ถูกนำมาใช้วัดสมรรถนะของการแบ่งนับเวกเตอร์ แต่ยังมีคุณสมบัติบางอย่างที่น่าสนใจต่อการแบ่งนับเวกเตอร์นั้นก็ คือ ความหนาแน่นหรือจำนวนเวกเตอร์ต่อผลแบ่งกั้น ความหนาแน่นสามารถบอกได้ถึงลักษณะการกระจายของปัญหาได้ ซึ่งจะทำให้เข้าใจปัญหาได้มากยิ่งขึ้นตามไปด้วย วิทยานิพนธ์นี้จึงได้ทำการเสนออัลกอริทึมสำหรับการแบ่งนับเวกเตอร์เพื่อที่จะให้หนังสือรหัสแสดงการกระจายความหนาแน่นของปัญหาได้ โดยใช้แนวคิดที่ว่าการกระจายความหนาแน่นสามารถแสดงให้เห็นถึงลักษณะการกระจายของปัญหาได้ อัลกอริทึมที่เสนอใช้แนวคิดของการแบ่งแยกและเอาชนะ โดยจะใช้ความสัมพันธ์เวียนบันเกิดทำการแบ่งกั้นปัญหาออกเป็นผลแบ่งกั้น 2 ผลแบ่งกั้นที่มีจำนวนใกล้เคียงกันจนกระทั้งได้จำนวนผลแบ่งกั้นเท่ากับขนาดหนังสือรหัสที่ต้องการ อัลกอริทึมที่เสนอนี้สามารถนำไปใช้เป็นตัวกำหนดค่าเริ่มต้นของอัลกอริทึมแอลบีจีได้อีกด้วย การเปรียบเทียบสมรรถนะของอัลกอริทึมที่เสนอนี้ทั้งในฐานะของอัลกอริทึมสำหรับการแบ่งนับเวกเตอร์และ ตัวกำหนดค่าเริ่มต้นของอัลกอริทึมแอลบีจี กับอัลกอริทึมแอลบีจีที่ใช้ค่าเริ่มต้นจากการสุ่ม และงานของซีออง จากการทดลองพบว่าอัลกอริทึมที่เสนอนี้ในฐานะของออัลกอริทึมสำหรับการแบ่งนับเวกเตอร์มีค่าความคลาดเคลื่อนความหนาแน่นเฉลี่ยต่ำที่สุด และเมื่อใช้อัลกอริทึมที่เสนอนี้ในฐานะของตัวกำหนดค่าเริ่มต้นของอัลกอริทึมแอลบีจีนั้นให้ค่าความผิดเพี้ยนเฉลี่ยต่ำที่สุด เมื่อทำการพิจารณาความซับซ้อนเชิงเวลาแล้วพบว่าอัลกอริทึมที่เสนอเท่ากับ $\Theta(dn\log m)$ อีก 3 อัลกอริทึมมีความซับซ้อนเชิงเวลาเท่ากัน คือ $\Theta(mn)$ โดยที่ n คือจำนวนเวกเตอร์ m คือจำนวนคำรหัส และ d คือ จำนวนมิติของเวกเตอร์ จากการทดสอบลักษณะการกระจายของเวกเตอร์พบว่าอัลกอริทึมที่เสนอสามารถให้หนังสือรหัสที่มีลักษณะการกระจายใกล้เคียงกับเซตของเวกเตอร์ โดยจะประมาณได้ใกล้เคียงมากขึ้นถ้าขนาดของหนังสือรหัสมากขึ้น ต่อมาจึงได้ทำการปรับปรุงให้อัลกอริทึมสามารถปรับการให้ความสำคัญต่อความหนาแน่นได้มากแค่ไหนโดยการเสนอกระบวนงานเพิ่มเติม ที่ความสามารถปรับค่าความยืดหยุ่นที่จะจำกัดจำนวนเวกเตอร์สูงสุดและต่ำสุดในแต่ละผลแบ่งกั้นได้

5.2 ข้อจำกัดของงานวิจัย

เมื่อใช้จำนวนคำรหัสที่ไม่สอดคล้องกับลักษณะของเวกเตอร์ทั้งหมดจะทำให้การแสดงให้เห็นถึงความหนาแน่นมีความคลาดเคลื่อน เช่น ดังในรูปที่ 14 เวกเตอร์แบ่งออกเป็น 5 กลุ่มที่มีจำนวนเวกเตอร์เท่ากัน เมื่อใช้อัลกอริทึม 4 แบบที่ได้ใช้ทดลองในบทที่ 3 (a) อัลกอริทึมแอลบีจีที่ได้หนังสือรหัสจากการสุ่ม (b) อัลกอริทึมแอลบีจีที่ได้หนังสือรหัสจากงานของซีออง (c) อัลกอริทึมสำหรับการแบ่งนับเวกเตอร์โดยการประมาณความหนาแน่นและ (d) อัลกอริทึมแอลบีจีที่ได้หนังสือรหัสจากอัลกอริทึมสำหรับการแบ่งนับเวกเตอร์โดยการประมาณความหนาแน่น และกำหนดให้ใช้หนังสือรหัสขนาด 4 คำรหัส คำรหัสจะไม่ได้อยู่ในบริเวณที่มีเวกเตอร์อยู่จริง และในกรณีของรูปที่ 15 ที่เปลี่ยนมาใช้ใช้คำรหัส 8 คำรหัส ก็มีคำรหัสที่ไม่ได้อยู่ในบริเวณที่มีเวกเตอร์อยู่จริงเช่นเดียวกัน จงกล่าวได้ว่าถ้ามีจำนวนคำรหัสที่ไม่เหมาะสมกับจำนวนกลุ่มของเวกเตอร์ก็อาจให้ผลลัพธ์ที่ไม่เหมาะสม เช่น มีค่าความผิดเพี้ยนมากกว่าปกติ ซึ่งสังเกตได้จากค่าความผิดเพี้ยนของหนังสือรหัสที่ได้จากอัลกอริทึมที่เสนอจากตารางที่ 1 เมื่อใช้คำรหัส 8 คำรหัสกับเวกเตอร์ 4 กลุ่มที่มีขนาดเท่ากัน มีค่าความผิดเพี้ยนเท่ากับ 1.1222 แต่เมื่อใช้กับเวกเตอร์ 5 กลุ่มที่มีขนาดเท่ากันแล้วค่าความผิดเพี้ยนมีค่าเท่ากับ 552.6625 และเช่นเดียวกับในกรณีที่ใช้อัลกอริทึมที่เสนอเป็นตัวกำหนดค่าเริ่มต้นให้กับอัลกอริทึมแอลบีจี ค่าความผิดเพี้ยนจะเปลี่ยนจาก 1.1077 เมื่อใช้กับเวกเตอร์ 4 กลุ่มขนาดเท่ากันกลายเป็น 317.0450 เมื่อใช้กับเวกเตอร์ 5 กลุ่มขนาดเท่ากัน



รูปที่ 14 คำรหัส 4 คำรหัสที่ได้จากเวกเตอร์ 5 กลุ่ม



รูปที่ 15 คำรหัส 8 คำรหัสที่ได้จากเวกเตอร์ 5 กลุ่ม

5.2 ข้อเสนอแนะ

1. อัลกอริทึมนี้มาจากการแบ่งแยกทีละสองจึงจะให้หนังสือรหัสที่มีขนาดเป็นเลขยกกำลังของ 2 ถ้าต้องการให้ได้หนังสือรหัสมีขนาดเป็นจำนวนเต็มบวกใดๆ สามารถทำได้โดยทำการแบ่งกั้นให้ได้ผลแบ่งกั้นมีจำนวนมากกว่าหนังสือรหัส จากนั้นจึงทำการรวมผลแบ่งกั้น 2 ผลแบ่งกั้นที่มีคำรหัสอยู่ใกล้กันมากที่สุด ทำไปจนกระทั่งได้หนังสือรหัสขนาดที่ต้องการ
2. อัลกอริทึมนี้อาจมีปัญหาการแบ่งกลุ่ม เนื่องจากอัลกอริทึมใช้หลักการแบ่งกั้นให้ผลแบ่งกั้นมีความหนาแน่นใกล้เคียงกัน จึงอาจแทนกลุ่มที่มีความหนาแน่นมากด้วยคำรหัสหลายคำรหัส หรือไม่แทนกลุ่มที่มีความหนาแน่นน้อยมากก็ได้ อาจปรับปรุงได้โดยการกำหนดขนาดของหนังสือรหัสให้เท่ากับจำนวนกลุ่มที่มีอยู่ จากนั้นจึงใช้ส่วนขยายของอัลกอริทึมโดยทำการปรับค่าความยืดหยุ่นจนได้ผลที่ต้องการ

รายการอ้างอิง

1. A. Buzo, A. Jr. Gray, R. Gray, and J. Markel. Speech coding based upon vector quantization. IEEE transactions on acoustics, speech, and signal processing. 5(October 1980):562-574.

2. A. Gersho. On the structure of vector quantizers. IEEE transactions on information theory. 2(March 1982):157-166.

3. R. Gray. Vector quantization. IEEE ASSP magazine. 2(April 1984):4-29.

4. Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. IEEE transactions on communications. 1(January 1980):84-95.

5. I. Katsavounidis, C.-C. J. Kuo, and Z. Zhang. A new initialization technique for generalized Lloyd iteration. IEEE signal processing letters. 10(October 1994):144-146.

6. Huilin Xiong, M. N. S. Swamy, and M. O. Ahmad. Competitive splitting for codebook initialization. IEEE signal processing letters. 5(May 2004):474-477.

7. Lixin Xu, W.Q. Liu, and V. Svetha. A two stage vector quantization approach via self-organizing map. Proceedings of international conference on signal processing, pp. 913-916, 2002.

8. T. Kohonen. Self-organization and associative memory. Springer series in information sciences. New York: Springer, 1988.

9. K. K. Paliwal and B. S. Atal. efficient vector quantization of LPC parameters at 24 bits/frame. IEEE transactions on speech and audio processing. 1(January 1993):3-14.

10. P. C. Cosman, R. M. Gray, and M. Vetterli. Vector quantization of image subbands: a survey. IEEE transactions on image processing. 2(February 1996):202-225.

11. K. Fukunaga. Introduction to Statistical Pattern Recognition Second Ed. Computer science and scientific computing series. New York: Academic Press, 1990.

12. J. M. Jolion, P. Meer, and S. Bataouche. Robust clustering with applications in computer vision. IEEE transactions on pattern analysis and machine intelligence. 8(August 1991):791-802.

13. J. C. Bezdek. Pattern recognition with fuzzy objective function algorithms. Advanced applications in pattern recognition. New York: Plenum Pub Corp, 1981.

14. N. B. Karayiannis and P.-I. Pai. Fuzzy vector quantization algorithms and their application in image compression. IEEE transactions on image processing. 9(September 1995):1193-1201.

15. N. B. Karayiannis and Pin.-I. Pai. Fuzzy algorithms for learning vector quantization. IEEE transactions on neural networks. 5(September 1996):1196-1211.

16. S. P. Lloyd. Least-squares quantization in PCM. IEEE transactions on information theory. 2(March 1982):129-137.

17. G. Patane and M. Russo. Fully automatic clustering system. IEEE transactions on neural networks. 6(November 2002):1285-1298.

18. A. Gercho and R. M. Gray. Vector quantization and signal compression. The Kluwer international series in engineering and computer science. Boston: Kluwer Academic, 1992.

19. T. Hofmann and J. M. Buhmann. Competitive Learning Algorithms for Robust Vector Quantization. IEEE transactions on signal processing. 6(June 1998):1665-1675.

20. B. Fritzke. The LBG-U method for vector quantization - an improvement over LBG inspired from neural network. Neural processing letters. 1(February 1997):35-45.

21. Daeryong Lee, Seonqioon Baek, and Koengmo Sung. Modified K-means algorithm for vector quantizer design. IEEE signal processing letters. 6(January 1997):2-4.

22. M. R. Anderberg. Cluster analysis for applications. Probability and mathematical statistics. New York: Academic Press, 1973.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

23. Zhong-Xuan Yuan, Bo-Ling Xu and Chong-Zhi Yu. Binary quantization of feature vectors for robust text-independent speaker identification. IEEE transactions on speech and audio processing. 1(January 1999):70-78.

24. S. Junsod and A. Surarerks. An approach of binary isomorphic quantization for speaker identification. Proceedings of the 3rd IEEE international symposium on signal processing and information technology, pp. 761-764, 2003.

25. คณาจารย์ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย. ความน่าจะเป็นและสถิติ. ห้างหุ้นส่วนจำกัดพิทักษ์การพิมพ์, 2544.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายอรรคพล นงนุช เกิดเมื่อวันที่ 14 กรกฎาคม พ.ศ. 2523 ที่กรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า จากภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ ในปีการศึกษา 2544 และเข้าศึกษาในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2546

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย