วิธีการจัดกลุ่มข้อมูลที่มีการซ้อนทับกันโดยใช้เทคนิคความหนาแน่น

CLASSIFY METHOD FOR OVERLAPPING DATA USING DENSITY TECHNIQUE

ธนวัฒน์ ภัทรวรเมธ

THANAWAT PHATTARAWORAMET

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2548

ISBN 974-15-1956-7

CLASSIFY METHOD FOR OVERLAPPING DATA USING DENSITY TECHNIQUE

THANAWAT PHATTARAWORAMET

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN COMPUTER ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2005

ISBN 974-15-1956-7

หัวข้อวิทยานิพนธ์	**วิธีการจัดกลุ่มข้อมูลที่มีการซ้อนทับกันโดยใช้เทคนิคความหนาแน่น**
นักศึกษา	**นายธนวัฒน์ ภัทรวรเมธ**
รหัสนักศึกษา	44061610
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
พ.ศ.	2548
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร. บุญธีร์ เครือตราชู

บทคัดย่อ

อัลกอริธึมการรวมกลุ่มข้อมูลมีบทบาทสำคัญสำหรับในงานวิจัยด้านการรู้จำรูปแบบโดยนำมาใช้ในการบ่งชี้ข้อมูลที่จะรู้จำ งานวิจัยนี้ได้ทำการประยุกต์ใช้เทคนิคของการจัดกลุ่มข้อมูลแบบไม่ผู้สอนที่เรียกว่า DBSCAN มาช่วยเพื่อให้การจัดกลุ่มสามารถทำได้กับข้อมูลที่มีการซ้อนทับกัน โดยประยุกต์ใช้พารามิเตอร์ที่เรียกว่า "class ratio" เพื่อให้สามารถจำแนกกลุ่มข้อมูลที่มีการซ้อนทับกันได้อย่างมีประสิทธิภาพ โดยในงานวิจัยนี้จะเรียกวิธีการที่นำเสนอว่า "ODBSCAN" ในการจัดกลุ่มและกำจัดข้อมูลที่เป็น ข้อมูลรบกวน ค่าความคล้ายคลึงกันของข้อมูลที่เพิ่มเข้ามาจะใช้เป็นพารามิเตอร์ที่ใช้ในการบ่งชี้ข้อมูลที่ซ้อนทับกัน

Thesis Title	Classify Method for Overlapping Data using Density Technique
Student	Mr. Thanawat Phattaraworamet
Student ID.	44061610
Degree	Master of Engineering
Programme	Computer engineering
Year	2005
Thesis Advisor	Assoc. Prof. Dr. Boontee Kruatrachue

ABSTRACT

Clustering algorithms are attractive task for identification in pattern recognition. In this paper we introduce a modification of DBSCAN for recognition purpose. The traditional DBSCAN is a unsupervised clustering technique. However, we can adapt it for classification purpose in overlapping data. We use similarity of class ratio applied for handling the overlapping problem, called ODBSCAN. In our approach, the ODBSCAN classifies data and eliminates noise. We use the similarity of class ratio to identify the overlapping data.

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงด้วยดีเนื่องจากกำลังใจและพระคุณอันหาที่สุดมิได้ จากคุณแม่ผู้ล่วงลับ คุณพ่อและพี่ ๆ ข้าพเจ้าขอสำนึกในพระคุณอย่างเป็นที่สุด

วิทยานิพนธ์นี้จะไม่สามารถสำเร็จลุล่วงไปได้หากปราศจากแรงผลักดัน และคำแนะนำที่มีประโยชน์ของ รศ.ดร. บุญธีร์ เครือตราชู ผู้ควบคุมวิทยานิพนธ์ ข้าพเจ้าขอกราบขอบพระคุณเป็นอย่างสูง

ข้าพเจ้าขอกราบขอบพระคุณ คุณครูและอาจารย์ทุกท่านตั้งแต่เล็กจนเติบใหญ่ ที่ได้มอบวิชาความรู้ให้แก่ข้าพเจ้า รวมทั้งคำสั่งสอนและอบรมให้ข้าพเจ้าเป็นคนดี ข้าพเจ้าขอกราบขอบพระคุณเป็นอย่างสูง

ข้าพเจ้าขอขอบคุณสำหรับกำลังใจ คำแนะนำ และประสบการณ์ที่ดีจากพี่ ๆ และเพื่อน ๆ นักศึกษาป.โททุกท่าน และขอขอบคุณ นางสาวปองเกษม พลสันติกุล และนายณรงค์ชัย มุ่งแฝงกลาง ที่ช่วยแก้ไขภาษาในการส่งบทความตีพิมพ์ต่างประเทศ และเรียบเรียงวิทยานิพนธ์ อีกทั้งขอขอบคุณบัณฑิตวิทยาลัย ที่ให้ทุนสนับสนุนการทำวิทยานิพนธ์

สุดท้ายนี้คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับผู้มีพระคุณทุกท่าน หากวิทยานิพนธ์ฉบับนี้มีข้อผิดพลาดประการใดข้าพเจ้าขอน้อมรับไว้เพียงผู้เดียว

ธนวัฒน์ ภัทรวรเมธ

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญรูป VIII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา 1

1.3 สมมุติฐานของการศึกษา 1

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในงานวิจัย 2

1.5 ขอบเขตของการศึกษา 4

1.6 ขั้นตอนของการศึกษา 4

1.7 รายละเอียดในแต่ละบท 5

บทที่ 2 การซ้อนทับกันของกลุ่มข้อมูล และทฤษฎีพื้นฐานของการจัดกลุ่มข้อมูลที่เกี่ยวข้อง 6

2.1 การซ้อนทับกันของกลุ่มข้อมูล 6

2.2 ทฤษฎีพื้นฐานของการจัดกลุ่มข้อมูลที่เกี่ยวข้องกับการจัดกลุ่มข้อมูลที่ซ้อนทับกัน 7

2.2.1 DBSCAN: A Density-Based Technique Clustering Method Based on Connected Regions with Sufficiently High Density 7

2.2.2 Fuzzy c-Means 12

2.2.3 ต้นแบบวิธีการจัดกลุ่มข้อมูลที่มีการซ้อนทับกัน 17

บทที่ 3 การจัดกลุ่มข้อมูลที่มีการซ้อนทับกันโดยใช้เทคนิคความหนาแน่น 24

3.1 อัตราส่วนคลาส (Class Ratio) 24

3.2 ความต่างของอัตราส่วนคลาส (Difference of Class Ratio:DCR) 25

3.2.1 หาจากผลรวมความต่างอัตราส่วนของแต่ละคลาส (DCR Type 1) 26

3.2.2 หาจากผลรวมความต่างของอัตราส่วนแต่ละคลาสที่มีการกำจัดข้อมูลรบกวน (DCR Type 2) 27

สารบัญ (ต่อ)

หน้า

3.2.3 หาจากความต่างสูงสุดของความต่างอัตราส่วนของแต่ละคลาส (DCR Type 3) 27

3.3 นิยามความหนาแน่นของข้อมูลที่มีการประยุกต์ใช้อัตราส่วนคลาสร่วมด้วย 28

3.4 อัลกอริธึม ODBSCAN 29

3.4.1 การทำงานของอัลกอริธึม 29

3.4.2 วิธีหาจุดตัวแทนของคลัสเตอร์ในการแพร่ (Point Representation of Cluster:PRC) 31

3.4.3 วิธีการหาจุดข้างเคียง 32

3.5 การระบุคลัสเตอร์ (Cluster Identification) 33

3.6 การวัดประสิทธิภาพของการจัดกลุ่ม 34

3.6.1 ค่าสัมประสิทธิประสิทธิภาพของการจัดกลุ่ม 34

3.7 การประยุกต์ใช้งาน 35

บทที่ 4 การทดลองการจัดกลุ่มข้อมูลที่มีการซ้อนทับกันเบื้องต้น ของวิธีการ ODBSCAN และวิธีการอื่น 36

4.1 การทดลองเกี่ยวกับลักษณะของผลต่างอัตราส่วนคลาส 36

4.1.1 ผลรวมความต่างอัตราส่วนของแต่ละคลาส (DCR Type 1) 37

4.1.2 ผลรวมความต่างอัตราส่วนของแต่ละคลาสที่กำจัดข้อมูลรบกวน (DCR Type 2) 38

4.1.3 ความต่างอัตราส่วนคลาสสูงสุดที่กำจัดข้อมูลรบกวน (DCR Type 3) 40

4.2 การเปรียบเทียบลักษณะการหาจุดข้างเคียง 41

4.3 การหาวิธีการที่เหมาะสมสำหรับตัวแทนที่ใช้ในการแพร่ 43

4.4 สรุปวิธีที่เหมาะสมสำหรับแต่ละพารามิเตอร์ 45

4.5 การทดลองเปรียบเทียบกับวิธีการอื่น 45

บทที่ 5 การทดลองเกี่ยวกับข้อมูลที่มีการกระจายแบบต่าง ๆ 47

5.1 การทดลองการจัดกลุ่มข้อมูลที่มีกระจายแบบลดลงสม่ำเสมอ (Constant Fade-out Distribution) 47

5.2 การทดลองการจัดกลุ่มข้อมูลที่มีการกระจายแบบเกาส์เซียน (Gaussian Distribution) 48

สารบัญ (ต่อ)

หน้า

5.3 ผลการทดลองการจัดกลุ่มข้อมูลลายมือเขียนภาษาไทย 50

บทที่ 6 สรุปผลการทดลอง และข้อเสนอแนะ 51

6.1 สรุปผลการวิจัย 51

6.2 ข้อเสนอแนะ 51

เอกสารอ้างอิง 52

ภาคผนวก งานวิจัยที่ได้รับการตีพิมพ์ 53

ประวัติผู้เขียน 59

สารบัญตาราง

ตารางที่ หน้า

3.1 แสดงอัตราส่วนคลาสของจุด p และ q 26

4.1 ผลการทดลองเปรียบเทียบค่า MaxDiff ของ DCR Type 1 38

4.2 ผลการทดลองเปรียบเทียบค่า MaxDiff ของ DCR Type 2 38

4.3 ผลการทดลองเปรียบเทียบค่า MaxDiff ของ DCR Type 3 40

4.4 ผลการทดลองของลักษณะการหาจุดข้างเคียง 43

4.5 ผลการทดลองของตัวแทนที่ใช้ในการแพร่ 44

4.6 ผลการทดลองเปรียบเทียบการจัดกลุ่มระหว่าง FCM และ ODBSCAN 45

สารบัญรูป

รูปที่ หน้า

1.1 แสดงรัศมี lower และ upper 2

1.2 แสดงการกำหนดพื้นที่ของส่วน lower และ upper สำหรับข้อมูลซ้อนทับกัน 2

1.3 แสดงความสัมพันธ์แบบความสามารถในการไปถึงโดยใช้ความหนาแน่น และความสัมพันธ์แบบการเชื่อมกันโดยใช้ความหนาแน่น 3

1.4 ผลการจัดกลุ่มของ DBSCAN 4

2.1 แสดงข้อมูลที่มีการซ้อนทับกันของกลุ่มผู้ชายและผู้หญิง 6

2.2 แสดงการระบุบริเวณที่มีการซ้อนทับกัน 7

2.3 เซตข้อมูลตัวอย่าง 8

2.4 แสดงคุณสมบัติ directly density-reachable แบบอสมมาตร [2] 9

2.5 แสดงคุณสมบัติ density-reachable แบบอสมมาตรและคุณสมบัติ density-connected [2] 10

2.6 แสดงอัลกอริธึมของ DBSCAN 11

2.7 แสดงการทำงานของ Fuzzy c-Mean 13

2.8 แสดงจุดของข้อมูลที่มีมิติเดียว 13

2.9 แสดงฟังก์ชันความเป็นสมาชิกของ A ของการจัดกลุ่มแบบ k-mean 13

2.10 แสดงฟังก์ชันความเป็นสมาชิกของ A ของการจัดกลุ่มแบบ Fuzzy c-Mean 14

2.11 แสดงเมตริกซ์ U ของ k-mean และ Fuzzy c-Mean 14

2.12 แสดงเงื่อนไขค่าความเป็นสมาชิกเริ่มต้นของแต่ละจุด 15

2.13 แสดง u_{ij} และจุด c_j เมื่ออัลกอริธึมทำงานได้ 8 รอบ โดยใช้ค่า m=2 และ $\varepsilon = 0.3$ 16

2.14 แสดง u_{ij} และจุด c_j เมื่ออัลกอริธึมทำงานได้ 37 รอบ โดยใช้ค่า m=2 และ $\varepsilon = 0.01$ 16

2.15 แสดงการเปรียบเทียบ prototype ของงานวิจัยของ M.A. Abou-Nasr (ซ้าย)และ prototype ของงานวิจัยต้นแบบสำหรับข้อมูลที่มีการซ้อนทับกัน (ขวา) 17

2.16 แสดงการกำหนดพื้นที่ของส่วน lower และ upper สำหรับข้อมูลซ้อนทับกัน 17

2.17 แสดงข้อมูลที่กำลังทดสอบเป็นกลุ่มเดียวกับกลุ่มทดสอบ 19

2.18 แสดงข้อมูลที่กำลังทดสอบต่างกลุ่มกับกลุ่มทดสอบ 19

2.19 แสดงการปรับค่า threshold เมื่อข้อมูลที่กำลังทดสอบอยู่กลุ่มเดียวกัน และอยู่ในพื้นที่ Lower 20

2.20 แสดงการปรับค่า threshold เมื่อข้อมูลที่กำลังทดสอบอยู่กลุ่มเดียวกัน และอยู่ในพื้นที่ Upper 20

สารบัญรูป (ต่อ)

รูปที่ หน้า

2.21 แสดงการปรับค่า threshold เมื่อข้อมูลที่กำลังทดสอบอยู่กลุ่มเดียวกัน และอยู่นอกพื้นที่ Upper (ระยะทางน้อยกว่า 1.2 เท่าของ Upper) 21

2.22 แสดงการปรับค่า threshold เมื่อข้อมูลที่กำลังทดสอบอยู่กลุ่มเดียวกัน และอยู่นอกพื้นที่ Upper (ระยะมากกว่า 1.2 เท่าของ Upper) 21

2.23 แสดงการปรับค่า threshold เมื่อข้อมูลที่กำลังทดสอบอยู่ต่างกลุ่มกัน กรณีข้อมูลใหม่อยู่ในพื้นที่ Lower22

2.24 แสดงการปรับค่า threshold เมื่อข้อมูลที่กำลังทดสอบอยู่ต่างกลุ่มกัน กรณีข้อมูลใหม่อยู่ในพื้นที่ Upper 22

2.25 แสดงการปรับค่า threshold เมื่อข้อมูลที่กำลังทดสอบอยู่ต่างกลุ่มกันกรณีข้อมูลใหม่อยู่นอกพื้นที่ Upper 23

3.1 แสดงอัตราส่วนคลาส 25

3.2 แสดงการค่าความต่างอัตราส่วนคลาสของจุด p และ q 25

3.3 แสดง $\varepsilon - neighborhood\ with\ class\ ratio$ ของจุด p โดยมีค่า MaxDiff = 0.4..... 28

3.4 แสดง $\varepsilon - neighborhood\ with\ class\ ratio$ ของจุด p โดยมีค่า MaxDiff = 0.2..... 29

3.5 แสดงอัลกอริธึมของ ODBSCAN 30

3.6 ฟังก์ชันหาจุดตัวแทนของคลัสเตอร์ในการแพร่โดยใช้จุดแรกที่ทำให้เกิดคลัสเตอร์ 31

3.7 ฟังก์ชันหาจุดตัวแทนของคลัสเตอร์ในการแพร่โดยใช้ค่าเฉลี่ย 31

3.8 ฟังก์ชัน QueryRegion ที่ใช้ลักษณะการหาจุดข้างเคียงเฉพาะจุดที่คล้ายกับจุดตัวแทนในการแพร่ 32

3.9 ฟังก์ชัน QueryRegion ที่ใช้ลักษณะการหาจุดข้างเคียงทั้งหมดเมื่อจุดที่คล้ายจุดตัวแทนในการแพร่เกินค่า MinPts 33

3.10 ฟังก์ชัน QueryRegion ที่ใช้ลักษณะการหาจุดข้างเคียงพื้นที่ทั้งหมดเมื่อจุดที่คล้ายจุดตัวแทนในการแพร่เป็นส่วนหลัก 33

4.1 แสดงข้อมูลที่ใช้ในการทดลอง 36

4.2 แสดงผลการทดลองจากการจัดกลุ่มด้วยลักษณะของผลต่างอัตราส่วนคลาสแบบผลรวมความต่างอัตราส่วนของแต่ละคลาส โดยใช้ MaxDiff ค่าต่าง ๆ 37

4.3 แสดงผลการทดลองจากการจัดกลุ่มด้วยลักษณะของผลต่างอัตราส่วนคลาสแบบผลรวมความต่างอัตราส่วนของแต่ละคลาสที่กำจัดข้อมูลรบกวน โดยใช้ MaxDiff ค่าต่าง ๆ 39

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.4 แสดงผลการทดลองจากการจัดกลุ่มด้วยลักษณะของผลต่างอัตราส่วนคลาสแบบความต่างอัตราส่วนคลาสสูงสุด โดยใช้ MaxDiff ค่าต่าง ๆ 40

4.5 แสดงผลการทดลองจากการจัดกลุ่มโดยใช้ลักษณะการหาจุดข้างเคียงเฉพาะจุดที่มีความคล้ายกับจุดที่เป็นจุดตัวแทน 41

4.6 แสดงผลการทดลองจากการจัดกลุ่มโดยใช้ลักษณะการหาจุดข้างเคียงทั้งหมดเมื่อจุดที่คล้ายกับจุดตัวแทนเกินค่า MinPts 42

4.7 แสดงผลการทดลองจากการจัดกลุ่มโดยใช้ลักษณะการหาจุดข้างเคียงทั้งหมดเมื่อมีจุดที่คล้ายกับจุดตัวแทนเป็นส่วนหลัก 42

4.8 แสดงผลการทดลองจากการจัดกลุ่มโดยการใช้ตัวแทนในการแพร่โดยใช้จุดแรกที่ทำให้เกิดคลัสเตอร์ 44

4.9 แสดงผลการทดลองจากการจัดกลุ่มโดยการใช้ตัวแทนในการแพร่โดยใช้ค่าเฉลี่ย 44

4.10 แสดงผลการจัดกลุ่มโดยใช้ FCM 46

4.11 แสดงผลการจัดกลุ่มโดยใช้ ODBSCAN 46

5.1 ข้อมูลที่ใช้ในการทดสอบที่มีการกระจายแบบลดลงสม่ำเสมอ 47

5.2 ผลการจัดกลุ่มข้อมูลที่มีการกระจายแบบลดลงสม่ำเสมอ 48

5.3 ข้อมูลที่ใช้ในการทดสอบที่มีการกระจายแบบเกาส์เซียน 49

5.4 ผลการจัดกลุ่มข้อมูลที่มีการกระจายแบบเกาส์เซียน 49

5.5 ข้อมูลลายมือเขียนภาษาไทยที่ใช้ในการทดลอง 50

5.6 ผลการทดลองจัดกลุ่มข้อมูลลายมือเขียนภาษาไทยตามอัลกอริธึมของ ODBSCAN 50

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

สืบเนื่องจากงานวิจัยที่ศึกษาเกี่ยวกับการจัดกลุ่มที่ใช้ feature based [1, 2, 3] เมื่อทำการจัดกลุ่มข้อมูลโดยมี feature ที่จำกัด เช่น การจัดกลุ่มเพื่อจำแนกลายมือเขียนระหว่างตัวอักษร ข กับ บ ของผู้เขียนบางส่วนมีลักษณะความคล้ายคลึงกัน เป็นผลให้การจัดกลุ่มด้วยวิธีการเดิม(ไม่มีการซ้อนทับกันของคลาส) ได้กลุ่มย่อยๆ จำนวนมาก ณ บริเวณที่มีความคล้ายคลึงกัน และกลุ่มที่ได้ไม่สามารถบ่งชี้ถึงความคลุมเครือของข้อมูลที่บริเวณนั้นได้ จากงานวิจัยของ B. Kruatrachue, K. Siriboon and K. Warunsin [1] ที่ได้ศึกษาการจัดกลุ่มข้อมูลในลักษณะดังกล่าว แต่พบว่าผลที่ได้จากการจัดกลุ่มของวิธีการนี้มีกลุ่มข้อมูลเป็นจำนวนมาก และไม่สามารถบ่งบอกถึงความคลุมเครือของข้อมูลได้ ซึ่งในปัจจุบันยังไม่มีวิธีการที่เหมาะสมในการจัดกลุ่มข้อมูลที่มีการปนหรือซ้อนทับกันได้อย่างมีประสิทธิภาพ วิธีการที่จะนำเสนอในวิทยานิพนธ์นี้สามารถบ่งบอกถึงความคลุมเครือของข้อมูลบริเวณดังกล่าว และให้กลุ่มข้อมูลจำนวนน้อยลง โดยใช้หลักการของอัตราส่วนคลาส (Class Ratio) เพื่อจัดการเกี่ยวกับความเหมือนกันภายในกลุ่มข้อมูล วิธีการที่นำเสนอเรียกว่า "Overlapping DBSCAN" หรือ "ODBSCAN" [4] ซึ่งเป็นวิธีการที่ประยุกต์มาจากการจัดกลุ่มข้อมูลแบบ Unsupervised Learning ที่มีชื่อว่า DBSCAN [2] ให้เป็นวิธีการแบบ Supervised Learning

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

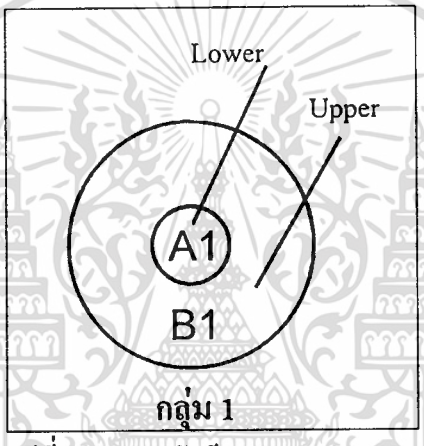
ความมุ่งหมายและวัตถุประสงค์ของวิทยานิพนธ์นี้เพื่อพัฒนาวิธีการจัดกลุ่มข้อมูลที่มีการซ้อนทับกัน ให้สามารถบ่งบอกถึงบริเวณของกลุ่มข้อมูลที่มีความคลุมเครือ ขอบเขตของบริเวณที่เกิดความคลุมเครือ ลักษณะของความคลุมเครือ และได้กลุ่มที่มีลักษณะความคลุมเครือเท่ากันทั้งกลุ่ม

1.3 สมมุติฐานของการศึกษา

ข้อมูลที่จะนำมาจัดกลุ่มมีการซ้อนทับกันมาก โดยข้อมูลที่ซ้อนทับกันนั้นไม่ถือว่าเป็นข้อมูลรบกวน (Noise) ดังนั้นผลของการจัดกลุ่มที่มีข้อมูลใน Feature Space ที่คล้ายกันและข้อมูลที่มีอัตราส่วนของคลาสลาเบลที่คล้ายกันให้อยู่ในกลุ่มข้อมูลเดียวกัน

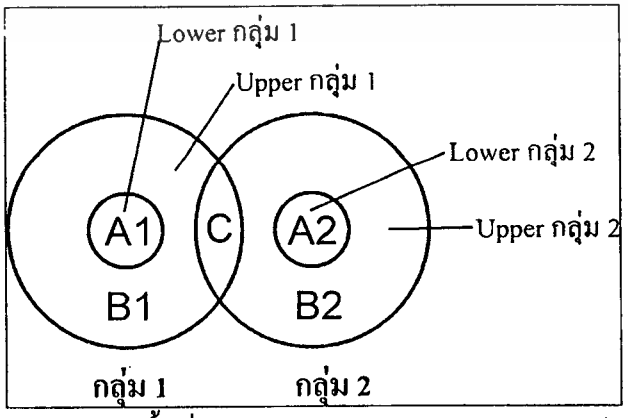
1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในงานวิจัย

วิทยานิพนธ์นี้นำเสนอแนวคิดการจัดกลุ่มข้อมูลที่มีการซ้อนทับกันโดยใช้เทคนิคความหนาแน่นที่เรียกว่า "Overlapping DBSCAN" หรือ "ODBSCAN" ซึ่งประยุกต์แนวคิดเริ่มต้นมาจากงานวิจัยของ B. Kruatrachue, K. Siriboon and K. Warunsin [1] ที่มีชื่อว่า "Modified Neural Network Classifier" หรือ "MNNC" ที่กล่าวถึงการจัดกลุ่มข้อมูลที่มีการซ้อนทับกัน โดยต้นแบบกลุ่มใดๆ ประกอบด้วยค่าของจุดศูนย์กลาง และรัศมี 2 รัศมี คือ lower และ upper ดังรูปที่ 1.1 รัศมี lower คือรัศมีที่ครอบคลุมพื้นที่ A1 โดยที่ข้อมูลที่อยู่ในพื้นที่ A1 เป็นข้อมูลที่มีความแน่นอนว่าเป็นข้อมูลกลุ่มที่ 1 ส่วน รัศมี upper คือรัศมีที่ครอบคลุมพื้นที่ B1 ข้อมูลที่อยู่ในส่วนนี้เป็นส่วนที่มีโอกาสเป็นข้อมูลกลุ่มที่ 1 หรือกลุ่มอื่นๆ



รูปที่ 1.1 แสดงรัศมี lower และ upper

จากรูปที่ 1.1 เมื่อนำไปใช้ในการจัดกลุ่มข้อมูลที่มีการซ้อนทับกันจะได้กลุ่มข้อมูลดังรูปที่ 1.2 ส่วน A1 และ A2 เป็นข้อมูลของกลุ่มที่ 1 และกลุ่มที่ 2 ตามลำดับ ส่วน B1 เป็นพื้นที่ที่มีโอกาสเป็นข้อมูลของกลุ่มอื่นๆ หรือ กลุ่มที่ 1 ส่วน B2 เป็นพื้นที่ที่มีโอกาสเป็นข้อมูลของกลุ่มอื่นๆ หรือกลุ่มที่ 2 ส่วน C เป็นพื้นที่ที่มีโอกาสเป็นข้อมูลของกลุ่มที่ 1 หรือ กลุ่มที่ 2 ก็ได้

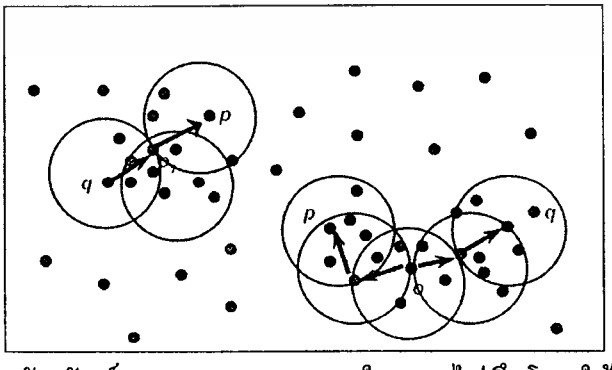


รูปที่ 1.2 แสดงการกำหนดพื้นที่ของส่วน lower และ upper สำหรับข้อมูลซ้อนทับกัน

วิธีการ MNNC จะนำค่าข้อมูลที่ใช้สำหรับการเรียนรู้มาทดสอบทีละค่าตามลำดับ โดยจะทำการทดสอบทั้งในส่วน lower และ upper ของทุกกลุ่มที่มีการสร้างขึ้นในอัลกอริธึม ถ้ามีข้อมูลอยู่ผิดกลุ่มในพื้นที่ lower หรือ upper ของกลุ่มใดๆ จะลดรัศมีดังกล่าวลง เพื่อให้ขอบเขตของกลุ่มอธิบายข้อมูลได้ถูกต้อง ถ้าข้อมูลใหม่ตกอยู่ในกลุ่มที่ถูกต้องแล้วจะทำการปรับค่าน้ำหนักเฉลี่ยของกลุ่มที่ถูกต้องนั้น และถ้าข้อมูลไม่ได้อยู่ในพื้นที่ของ lower และ upper กลุ่มใดๆ จะมีการสร้างกลุ่มใหม่ให้กับข้อมูล ขั้นตอนการเรียนรู้นี้จะทำซ้ำกันจนกระทั่งไม่มีการสร้างกลุ่มใหม่หรือรัศมีของทุกกลุ่มไม่มีการเปลี่ยนแปลง ในงานวิจัย [1] มีข้อจำกัดคือ รูปร่างของกลุ่มข้อมูลที่ได้จะเป็นวงกลม (สำหรับข้อมูล 2 มิติ) ซึ่งถ้าข้อมูลมีการซ้อนทับกันมากและมีการกระจายสูงจะทำให้การจัดกลุ่มไม่มีประสิทธิภาพ

อย่างไรก็ตามได้มีงานวิจัยของ Easter M., Kriegel H.-P., Sander J. and Xu X.[2] ที่นำเสนออัลกอริธึมที่มีชื่อว่า "Density-Based Spatial Clustering of Applications with Noise" หรือ "DBSCAN" ซึ่งสามารถจัดกลุ่มของข้อมูลโดยไม่จำกัดรูปร่างของกลุ่มข้อมูลได้ โดยมีแนวคิดคือจะทำการขยายขอบเขตของการรวมกลุ่มให้ครอบคลุมไปยังกลุ่มข้อมูลที่มีความหนาแน่นสูงพอที่จะเป็นสมาชิกของกลุ่มได้ และทำการค้นหากลุ่ม (Cluster) ที่มีรูปแบบต่างๆในฐานข้อมูลแบบสปาเชียล (Spatial) ที่มีข้อมูลรบกวน โดยอัลกอริธึมจะทำการนิยามกลุ่มเสมือนเป็นเซตที่มีความเป็นไปได้มากที่สุดของจุดที่ความหนาแน่นสามารถเชื่อมต่อไปถึง (Maximal set of density-connected points)

ความสามารถในการแผ่ไปถึงข้อมูลโดยใช้ความหนาแน่น (Density reachability) คือ ความสามารถในการเชื่อมโยงถึงหรือถ่ายทอดความสัมพันธ์ไปถึงซึ่งกันและกันของกลุ่มที่มีความหนาแน่น ความสัมพันธ์ดังกล่าวเป็นลักษณะความสัมพันธ์แบบอสมมาตร โดยวัตถุแกน (Core object) เท่านั้นที่มีความสามารถในการโยงไปถึงกันโดยใช้ความหนาแน่นได้ แต่ความสัมพันธ์แบบการเชื่อมกันโดยใช้ความหนาแน่น (Density connectivity) เป็นความสัมพันธ์แบบสมมาตรกัน ดังแสดงในรูปที่ 1.3



รูปที่ 1.3 แสดงความสัมพันธ์แบบความสามารถในการไปถึงโดยใช้ความหนาแน่นและความสัมพันธ์แบบการเชื่อมกันโดยใช้ความหนาแน่น

DBSCAN จะทำการค้นหาจุดข้อมูลเพื่อจัดกลุ่ม โดยทำการตรวจสอบค่า $\varepsilon - neighborhood$ ของแต่ละจุดในฐานข้อมูล ถ้า $\varepsilon - neighborhood$ ของจุด p นั้นๆ มีค่ามากกว่า *MinPts* (ค่าของจำนวนจุดที่น้อยที่สุดที่สามารถถือเป็นคลัสเตอร์) กลุ่มใหม่ที่มีจุด p อยู่จะถูกสร้างขึ้นมา จากนั้น DBSCAN จะทำการ วนทำซ้ำเพื่อหาวัตถุที่สามารถเป็นความสามารถแผ่ถึงโดยตรง (directly density-reachable) สำหรับวัตถุแกน p โดยทั้งนี้อาจต้องทำการรวมกลุ่มที่มีค่า density-reachable เล็ก ๆ เข้าไปด้วย การทำงานจะสิ้นสุดเมื่อไม่มีจุดใหม่ที่จะเพิ่มเข้าไปในกลุ่มแล้ว

ประสิทธิภาพของอัลกอริธึม DBSCAN ทั้งในการทดลองกับข้อมูลที่ทำการสังเคราะห์และจากฐานข้อมูลจริง ซึ่งได้จาก SEQUOIA 2000 [5] สามารถจัดกลุ่มข้อมูลที่มีรูปร่างไม่จำกัดได้อย่างมีประสิทธิภาพ ดังรูปที่ 1.4

รูปที่ 1.4 ผลการจัดกลุ่มของ DBSCAN

1.5 ขอบเขตของการศึกษา

ขอบเขตของวิทยานิพนธ์นี้คือ การศึกษาการจัดกลุ่มข้อมูลที่มีการซ้อนทับกัน และพัฒนาวิธีการจัดกลุ่มข้อมูลให้สามารถระบุบริเวณที่มีการซ้อนทับกันได้ โดยจะทำการทดสอบวิธีการจัดกลุ่มกบัข้อมูลที่มีการสังเคราะห์ให้มีการซ้อนทับกัน และข้อมูลที่ได้มาจาก Feature ขอวตัวอักษรลายมือเขียนที่มีการซ้อนทับกัน

1.6 ขั้นตอนของการศึกษา

ศึกษาวิธีการจัดกลุ่มข้อมูลแบบต่างๆ เพื่อกำหนดหัวข้อ เป้าหมาย วัตถุประสงค์ และขอบเขตของการทำวิทยานิพนธ์

ทำการทดลอง นำวิธีการที่ได้ศึกษามาทำการทดลอง เพื่อหาแนวทางใหม่ในการจัดกลุ่มข้อมูลที่มีการซ้อนทับกัน

ทดลองตามแนวทางที่ได้วางไว้ เพื่อพิสูจน์แนวคิดว่าได้ผลตามที่ต้องการหรือไม่ พร้อมทั้งเปรียบเทียบกับการจัดกลุ่มแบบ Fuzzy c-Mean

สรุปผลการทดลอง

1.7 รายละเอียดในแต่ละบท

ในวิทยานิพนธ์นี้แบ่งเนื้อหาการนำเสนอออกเป็น 6 บทดังนี้

บทที่ 1 กล่าวถึงความเป็นมาและความสำคัญของปัญหา แนวความคิดที่นำเสนอเพื่อแก้ไขปัญหา วัตถุประสงค์ ขอบเขตของงานวิจัย ขั้นตอนการศึกษา และรายละเอียดของเนื้อหาต่างๆ ในวิทยานิพนธ์

บทที่ 2 กล่าวถึงการซ้อนทับกันของกลุ่มข้อมูล และทฤษฎีพื้นฐานของจัดกลุ่มข้อมูล โดยจะกล่าวถึงการซ้อนทับกันของกลุ่มข้อมูล แนวคิดของวิธีการต้นแบบวิธีการจัดกลุ่มข้อมูลที่มีการซ้อนทับกัน DBSCAN และ Fuzzy c-Mean และ MNNC ตามลำดับ

บทที่ 3 กล่าวถึงหลักการทำงานของ ODBSCAN โดยอธิบายลักษณะการจัดกลุ่ม โดยใช้ความหนาแน่น การหาค่าอัตราส่วนของกลุ่มข้อมูล และการหาตัวแทนกลุ่มที่ใช้ในการแพร่

บทที่ 4 กล่าวถึงการทดลองเพื่อหาค่าพารามิเตอร์สำหรับวิธีการ ODBSCAN และการทดลองกับวิธีการอื่น

บทที่ 5 การทดลองเกี่ยวกับข้อมูลที่มีการกระจายแบบต่าง ๆ

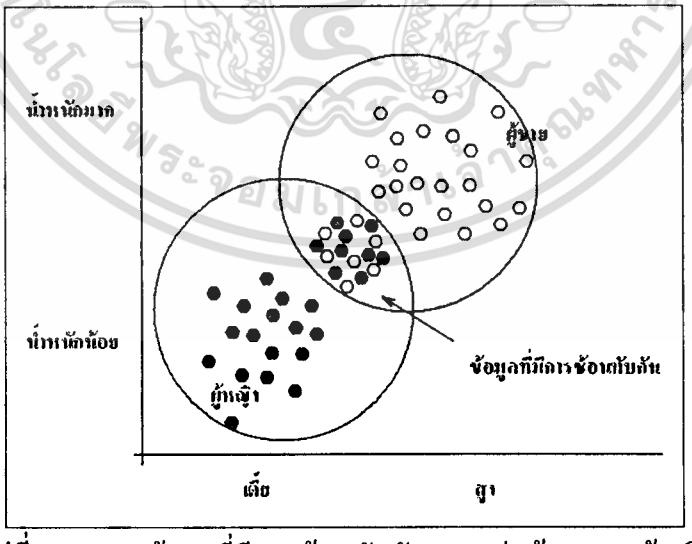
บทที่ 6 สรุปผลการทดลอง และข้อเสนอแนะ

บทที่ 2
การซ้อนทับกันของกลุ่มข้อมูล และทฤษฎีพื้นฐานของการจัดกลุ่มข้อมูลที่เกี่ยวข้อง

บทนี้จะกล่าวถึงความหมายของการซ้อนทับกันของกลุ่มข้อมูล ความสำคัญของการจัดกลุ่มข้อมูลที่มีการซ้อนทับกัน และทฤษฎีพื้นฐานที่เป็นแนวคิดเริ่มต้นของวิทยานิพนธ์นี้ ซึ่งประกอบด้วย 3 วิธีการ คือ DBSCAN [2] Fuzzy c-Mean[6] และต้นแบบวิธีการจัดกลุ่มข้อมูลที่มีการซ้อนทับกัน [1] ตามลำดับ

2.1 การซ้อนทับกันของกลุ่มข้อมูล

วิธีการจัดกลุ่มข้อมูลในปัจจุบันจะมีลักษณะผลลัพธ์ของกลุ่มข้อมูลซึ่งไม่ยอมให้ข้อมูลที่ไม่ใช่ชนิดเดียวกันอยู่ในกลุ่มเดียวกัน ในทางปฏิบัติถ้าหากข้อมูลต่างชนิดกันมีความคล้ายคลึงกันตามคุณลักษณะที่ทำการพิจารณา ตัวอย่างเช่น การจำแนกชายหญิงโดยใช้คุณลักษณะของน้ำหนักและส่วนสูง ดังแสดงในรูปที่ 2.1 จะพบว่าเราสามารถแบ่งข้อมูลได้ 2 กลุ่มคือ กลุ่มผู้ชาย และกลุ่มผู้หญิง และส่วนที่ไม่สามารถระบุได้ว่าเป็นกลุ่มผู้ชาย หรือผู้หญิง เนื่องจากข้อมูลซ้อนทับกัน เช่น ลักษณะของผู้ชายที่ตัวเตี้ย และน้ำหนักน้อย กับผู้หญิงที่ตัวสูง และน้ำหนักมาก ดังนั้นถ้าใช้วิธีการจัดกลุ่มข้อมูลแบบเดิมจะไม่สามารถจัดกลุ่มข้อมูลที่มีการซ้อนทับกันได้



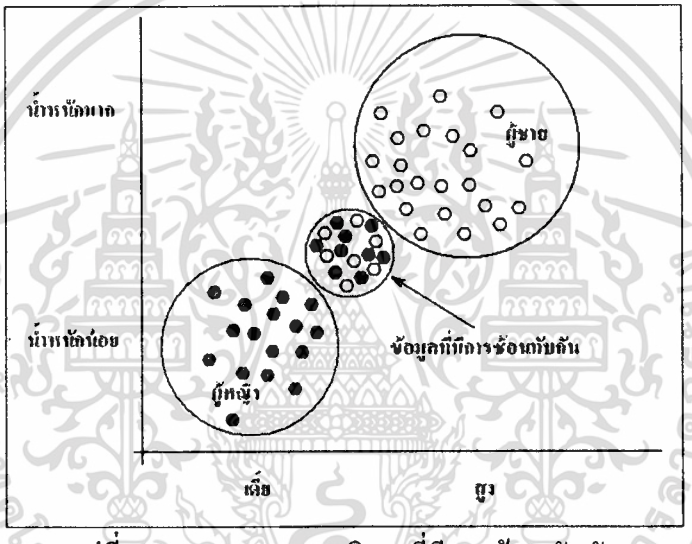
รูปที่ 2.1 แสดงข้อมูลที่มีการซ้อนทับกันของกลุ่มผู้ชายและผู้หญิง

จากตัวอย่างเรื่องการระบุชายหญิงดังที่กล่าวมาข้างต้น สามารถนิยามการจัดกลุ่มข้อมูลที่มีการซ้อนทับกันได้ดังนี้

นิยาม 2.1 บริเวณพื้นที่รัศมีขนาดเล็กใดๆ ที่มีข้อมูลของคลาสที่ต่างกันอยู่ภายใน จะเรียกบริเวณนี้ว่าพื้นที่ที่เกิดการซ้อนทับกันของข้อมูล

นิยาม 2.2 วิธีจัดกลุ่มข้อมูลที่มีการซ้อนทับกันคือ วิธีการที่จะบ่งชี้ถึงลักษณะและพื้นที่ที่เกิดการซ้อนทับกันของข้อมูลกันได้

ในรูปที่ 2.2 แสดงตัวอย่างของกลุ่มข้อมูลที่มีการซ้อนทับกันคือ ข้อมูลที่อยู่ในวงกลมเล็ก การจัดกลุ่มข้อมูลแบบนี้จะมีประโยชน์ในการลดความผิดพลาดจากการจำแนกกลุ่มข้อมูลในบริเวณนี้ได้ เนื่องจากโดยทั่วไป บริเวณดังกล่าวไม่ควรจะถูกระบุว่าเป็นข้อมูลชนิดใดชนิดหนึ่งเพียงอย่างเดียว อีกทั้งยังสามารถบ่งบอกถึงคุณลักษณะของการซ้อนทับกันได้ กล่าวคือการบ่งบอกถึงความน่าจะเป็นของข้อมูลแต่ละชนิดในบริเวณที่มีการซ้อนทับกัน



รูปที่ 2.2 แสดงการระบุบริเวณที่มีการซ้อนทับกัน

2.2 ทฤษฎีพื้นฐานของการจัดกลุ่มข้อมูลที่เกี่ยวข้องกับการจัดกลุ่มข้อมูลที่ซ้อนทับกัน

ในส่วนนี้จะกล่าวถึงทฤษฎีพื้นฐานของการจัดกลุ่มข้อมูลที่เป็นแนวความคิดของวิธีการที่นำเสนอของวิทยานิพนธ์นี้ ซึ่งประกอบด้วย วิธีการ DBSCAN [2] Fuzzy c-Mean [6] และต้นแบบวิธีการจัดกลุ่มข้อมูลที่มีการซ้อนทับกัน [1]

2.2.1 DBSCAN: A Density-Based Technique Clustering Method Based on Connected Regions with Sufficiently High Density

อัลกอริธึมการจัดกลุ่มที่อาศัยความหนาแน่นของข้อมูลที่เป็นที่รู้จักกันดีคือ อัลกอริธึม DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [2] ซึ่งในวิทยานิพนธ์นี้ได้ใช้เป็นพื้นฐานในการจัดกลุ่มข้อมูล โดยจะอธิบายคำนิยามพื้นฐานต่างๆ ที่เกี่ยวข้องกับการจัดกลุ่มข้อมูลที่ใช้ความหนาแน่นของข้อมูล และกระบวนการของอัลกอริธึมตามลำดับ

โดยก่อนที่จะกล่าวถึงกระบวนการต่างๆ ของอัลกอริธึมที่นำมาใช้ในการจัดกลุ่มนั้น เราต้องทราบนิยามที่เกี่ยวข้องกับความหนาแน่นของข้อมูล ซึ่งเป็นพื้นฐานในการจัดกลุ่มข้อมูลโดย DBSCAN ซึ่งจะกล่าวถึงในส่วนต่อไปนี้

2.2.1.1 คำนิยามต่างๆ ของการจัดกลุ่มโดยใช้ความหนาแน่น

เมื่อพิจารณาข้อมูลดังรูปที่ 2.3 เราสามารถที่จะระบุข้อมูลเป็นคลัสเตอร์ (กลุ่มข้อมูลที่มีความคล้ายกันจะอยู่ในกลุ่มเดียวกัน) และข้อมูลรบกวนได้โดยง่าย เหตุผลที่ทำให้เราสามารถรู้ได้ว่าจุดใดพิจารณาเป็นคลัสเตอร์และจุดใดที่พิจารณาเป็นข้อมูลรบกวน ก็เนื่องมาจากความหนาแน่นของจุดที่ปรากฏเป็นคลัสเตอร์นั้นจะมีความหนาแน่นสูงกว่าบริเวณอื่นๆ

รูปที่ 2.3 เซตข้อมูลตัวอย่าง

ในส่วนต่อไปนี้จะเป็นการกล่าวถึงนิยามและความหมายของคลัสเตอร์ และข้อมูลรบกวน นิยามของกลุ่มและอัลกอริธึมที่จะกล่าวต่อไปในงานวิจัยนี้ได้นำยูคลิเดียนสเปซ (Euclidean space) สองและสามมิติมาประยุกต์ใช้ โดยแนวคิดหลักมาจากแต่ละจำนวนจุดใกล้เคียงกันที่อยู่ในรัศมี Eps ($\varepsilon - neighborhood$) จะต้องมีจำนวนอย่างน้อยเท่ากับจำนวนจุดที่น้อยที่สุดเกินค่า threshold ที่กำหนด

นิยาม 2.3 ε **-neighborhood**

$\varepsilon - neighborhood$ ของจุด p แสดงโดยใช้ $N_{Eps}(p)$ ที่ถูกกำหนดโดย

$$N_{Eps}(p) = \{q \in D \mid dist(p,q) \le Eps\}.$$

โดยที่

D	คือ ฐานข้อมูลทั้งหมด
Eps	คือ ระยะรัศมีที่กำหนด
dist(p,q)	คือ ฟังก์ชันการหาระยะห่างระหว่างจุด p และ q

$\varepsilon - neighborhood$ ของจุด p คือ จุดที่อยู่ภายในรัศมี Eps ของจุด p ซึ่งในการที่จะทำการสร้างคลัสเตอร์ขึ้นมาได้นั้น จำเป็นจะต้องมีจำนวนจุดของ $\varepsilon - neighborhood$ มากกว่าหรือ

เท่ากับ MinPts ซึ่งเรียกจุด p ว่า "จุดแกน (Core point)" ส่วนจุดที่มีจำนวนจุดของ $\varepsilon - neighborhood$ น้อยกว่า MinPts เราเรียกว่า "จุดขอบ (Border point)"

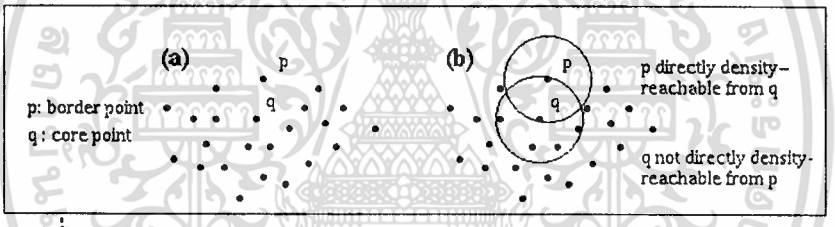
นิยาม 2.4 Directly density-reachable

จุด p จะเป็น directly density-reachable จากจุด p ตามค่า Eps และ MinPts ถ้า

1) $q \in N_{Eps}(p)$ and

2) $|N_{Eps}(p)| \geq$ MinPts

directly density-reachable คือ คุณสมบัติของจุดใดๆ ที่เป็น $\varepsilon - neighborhood$ ของจุด p โดยที่จำนวนจุดใน $\varepsilon - neighborhood$ มีมากกว่าหรือเท่ากับ MinPts ซึ่งคุณสมบัตินี้จะเป็นคุณสมบัติแบบสมมาตรเมื่อเป็นคู่ของจุดใดๆ ที่เป็นจุดแกน (จุดที่มี $\varepsilon - neighborhood$ มากกว่าหรือเท่ากับ MinPts) กล่าวคือ เมื่อ p เป็นจุดแกนและเป็น directly density-reachable จากจุด q ที่เป็นจุดแกนแล้ว จุด q จะเป็น directly density-reachable จากจุด p ด้วยเช่นกัน แต่จะเป็นคุณสมบัติแบบอสมมาตรเมื่อจุดใดจุดหนึ่งนั้นเป็นจุดขอบ ดังแสดงในรูปที่ 2.4

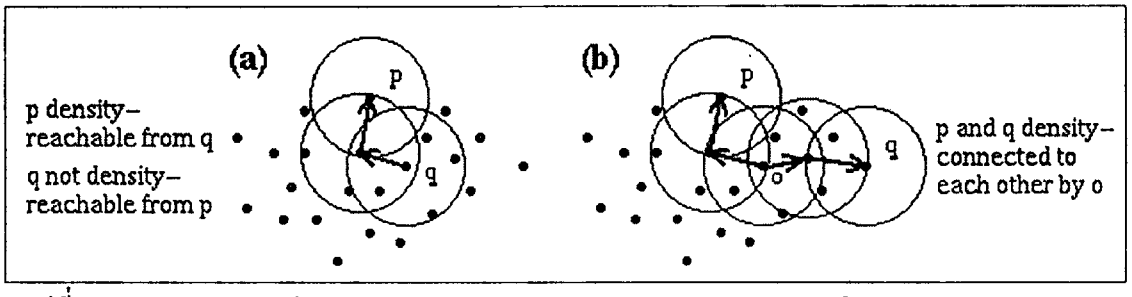


รูปที่ 2.4 แสดงคุณสมบัติ directly density-reachable แบบอสมมาตร [2]

นิยาม 2.5 Density-reachable

จุด p จะเป็น density-reachable จากจุด q ถ้ามีความต่อเนื่องของจุด $p_1, ..., p_n$, $p_1 = q$, $p_n = p$ โดยที่ p_{i+1} เป็น directly density-reachable จากจุด p_i

คุณสมบัติ Density-reachable เป็นคุณสมบัติที่ขยายจากคุณสมบัติ directly density-reachable โดยรูปที่ 2.5 แสดงความสัมพันธ์ของจุดตัวอย่างที่เป็นแบบอสมมาตร แต่จะเป็นแบบสมมาตรเมื่อจุดทั้งคู่เป็นจุดแกน สำหรับจุดที่เป็นจุดขอบที่อยู่ในคลัสเตอร์เดียวกันอาจจะไม่มีความสัมพันธ์แบบ density reachable กันก็ได้ อย่างไรก็ตามอาจจะมีจุดแกนที่อยู่ในคลัสเตอร์ดังกล่าวมีความสัมพันธ์แบบ density-reachable ของทั้งสองจุด ซึ่งคุณสมบัตินี้เรียกว่า density-connected



รูปที่ 2.5 แสดงคุณสมบัติ density-reachable แบบอสมมาตรและคุณสมบัติ density-connected [2]

นิยาม 2.6 Density-connected

จุด p จะเป็น density-connected ถึงจุด q ซึ่งเป็นไปตามค่า Eps และ MinPts ถ้ามีจุด o ซึ่งทั้งจุด p และ q เป็น density-reachable จากจุด o ซึ่งเป็นไปตามค่า Eps และ MinPts

Density-connected เป็นความสัมพันธ์แบบสมมาตร สำหรับจุดที่คุณสมบัติ density-reachable ความสัมพันธ์ของ density-connected ก็จะสะท้อนไปได้ดังรูปที่ 2.5 b

นิยาม 2.7 คลัสเตอร์ (Cluster)

ให้ D เป็นเซตของจุดทั้งหมด คลัสเตอร์ C ซึ่งเป็นไปตามค่า Eps และ MinPts จะเป็นซับเซตไม่ว่างของ D และเป็นไปตามเงื่อนไขต่อไปนี้

1) $\forall p,q : \text{if } p \in C$ และ q เป็น density-reachable จาก p ซึ่งเป็นไปตามค่า Eps และ MinPts ดังนั้น $q \in C$ จะเรียกว่า "Maximality"

2) $\forall p, q \in C$: q เป็น density-connected ถึง p ซึ่งเป็นไปตามค่า Eps และ MinPts จะเรียกว่า "Connectivity"

ดังนั้นการนิยามคลัสเตอร์โดยใช้ความหนาแน่น จึงสามารถนิยามโดย คลัสเตอร์ คือ เซตของจุดที่เป็น density-connected กัน

นิยาม 2.8 ข้อมูลรบกวน (Noise)

ให้ $C_1, \ldots, C_k$ เป็นคลัสเตอร์ของ D ซึ่งเป็นไปตามค่า Eps, MinPts และ MaxDiff โดยที่ i = 1,...., k. แล้วเราสามารถนิยามข้อมูลรบกวน คือ จุดที่อยู่ใน D แต่ไม่เป็นสมาชิกของคลัสเตอร์ใดเลย ซึ่งจะเขียนได้ดังนี้ $\text{noise} = \{p \in D \mid \forall i : p \notin C_i\}$.

2.2.1.2 อัลกอริธึมของ DBSCAN

ในส่วนนี้จะทำการอธิบายอัลกอริธึมของ DBSCAN โดยการทำงานของอัลกอริธึมนี้จะเริ่มจากทำการหาจุด p ที่ยังไม่ได้ทำการจัดกลุ่ม แล้วทำการตรวจสอบว่าจุด p เป็นจุดแกนหรือไม่ ถ้าไม่จะทำการกำหนดให้จุด p เป็นข้อมูลรบกวน แต่ถ้าเป็นจุดแกนอัลกอริธึมจะทำการสร้างคลัสเตอร์ขึ้นใหม่แล้วทำการกำหนดจุด p และ $\varepsilon - neighborhood$ ของจุด p ให้เป็นคลัสเตอร์ที่สร้างใหม่นั้น จากนั้นทำการสร้างคิว (Queue) ขึ้นใหม่โดยมีสมาชิกเป็น $\varepsilon - neighborhood$ ของ p แล้วทำ

การดึงจุด q จากคิว ทำการตรวจสอบว่าจุด q เป็นจุดแกนหรือไม่ ถ้าใช่ให้ทำการกำหนด $\varepsilon - neighborhood$ ของจุด q ให้เป็นคลัสเตอร์ใหม่นั้น และทำการเพิ่ม $\varepsilon - neighborhood$ ของ q เข้าไปในคิว ทำการหาจุด q ใหม่จนกว่าคิวจะว่าง และทำการหาจุด p จนกว่าไม่มีจุด p ที่ยังไม่ได้จัดกลุ่ม โดยมี Psudo code ดังรูปที่ 2.6

```
1.  DBSCAN
2.     ClusterID=0
3.     FOR p=1 TO N DO
4.        IF p is CorePoint THEN
5.           ClusterID=ClusterID+1
6.           Assign Cluster of p = ClusterID
7.           Assign Cluster of ε – neighborhood of p = ClusterID
8.           Add ε – neighborhood of p to Queue
9.           DO
10.             q = Dequeue Queue
11.             IF q is Core Point Then
12.                Assign Cluster of ε – neighborhood of q = ClusterID
13.                Add ε – neighborhood of q to Queue
14.             END IF
15.          LOOP UNTIL Queue empty
16.       ELSE
17.          Assign Cluster of p = NOISE
18.       END IF
19.    END FOR
20. END
```

รูปที่ 2.6 แสดงอัลกอริธึมของ DBSCAN

2.2.2 Fuzzy c-Means [6]

Fuzzy c-Means (FCM) เป็นวิธีการของการรวมกลุ่ม (Clustering) ที่อนุญาตให้ข้อมูลหนึ่งข้อมูลสามารถเป็นสมาชิกของคลัสเตอร์ได้มากกว่าหนึ่งคลัสเตอร์ โดยใช้ membership เป็นตัวบอกว่ามีความเป็นสมาชิกในคลัสเตอร์ใดเท่าไหร่ จากลักษณะเด่นของ FCM สามารถที่จะนำมาประยุกต์ใช้ในการจัดกลุ่มข้อมูลที่ซ้อนทับกันได้

โดยวิธีการนี้ (ซึ่งพัฒนาโดย Dunn [9] และปรับปรุงโดย Bezdek [6]) นิยมใช้ในการรู้จำรูปแบบ ซึ่งอยู่บนพื้นฐานการหาค่าต่ำสุดของฟังก์ชันวัตถุประสงค์ดังสมการ (2.1)

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m} \| x_i - c_j \|^2 , \quad 1 \le m < \infty \tag{2.1}$$

โดยที่

m	คือ จำนวนจริงที่มากกว่า 1
u_{ij}	คือ ค่าความเป็นสมาชิกของ x_i ในคลัสเตอร์ j
x_i	คือ ข้อมูลที่ i
c_j	คือ ศูนย์กลางของคลัสเตอร์ j และ
‖*‖	คือ norm ของค่าความเหมือนระหว่างข้อมูลและศูนย์กลาง
C	คือ จำนวนคลัสเตอร์ทั้งหมด
N	คือ จำนวนจุดทั้งหมด

การแบ่งแบบฟัซซีในแต่ละรอบ ทำเพื่อให้ได้ที่ค่าที่ดีที่สุดของฟังก์ชันวัตถุประสงค์ที่แสดงข้างบน โดยอาศัยการปรับค่าความเป็นสมาชิกของ u_{ij} และศูนย์กลางคลัสเตอร์ c_j ดังนี้

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{\| x_i - c_j \|}{\| x_i - c_k \|}\right)^{\frac{2}{m-1}}} , \quad c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \tag{2.2}$$

การทำงานจะสิ้นสุดเมื่อ $\max_{ij}\{|u_{ij}^{(k+1)} - u_{ij}^{(k)}|\} < \varepsilon$ โดยที่ ε เป็นค่าสำหรับการหยุดทำงานโดยค่าอยู่ระหว่าง 0 กับ 1 และ k เป็นหมายเลขของรอบการทำงาน กระบวนการนี้จะลู่เข้าสู่ค่าต่ำสุดของ J_m

วิธีการนี้จะเริ่มต้น โดยต้องทำการกำหนดว่าข้อมูลที่จัดมีจำนวนทั้งหมดกี่คลัสเตอร์ จากนั้นจะทำการกำหนดค่าเริ่มต้นของคลัสเตอร์ โดยอาจจะกำหนดค่าความเป็นสมาชิกของจุดทั้งหมด

(กำหนด u) หรือทำการกำหนดศูนย์กลางของคลัสเตอร์ (กำหนด c) ก่อน โดยในรูปที่ 2.7 เป็นการกำหนดค่าการทำงานโดยทำการกำหนดค่าความเป็นสมาชิกของจุดทุดจุดก่อน

1. Initialize $U=[u_{ij}]$ matrix, $U^{(0)}$
2. *At k-step: calculate the centers vectors* $C^{(k)}=[c_j]$ *with* $U^{(k)}$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m}$$

3. *Update* $U^{(k)}$, $U^{(k+1)}$

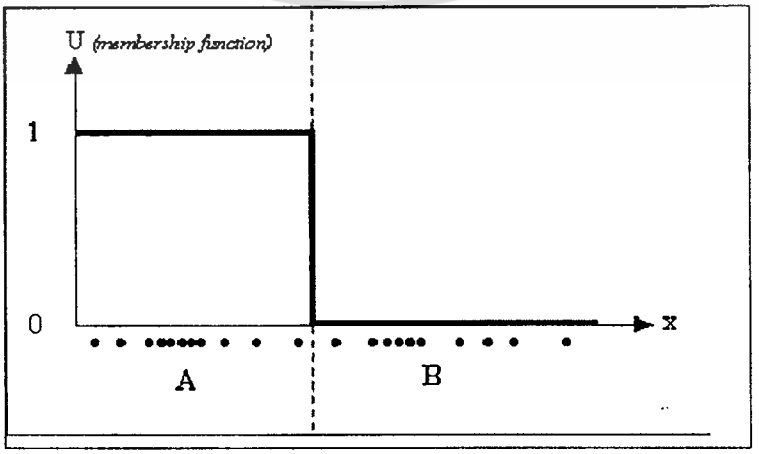
$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\| x_i - c_j \|}{\| x_i - c_k \|} \right)^{\frac{2}{m-1}}}$$

4. *If* $\| U^{(k+1)} - U^{(k)} \| < \varepsilon$ *then STOP; otherwise return to step 2.*

รูปที่ 2.7 แสดงการทำงานของ Fuzzy c-Mean

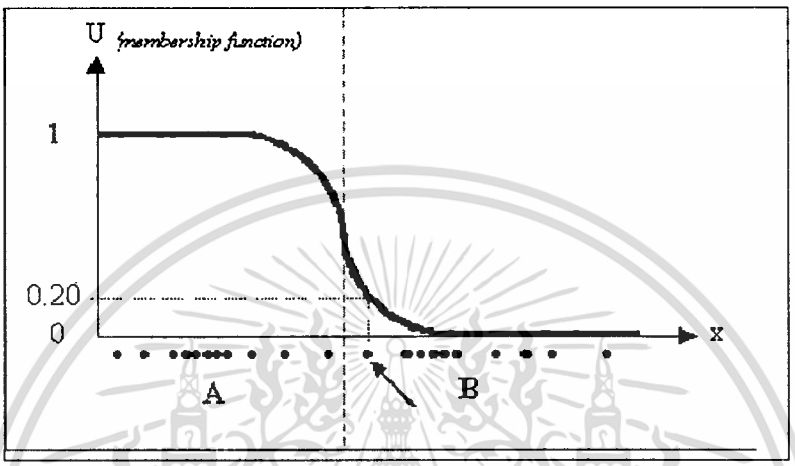
รูปที่ 2.8 แสดงจุดของข้อมูลที่มีมิติเดียว

จากรูปที่ 2.8 เราสามารถที่จะแบ่งข้อมูลออกเป็นสองคลัสเตอร์ได้โดยการประมาณความหนาแน่นของข้อมูล ซึ่งจะเรียกคลัสเตอร์ทั้งสองว่า "คลัสเตอร์ A และคลัสเตอร์ B" ในกรณีแรกที่จะแสดงนั้นเป็นวิธีการจัดกลุ่มแบบ k-Mean ซึ่งข้อมูลแต่ละข้อมูลจะถูกกำหนดไปยังจุดกึ่งกลางใดจุดกึ่งกลางเดียว ดังนั้นฟังก์ชันความเป็นสมาชิกจะมีลักษณะดังรูปที่ 2.9



รูปที่ 2.9 แสดงฟังก์ชันความเป็นสมาชิกของ A ของการจัดกลุ่มแบบ k-mean

ในวิธีการของ FCM แทนที่จะทำการกำหนดให้แต่ละจุดเป็นสมาชิกของคลัสเตอร์ใดคลัสเตอร์หนึ่งเพียงคลัสเตอร์เดียว อัลกอริธึมนี้ทำการกำหนดค่าความเป็นสมาชิกของแต่ละข้อมูลในแต่ละคลัสเตอร์ตามเส้นที่ราบเรียบกว่าของ k-Mean ซึ่งแต่ละข้อมูลอาจจะเป็นสมาชิกของคลัสเตอร์ได้หลายคลัสเตอร์ด้วยค่าสัมประสิทธิ์ความเป็นสมาชิก



รูปที่ 2.10 แสดงฟังก์ชันความเป็นสมาชิกของ A ของการจัดกลุ่มแบบ Fuzzy c-Mean

ในรูปที่ 2.10 ข้อมูลที่เป็นจุดที่ลูกศรชี้จะเป็นสมาชิกของคลัสเตอร์ B มากกว่าคลัสเตอร์ A โดยค่า 0.2 ของ 'u' แสดงค่าความเป็นสมาชิกของคลัสเตอร์ A ของข้อมูล จากนั้นจะใช้เมตริกซ์ U ซึ่งแฟกเตอร์ภายในนำมาจากฟังก์ชันการเป็นสมาชิกของข้อมูลของแต่ละคลัสเตอร์

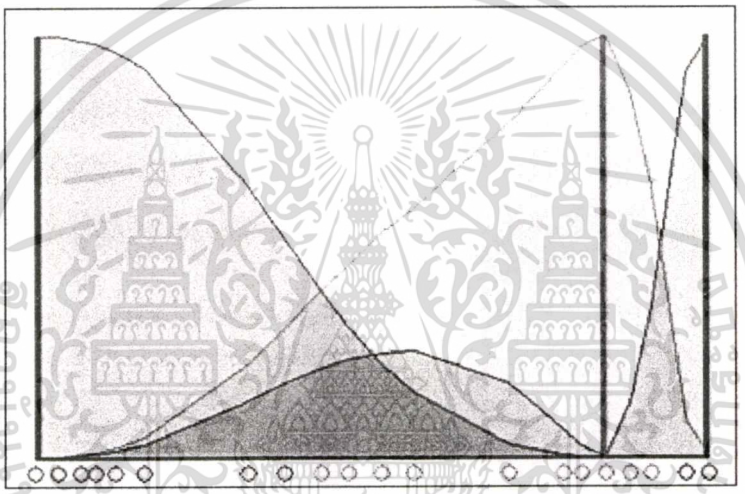
$U_{Nxc} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ \dots & \dots \\ 0 & 1 \end{bmatrix}$	$U_{Nxc} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \\ 0.6 & 0.4 \\ \dots & \dots \\ 0.9 & 0.1 \end{bmatrix}$
(a) k-mean	(b) Fuzzy c-Mean

รูปที่ 2.11 แสดงเมตริกซ์ U ของ k-mean และ Fuzzy c-Mean

จำนวนแถวและคอลัมน์ในเมตริกซ์ U ขึ้นอยู่กับจำนวนข้อมูลและจำนวนของคลัสเตอร์ที่พิจารณา จากตัวอย่างในรูปที่ 2.11 มีคอลัมน์อยู่สองคอลัมน์ (C=2 คลัสเตอร์) และ N แถว โดยที่ C เป็นจำนวนของคลัสเตอร์ทั้งหมด และ N เป็นจำนวนข้อมูลทั้งหมด ซึ่งข้อมูลภายในจะแสดงโดยใช้ u_{ij}

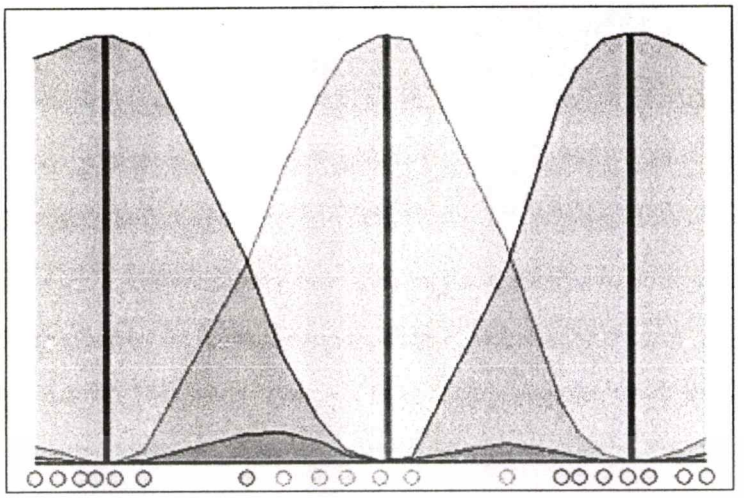
จากรูปที่ 2.11 (a) เป็นเมตริกซ์ของ k-Mean และ (b) เป็น FCM จะสังเกตว่าในกรณี (a) ค่าสัมประสิทธิ์ภายในแต่ละแถวจะมีค่า 1 ได้ค่าเดียวเท่านั้นนอกนั้นเป็น 0 ซึ่งแสดงให้เห็นว่าแต่ละจุดสามารถเป็นสมาชิกได้เพียงคลัสเตอร์เดียว

พิจารณาการจัดกลุ่มข้อมูลโดยใช้ข้อมูลแบบหนึ่งมิติของ FCM โดยข้อมูลจำนวน 20 ข้อมูล และมีคลัสเตอร์ทั้งหมด 3 คลัสเตอร์ถูก ใช้เป็นค่าเริ่มต้นของอัลกอริธึม และใช้ในการคำนวณเมตริกซ์ U ตามสมการที่ (2.2) จะได้ค่า u_{ij}ของแต่ละข้อมูลตามรูปที่ 2.12 ซึ่งแสดงค่าความเป็นสมาชิกของแต่ละข้อมูลของแต่ละคลัสเตอร์ สีของข้อมูลแสดงคลัสเตอร์ที่ใกล้ที่สุดตามฟังก์ชันความเป็นสมาชิก



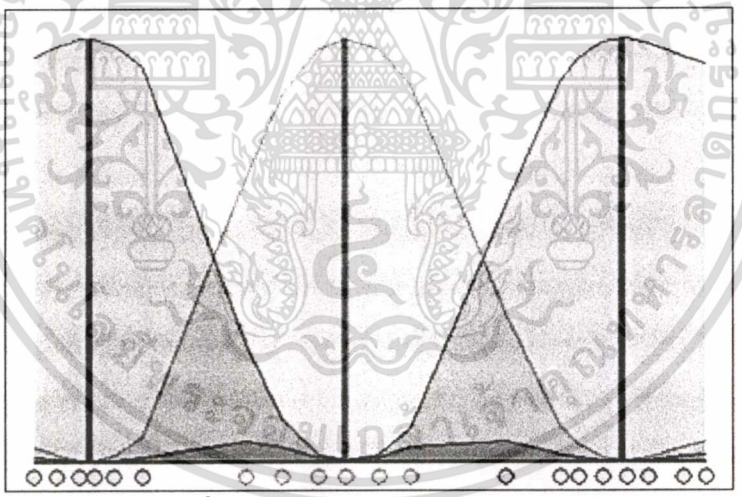
รูปที่ 2.12 แสดงเงื่อนไขค่าความเป็นสมาชิกเริ่มต้นของแต่ละจุด

ในการจำลองการทำงานที่แสดงดังรูปที่ 2.12 ใช้ค่าสัมประสิทธิ์ความคลุมเครือ (fuzziness coefficient) m=2 และอัลกอริธึมสิ้นสุดเมื่อ $\max_{ij}\{|u_{ij}^{(k+1)} - u_{ij}^{(k)}|\} < 0.3$ และแสดงเงื่อนไขเริ่มต้น โดยการกระจายของความคลุมเครือเป็นไปตามแต่ละตำแหน่ง อัลกอริธึมที่ใช้ในตอนแรกยังไม่สามารถบ่งชี้แต่ละคลัสเตอร์ได้ดีนัก ในการทำงานจะทำการทำซ้ำอัลกอริธึมไปกระทั่งเงื่อนไขหยุดเป็นจริง โดยจะได้ผลลัพธ์ดังรูปที่ 2.13



รูปที่ 2.13 แสดง u_i และจุด c_j เมื่ออัลกอริธึมทำงานได้ 8 รอบ โดยใช้ค่า m=2 และ $\varepsilon = 0.3$

เราสามารถที่จะทำให้ได้กลุ่มที่ดีกว่านั้น โดยการใช้ค่า $\varepsilon = 0.01$ แต่เวลาในการคำนวณจะเพิ่มขึ้น ในรูปที่ 2.14 เราจะพบผลลัพธ์ที่ดีกว่าเมื่อเราใช้เงื่อนไขเป็น $\varepsilon = 0.01$ ซึ่งต้องทำการคำนวณทั้งหมด 37 รอบการทำงาน

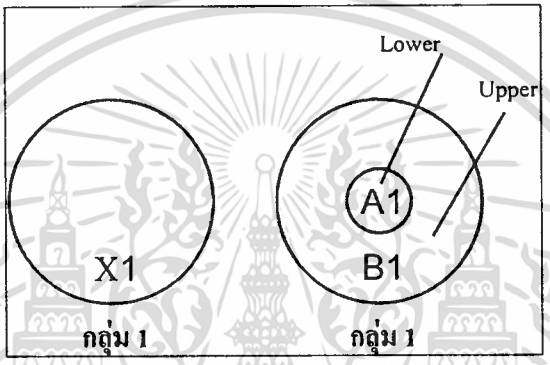


รูปที่ 2.14 แสดง u_i และจุด c_j เมื่ออัลกอริธึมทำงานได้ 37 รอบ โดยใช้ค่า m=2 และ ε =0.01

ถ้ากำหนดค่าเริ่มต้นที่แตกต่างกัน ส่วนใหญ่จะได้ผลลัพธ์ที่คล้ายกัน แต่จำนวนรอบการทำงานอาจไม่เท่ากัน

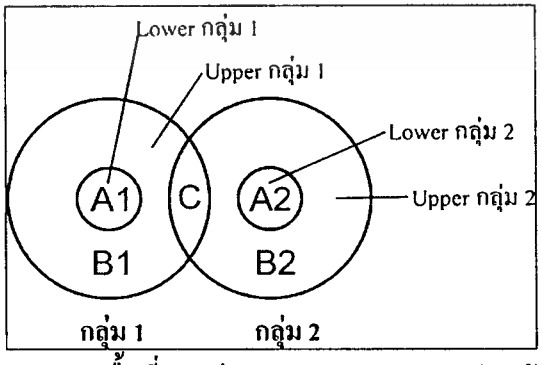
2.2.3 ต้นแบบวิธีการจัดกลุ่มข้อมูลที่มีการซ้อนทับกัน [1]

ต้นแบบสำหรับข้อมูลที่มีการซ้อนทับกัน (Overlap data pattern presentation) ดังรูปที่ 2.15 ด้านขวา จากรูปสามารถแบ่งกลุ่มของข้อมูลออกเป็น 2 ส่วน คือ ส่วน lower และ ส่วน upper โดย ส่วน lower คือ พื้นที่ A1 หมายความว่าข้อมูลที่อยู่ในพื้นที่ A1 เป็นข้อมูลที่มีความแน่นอนว่าเป็นข้อมูลของกลุ่ม 1 ส่วน upper คือ พื้นที่ B1 หมายความว่าข้อมูลที่อยู่ในส่วนของ B1 เป็นส่วนที่มีโอกาสเป็นข้อมูลของกลุ่ม 1 และกลุ่มอื่นๆ สำหรับงานวิจัยของ M.A. Abou-Nasr [10] นำเสนอต้นแบบกลุ่มหนึ่งจะมีข้อมูลแค่ส่วนเดียวดังรูปที่ 2.15 ซ้าย หมายความว่าข้อมูลที่อยู่ในพื้นที่ X1 เป็นข้อมูลของกลุ่ม 1



รูปที่ 2.15 แสดงการเปรียบเทียบ prototype ของงานวิจัยของ M.A. Abou-Nasr (ซ้าย)และ prototype ของงานวิจัยต้นแบบสำหรับข้อมูลที่มีการซ้อนทับกัน (ขวา)

จากต้นแบบที่ได้นำเสนอในรูปที่ 2.15 เมื่อนำไปใช้จัดกลุ่มข้อมูลที่มีการซ้อนทับกันจะได้ผลลัพธ์ดังรูปที่ 2.16 ส่วน A1 และ A2 เป็นข้อมูลของกลุ่มที่ 1 และกลุ่มที่ 2 ตามลำดับ ส่วน B1 เป็นพื้นที่ที่มีโอกาสเป็นข้อมูลของกลุ่มที่ 1 หรือกลุ่มอื่น ๆ ส่วน B2 เป็นพื้นที่ที่มีโอกาสเป็นข้อมูลของกลุ่มที่ 2 หรือกลุ่มอื่น ๆ ส่วน C เป็นพื้นที่ที่มีโอกาสเป็นข้อมูลของกลุ่มที่ 1 หรือกลุ่มที่ 2 ก็ได้



รูปที่ 2.16 แสดงการกำหนดพื้นที่ของส่วน lower และ upper สำหรับข้อมูลซ้อนทับกัน

การทำงานของอัลกอริธึมในการจัดกลุ่มใช้หลักการของ NNC [10] จะประกอบไปด้วยการกำหนดรัศมีหรือค่า threshold ของกลุ่มทั้งสองค่า และการปรับค่าจุดศูนย์กลางหรือน้ำหนักของกลุ่ม สำหรับการปรับค่า threshold (lower และ upper) จะใช้ฟังก์ชันการวัดระยะห่างแบบยูคลิด (Euclidean distance) โดยสามารถคำนวณได้ตามสมการ (2.3) ซึ่งเป็นการวัดระยะห่างของข้อมูลใหม่กับค่าจุดศูนย์กลางหรือน้ำหนักของกลุ่ม

$$b_{jT} = \sqrt{\sum_{s=1}^{n} (W_{b_j a_s} - P_s)^2} \qquad (2.3)$$

โดยที่

$W_{b_j a_s}$ คือ ค่าน้ำหนักของกลุ่มที่กำลังพิจารณา

P_s คือ ค่าคุณลักษณะ(feature)ของข้อมูล

b_j คือ ค่าระยะห่างของกลุ่มที่กำลังพิจารณากับข้อมูลใหม่

สำหรับการปรับค่าน้ำหนักของกลุ่ม (center ของกลุ่ม) สามารถคำนวณได้ตามสมการ (2.4)

$$W_{b_j a_{new}} = \frac{(W_{b_j a_{old}}) * M + P_i}{M+1}, \forall i = 1,2,\ldots,n \qquad (2.4)$$

โดยที่

$W_{b_j a_{new}}$ คือ ค่าน้ำหนักเฉลี่ยใหม่ของกลุ่ม

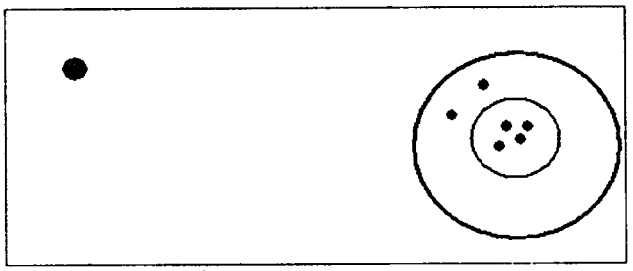
$W_{b_j a_{old}}$ คือ ค่าน้ำหนักเฉลี่ยเดิมของกลุ่ม

p_i คือ ค่าคุณลักษณะ(feature) ของข้อมูลคือจุดใหม่ที่เป็นสมาชิกของกล่มนี้

M คือ ค่าจำนวนข้อมูลที่เข้ามาทดสอบในอัลกอริธึม

เริ่มต้นอัลกอริธึมจะทำการพิจารณาข้อมูลของการเรียนรู้ทีละค่า เมื่อข้อมูลตัวแรกที่เข้ามาอัลกอริธึมจะกำหนดข้อมูลนี้เป็นน้ำหนักของกลุ่ม และกำหนด ค่า threshold ของกลุ่มทั้งสองค่าคือ lower threshold และ upper threshold โดยครั้งแรกจะกำหนดให้มีค่าเท่ากัน จากนั้นเมื่อข้อมูลตัวที่สองเข้ามาอัลกอริธึมจะทำการหาระยะห่างของข้อมูลกับทุกกลุ่ม ข้อมูลใหม่จะถูกนำมารวมกับกลุ่มที่มีค่าเฉลี่ย (W) ใกล้จุดข้อมูลที่สุด ซึ่งจะทำให้ค่าของ W เปลี่ยนไปตามสมการที่ (2.4) จากนั้นปรับค่า threshold ของกลุ่มโดยสามารถพิจารณาภาพรวมได้ดังต่อไปนี้

เมื่อข้อมูลที่กำลังทดสอบอยู่เป็นกลุ่มเดียวกันดังรูปที่ 2.17 ใช้สัญลักษณ์จุดทึบสีดำแทนข้อมูลใหม่และกลุ่มข้อมูลที่กำลังพิจารณาคือจุดสีดำทึบเหมือนกัน



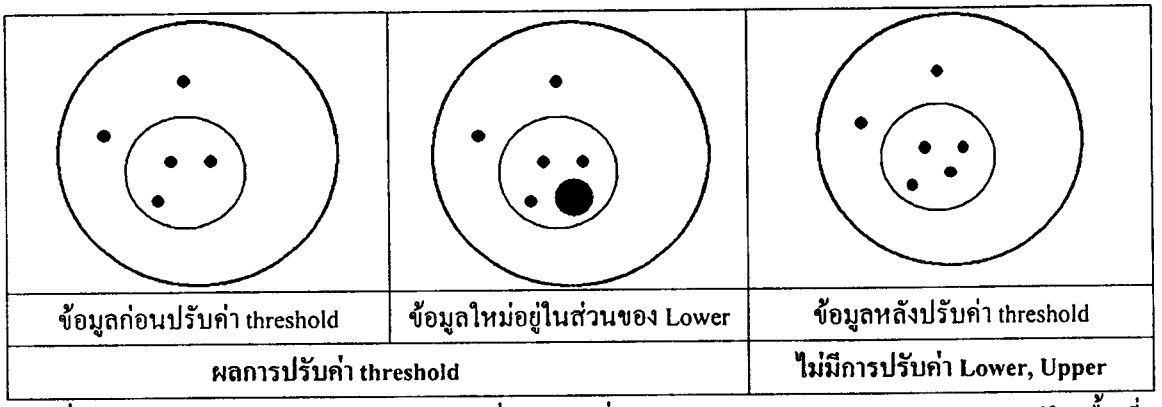
รูปที่ 2.17 แสดงข้อมูลที่กำลังทดสอบเป็นกลุ่มเดียวกับกลุ่มทดสอบ

เมื่อข้อมูลที่กำลังทดสอบอยู่ต่างกลุ่มกันดังรูปที่ 2.18 ใช้สัญลักษณ์จุดสีดำขาวแทนข้อมูลใหม่และกลุ่มข้อมูลที่กำลังพิจารณาคือ จุดสีดำทึบซึ่งต่างกลุ่มกัน

รูปที่ 2.18 แสดงข้อมูลที่กำลังทดสอบต่างกลุ่มกับกลุ่มทดสอบ

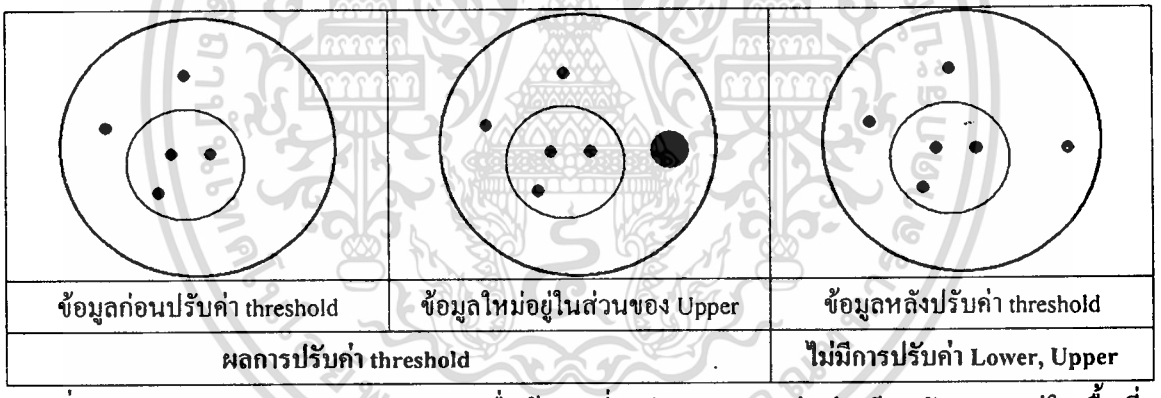
การปรับค่า threshold เมื่อข้อมูลที่กำลังทดสอบอยู่กลุ่มเดียวกันจะเป็นไปตามรูปที่ 2.19-2.22 ซึ่งสามารถอธิบายรายละเอียดดังนี้

การปรับค่าเมื่อข้อมูลที่กำลังทดสอบอยู่ใน lower threshold ดังรูปที่ 2.19 ด้านซ้ายคือ กลุ่มข้อมูลเดิม รูปกลางคือ ข้อมูลใหม่ที่กำลังพิจารณา (วงกลมทึบใหญ่) เมื่อคำนวณระยะห่างจากข้อมูลถึงกลุ่มที่กำลังทดสอบแล้วอยู่ในพื้นที่ lower ของกลุ่ม อัลกอริธึมจะไม่มีการปรับค่า threshold ใดๆ ของกลุ่มดังแสดงในรูปด้านขวา เพราะข้อมูลอยู่ในพื้นที่ที่ถูกต้องแล้ว แต่อัลกอริธึมจะทำการคำนวณเฉลี่ยน้ำหนักของกลุ่มนี้ใหม่



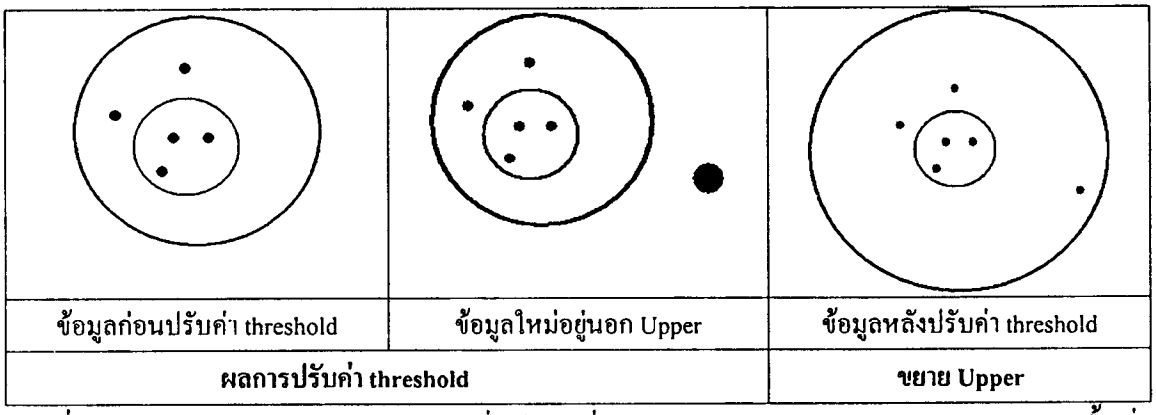
รูปที่ 2.19 แสดงการปรับค่า threshold เมื่อข้อมูลที่กำลังทดสอบอยู่กลุ่มเดียวกัน และอยู่ในพื้นที่ Lower [1]

การปรับค่าเมื่อข้อมูลที่กำลังทดสอบอยู่ระหว่าง lower threshold และ upper threshold ดังรูปที่ 2.20 เนื่องจากข้อมูลใหม่ที่เมื่อคำนวณระยะห่างจากข้อมูลถึงกลุ่มที่กำลังทดสอบแล้วอยู่ในพื้นที่ระหว่างค่า lower และค่า upper ของกลุ่ม อัลกอริธึมจะไม่มีการปรับค่า threshold ใดๆ ของกลุ่ม ดังรูปด้านซ้าย เพราะข้อมูลอยู่ในพื้นที่ที่ถูกต้องแล้ว อัลกอริธึมจะทำการคำนวณเฉลี่ยน้ำหนักของกลุ่มเท่านั้น

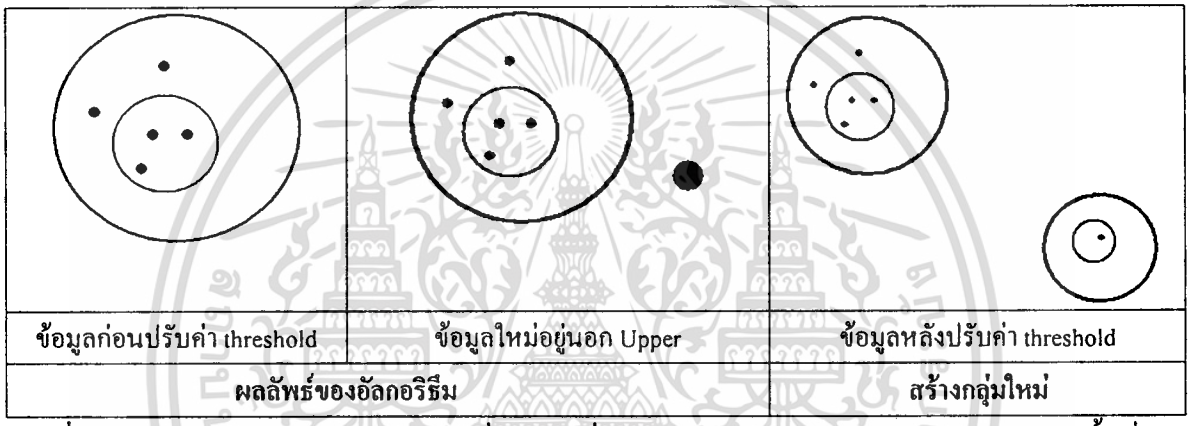


รูปที่ 2.20 แสดงการปรับค่า threshold เมื่อข้อมูลที่กำลังทดสอบอยู่กลุ่มเดียวกัน และอยู่ในพื้นที่ Upper [1]

การปรับค่าเมื่อข้อมูลที่กำลังทดสอบอยู่นอก upper threshold ดังรูปที่ 2.21 เมื่อคำนวณระยะห่างจากข้อมูลถึงกลุ่มที่กำลังทดสอบแล้วอยู่นอกพื้นที่ของกลุ่ม อัลกอริธึมจะทำการปรับค่า threshold โดยแบ่งเป็น 2 กรณีคือ กรณีที่ข้อมูลใหม่อยู่นอก upper threshold มีระยะทางน้อยกว่า 1.2 เท่าของ upper threshold ของกลุ่ม อัลกอริธึมจะขยาย upper threshold ของกลุ่มนั้นดังรูปที่ 2.21 ด้านขวา ส่วนกรณีที่ข้อมูลใหม่อยู่นอก upper threshold มากกว่า 1.2 เท่าอัลกอริธึมจะสร้างกลุ่มใหม่ดังรูปที่ 2.22



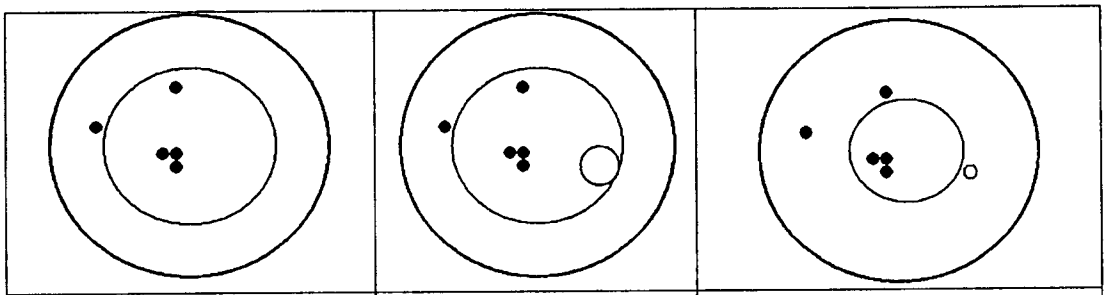
รูปที่ 2.21 แสดงการปรับค่า threshold เมื่อข้อมูลที่กำลังทดสอบอยู่กลุ่มเดียวกัน และอยู่นอกพื้นที่ Upper (ระยะทางน้อยกว่า 1.2 เท่าของ Upper) [1]



รูปที่ 2.22 แสดงการปรับค่า threshold เมื่อข้อมูลที่กำลังทดสอบอยู่กลุ่มเดียวกัน และอยู่นอกพื้นที่ Upper (ระยะมากกว่า 1.2 เท่าของ Upper) [1]

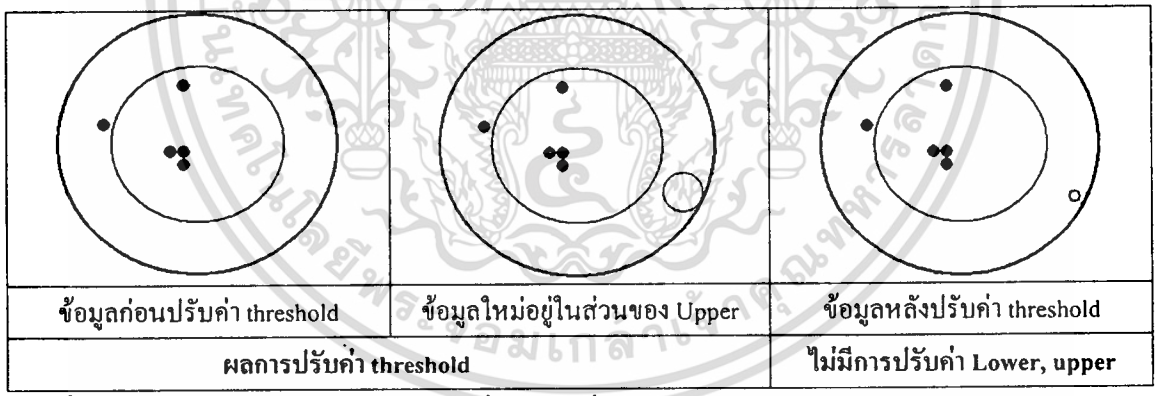
การปรับค่า threshold เมื่อข้อมูลที่กำลังทดสอบอยู่ต่างกลุ่มกัน เป็นไปตามรูปที่ 2.23-2.27 และสามารถอธิบายรายละเอียดเป็นหัวข้อย่อยดังนี้

การปรับค่าเมื่อข้อมูลใหม่ที่กำลังทดสอบอยู่ใน lower threshold ดังรูปที่ 2.23 เมื่อคำนวณระยะห่างจากข้อมูลใหม่ (วงกลมใหญ่สีขาว) ถึงกลุ่มที่กำลังทดสอบแล้วข้อมูลอยู่ในพื้นที่ lower ของกลุ่ม อัลกอริธึมจะปรับลดค่า lower threshold เพื่อให้เซตของ lower threshold ถูกต้อง (ข้อมูลใน lower จะประกอบด้วยกลุ่มข้อมูลแบบเดียวกัน) ดังรูปที่ 2.23 ด้านขวา



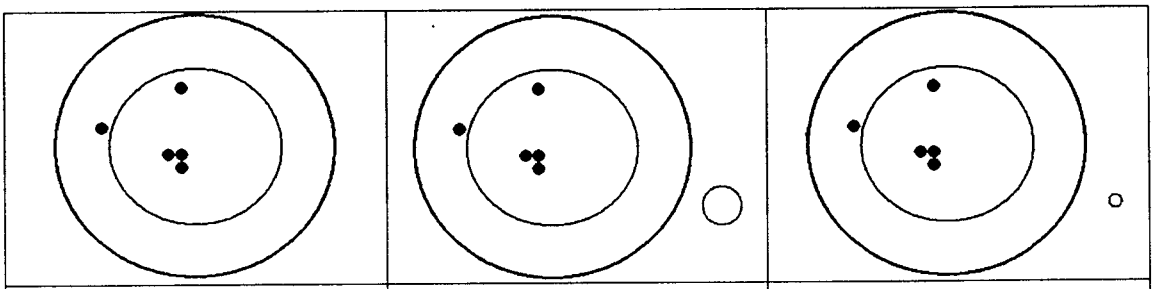
รูปที่ 2.23 แสดงการปรับค่า threshold เมื่อข้อมูลที่กำลังทดสอบอยู่ต่างกลุ่มกัน กรณีข้อมูลใหม่อยู่ในพื้นที่ Lower [1]

การปรับค่าเมื่อข้อมูลที่กำลังทดสอบอยู่ระหว่าง lower threshold และ upper threshold ดังรูปที่ 2.24 ด้านซ้ายคือ กลุ่มข้อมูลเดิม รูปกลางคือ ข้อมูลใหม่ เมื่อคำนวณระยะห่างจากข้อมูลถึงกลุ่มที่กำลังทดสอบแล้วข้อมูลอยู่ในพื้นที่ระหว่าง lower และ upper ของกลุ่ม อัลกอริธึมจะทดสอบปรับลดค่า upper threshold แล้วตรวจสอบว่าข้อมูลเดิมของกลุ่มหายไปน้อยกว่า 20% จะลด upper threshold ดังรูปที่ 2.24 ขวา แต่ถ้าข้อมูลเดิมหายไปมากกว่า 20 % จะไม่ปรับลดค่า upper threshold



รูปที่ 2.24 แสดงการปรับค่า threshold เมื่อข้อมูลที่กำลังทดสอบอยู่ต่างกลุ่มกัน กรณีข้อมูลใหม่อยู่ในพื้นที่ Upper [1]

การปรับค่าเมื่อข้อมูลที่กำลังทดสอบอยู่นอก upper threshold ดังรูปที่ 2.25 ด้านซ้ายคือ กลุ่มข้อมูลเดิม รูปกลางคือ ข้อมูลใหม่ เมื่อคำนวณระยะห่างจากข้อมูลถึงกลุ่มที่กำลังทดสอบแล้วข้อมูลอยู่นอกพื้นที่ upper threshold ของกลุ่ม อัลกอริธึมจะไม่ปรับลดค่า threshold ใดๆ เพื่อให้ข้อมูลของกลุ่มถูกต้อง ดังรูปที่ 2.25 ด้านขวา



ข้อมูลก่อนปรับค่า threshold	ข้อมูลใหม่อยู่นอกส่วนของ Upper	ข้อมูลหลังปรับค่า threshold
ผลการปรับค่า threshold		**ไม่มีการปรับค่า Lower, upper**

รูปที่ 2.25 แสดงการปรับค่า threshold เมื่อข้อมูลที่กำลังทดสอบอยู่ต่างกลุ่มกันกรณีข้อมูลใหม่อยู่นอกพื้นที่ Upper [1]

ในกรณีจุดเข้ามาใหม่ไม่ตกอยู่ใน upper หรือ lower ของกลุ่มที่มีอยู่แล้วเลย จะทำการสร้างโดยมีศูนย์กลางอยู่ที่เข้ามาใหม่นั้น

บทที่ 3
การจัดกลุ่มข้อมูลที่มีการซ้อนทับกันโดยใช้เทคนิคความหนาแน่น

ในบทนี้จะกล่าวถึงวิธีการจัดกลุ่มข้อมูลที่มีการซ้อนทับกันที่ใช้ในวิทยานิพนธ์ วิธีการนี้เรียกว่า "Overlapping DBSCAN" หรือ "ODBSCAN" ซึ่งเป็นวิธีการจัดกลุ่มข้อมูลที่มีการซ้อนทับกันโดยใช้เทคนิคความหนาแน่น ที่ดัดแปลงมาจากวิธีการ DBSCAN เพื่อให้สามารถจัดกลุ่มข้อมูลที่มีการซ้อนทับกันได้ โดยอาศัยอัตราส่วนคลาส (Class Ratio) และความต่างของอัตราส่วนคลาส (Difference of Class Ratio)

ส่วนประกอบของเนื้อหาภายในบท เริ่มจากนิยามของอัตราส่วนคลาส นิยามความต่างของอัตราส่วนคลาสและการหาความต่างอัตราส่วนคลาสในรูปแบบต่างๆ นิยามของความหนาแน่นของข้อมูลที่มีการประยุกต์ใช้อัตราส่วนคลาสร่วมด้วย จากนั้นจะนำเสนอหลักการของวิธีการ ODBSCAN วิธีการหาตัวแทนในการแพร่ และลักษณะการหาจุดข้างเคียงแบบต่างๆ วิธีการระบุคลาส และการวัดประสิทธิภาพการจัดกลุ่ม

3.1 อัตราส่วนคลาส (Class Ratio)

ในส่วนนี้จะกล่าวถึงนิยามและความสำคัญของอัตราส่วนคลาส ตัวอย่างการหาอัตราส่วนคลาส

นิยาม อัตราส่วนคลาส คือ เวกเตอร์ของค่าอัตราส่วนของจำนวนจุดของแต่ละคลาสที่อยู่ภายในกลุ่มเดียวกัน แสดงโดยค่า

$$CR(p) = [r_a, r_b, ..r_m] \tag{3.1}$$

$$r_a = \frac{n_a}{\sum_{i=1}^{m} n_i} \tag{3.2}$$

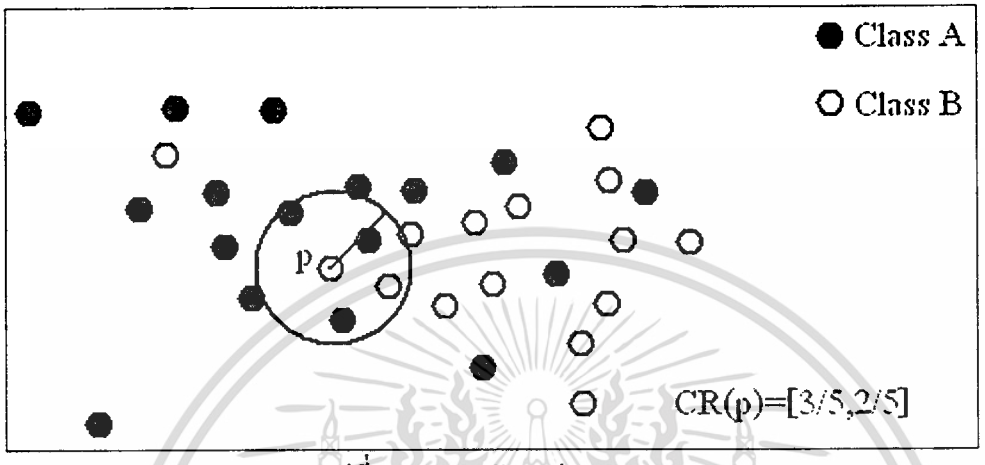
r_a คือ อัตราส่วนคลาสของคลาส a

n_a คือ จำนวนจุดที่เป็นคลาส a

m คือ จำนวนคลาสทั้งหมด

ค่าของอัตราส่วนคลาสเป็นค่าที่บ่งบอกถึงลักษณะของข้อมูลภายใน ว่าประกอบด้วยคลาสใดบ้าง ในอัตราส่วนเท่าใด ซึ่งค่านี้จะเป็นค่าที่ใช้เปรียบเทียบความเป็นกลุ่มเดียวกันของกลุ่มข้อมูลข้างเคียง โดยที่ถ้ากลุ่มข้อมูลใด 2 กลุ่มข้อมูลที่ติดกันเป็นกลุ่มข้อมูลที่คล้ายกัน จะมีค่าของอัตราส่วนคลาสนี้ใกล้เคียงกัน

อัตราส่วนคลาสของแต่ละคลาส สามารถหาได้จากอัตราส่วนระหว่างจำนวนจุดของคลาสนั้นๆ กับผลรวมจำนวนจุดทั้งหมด จากรูปที่ 3.1 สามารถหาอัตราส่วนคลาสของคลาส A = 3/5 และอัตราส่วนคลาสของคลาส B = 2/5 ดังนั้นอัตราส่วนคลาสของจุด p จึงมีค่าเป็น CR(p) = [3/5, 2/5]

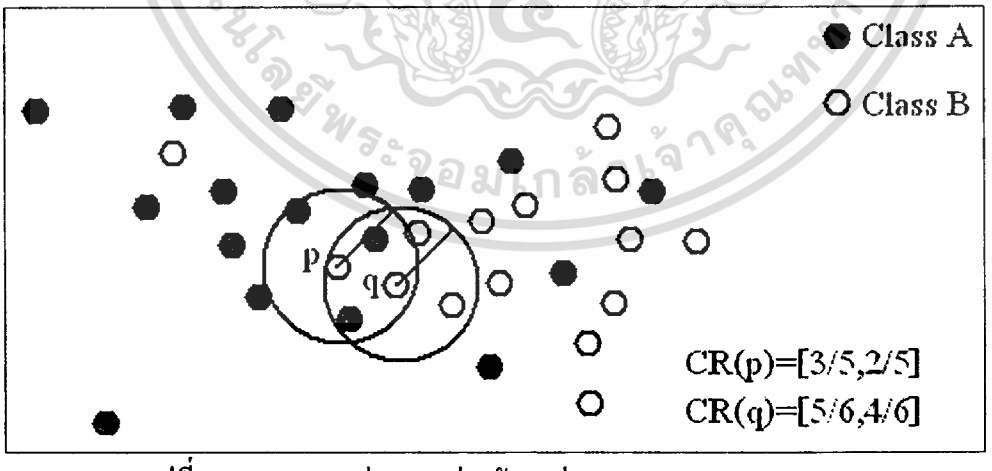


รูปที่ 3.1 แสดงอัตราส่วนคลาส

3.2 ความต่างของอัตราส่วนคลาส (Difference of Class Ratio:DCR)

นิยาม ความต่างอัตราส่วนคลาส คือ ค่าความแตกต่างของค่าอัตราส่วนคลาสระหว่างจุดสองจุด โดยค่าความแตกต่างของอัตราส่วนคลาสของจุด p และ q สามารถเขียนแทนโดย

$$DCR(p,q) = |CR(p) - CR(q)| \qquad (3.3)$$



รูปที่ 3.2 แสดงการค่าความต่างอัตราส่วนคลาสของจุด p และ q

ความต่างของอัตราส่วนคลาสเป็นค่าที่ใช้บอกความแตกต่างระหว่างอัตราส่วนคลาสของกลุ่ม 2 กลุ่ม โดยถ้ากลุ่ม 2 กลุ่มใดมีค่าความต่างของอัตราส่วนคลาสน้อย นั่นหมายถึงกลุ่มสองกลุ่มนั้นมีความคล้ายคลึงกันมาก ในทางตรงกันข้าม ถ้ามีความต่างของอัตราส่วนคลาสสูงก็หมายถึงมีความ

คล้ายคลึงกันน้อย และการหาค่าของความต่างของอัตราส่วนคลาสนั้นสามารถหาได้หลายแบบ โดยแต่ละแบบมีวิธีการแตกต่างกัน ดังนี้

3.2.1 หาจากผลรวมความต่างอัตราส่วนของแต่ละคลาส (DCR Type 1)

ค่าความต่างของอัตราส่วนคลาสลักษณะนี้ หาได้จากผลรวมความแตกต่างของอัตราส่วนคลาสแต่ละคลาสของจุดสองจุด

$$|CR(p)-CR(q)| = \sum_{i=1}^{m} |r_i^p - r_i^q| \qquad (3.4)$$

โดยที่

r_i^p, r_i^q คือ ค่าอัตราส่วนคลาสของคลาส i ของจุด p และ q ตามลำดับ

m คือ จำนวนคลาสทั้งหมด

ตารางที่ 3.1 แสดงอัตราส่วนคลาสของจุด p และ q

คลาส	จุด p	จุด q
a	0.3	0.4
b	0.2	0.5
c	0.5	0.3
d	0.0	0.1

จากตารางที่ 3.1 สามารถหาความต่างอัตราส่วนคลาสโดยวิธีผลรวมความต่างอัตราส่วนคลาสของจุด p และ q จากสมการ 3.4 ได้ดังนี้

$$\begin{aligned} DCR(p,q) &= |p_a-q_a| + |p_b-q_b| + |p_c-q_c| + |p_d-q_d| \\ &= |0.3-0.4| + |0.2-0.5| + |0.5-0.3| + |0.0-0.1| \\ &= 0.1+0.3+0.2+0.1 \\ &= 0.7 \end{aligned}$$

3.2.2 หาจากผลรวมความต่างของอัตราส่วนแต่ละคลาสที่มีการกำจัดข้อมูลรบกวน (DCR Type 2)

ค่าความต่างอัตราส่วนคลาสลักษณะนี้ หาได้จากผลรวมของค่าความต่างอัตราส่วนคลาสแต่ละคลาสของจุดสองจุด ที่มีค่าเกินค่า α ค่าหนึ่งที่กำหนดสำหรับค่าของข้อมูลรบกวน

$$|CR(p)-CR(q)| = \sum_{i=1}^{m} d(|r_i^p - r_i^q|) \tag{3.5}$$

$$d(|r_i^p - r_i^q|) = \begin{cases} 0 & where\ |r_i^p - r_i^q| \le \alpha \\ |r_i^p - r_i^q| - \alpha & otherwise \end{cases} \tag{3.6}$$

โดยที่

r_i^p, r_i^q คือ ค่าอัตราส่วนคลาสของคลาส i ของจุด p และ q ตามลำดับ

m คือ จำนวนคลาสทั้งหมด

α คือ ค่ากำหนดของข้อมูลรบกวน

จากตารางที่ 3.1 สามารถหาความต่างอัตราส่วนคลาสโดยวิธีผลรวมความต่างอัตราส่วนคลาสของจุด p และ q จากสมการ 3.6 และ 3.7 โดยกำหนดค่า α =0.1 ได้ดังนี้

$$\begin{aligned} DCR(p,q) &= d(|p_a-q_a|) + d(|p_b-q_b|) + d(|p_c-q_c|) + d(|p_d-q_d|) \\ &= d(|0.3-0.4|) + d(|0.2-0.5|) + d(|0.5-0.3|) + d(|0.0-0.1|) \\ &= 0.0+0.2+0.1+0.0 \\ &= 0.3 \end{aligned}$$

วิธีการนี้จะเห็นว่าแตกต่างจากวิธีการแรกคือมีการกำจัดค่าของข้อมูลรบกวนออกไปจากข้อมูลซึ่งจะมีผลดีถ้าข้อมูลที่ใช้ในการจัดกลุ่มมีข้อมูลรบกวนอยู่

3.2.3 หาจากความต่างสูงสุดของความต่างอัตราส่วนของแต่ละคลาส (DCR Type 3)

ค่าความต่างของอัตราส่วนคลาสลักษณะนี้ หาได้จากค่าความต่างอัตราส่วนคลาสแต่ละคลาสที่มีค่าความต่างสูงสุด

$$|CR(p)-CR(q)| = \max_i(|r_i^p - r_i^q|), i = [1,m] \tag{3.7}$$

โดยที่

r_i^p, r_i^q คือ ค่าอัตราส่วนคลาสของคลาส i ของจุด p และ q ตามลำดับ

m คือ จำนวนคลาสทั้งหมด

จากตารางที่ 3.1 สามารถหาความต่างอัตราส่วนคลาสโดยวิธีผลรวมความต่างอัตราส่วนคลาสของจุด p และ q จากสมการ 3.5 ได้ดังนี้

$$DCR(p,q) = \max(|p_a-q_a| , |p_b-q_b| , |p_c-q_c| , |p_d-q_d|)$$
$$= \max(|0.3-0.4| , |0.2-0.5| , |0.5-0.3| , |0.0-0.1|)$$
$$= 0.3$$

3.3 นิยามความหนาแน่นของข้อมูลที่มีการประยุกต์ใช้อัตราส่วนคลาสร่วมด้วย

ในส่วนนี้จะทำการกล่าวถึง นิยามของความหนาแน่นของข้อมูลที่มีการประยุกต์อัตราส่วนคลาสร่วมด้วย ซึ่งจะกล่าวถึงนิยามที่แตกต่างจากความหนาแน่นของข้อมูลแบบเดิม ดังนี้

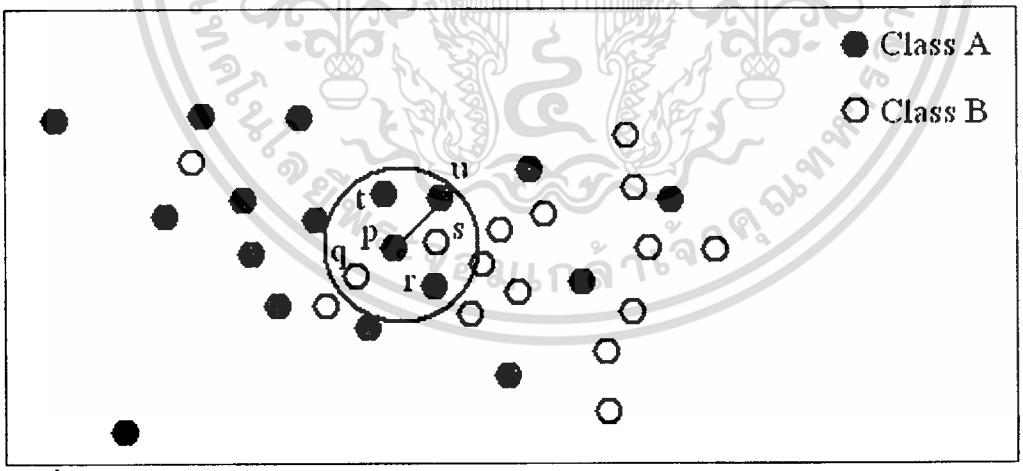
นิยาม 3.1: *ε – neighborhood with class ratio* คือ *ε – neighborhood* ของจุด p ซึ่งค่า DCR(p,q) ≤ MaxDiff จะเขียนแทนด้วย

$$NCR_{Eps}(p) = \{q \in D \mid dist(p,q) \le Eps \text{ and } DCR(p,q) \le MaxDiff\} \quad (3.8)$$

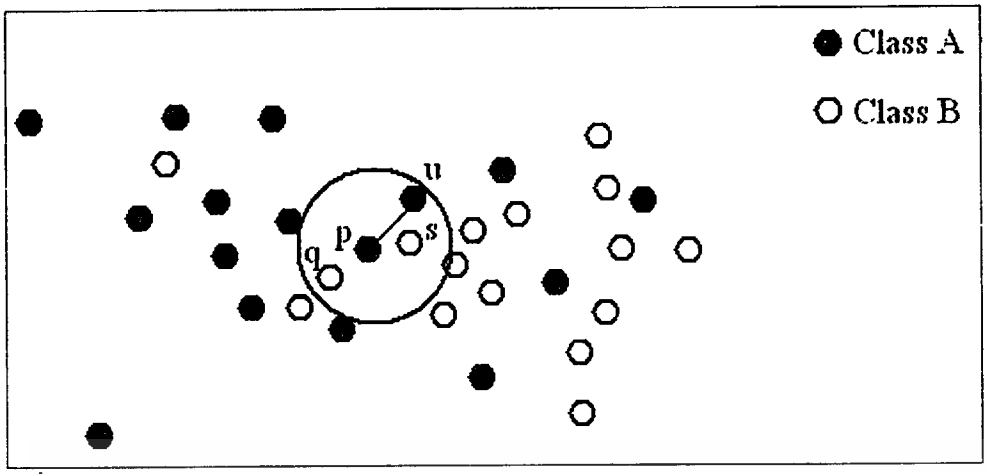
โดยที่

$NCR_{Eps}(q)$ คือ *ε – neighborhood with class ratio* ของจุด q

MaxDiff คือ ค่าความต่างสูงสุดที่ถือว่าจุดสองจุดมีส่วนประกอบคลาสคล้ายกัน



รูปที่ 3.3 แสดง *ε – neighborhood with class ratio* ของจุด p โดยมีค่า MaxDiff = 0.4



รูปที่ 3.4 แสดง $\varepsilon - neighborhood\ with\ class\ ratio$ ของจุด p โดยมีค่า MaxDiff = 0.2

นิยาม 3.2: จุดแกน (Core Point): จุด p จะเป็นจุดแกน ถ้า

1) $|NCR_{Eps}(p)| \geq MinPts$

นิยาม 3.2: Directly density-reachable: จุด ๆ จะเป็น directly density-reachable จากจุด ย ถ้า

1) $q \in NCR_{Eps}(p)$ and

2) $|NCR_{Eps}(p)| \geq MinPts$

ส่วนนิยามอื่นที่ไม่ได้กล่าวถึงในส่วนนี้ จะใช้นิยามความหนาแน่นของข้อมูลแบบเดิม ตามที่เคยกล่าวไว้แล้วในหัวข้อ 2.2.1.1

3.4 อัลกอริธึม ODBSCAN

3.4.1 การทำงานของอัลกอริธึม

ในส่วนนี้จะทำการอธิบายอัลกอริธึมของ ODBSCAN โดยการทำงานของอัลกอริธึมนี้จะเริ่มจากทำการหาจุด p ที่ยังไม่ได้ทำการจัดกลุ่ม แล้วทำการตรวจสอบว่าจุด p เป็นจุดแกนหรือไม่ ถ้าไม่จะทำการกำหนดให้จุด p เป็นข้อมูลรบกวน แต่ถ้าเป็นจุดแกนอัลกอริธึมจะทำการสร้างคลัสเตอร์ขึ้นใหม่แล้วทำการกำหนดจุด p และ $\varepsilon - neighborhood\ with\ class\ ratio$ (หรือ $\varepsilon - neighborhood$ ซึ่งขึ้นอยู่กับวิธีการหาจุดข้างเคียง โดยจะอธิบายรายละเอียดในหัวข้อที่ 3.4.3) ของจุด p ให้เป็นคลัสเตอร์ที่สร้างใหม่ จากนั้นทำการสร้างคิว (Queue) ขึ้นใหม่โดยมีสมาชิกเป็น $\varepsilon - neighborhood\ with\ class\ ratio$ ของ p จากนั้นทำการดึงจุด q จากคิว ทำการตรวจสอบว่าจุด q เป็นจุดแกนหรือไม่ โดยใช้ $\varepsilon - neighborhood$ ของ q เปรียบเทียบกับจุดที่เป็นตัวแทนของคลัสเตอร์ (ซึ่งจะอธิบายวิธีการหาตัวแทนของคลัสเตอร์ในหัวข้อ 3.4.2) ถ้าใช่ให้ทำการกำหนด $\varepsilon - neighborhood\ with\ class\ ratio$ (หรือ $\varepsilon - neighborhood$ ซึ่งขึ้นอยู่กับวิธีการหาจุด

ข้างเคียง) ของจุด q ให้เป็นคลัสเตอร์เดียวกับคลัสเตอร์ของจุด p ที่แล้ว และทำการเพิ่ม $\varepsilon - neighborhood\ with\ class\ ratio$ ของ q เทียบกับจุดตัวแทนคลัสเตอร์ เข้าไปในคิว ทำขบวนการหาจุด q มาทำเป็นจุดแกนจนกว่าคิวจะว่าง จากนั้นจะเริ่มทำการหาจุด p จนกว่าไม่มีจุด p ที่ยังไม่ได้จัดกลุ่ม โดยมี Psudo code ดังรูปที่ 3.5

```
1.  ODBSCAN
2.      ClusterID=0
3.      FOR p=1 TO N DO
4.          IF p is not label THEN
5.              IF p is CorePoint THEN
6.                  ClusterID=ClusterID+1
7.                  Assign Cluster of p = ClusterID
8.                      Assign Cluster of ε − neighborhood of p
                            (or ε − neighborhood with class ratio according to RegionQuery
                            function) = ClusterID
9.                      Add ε − neighborhood with class ratio of p to Queue
10.                     Finding Point Representation of Cluster assign to PRC
11.                     DO
12.                       q = Dequeue Queue
13.                       IF q is Core Point compare with PRC Then
14.                           Assign Cluster of ε − neighborhood of q
                                  (or ε − neighborhood with class ratio according to RegionQuery
                                  function compare with PRC) = ClusterID
15.                           Add ε − neighborhood with class ratio of q compare with PRC to
                              Queue
16.                       END IF
17.                     LOOP UNTIL Queue empty
18.             ELSE
19.                 Assign Cluster of p = NOISE
20.             END IF
21.         END IF
22.     END FOR
23. END
```

รูปที่ 3.5 แสดงอัลกอริธึมของ ODBSCAN

3.4.2 วิธีหาจุดตัวแทนของคลัสเตอร์ในการแพร่ (Point Representation of Cluster:PRC)

การหาจุดตัวแทนของคลัสเตอร์ในการแพร่ เป็นส่วนสำคัญส่วนหนึ่งในการที่จะทำให้ได้คลัสเตอร์ที่มีความคล้ายคลึงของข้อมูลภายในคลัสเตอร์ เนื่องจากตัวแทนนี้จะถูกใช้ในการเปรียบเทียบความต่างอัตราส่วนคลาสกับจุดที่จะแพร่ไป โดยมีการเลือกจุดตัวแทนได้ 2 รูปแบบด้วยกันคือ

3.4.2.1 จุดตัวแทนของคลัสเตอร์ในการแพร่โดยใช้จุดแรกที่ทำให้เกิดคลัสเตอร์ (PRC Type 1)

เป็นการใช้จุดแรกที่ทำให้เกิดเป็นคลัสเตอร์เป็นตัวแทน เพื่อใช้ในการเปรียบเทียบค่าความต่างอัตราส่วนคลาสกับจุดที่จะถือเป็นสมาชิกของคลัสเตอร์ ดังแสดงในรูปที่ 3.6

```
1.  CalRepresentativePoint(currentP, representativePointList): Point
2.      RETURN representativePointList.first()
3.  END // CalRepresentativePoint
```

รูปที่ 3.6 ฟังก์ชันหาจุดตัวแทนของคลัสเตอร์ในการแพร่โดยใช้จุดแรกที่ทำให้เกิดคลัสเตอร์

3.4.2.2 จุดตัวแทนของคลัสเตอร์ในการแพร่โดยใช้ค่าเฉลี่ย (PRC Type 2)

เป็นการหาจุดตัวแทนของคลัสเตอร์ที่ใช้ในการแพร่ โดยจะทำการหาค่าเฉลี่ยของจุดทั้งหมดที่มีอยู่ในคลัสเตอร์ปัจจุบัน ดังแสดงในรูปที่ 3.7

```
1.  CalRepresentativePoint(currentP, representativePointList) : Point
2.      representativePointList.add(currentP)
3.      FOR i=0 TO representativePointList.size() DO
4.          returnPoint+=representativePointList.elementAt(i)
5.      END FOR
6.      returnPoint/=representativePointList.size()
7.      RETURN returnPoint
8.  END // CalRepresentativePoint
```

รูปที่ 3.7 ฟังก์ชันหาจุดตัวแทนของคลัสเตอร์ในการแพร่โดยใช้ค่าเฉลี่ย

3.4.3 วิธีการหาจุดข้างเคียง

วิธีการหาจุดข้างเคียงเป็นการบ่งบอกถึงจุดที่จะถูกกำหนดให้เป็นคลัสเตอร์เดียวกันกับจุดตัวแทนคลัสเตอร์หรือไม่ โดยจุดข้างเคียงเป็นผลที่เกิดจากการคืนค่าจุดที่ได้จากฟังก์ชัน QueryRegion (QR) โดยที่การคืนค่าจุดของแต่ละแบบจะมีความต่างกัน ดังจะกล่าวต่อไปนี้

3.4.3.1 ลักษณะจุดข้างเคียงเฉพาะจุดที่มีความคล้ายกับจุดที่เป็นจุดตัวแทนในการแพร่ (QR Type 1)

เป็นการหาจุดข้างเคียงเฉพาะจุดที่มีค่าความต่างอัตราส่วนคลาสระหว่างจุดตัวแทนกับจุดนั้นๆ ไม่เกินค่า MaxDiff ที่กำหนด ดังแสดงในรูปที่ 3.8

```
1.  QueryRegion(currentP, representativeP, Eps, MinPts, MaxDiff): SetOfPoint
2.      RETURN NCR_Eps(representativeP)
3.  END // QueryRegion
```

รูปที่ 3.8 ฟังก์ชัน QueryRegion ที่ใช้ลักษณะการหาจุดข้างเคียงเฉพาะจุดที่คล้ายกับจุดตัวแทนในการแพร่

ลักษณะการหาจุดข้างเคียงแบบนี้ จุดที่จะได้จากฟังก์ชัน QueryRegion จะมีเฉพาะจุดที่มีค่าความต่างอัตราส่วนคลาสกับจุดที่เป็นตัวแทนของคลัสเตอร์ ไม่เกินค่า MaxDiff

3.4.3.2 ลักษณะจุดข้างเคียงทั้งหมดเมื่อจุดที่คล้ายกับจุดตัวแทนในการแพร่เกินค่า MinPts (QR Type 2)

เป็นการหาจุดข้างเคียงทั้งหมดที่อยู่ในพื้นที่ ถ้าจุดข้างเคียงมีจำนวนจุดที่มีความต่างอัตราส่วนคลาสไม่เกิน MaxDiff เกินจำนวน MinPts ที่กำหนด แต่ถ้าไม่เป็นไปตามเงื่อนไขดังกล่าวจะกำหนดจุดข้างเคียงเป็น NULL ดังแสดงในรูปที่ 3.9

```
1.  QueryRegion(currentP, representativeP, Eps, MinPts, MaxDiff): SetOfPoint
2.      IF NCR_Esp(representativeP)>MinPts THEN
3.          RETURN NULL
4.      ELSE
5.          RETURN N_Esp(currentP)
6.      ENDIF
7.  END // QueryRegion
```

รูปที่ 3.9 ฟังก์ชัน QueryRegion ที่ใช้ลักษณะการหาจุดข้างเคียงทั้งหมดเมื่อจุดที่คล้ายจุดตัวแทนในการแพร่เกินค่า MinPts

3.4.3.3 ลักษณะจุดข้างเคียงทั้งหมดเมื่อมีจุดที่คล้ายจุดตัวแทนในการแพร่เป็นส่วนหลัก (QR Type3)

เป็นการหาจุดข้างเคียงทั้งหมดถ้าจำนวนจุดที่มีค่าความต่างอัตราส่วนคลาสไม่เกินค่า MaxDiff มีมากกว่าครึ่งหนึ่งของจำนวนทุกจำนวนที่มีอยู่ทั้งหมดในพื้นที่ แต่ถ้าไม่เป็นไปตามเงื่อนไขดังกล่าวจะกำหนดการจุดข้างเคียงเป็น NULL ดังแสดงในรูปที่ 3.10

```
1.  QueryRegion(currentP, representativeP, Eps, MinPts, MaxDiff): SetOfPoint
2.      IF NCR_Esp(representativeP)>N_Eps(currentP)/2 AND sofMatch.size()>=MintPts THEN
3.          RETURN N_Eps(currentP)
4.      ELSE
5.          RETURN NULL
6.      ENDIF
7.  END // QueryRegion
```

รูปที่ 3.10 ฟังก์ชัน QueryRegion ที่ใช้ลักษณะการหาจุดข้างเคียงทั้งหมดเมื่อจุดที่คล้ายจุดตัวแทนในการแพร่เป็นส่วนหลัก

3.5 การระบุคลัสเตอร์ (Cluster Identification)

การระบุคลัสเตอร์คือการบ่งชี้จุด ว่าจุดดังกล่าวอยู่ในคลัสเตอร์ใด โดยจุดนั้นจะอยู่ในคลัสเตอร์ใดนั้น ระยะห่างระหว่างจุดกับคลัสเตอร์นั้น จะต้องน้อยที่สุดเมื่อเทียบกับคลัสเตอร์อื่น ซึ่งเป็นไปตาม (3.9)

$$q \in C_i \leftrightarrow \exists p \in C_i \; and \; \forall o \in C_j \mid dist(p,q) < dist(o,q) \qquad (3.9)$$

โดยที่

C_i	คือ คลัสเตอร์ i
C_j	คือ คลัสเตอร์ j
dist(p,q)	คือ ระยะระหว่างจุด p และ q
dist(o,q)	คือ ระยะระหว่างจุด o และ q

เมื่อทำการตรวจสอบจุด q ตาม (3.9) แล้วจะได้ว่า q อยู่ในคลัสเตอร์ใด และการหาว่าอัตราส่วนคลาสของจุด q นั้นเป็นเท่าใดสามารถหาได้ตามสมการ (3.10) ดังนี้

$$CR(q) = \frac{\sum_{p \in C_i} CR(p)}{N_{C_i}} \qquad (3.10)$$

โดยที่

CR(q)	คือ อัตราส่วนคลาสของจุด q
i	คือ คลัสเตอร์ที่จุด q เป็นสมาชิก
CR(p)	คือ อัตราส่วนคลาสของจุด p
N_{Ci}	คือ จำนวนจุดทั้งหมดของคลัสเตอร์ C_i

3.6 การวัดประสิทธิภาพของการจัดกลุ่ม

การวัดความถูกต้องของการจัดกลุ่มข้อมูลที่มีการซ้อนทับกันในวิทยานิพนธ์นี้ใช้สัมประสิทธิ์ประสิทธิภาพการจัดกลุ่ม

3.6.1 ค่าสัมประสิทธิประสิทธิภาพของการจัดกลุ่ม

เนื่องจากในการจัดกลุ่มข้อมูล ความต้องการในการจัดกลุ่มก็คือ เพื่อต้องการให้ได้ คลัสเตอร์ที่มีความคล้ายกัน และจำนวนของคลัสเตอร์น้อยที่สุดเท่าที่จะเป็นไปได้โดยยังคงความคล้ายคลึงกันของคลัสเตอร์อยู่ ดังนั้นในงานวิจัยนี้จึงได้เสนอสัมประสิทธิ์ประสิทธิภาพของการจัดกลุ่ม (Coefficient of Clustering Efficiency:CCE) โดยมีสามารถหาได้ดังสมการ (3.11)

$$CCE = N_{Clusters} * SSE_{norm} \qquad (3.11)$$

โดยที่

$N_{Clusters}$	คือ จำนวนคลัสเตอร์ที่เกิดขึ้นของการจัดกลุ่ม
SSE_{norm}	คือ Normalize of Sum Square Error ของการจัดกลุ่ม

โดย Normalize of Sum Square Error คือ ค่าเฉลี่ยของผลรวมของความผิดพลาดของอัตราส่วนคลาสของแต่ละจุดกับค่าเฉลี่ยอัตราส่วนคลาสของทั้งคลัสเตอร์ที่จุดนั้นเป็นสมาชิกอยู่ การวัดความผิดพลาดสามารถวัดได้จากการพิจารณาว่าข้อมูลนั้นแตกต่างจากค่าเฉลี่ยของคลัสเตอร์เท่าไหร่ ซึ่งสามารถแสดงได้ดังสมการ (3.12)

$$SSE_{norm} = \frac{\sum_{i=1}^{m}\sum_{x \in C_i} \| x - \bar{x}_i \|^2}{N} \qquad (3.12)$$

โดยที่

m	คือ จำนวนกลุ่มทั้งหมด
C_i	คือ คลัสเตอร์แต่ละคลัสเตอร์ i
x	คือ อัตราส่วนคลาสของสมาชิกของคลัสเตอร์นั้น
$\| x - \bar{x}_i \|$	คือ ความต่างของจุดนั้นกับค่าเฉลี่ยของคลัสเตอร์
N	คือ จำนวนจุดทั้งหมด

สัมประสิทธิ์ประสิทธิภาพของการจัดกลุ่ม จะบ่งบอกว่า วิธีการจัดกลุ่มนั้น ๆ มีความผิดพลาดเมื่อเทียบกับจำนวนกลุ่มที่ได้ และจำนวนข้อมูลทั้งหมดเป็นเท่าใด ดังนั้นถ้าค่าสัมประสิทธิ์ประสิทธิภาพการจัดกลุ่มต่ำ แสดงว่ามีประสิทธิภาพดี

3.7 การประยุกต์ใช้งาน

ในการประยุกต์ใช้งานของวิธีการ ODBSCAN ที่เสนอในวิทยานิพนธ์นี้ สามารถทำได้ตามขั้นตอนดังต่อไปนี้

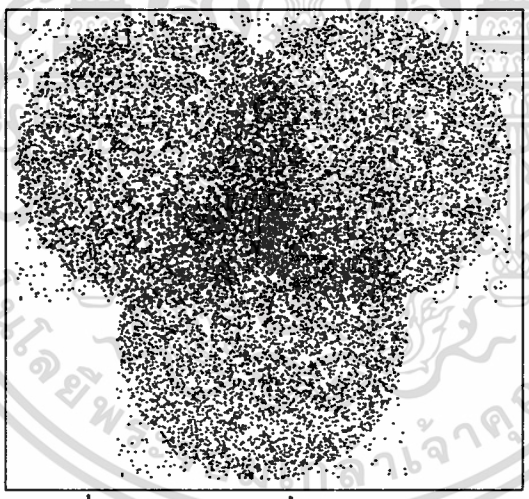
1. นำข้อมูลที่ใช้ในการเรียนรู้มาจัดกลุ่ม โดยกำหนดค่าพารามิเตอร์ต่าง ๆ ให้เหมาะสมเพื่อให้ได้ค่าสัมประสิทธิ์ประสิทธิภาพการจัดกลุ่ม (CCE) ต่ำที่สุด ผลที่ได้คือสามารถบอกกลุ่มของข้อมูลที่นำมาเรียนรู้ว่าเป็นกลุ่มใด โดยไม่ต้องบอกกลุ่มเริ่มต้น แต่ข้อมูลจะต้องบอกว่าเป็นข้อมูลคลาสใด

2. นำข้อมูลทดสอบที่ไม่ทราบคลาสมาระบุคลัสเตอร์ (Cluster Identification)

3. นำผลลัพธ์ที่ได้จากการระบุคลัสเตอร์ คืออัตราส่วนคลาสมาเป็นคำตอบในการที่จะบ่งบอกว่า ข้อมูลทดสอบมีความน่าจะเป็นที่จะเป็นคลาสนั้น ๆ ตามอัตราส่วนคลาสนั้น เช่น ถ้าอัตราส่วนคลาสที่ได้จากการระบุคลัสเตอร์เป็น คลาส A คือ 0.2 คลาส B คือ 0.5 และคลาส C คือ 0.3 ดังนั้นข้อมูลทดสอบมีความน่าจะเป็นคลาส A 20% คลาส B 50% และคลาส C 30% เป็นต้น

บทที่ 4
การทดลองการจัดกลุ่มข้อมูลที่มีการซ้อนทับกันเบื้องต้น ของวิธีการ ODBSCAN และวิธีการอื่น

ในบทนี้จะกล่าวถึงการทดลองหาค่าพารามิเตอร์ของ ODBSCAN ที่เหมาะสมในการจัดกลุ่ม ซึ่งประกอบด้วย ลักษณะค่าผลต่างอัตราส่วนคลาส ลักษณะการหาจุดข้างเคียง และตัวแทนที่ใช้ในการแพร่ และการทดลองเปรียบเทียบกับวิธีการอื่น ในการจัดกลุ่มข้อมูลที่มีการกระจายแบบสม่ำเสมอที่มีการซ้อนทับกัน โดยข้อมูลที่ใช้ในการทดลองประกอบด้วยข้อมูลจุดจำนวน 3 คลาส จำนวนทั้งหมด 17,257 จุด ซึ่งเป็นข้อมูลที่มีการกระจายแบบสม่ำเสมอ คลาสแต่ละมีจำนวนจุด 5,500 จุด และมีข้อมูลรบกวนที่อยู่บริเวณรอบข้อมูลทั้งหมด 575 จุดโดยมีลักษณะของข้อมูลเป็นดังรูปที่ 4.1 เนื่องจากเป็นข้อมูลที่เราทราบว่าควรจะแบ่งออกเป็น 7 คลัสเตอร์ ดังนั้นผลของการจัดกลุ่มสามารถหาได้ว่าผิดไปจากที่ควรจะเป็นหรือไม่



รูปที่ 4.1 แสดงข้อมูลที่ใช้ในการทดลอง

4.1 การทดลองเกี่ยวกับลักษณะของผลต่างอัตราส่วนคลาส

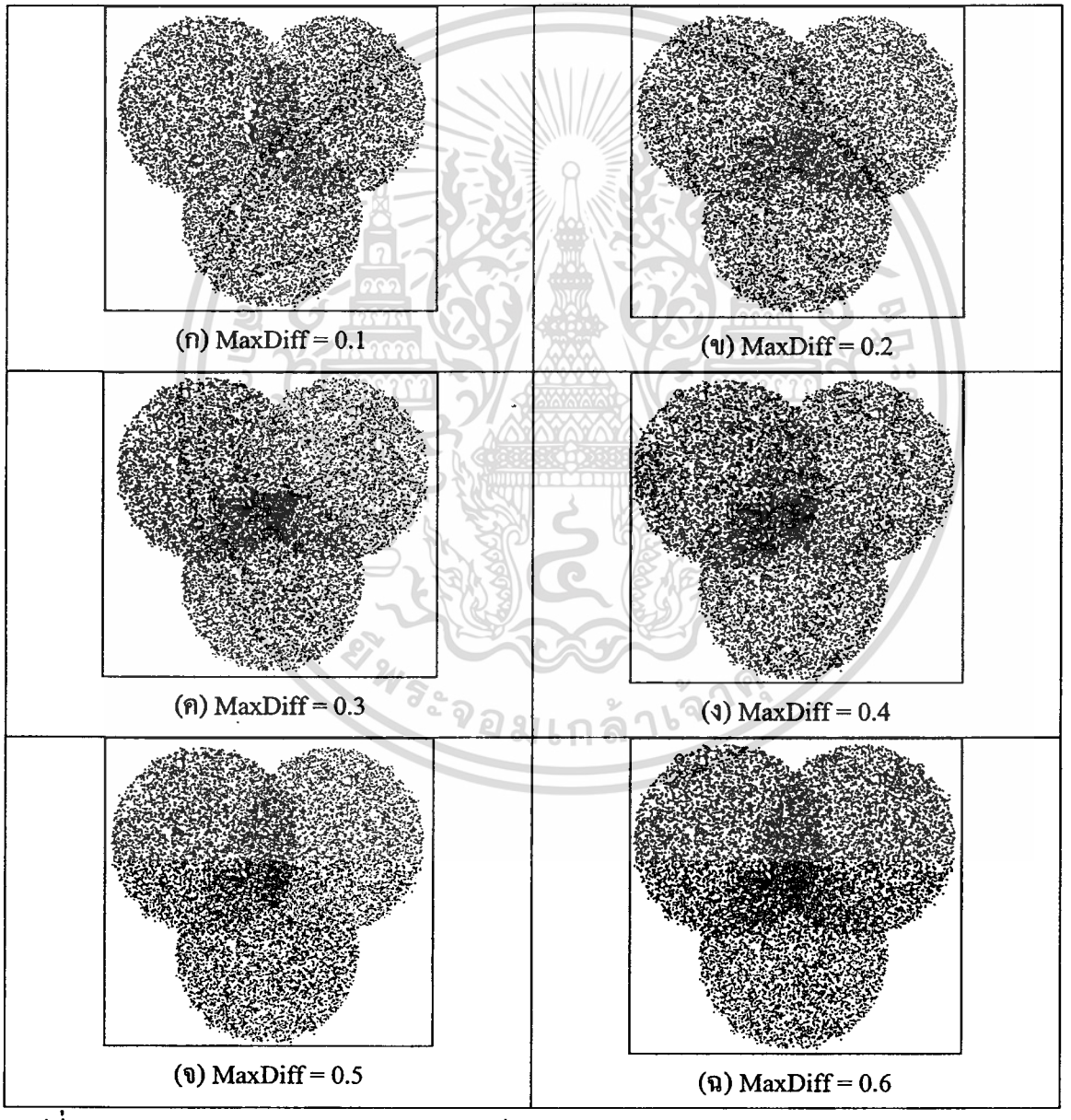
การหาค่าของความต่างอัตราส่วนคลาสนั้นสามารถหาได้ 3 วิธีดังนี้

1. หาจากผลรวมความต่างอัตราส่วนของแต่ละคลาส (DCR Type 1)
2. หาจากความต่างสูงสุดของความต่างอัตราส่วนของแต่ละคลาส (DCR Type 2)
3. หาจากผลรวมความต่างของอัตราส่วนแต่ละคลาสที่มีการกำจัดข้อมูลรบกวน (DCR Type 3)

ในการทดลองนี้จะเป็นการทดลองเพื่อหาค่า MaxDiff ที่เหมาะสมสำหรับข้อมูลที่มีการกระจายแบบสม่ำเสมอ โดยในการทดลองจะทำการกำหนดพารามิเตอร์การหาจุดข้างเคียง และกำหนดตัวแทนในการแพร่ เป็น QR Type 2 และ PRC Type 2 ตามลำดับ

4.1.1 ผลรวมความต่างอัตราส่วนของแต่ละคลาส (DCR Type 1)

การทดลองจะทำการทดลอง โดยใช้ค่า MaxDiff เป็น 0.1, 0.2, 0.3, 0.4, 0.5 และ 0.6 ของลักษณะความต่างอัตราส่วนคลาสแบบผลรวมความต่างอัตราส่วนของแต่ละคลาส โดยผลการจัดกลุ่มแต่ละค่าได้ผลดังรูปที่ 4.2 และตารางที่ 4.1



รูปที่ 4.2แสดงผลการทดลองจากการจัดกลุ่มด้วยลักษณะของผลต่างอัตราส่วนคลาสแบบผลรวมความต่างอัตราส่วนของแต่ละคลาส โดยใช้ MaxDiff ค่าต่าง ๆ

ตารางที่ 4.1 ผลการทดลองเปรียบเทียบค่า MaxDiff ของ DCR Type 1

MaxDiff	จำนวนคลัสเตอร์	SSE	CCE
0.1	90	92.30	0.4814
0.2	32	128.85	0.2389
0.3	13	156.08	0.1176
0.4	8	206.63	0.0958
0.5	8	258.21	0.1322
0.6	6	384.56	0.1338

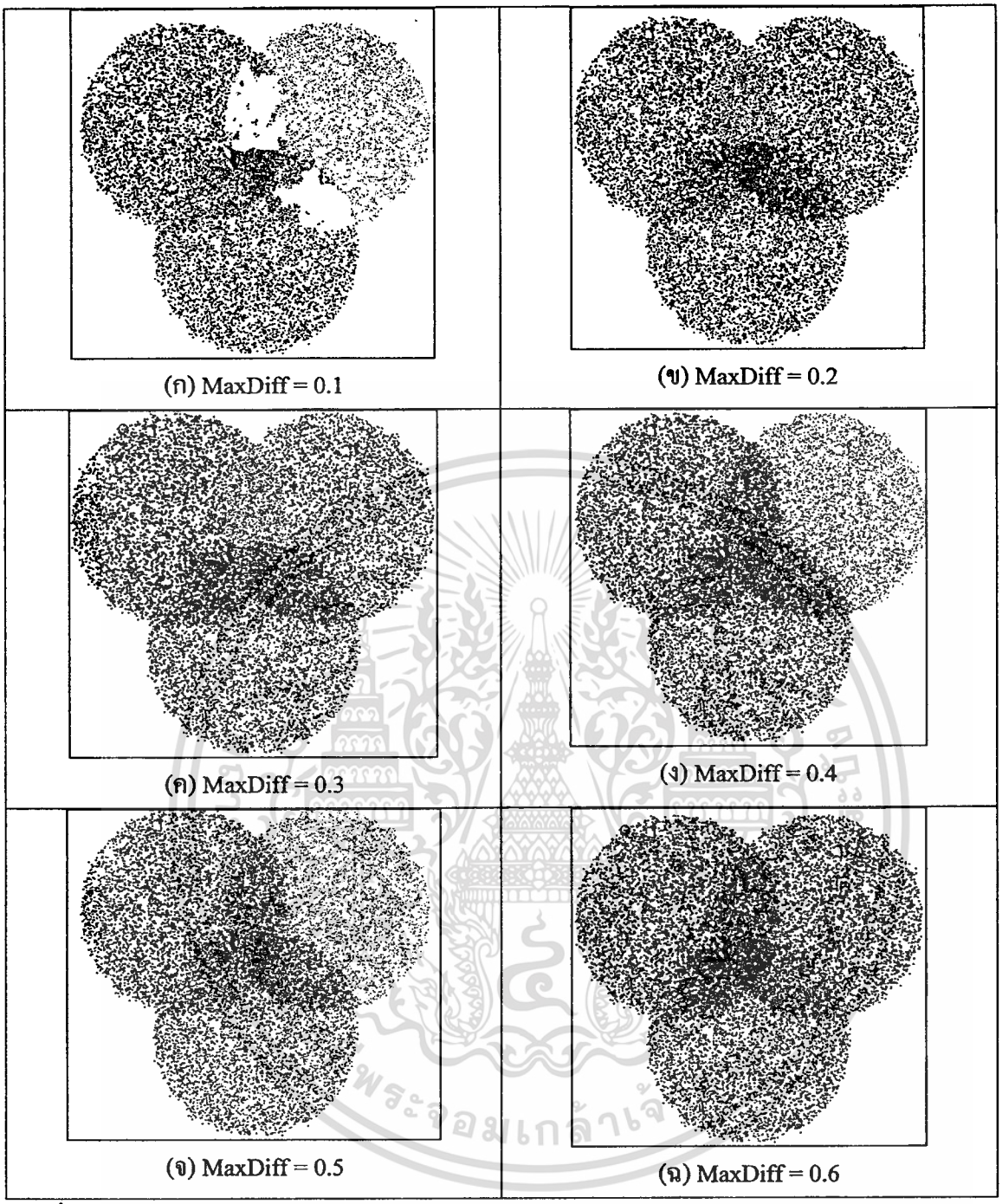
จากผลการทดลองพบว่าค่าของ MaxDiff ที่ให้ผลการจัดกลุ่มที่ดีที่สุดของ DCR Type 1 สำหรับข้อมูลชุดนี้มีค่าเป็น 0.4 เนื่องจากให้ค่าสัมประสิทธิ์ประสิทธิภาพการจัดกลุ่มต่ำที่สุด คือ 0.0958 ซึ่งเมื่อพิจารณารูปที่ 4.2 (ง) จะพบว่าจะได้ให้ผลการจัดกลุ่มที่ดี แต่อย่างไรก็ตามยังคงมีคลัสเตอร์เล็ก ๆ ซึ่งเป็นผลมาจากข้อมูลรบกวนที่บริเวณขอบของข้อมูลที่ซ้อนทับกัน

4.1.2 ผลรวมความต่างอัตราส่วนของแต่ละคลาสที่กำจัดข้อมูลรบกวน (DCR Type 2)

การทดลองจะทำการทดลอง โดยใช้ค่า MaxDiff เป็น 0.1,0.2,0.3, 0.4, 0.5 และ 0.6 ของลักษณะความต่างอัตราส่วนคลาสแบบผลรวมความต่างอัตราส่วนของแต่ละคลาสที่กำจัดข้อมูลรบกวน ซึ่งมีค่ากำหนดของข้อมูลรบกวน (α) เท่ากับ 0.05 โดยผลลัพธ์การจัดกลุ่มแต่ละค่า ดังแสดงในตารางที่ 4.2 และรูปที่ 4.3

ตารางที่ 4.2 ผลการทดลองเปรียบเทียบค่า MaxDiff ของ DCR Type 2

MaxDiff	จำนวนคลัสเตอร์	SSE	CCE
0.1	81	97.42	0.4573
0.2	27	138.16	0.2162
0.3	12	172.41	0.1199
0.4	7	223.27	0.0906
0.5	7	351.03	0.1424
0.6	6	418.51	0.1455

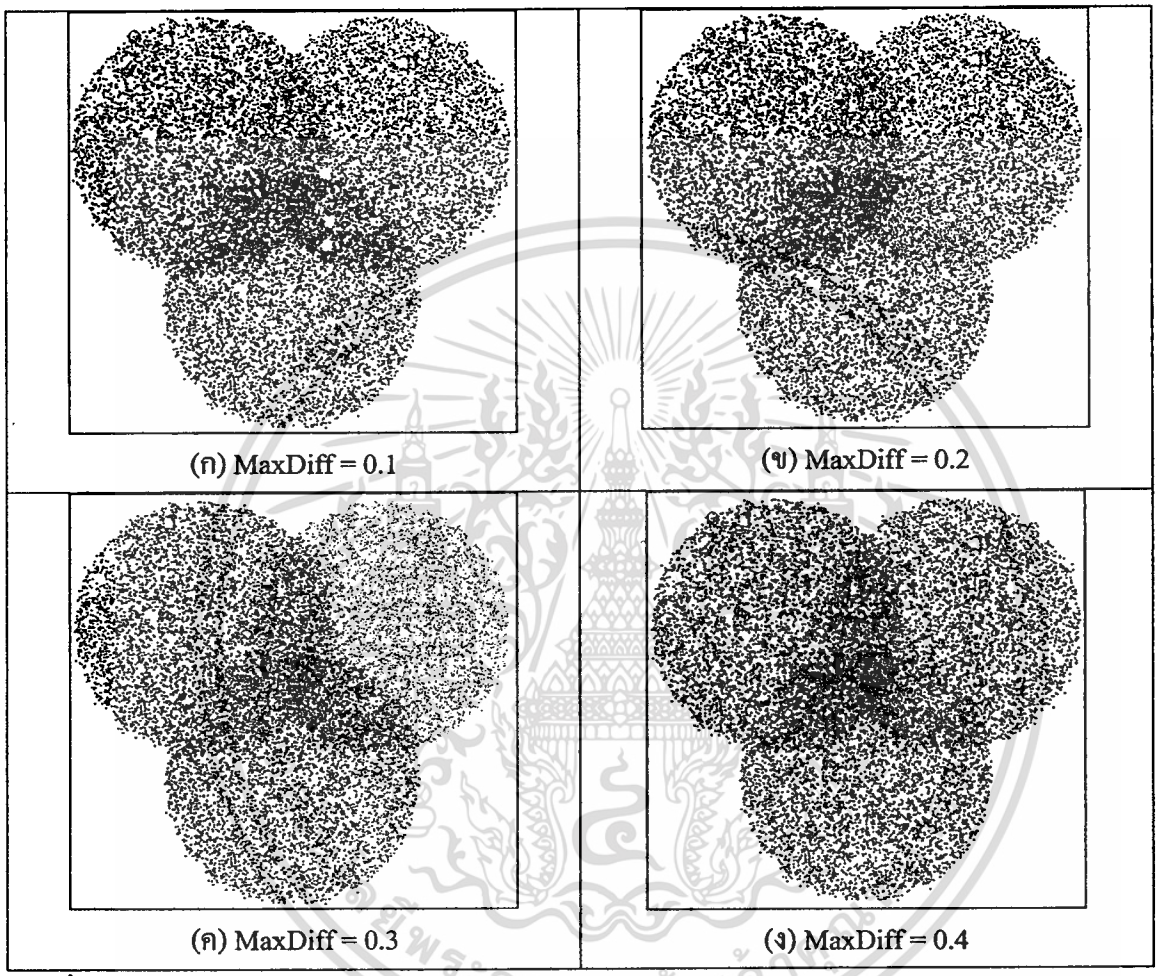


รูปที่ 4.3แสดงผลการทดลองจากการจัดกลุ่มด้วยลักษณะของผลต่างอัตราส่วนคลาสแบบผลรวมความต่างอัตราส่วนของแต่ละคลาสที่กำจัดข้อมูลรบกวน โดยใช้ MaxDiff ค่าต่าง ๆ

จากผลการทดลองพบว่าค่าของ MaxDiff ที่ให้ผลการจัดกลุ่มที่ดีที่สุดของ DCR Type 2 สำหรับข้อมูลชุดนี้มีค่าเป็น 0.4 เนื่องจากให้ค่าสัมประสิทธิ์ประสิทธิภาพการจัดกลุ่มต่ำที่สุด คือ 0.0906 เนื่องจากให้ค่าสัมประสิทธิ์ประสิทธิภาพการจัดกลุ่มต่ำที่สุด คือ 0.0958 ซึ่งเมื่อพิจารณารูปที่ 4.3 (ง) จะพบว่าจะได้ให้ผลการจัดกลุ่มที่ดี มีจำนวนของคลัสเตอร์เท่ากับจำนวนคลัสเตอร์ที่ควรจะแบ่งได้

4.1.3 ความต่างอัตราส่วนคลาสสูงสุดที่กำจัดข้อมูลรบกวน (DCR Type 3)

การทดลองจะทำการทดลอง โดยใช้ค่า MaxDiff เป็น 0.1, 0.2, 0.3 และ 0.4 ของลักษณะความต่างอัตราส่วนคลาสแบบความต่างอัตราส่วนคลาสสูงสุด โดยผลการจัดกลุ่มแต่ละค่าได้ผลดังรูปที่ 4.4



รูปที่ 4.4 แสดงผลการทดลองจากการจัดกลุ่มด้วยลักษณะของผลต่างอัตราส่วนคลาสแบบความต่างอัตราส่วนคลาสสูงสุด โดยใช้ MaxDiff ค่าต่าง ๆ

ตารางที่ 4.3 ผลการทดลองเปรียบเทียบค่า MaxDiff ของ DCR Type 3

MaxDiff	จำนวนคลัสเตอร์	SSE	CCE
0.1	32	113.90	0.2112
0.2	8	206.63	0.0958
0.3	6	315.03	0.1220
0.4	6	859.08	0.2987

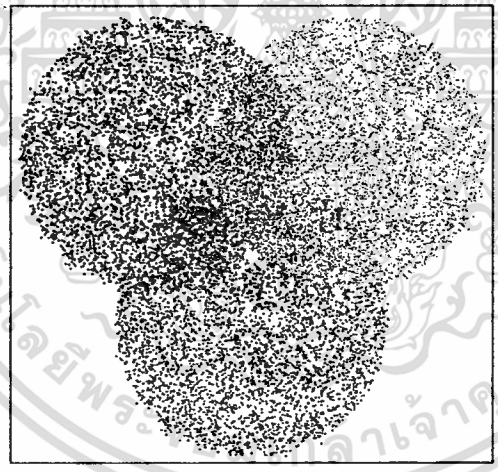
จากผลการทดลองพบว่าค่าของ MaxDiff ที่ให้ผลการจัดกลุ่มที่ดีที่สุดของ DCR Type 3 สำหรับข้อมูลชุดนี้มีค่าเป็น 0.2 เนื่องจากมีค่าสัมประสิทธิ์ประสิทธิภาพการจัดกลุ่มต่ำที่สุด คือ 0.0958 ซึ่งเมื่อพิจารณารูปที่ 4.4 (ข) จะพบว่าจะได้ให้ผลการจัดกลุ่มที่ดี แต่อย่างไรก็ตามยังคงมีคลัสเตอร์เล็ก ๆ ซึ่งเป็นผลมาจากข้อมูลรบกวนที่บริเวณขอบของข้อมูลที่ซ้อนทับกัน

4.2 การเปรียบเทียบลักษณะการหาจุดข้างเคียง

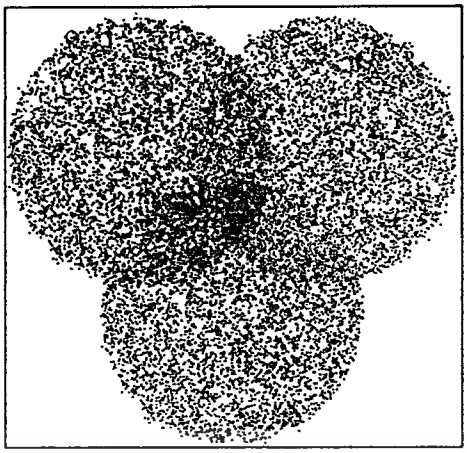
ดังที่กล่าวไว้ในหัวข้อ 3.4.3 เกี่ยวกับลักษณะการหาจุดข้างเคียง ซึ่งมีลักษณะการหาอยู่ 3 รูปแบบดังนี้

1. ลักษณะจุดข้างเคียงเฉพาะจุดที่มีความคล้ายกับจุดที่เป็นจุดตัวแทน
2. ลักษณะจุดข้างเคียงทั้งหมดเมื่อจุดที่คล้ายกับจุดตัวแทนเกินค่า MinPts
3. ลักษณะจุดข้างเคียงทั้งหมดเมื่อมีจุดที่คล้ายจุดตัวแทนเป็นส่วนหลัก

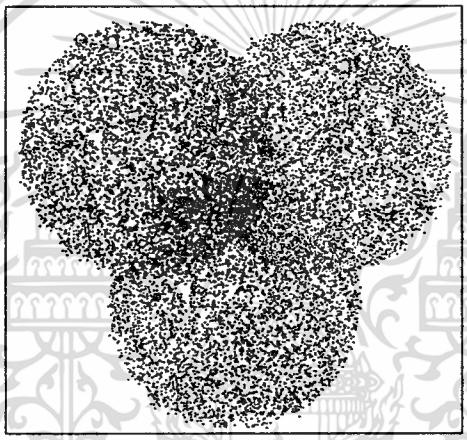
ในการทดลองจะทำการกำหนดพารามิเตอร์ค่าผลต่างอัตราส่วนคลาส และตัวแทนในการแพร่ให้เป็น DCR Type 1 และ PRC Type 2 ตามลำดับ และค่าความต่างสูงสุดของอัตราส่วนคลาส (MaxDiff) เท่ากับ 0.4 ซึ่งผลการทดลองได้แสดงในรูปที่ 4.5 - 4.7 ตามลำดับ



รูปที่ 4.5 แสดงผลการทดลองจากการจัดกลุ่มโดยใช้ลักษณะการหาจุดข้างเคียงเฉพาะจุดที่มีความคล้ายกับจุดที่เป็นจุดตัวแทน



รูปที่ 4.6แสดงผลการทดลองจากการจัดกลุ่มโดยใช้ลักษณะการหาจุดข้างเคียงทั้งหมดเมื่อจุดที่คล้ายกับจุดตัวแทนเกินค่า MinPts



รูปที่ 4.7แสดงผลการทดลองจากการจัดกลุ่มโดยใช้ลักษณะการหาจุดข้างเคียงทั้งหมดเมื่อมีจุดที่คล้ายกับจุดตัวแทนเป็นส่วนหลัก

ตารางที่ 4.4 ผลการทดลองของลักษณะการหาจุดข้างเคียง

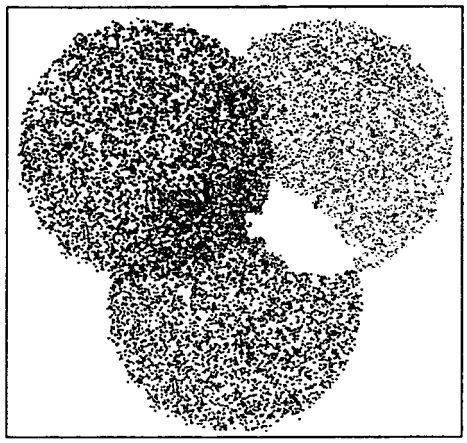
วิธีการ	จำนวนคลัสเตอร์	SSE	CCE
ลักษณะจุดข้างเคียงเฉพาะจุดที่มีความคล้ายกับจุดที่เป็นจุดตัวแทน (QR Type 1)	50	93.64	0.2713
ลักษณะจุดข้างเคียงทั้งหมดเมื่อจุดที่คล้ายกับจุดตัวแทนเกินค่า MinPts (QR Type 2)	8	206.63	0.0958
ลักษณะจุดข้างเคียงทั้งหมดเมื่อมีจุดที่คล้ายกับจุดตัวแทนเป็นส่วนหลัก (QR Type 3)	8	218.80	0.1014

4.3 การหาวิธีการที่เหมาะสมสำหรับตัวแทนที่ใช้ในการแพร่

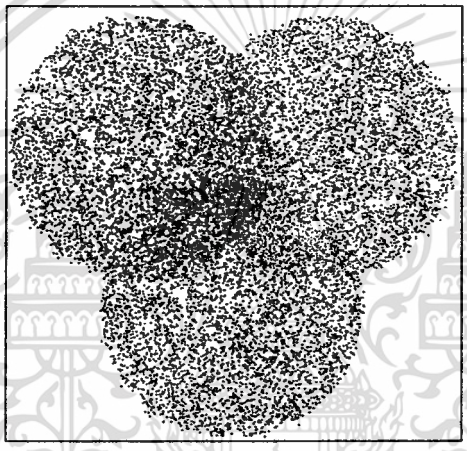
การหาตัวแทนในการแพร่ เป็นส่วนสำคัญส่วนหนึ่งในการที่จะทำให้ได้คลัสเตอร์ที่มีคุณภาพ เนื่องจากตัวแทนนี้จะถูกใช้ในการเปรียบเทียบความต่างอัตราส่วนคลาสกับจุดที่จะแพร่ไป โดยมีการเลือกจุดตัวแทนได้ 4 รูปแบบด้วยกันคือ

1. ตัวแทนในการแพร่โดยใช้จุดแรกที่ทำให้เกิดคลัสเตอร์
2. ตัวแทนในการแพร่โดยใช้ค่าเฉลี่ย

ในการทดลองจะทำการกำหนดพารามิเตอร์การหาจุดข้างเคียง และค่าความต่างอัตราส่วนคลาส ให้เป็นลักษณะจุดข้างเคียงทั้งหมดเมื่อจุดที่คล้ายกับจุดตัวแทนเกินค่า MinPts และตัวแทนในการแพร่โดยใช้ค่าเฉลี่ย ซึ่งผลการทดลองได้แสดงในรูปที่ 4.8 และ 4.9 ตามลำดับ



รูปที่ 4.8แสดงผลการทดลองจากการจัดกลุ่มโดยการใช้ตัวแทนในการแพร่โดยใช้จุดแรกที่ทำให้เกิดคลัสเตอร์



รูปที่ 4.9 แสดงผลการทดลองจากการจัดกลุ่มโดยการใช้ตัวแทนในการแพร่โดยใช้ค่าเฉลี่ย

ตารางที่ 4.5 ผลการทดลองของตัวแทนที่ใช้ในการแพร่

วิธีการ	จำนวนคลัสเตอร์	SSE	CCE
ตัวแทนในการแพร่โดยใช้จุดแรกที่ทำให้เกิดคลัสเตอร์ (PRC Type 1)	25	178.19	0.2581
ตัวแทนในการแพร่โดยใช้ค่าเฉลี่ย (PRC Type 2)	8	206.63	0.0958

4.4 สรุปวิธีที่เหมาะสมสำหรับแต่ละพารามิเตอร์

จากการทดลองที่เกี่ยวกับการหาความต่างอัตราส่วนคลาสทั้งสามวิธี ตามตารางที่ 4.1-4.3 จะพบว่าค่าความต่างสูงสุดที่จะให้ได้ผลลัพธ์ในการจัดกลุ่มที่ดี มีค่าแตกต่างกันขึ้นอยู่กับวิธีการหาค่าผลต่างอัตราส่วนคลาส โดยรวมแล้วได้ผลการจัดกลุ่มที่ได้ผลดีใกล้เคียงกัน แต่เมื่อพิจารณาค่าสัมประสิทธิ์ประสิทธิภาพการจัดกลุ่มแล้ว จะพบว่าวิธีการหาความต่างอัตราส่วนคลาสแบบ DCR Type 2 และค่าเป็น MaxDiff เท่ากับ 0.4 ได้ค่าที่ต่ำที่สุด ซึ่งหมายความว่าเป็นวิธีการจัดกลุ่มที่ดีที่สุดสำหรับข้อมูลชุดนี้

ผลการทดลองเกี่ยวกับลักษณะการหาจุดข้างเคียง ดังตารางที่ 4.4 พบว่า QR Type 2 และ QR Type 3 มีค่าสัมประสิทธิ์ประสิทธิภาพการจัดกลุ่มใกล้เคียงกัน แต่วิธีการ QR Type 2 มีค่าสัมประสิทธิ์ประสิทธิภาพการจัดกลุ่มต่ำที่สุด เนื่องจาก QR Type 3 จะมีบริเวณที่ไม่พบข้อมูลที่เป็นส่วนหลัก ดังนั้นจึงทำให้ค่าข้อมูลบริเวณนั้นเป็นข้อมูลรบกวน ซึ่งเป็นผลให้การจัดกลุ่มมีประสิทธิภาพด้อยกว่า

ผลการทดลองเกี่ยวกับการหาตัวแทนในการแพร่ ดังตารางที่ 4.5 พบว่าวิธีการใช้ตัวแทนในการแพร่โดยใช้ค่าเฉลี่ย (PRC Type 2) มีค่าสัมประสิทธิ์ประสิทธิภาพการจัดกลุ่มต่ำที่สุด เนื่องจากจุดตัวแทนที่ใช้ในการแพร่ใช้อัตราส่วนคลาสเฉลี่ยของข้อมูลทั้งคลัสเตอร์ ทำให้ความผิดพลาดของอัตราส่วนคลาสของจุดที่แพร่ไปกับอัตราส่วนคลาสของทั้งคลัสเตอร์น้อย

ดังนั้นวิธีการที่เหมาะสมสำหรับพารามิเตอร์ต่างๆ ของ ODBSCAN ของข้อมูลที่ใช้ในการทดลองนี้คือ ลักษณะการหาจุดข้างเคียงคือวิธีการ QR Type 2 ลักษณะผลต่างอัตราส่วนคลาสคือวิธีการ DCR Type 2 และตัวแทนในการแพร่ คือ วิธีการ PRC Type 2

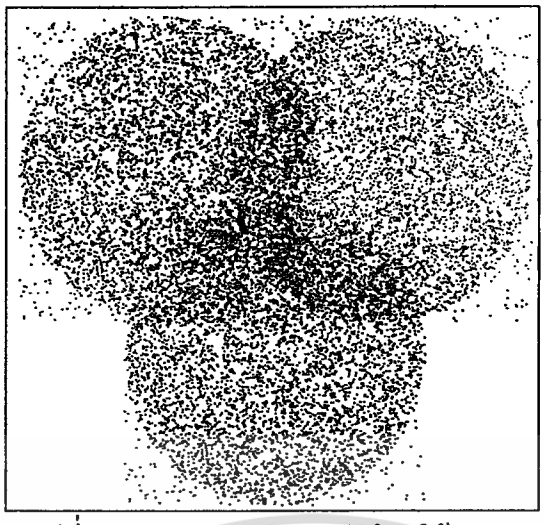
4.5 การทดลองเปรียบเทียบกับวิธีการอื่น

ในส่วนนี้จะทำการทดลองเปรียบเทียบการจัดกลุ่มวิธีการ Fuzzy c-Mean (FCM) และ ODBSCAN

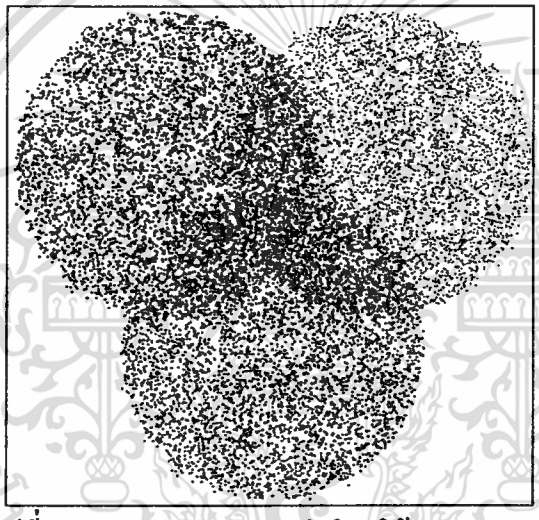
ในการทดลอง สำหรับ FCM ได้ทำการกำหนดค่าของจำนวนคลัสเตอร์เป็น 7 และมีค่าหยุด ε=0.6 และวิธีการ ODBSCAN ใช้พารามิเตอร์เป็น DCR Type 2 PRC Type 2 QR Type 2 และ MaxDiff=0.4 ได้ผลการทดลองดังตารางที่ 4.6 และรูปที่ 4.10 และ 4.11

ตารางที่ 4.6 ผลการทดลองเปรียบเทียบการจัดกลุ่มระหว่าง FCM และ ODBSCAN

วิธีการ	จำนวนคลัสเตอร์	SSE	CCE
FCM	7	1481.87	0.6011
ODBSCAN	7	223.27	0.0906



รูปที่ 4.10 แสดงผลการจัดกลุ่มโดยใช้ FCM



รูปที่ 4.11 แสดงผลการจัดกลุ่มโดยใช้ ODBSCAN

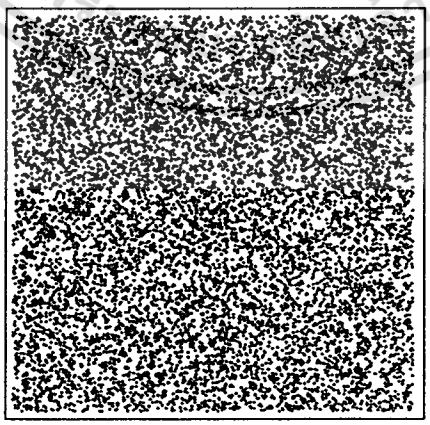
จากผลการทดลองจะพบได้ว่าวิธีการ FCM ไม่สามารถที่จะทำให้เกิดคลัสเตอร์ที่ดีได้ เนื่องจากค่าสัมประสิทธิ์ประสิทธิภาพการจัดกลุ่มที่ได้จากการทดลองมีค่าสูงมาก เมื่อเทียบกับการจัดกลุ่มโดยใช้ ODBSCAN และรูปร่างของคลัสเตอร์ที่ได้ ไม่เป็นไปตามแนวโน้มที่ต้องการในการจัดกลุ่มข้อมูลชุดนี้

บทที่ 5
การทดลองเกี่ยวกับข้อมูลที่มีการกระจายแบบต่าง ๆ

จากบทที่ 4 จะพบว่าค่าพารามิเตอร์ต่างๆ ที่ให้ผลลัพธ์ที่เหมาะสมที่สุดสำหรับการจัดกลุ่มข้อมูลที่มีการซ้อนทับกันคือ DCR Type 2, PRC Type 2 และ QR Type 2 ซึ่งในบทนี้จะนำพารามิเตอร์เหล่านั้นไปใช้ในการทดลองกับข้อมูลที่มีการกระจายแบบต่าง ๆ ได้แก่ข้อมูลที่มีการกระจายแบบลดลงสม่ำเสมอ ข้อมูลที่มีการกระจายแบบเกาส์เซียน และข้อมูลลายมือเขียนภาษาไทย เพื่อทดสอบว่าวิธีการที่ได้นำเสนอในวิทยานิพนธ์นี้สามารถประยุกต์ใช้กับการกระจายของข้อมูลแบบต่างๆ ดังที่กล่าวมาได้หรือไม่

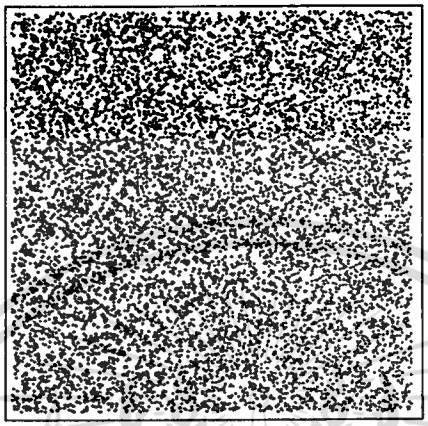
5.1 การทดลองการจัดกลุ่มข้อมูลที่มีกระจายแบบลดลงสม่ำเสมอ (Constant Fade-out Distribution)

สำหรับข้อมูลที่ใช้ในการทดลองในส่วนนี้ได้มาจากการจำลองการสร้างข้อมูลที่มีการกระจายแบบลดลงสม่ำเสมอ จำนวน 2 คลาส จำนวนคลาสละ 5,500 จุด รวมเป็น 11,000 จุด โดยข้อมูลแต่ละคลาสจะให้ทำการลดลงแบบขั้นบันได กล่าวคือจะทำการแบ่งช่วงความหนาแน่นออกเป็น 10 ระดับ โดยมีความหนาแน่นลดลั่นไปเป็นช่วง ๆ ซึ่งสีน้ำเงินระยะแกนนอนตั้งแต่ 0 ถึง 40 จะมีจำนวนจุดเป็น 1000 ระยะแกนนอนตั้งแต่ 40 ถึง 80 จะมีจุดทั้งหมด 900 จุด และทุก ๆ ระยะ 40 หน่วยจะลดจำนวนจุดลงทีละ 100 จุด เช่นเดียวกันสีแดงจะเริ่มจากระยะที่ 400 ถึง 360 จะมีจำนวนจุด 1000 จุด ระยะที่ 360 ถึง 320 จะมีจำนวนจุดเป็น 900 จุด และลดลงเรื่อย ๆ ตามแกนนอนทุก ๆ ระยะ 40 หน่วยเป็นจำนวน 100 จุด เช่นกัน ดังรูปที่ 5.1



รูปที่ 5.1 ข้อมูลที่ใช้ในการทดสอบที่มีการกระจายแบบลดลงสม่ำเสมอ

การทดลองได้ทำการจัดกลุ่มด้วยพารามิเตอร์ที่เหมาะสมสำหรับดังที่กล่าวมาแล้วในหัวข้อที่ 4.4 ซึ่งมีพารามิเตอร์เป็นดังนี้ QR Type 2, DCR Type 2 และ PRC Type 2 ส่วนค่า MaxDiff มีค่าเป็น 0.2 และ Eps=1.0 ทั้งสองค่านี้ได้จากการทดลองแล้วเลือกค่าที่ดีที่สุดมาใช้ในการทดลองนี้ โดยผลการจัดกลุ่มด้วยพารามิเตอร์ดังกล่าวได้ผลดังรูปที่ 5.2

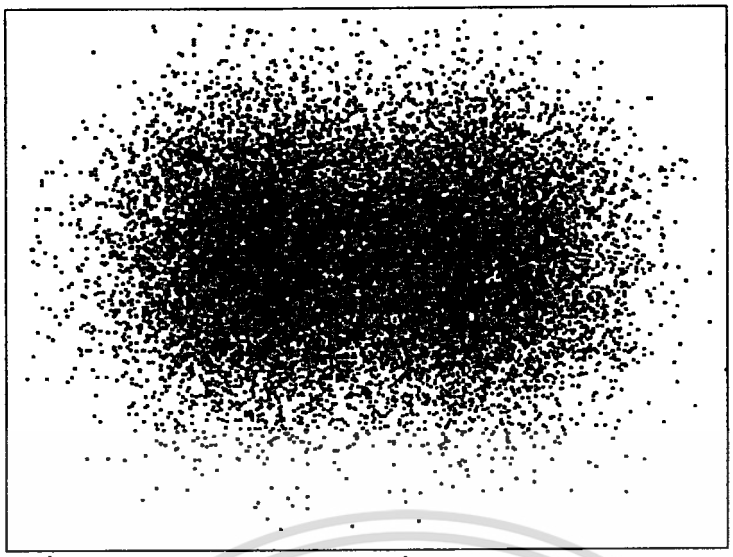


รูปที่ 5.2 ผลการจัดกลุ่มข้อมูลที่มีการกระจายแบบลดลงสม่ำเสมอ

ในการจัดกลุ่มที่ทำการทดลองได้จำนวนคลัสเตอร์ทั้งหมด 12 คลัสเตอร์ โดยมีค่าสัมประสิทธิ์ประสิทธิภาพการจัดกลุ่มเท่ากับ 0.1141 และจากรูปผลการจัดกลุ่ม จะพบได้ว่ารูปร่างของคลัสเตอร์เป็นรูปทรงแบบแท่งในแนวตั้ง ซึ่งมีแนวโน้มตามคลัสเตอร์ที่ควรจะเป็นนั้นเอง

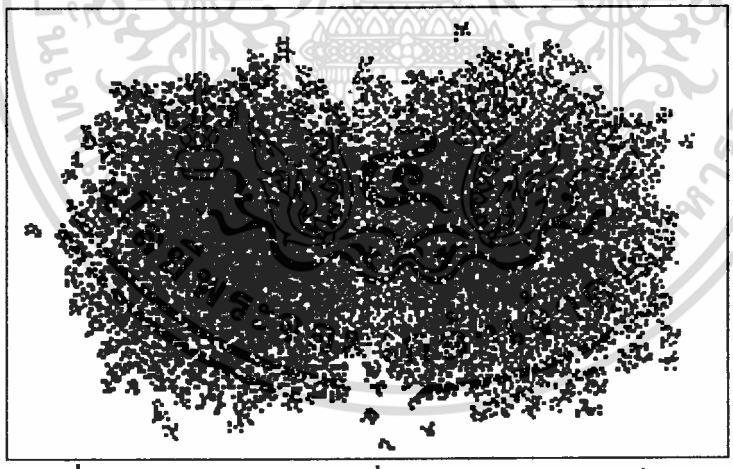
5.2 การทดลองการจัดกลุ่มข้อมูลที่มีการกระจายแบบเกาส์เซียน (Gaussian Distribution)

สำหรับข้อมูลที่ใช้ในการทดลองในส่วนนี้ได้มาจากการจำลองการสร้างข้อมูลที่มีการกระจายแบบเกาส์เซียนที่มีค่า $\sigma_x = 1$ โดยมีค่าเฉลี่ยอยู่ที่ 100 และ 150 ของคลาสสีน้ำเงินและสีแดงตามลำดับ ส่วนในด้านแกน y จะเป็นการกระจายแบบสม่ำเสมอ ซึ่งแต่ละคลาสจะมีจำนวน 10,000 จุด รวมทั้งเป็น 20,000 จุด ดังรูปที่ 5.3



รูปที่ 5.3 ข้อมูลที่ใช้ในการทดสอบที่มีการกระจายแบบเกาส์เซียน

การทดลองได้ทำการจัดกลุ่มด้วยพารามิเตอร์ที่เหมาะสมสำหรับดังที่กล่าวมาแล้วในหัวข้อที่ 4.4 ซึ่งมีพารามิเตอร์เป็นดังนี้ QR Type 2, DCR Type 2 และ PRC Type 2 ส่วนค่า MaxDiff มีค่าเป็น 0.2 และ Eps=1.0 ทั้งสองค่านี้ได้จากการทดลองแล้วเลือกค่าที่ดีที่สุดมาใช้ในการทดลองนี้ โดยผลการจัดกลุ่มด้วยพารามิเตอร์ดังกล่าวได้ผลดังรูปที่ 5.4

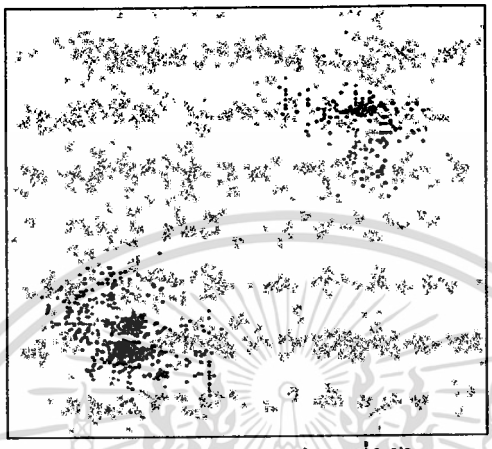


รูปที่ 5.4 ผลการจัดกลุ่มข้อมูลที่มีการกระจายแบบเกาส์เซียน

ในการจัดกลุ่มที่ทำการทดลองได้จำนวนคลัสเตอร์ทั้งหมด 59 คลัสเตอร์ โดยมีค่าสัมประสิทธิ์ประสิทธิภาพการจัดกลุ่มเท่ากับ 0.2825 ซึ่งจากรูปผลการทดลองที่แสดงในรูปที่ 5.4 จะพบว่าบริเวณที่เกิดการซ้อนทับตรงกลางระหว่างสองคลาสจะถูกแบ่งเป็นข้อมูลคลัสเตอร์ใหญ่ ๆ สามคลาสในแนวตั้ง โดยรูปร่างที่ได้นี้มีแนวโน้มใกล้เคียงกับที่ต้องการ

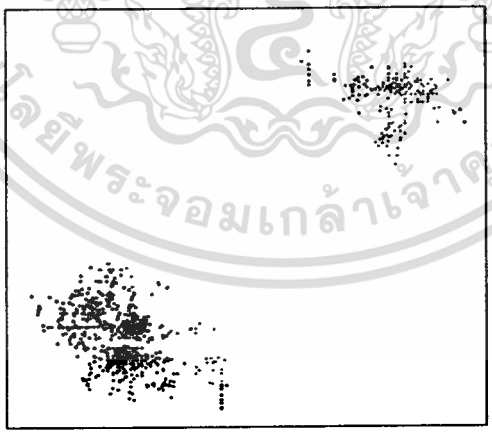
5.3 ผลการทดลองการจัดกลุ่มข้อมูลลายมือเขียนภาษาไทย

สำหรับข้อมูลที่ใช้ในส่วนนี้ใช้ข้อมูลลายมือเขียนอักษรภาษาไทยที่มี 2 คุณลักษณะ คือแกนนอนเป็นผลต่างความโค้งรวมของส่วนที่ 1 และแนวตั้งเป็นผลต่างความโค้งรวมของส่วนที่ 2 ของตัวอักษร ทั้งหมด 34 คลาส และมีจำนวนข้อมูลทั้งหมด 1150 ข้อมูล ดังแสดงในรูปที่ 5.5



รูปที่ 5.5 ข้อมูลลายมือเขียนภาษาไทยที่ใช้ในการทดลอง

การทดลองได้ทำการจัดกลุ่มด้วยพารามิเตอร์ที่เหมาะสมสำหรับดังที่กล่าวมาแล้วในหัวข้อที่ 4.4 ซึ่งมีพารามิเตอร์เป็นดังนี้ MaxDiff=0.2, Eps=0.06, QR Type 2, DCR Type 2 และ PRC Type 2 โดยผลการจัดกลุ่มด้วยพารามิเตอร์ดังกล่าวได้ผลดังรูปที่ 5.6



รูปที่ 5.6 ผลการทดลองจัดกลุ่มข้อมูลลายมือเขียนภาษาไทยตามอัลกอริธึมของ ODBSCAN

ในการจัดกลุ่มที่ทำการทดลองได้จำนวนคลัสเตอร์ทั้งหมด 27 คลัสเตอร์ SSE=20.14 และ CCE=0.4729 ซึ่งจากรูปที่ 5.6 จะพบว่าบริเวณที่เกิดการซ้อนทับกันในรูปที่ 5.5 จะถูกจัดเป็นคลัสเตอร์ได้อย่างมีประสิทธิภาพ

บทที่ 6
สรุปผลการทดลอง และข้อเสนอแนะ

6.1 สรุปผลการวิจัย

ในทดลองการจัดกลุ่มข้อมูลที่มีการกระจายแบบสม่ำเสมอ การกระจายแบบลดลงคงที่ การกระจายแบบเกาส์เซียน และการจัดกลุ่มข้อมูลลายมือเขียนภาษาไทย โดยใช้วิธีการ ODBSCAN จะให้ค่าสัมประสิทธิ์ประสิทธิภาพการจัดกลุ่ม (CCE) ต่ำ และจากผลการทดลองเปรียบเทียบวิธีการ FCM กับวิธีการ ODBSCAN ปรากฏว่าวิธีการ ODBSCAN สามารถจัดกลุ่มข้อมูลที่มีการซ้อนทับกันได้ดีกว่า

ดังนั้นวิธีการจัดกลุ่มข้อมูลที่มีการซ้อนทับกันโดยอาศัยเทคนิคความหนาแน่น จึงสามารถจัดกลุ่มข้อมูลที่มีการซ้อนทับกันที่มีการกระจายแบบต่างๆ เช่น ข้อมูลที่มีการกระจายแบบสม่ำเสมอ ข้อมูลที่มีการกระจายแบบลดลงสม่ำเสมอ ข้อมูลที่มีการกระจายแบบเกาส์เซียน รวมทั้งข้อมูลรายมือเขียนภาษาไทยได้อย่างเหมาะสม แต่ถึงอย่างไรก็ตามในการทดลองดังกล่าวข้างต้น ค่าพารามิเตอร์ต่างๆ ที่จะทำให้ได้ผลลัพธ์ที่เหมาะสมที่สุด เกิดจากการทดลองปรับเปลี่ยนค่าพารามิเตอร์ต่างๆ ได้แก่ ค่ารัศมี Eps ค่า MinPts และค่า MaxDiff โดยอาศัยผู้ทดลอง ซึ่งต้องอาศัยการทำลองสุ่มเพื่อหาค่าที่ดีที่สุด

6.2 ข้อเสนอแนะ

วิทยานิพนธ์นี้เป็นการจัดกลุ่มข้อมูลที่มีการซ้อนทับกัน โดยใช้เทคนิคความหนาแน่น ซึ่งถึงแม้ว่าจะสามารถจัดกลุ่มข้อมูลได้เหมาะสม แต่ยังคงมีบางประเด็นที่ต้องปรับปรุง ซึ่งผู้วิจัยขอเสนอแนะดังนี้

- ในการหาอัตราส่วนคลาสในวิทยานิพนธ์นี้จำเป็นต้องหาแต่ละจุดซึ่งใช้เวลามาก ดังนั้นจึงควรหาวิธีการปรับปรุงการหาค่าอัตราส่วนคลาสโดยไม่จำเป็นต้องหาทุกจุด เช่น การหาเป็นกลุ่มของจุด หรือใช้การหาโดยกำหนดขอบเขตในการคำนวณ เป็นต้น
- อีกหัวข้อที่ควรปรับปรุงเพื่อที่จะให้ได้การจัดกลุ่มที่เหมาะสม คือ การหาค่าพารามิเตอร์ ได้แก่ รัศมี Eps ค่า MinPts และ MaxDiff โดยอัตโนมัติ

เอกสารอ้างอิง

[1] Boontee Kruatrachue, Kulwarun Warunsin, Kritawan Siriboon, "The classified method for overlapping data". ICCA 2004,

[2] Easter M., Kriegel H.-P., Sander J. and Xu X. "A Density-Based Algorithm for Discovery Clusters in Large Spatial Databases with Noise," International Conference on Knowledge Discovery and Data Mining, Portland, OR, AAAI Press, 1996, pp-226-231.

[3] Andrew W Moore. "K-means and Hierarchical Clustering", http://www-2.cs.cmu.edu /~awm/ tutorials/kmeans.html.

[4] Thanawat Phattaraworamet, Boontee Kruatrachue and Kreangsak Tamee, "Prototype-based Classifier Using Density Technique", Advances in Intelligent System – Theory and Application, Luxembourg, 2004, pp.155-04.

[5] "SEQUOIA 2000", http://meteora.ucsd.edu/s2k/s2k_home.html.

[6] J. C. Bezdek "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York, Tariq Rashid: "Clustering", 1981.

[7] Beckmann N., Kriegel H.-P., Schneider R, and Seeger B. "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", Proc. ACM SIGMOD Int. Conf. on Management of Data, Atlantic City, NJ, 1990, pp. 322-331.

[8] Brinkhoff T., Kriegel H.-P., Schneider R., and Seeger B. "Efficient Multi-Step Processing of Spatial Joins", Proc. ACM SIGMOD Int. Conf. on Management of Data, Minneapolis, MN, 1994, pp. 197-208.

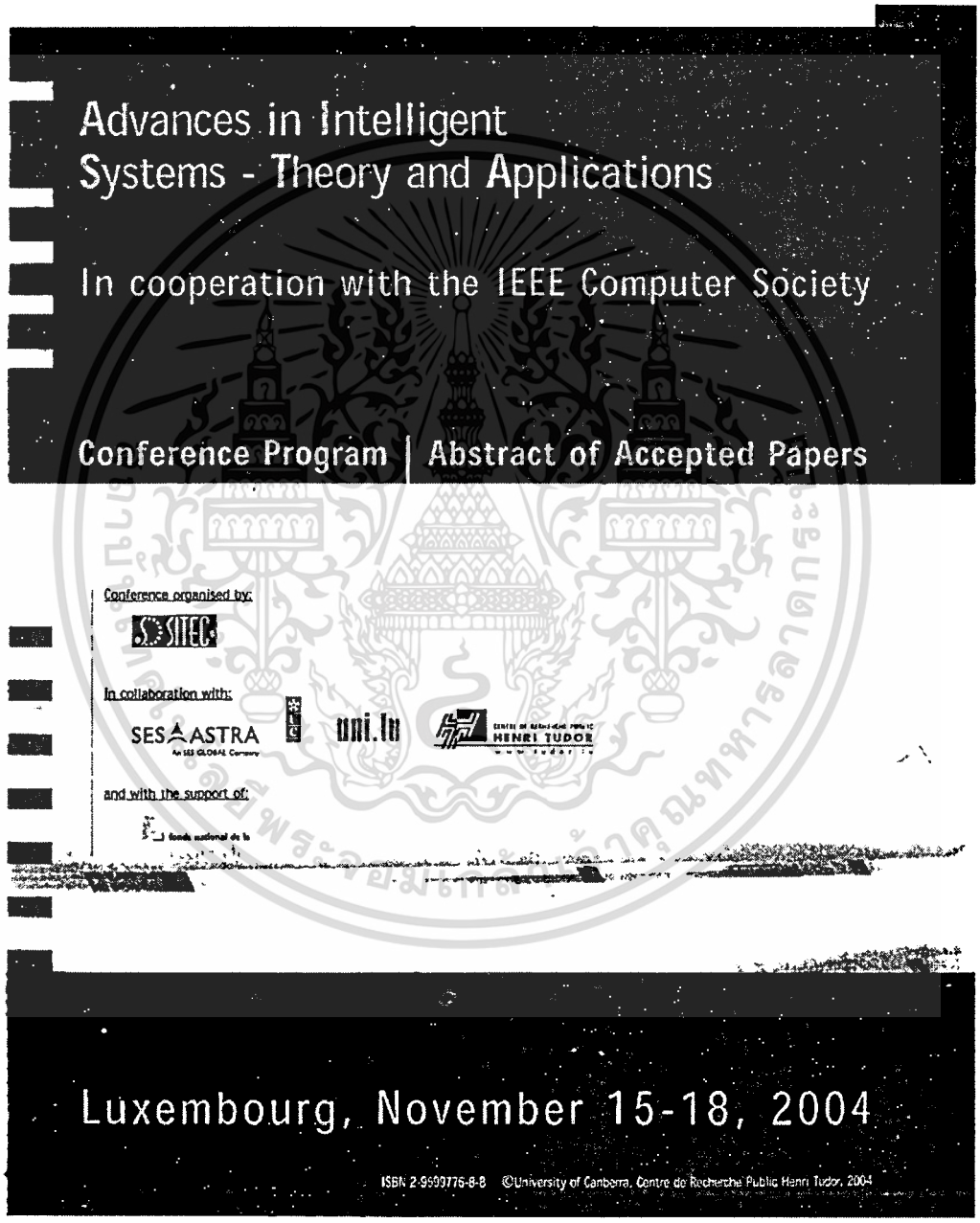
[9] J. C. Dunn "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", Journal of Cybernetics 3, 1973, pp 32-57

[10] M.A. Abou-Nasr and M.A. Sid-Ahmed, "Fast Learning and efficient memory utilization with a prototype base neural classifier". Pattern Reconition, Vol.28, No.4, pp. 581-593, 1995.

ภาคผนวก งานวิจัยที่ได้รับการตีพิมพ์

AISTA 2004

International Conference



AISTA2004 ID: 155-04

Prototype-based Classifier Using Density Technique

Thanawat PHATTARAWORAMET, Boontee KRUATRACHUE, and Kreangsak TAMEE

Abstract— **Clustering algorithms are attractive task for identification in pattern recognition. In this paper we introduce a modification of DBSCAN [1] for recognition purpose. The traditional DBSCAN is a clustering technique. However, we can adapt it for classification purpose in non-overlapping data. Thus, we use similarity of class ratio applied to traditional DBSCAN for handling this overlapping problem called O-DBSCAN. In our approach, the DBSCAN classifies data and eliminates noise. We use the similarity of class ratio to identify the overlapping data. In classification, points which are not the noise from O-DBSCAN, will be used as the prototypes.**

Index Terms — **DBSCAN, Overlap Classification, Density-based Clustering, Prototype-based classification.**

I. Introduction

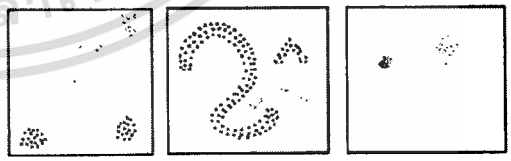
A Class is a collection of data that are similar to one another within the same class and are dissimilar to the objects in other classes. The process of grouping a set of physical or abstract objects into classes of similar objects is called classification. Classification is very close to clustering except that the classification is a supervise learning where classifier is trained with known class. Classification has wide applications including market or customer segmentation, pattern recognition, biological studies, spatial data analysis, Web document classification, and many others. Classification can be used as a standalone data mining tool to gain insight into the data distribution, or serve as a preprocessing step for other data mining algorithms operating on the detected clusters. Classification is a dynamic field of research in data mining. Many clustering algorithms have been developed. These can be categorized into hierarchical, partitioning, density-based, grid-based, and model-based methods [2].

The traditional DBSCAN is a clustering technique. However, we can adapt it for classification purpose in non-overlapping data. Thus, we use similarity of class ratio applied to traditional DBSCAN for handling this overlapping problem called O-DBSCAN. In classification, points, those are not the noise from O-DBSCAN, will be used as the prototypes.

The rest of paper is organized as follows. We discuss the previous work of clustering algorithm in section II. In section III, we describe the density based notion for overlapping classification. Section IV, we introduce our algorithm, which discovers the overlapping data. In section V, we explain the method to identify the class of unknown data. Section VI, we present a result and discussion of our method using syntactic data and the last section is the conclusion.

II. Previous Work

There are two basic types of clustering algorithms: partitioning and hierarchical algorithm. Partitioning algorithms construct a partition of a database of objects into a set of some clusters. Hierarchical algorithms create a hierarchical decomposition of database. The Hierarchical decomposition is represented by a dendrogram, a tree that iteratively splits database into smaller subsets until each subset consists of only one subject. In research of [1], they proposed clustering algorithm, called DBSCAN, for arbitrary shape. The proposed DBSCAN rely on a density based notation of clusters. It requires only one input parameter and supports the user to determine an appropriate value for it. The key idea is that for each point of a cluster the neighborhood of a given radius has to contain at least a minimum number of points. The shape of a neighborhood is determined by the choice of a distance function for two points.



database 1 **database 2** **database 3**

Fig.1 Clusters discovered by DBSCAN

There are some previous researches of classification for overlapping data such as the classification method for overlapping data [3]. This research used neural network

Manuscript received September 30, 2004.

Thanawat PHATTARAWORAMET is a master student in graduated school. He enrolls as a full time student in Department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang Bangkok, 10520 Thailand. (e-mail: phattaraw@ametit.yahoo.com).

Boontee KRUATRACHUE is Assoc. Prof. Dr in Department of Computer Engineering Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang Bangkok, 10520 Thailand. (e-mail: boontee@kmitl.ac.th)

Kreangsak TAMEE is a doctoral student in graduated school. He enrolls as a full time student in Department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang Bangkok, 10520 Thailand. (e-mail: kreangsak@hotmail.com).

classifier (NCC) and prototype to automatically classify the overlapping data. The center is used as a mean representative of training data for each class. The unclassified pattern is classified by measure distance from the class center. If the distance is in the lower (short radius), the unknown pattern has the high percentage of being in this class. If the distance is between the lower and upper (further radius), the pattern has the probability of being in this class or others. If the distance is outside the upper, the pattern is not in the class.

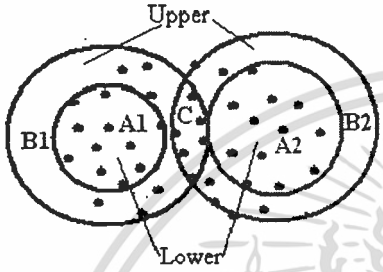


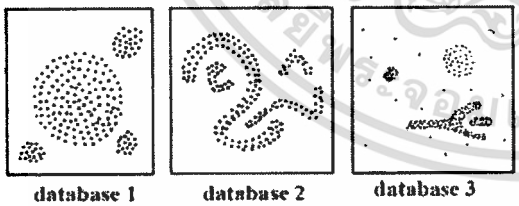
Fig. 2 Area of lower and upper in class A and class B

From fig. 2 A1 and A2 is area of class 1 and class 2. B1 is area of probability data of class 1 and others, B2 is area of probability fata of class 2 and others, C is area of data class 1 and class 2.

III. Density Based Notion for Overlapping Classification

When looking at the sample sets of points depicted in fig.3, we can easily and unambiguously detect clusters of points and noise.

The main reason why we recognize the clusters is that within each cluster we have a typical density of points which is considerably higher than outside of the cluster. Furthermore, the density within the areas of noise is lower than the density in any of the clusters.



database 1 database 2 database 3

Fig3. Sample Database

We try to formalize this intuitive notion of "clusters" and "noise" in a database D of points of some k-dimensional space S. The key idea is that for each point of a cluster the neighborhood of a given radius has to contain at least a minimum number of points, i.e. the density in the neighborhood has to exceed some threshold.

Besides the density of data, in overlap clustering, the other important parameter is the difference of class proportion in eps-neighborhood as definitions below:

Definition 1: (Eps-neighborhood of a point) The Eps-neighborhood of a point p, denoted by $N_{Eps}(p)$, is defined by $N_{Eps}(p) = \{q \in D \mid dist(p,q) = Eps\}$.

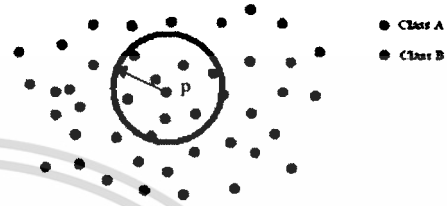


Fig. 3 Eps-neighborhood

Definition 2: Class ratio is the proportional of number of points of each class that is in Eps-neighborhood of a point p, denoted by

$CR(p) = [r_a, r_b, ..r_m]$ where $r_a = n_a/(n_a + n_b + ... + n_m)$

The parameter, Class ratio, identifies probability of point p, being the member of each class. n_a is the number of neighborhood points of point p, belonging to class a. In fig.3 the class ratio of point p is CR(p)=[4/9,5,9].

Definition 3: Dissimilarity of class ratio is the difference of class ratio between point p and q, denoted by

$DCR(p,q) = |CR(p) - CR(q)|$, where

$$|CR(p) - CR(q)| = \sum_{i=a}^{m} |r_i^p - r_i^q|$$

Dissimilarity of class ratio is the summation of the difference of class ratio between two neighborhood points. If this value is low, this means these two neighborhood points have a high probability being the same class.

Definition 4: Eps-neighborhood with similar class ratio is Eps-neighborhood of point p which $DCR(p,q) \le MaxDiff$, denoted by

$NC_{Eps}(p) = \{q \in D \mid dist(p,q) = Eps, DCR(p,q) \le MaxDiff\}$

Definition 5: (directly density-reachable) A point p is directly density-reachable from a point q with respect to Eps, MinPts if

1) $p \in NC_{Eps}(q)$ and

2) $|NC_{Eps}(q)| \ge MinPts$ (Core point condition)

Definition 6: (density-reachable) A point p is density-reachable from a point q with respect to Eps and MinPts if there is a chain of points $p_1, ..., p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i.

Definition 7: (density-connected) A point p is density-connected to a point q with respect to Eps and MinPts if there is a point o such that both, p and q are density-reachable from o with respect to Eps and MinPts.

Definition 8: (cluster) Let D be a database of points. A cluster C with respect to Eps and MinPts is a non-empty subset of D satisfying the following conditions.

1) $\forall p,q : if\ p \in C$ and q is density-reachable from p with respect to Eps and MinPts, then $q \in C$. (Maximality)

2) $\forall p, q \in C$: p is density-connected to q with respect to EPS and MinPts. (Connectivity)

Definition 9: (noise) Let $C_1, \ldots, C_k$ be the clusters of the database D with respect to parameters Eps_i and $MinPts_i$, i = 1,....., k. Then we define the noise as the set of points in the database D not belonging to any cluster Ci, i.e. $noise = \{p \in D \mid \forall i : p \notin C_i\}$.

IV. O-DBSCAN: OVERLAPPING-DBSCAN

To find a cluster, O-DBSCAN starts with an arbitrary point p and retrieves all points density-reachable from p with respect to Eps and MinPts. If p is a core point, this procedure yields a cluster with respect to Eps and MinPts. If p is a border point, no points are density-reachable from p and O-DBSCAN visits the next point of the database. In the following, we present a basic version of O-DBSCAN omitting details of data types and generation of additional information about clusters. In the following, we present the algorithm of O-DBSCAN:

```
O_DBSCAN (SetOfPoints, Eps, MinPts,MaxDiff)
  // SetOfPoints is UNCLASSIFIED
  ClusterId := nextId(NOISE);
  FOR i FROM 1 TO SetOfPoints.size DO
    Point := SetOfPoints.get(i);
    IF Point.ClId = UNCLASSIFIED THEN
      IF ExpandCluster(SetOfPoints, Point,
      ClusterId, Eps, MinPts, MaxDiff) THEN
        ClusterId := nextId(ClusterId)
      END IF
    END IF
  END FOR
END; // O_DBSCAN
```

SetOfPoints is the whole database. Eps and MinPts are global density parameters. MaxDiff is a global dissimilarity value to limit that those two points can be neighborhood for each other or not. Function ExpandCluster play very important role to emerge points in the same cluster which is presented below:

```
ExpandCluster(SetOfPoints, Point, ClId, Eps,
MinPts, MaxDiff) : Boolean;
  seeds:=SetOfPoints.regionQuery(Point,Eps);
  IF seeds.size<MinPts THEN // no core point
    SetOfPoint.changeClId(Point,NOISE);
    RETURN False;
  ELSE // all points in seeds are density-
    // reachable from Point
    SetOfPoints.changeClIds(seeds,ClId);
    seeds.delete(Point);
    WHILE seeds <> Empty DO
      currentP := seeds.first();
      result := SetOfPoints.regionQuery(currentP,
      Eps);
      IF result.size >= MinPts THEN
        FOR i FROM 1 TO result.size DO
          resultP := result.get(i);
          IF resultP.ClId
          IN (UNCLASSIFIED, NOISE) THEN
            IF resultP.ClId = UNCLASSIFIED THEN
              seeds.append(resultP);
            END IF;
          SetOfPoints.changeClId(resultP,ClId);
          END IF; // UNCLASSIFIED or NOISE
        END FOR;
      END IF; // result.size >= MinPts
      seeds.delete(currentP);
    END WHILE; // seeds <> Empty
    RETURN True;
  END IF
END; // ExpandCluster
```

SetOfPoints.regionQuery returns the Eps-neighbourhood with similar class ratio of point P in SetOfPoints as a list of points. Before returning from SetOfPoints.regionQuery (spatial function), this function has to check the dissimilarity of class ratio of each point. If any point has dissimilarity of class ratio more than MaxDiff value, this point is not included into the returned list. Dissimilarity has been shown as below:

```
Dissimilarity(Point1, Point2, MaxDiff) : Boolean;
  Diff=0
  FOR i FROM CLASS_A TO CLASS_M DO
    Diff := Diff + abs(Point1.getClassCount(i)-
      Point2.getClassCount(i));
  END FOR
  IF Diff>MaxDiff THEN
    RETURN False;
  ELSE
    RETURN True;
  END IF
END; // Dissimilarity
```

V. IDENTIFICATION METHOD

We propose the method to identify which class the unknown data belong to, as a detail shown in this section.

Trained points are used as prototypes to indicate unknown data which identification method uses the Eps-neighborhood of prototypes to identify the class for unknown data. The percentage of the probability to be that class has been shown as the identification value. This percentage can be calculated as:

$$P(a) = \frac{\sum_{i \in N_{eps}(P)} r_a^i}{|N_{eps}(P)|}$$

, where $P(a)$ is the percentage to be *class a* called class-percentage of a.

Identification value is represented as a list of class-percentages, $[P(a), P(b), \ldots, P(m)]$. For example, assume that there are 3 classes A, B, and C in class domain. If a list is

25%, 75%, 0%] , it means that this unknown point has a robability belonging to class A equal to 25%, class B 75% nd class C 0%.

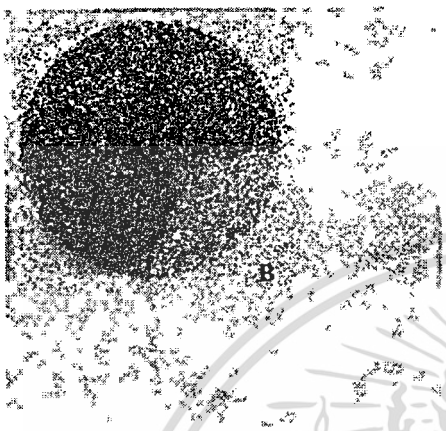


Fig. 4 Un-clustering data

VI. Result and Discussion

In our experiment, we used the syntactic data of two classes with 2516 overlapping data in 11670 points as shown in fig.4. There are also some noises in this data. Our algorithm can give a good performance to handle overlapping data. For 11670 points, there are 24 clusters with 2 non-overlapping clusters, cluster 1st and 2nd. Cluster 3rd to 24th are overlapping clusters. This algorithm can detect 301 noise points as shown in fig.5.

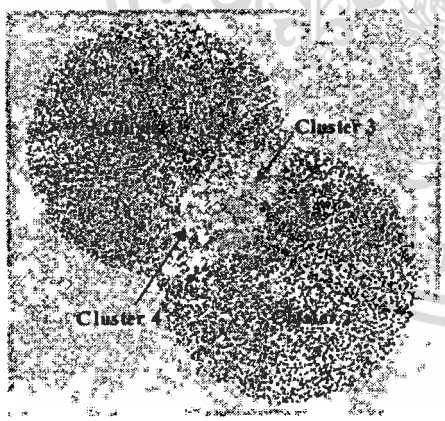


Fig. 5 Clustered data.

VII. Conclusion

Our research proposes a prototype based classifiers and algorithm for overlapping data. Our algorithm is an extension of DBSCAN [1]. We apply the dissimilarity of class ratio for clustering data. Class-percentage is used in identification method to identify class for unknown point. Our algorithm in this research gives satisfying results.

Acknowledgment

The authors would like to thank our colleague, Pongkasem Polsuntikul, to audit this document during our writing of this paper and AISTA chair man and committees for organizing this conference.

References

[1] Easter M., Kriegel H.-P., Sander J., Xu X. "A Density-Based Algorithm for Discovery Clusters in Large Spatial Databases with Noise," International Conference on Knowledge Discovery and Data Mining, Portland, OR, AAAI Press, 1996, pp-226-231.

[2] Jiawei Han and Micheline Kamber, Data Mining Concepts and Techniques, Morgan Kaufmann, pp. 335-393, 2001.

[3] Boonlee Kruatrachue, Kulwaran Warunsin, Kritawan Siriboon, "The classified method for overlapping data", ICCA 2004.

[4] M.A. Abou-Nasr and M.A. Sid-Ahmed, "Fast Learning and efficient memory utilization with a prototype base neural classifier" Pattern Recognition, Vol.28, No.4, pp. 581-593, 1995.

ประวัติผู้เขียน

ชื่อ-นามสกุล นายธนวัฒน์ ภัทรวรเมธ

วันเดือนปีเกิด วันที่ 19 กันยายน 2518 ที่จังหวัดหนองคาย

ที่อยู่ 137/3 ถนนผดุงสามัคคี ต.ท่าบ่อ อ.ท่าบ่อ จ.หนองคาย

ประวัติการศึกษา

พ.ศ. 2531 จบการศึกษาระดับประถมศึกษา จากโรงเรียนโกมลวิทยาคาร อ.ท่าบ่อ จ.หนองคาย

พ.ศ. 2534 จบการศึกษาระดับมัธยมศึกษาตอนต้น จากโรงเรียนท่าบ่อวิทยาคม อ.ท่าบ่อ จ.หนองคาย

พ.ศ. 2537 จบการศึกษาระดับมัธยมศึกษาตอนปลาย จากโรงเรียนท่าบ่อวิทยาคม อ.ท่าบ่อ จ.หนองคาย

พ.ศ. 2542 จบการศึกษาวิศวกรรมศาสตรบัณฑิต (วิศวกรรมคอมพิวเตอร์) จากคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ อ.เมือง จ.เชียงใหม่

ประสบการณ์ทำงาน

พ.ศ. 2542-2544 ลูกจ้างชั่วคราว ตำแหน่งอาจารย์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่