

วิธีการลดข้อมูลสำหรับการเรียนรู้ทำงานประสาทเทียม



นาย ไพริชน์ นิตยานุภาพ

สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2549

ลิขสิทธิ์ของ จุฬาลงกรณ์มหาวิทยาลัย

A DATA REDUCTION METHOD FOR ARTIFICIAL NEURAL NETWORK LEARNING



Mr. Pairit Nittayanuparp

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Computer Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2006

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์ วิธีการลดข้อมูลสำหรับการเรียนรู้ถ่ายงานประสาทเทียม
โดย นายไพรัช นิตยานุภาพ
สาขาวิชา วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา รองศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วน
หนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

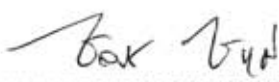

..... คณะบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์


..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ญาใจ ลิมปิยะกรณ์)


..... อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล)


..... กรรมการ
(อาจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์)


..... กรรมการ
(อาจารย์ ดร.ชลวิษ นัทธิ)

ไพริชน์ นิตยานุภาพ : วิธีการลดข้อมูลสำหรับการเรียนรู้ข่ายงานประสาทเทียม. (A DATA REDUCTION METHOD FOR ARTIFICIAL NEURAL NETWORK LEARNING)
 อ.ที่ปรึกษา : รศ.ดร. บุญเสริม กิจศิริกุล, 51 หน้า.

เป็นที่ยอมรับโดยทั่วไปว่าข่ายงานประสาทเทียมเป็นเครื่องมือจำแนกประเภทข้อมูลที่มีความถูกต้องยอดเยี่ยมมาก และสามารถนำไปประยุกต์ใช้ได้ดีในสาขาวิชาต่างๆ แต่ในกรณีที่ใช้ข้อมูลฝึกในปริมาณมากเกินไป พบว่าโปรแกรมฝึกข่ายงานประสาทบางตัวไม่สามารถเรียนรู้ข้อมูลได้ เนื่องจากโปรแกรมเกิดการล้มเหลวขึ้นเสียก่อน และถึงแม้ว่าจะเรียนรู้ข้อมูลได้สำเร็จ ก็ยังต้องใช้เวลาเรียนรู้ข้อมูลนานมาก วิทยานิพนธ์ฉบับนี้เสนอวิธีการลดข้อมูลแบบสุ่มตามความหนาแน่น โดยใช้แนวคิดการลดข้อมูลที่ซ้ำซ้อน และวิธีการลดข้อมูลแบบระเบิดลูกโซ่ที่เจาะจงเลือกเก็บเฉพาะข้อมูลตามขอบของข้อมูลแต่ละประเภท ซึ่งผลที่ได้จากการลดข้อมูลนี้ ไม่เพียงแต่จะช่วยลดโอกาสที่โปรแกรมจะเกิดการล้มเหลวในขั้นตอนการฝึกเท่านั้น แต่ยังช่วยลดเวลาในการฝึกให้น้อยลงอีกด้วย นอกจากนี้ผู้วิจัยยังได้นำอัลกอริทึมสำหรับลดจำนวนคุณสมบัติข้อมูลที่มีชื่อว่า อัลกอริทึม Relief มาช่วยในการลดข้อมูล ทำให้การลดข้อมูลแบบระเบิดลูกโซ่สามารถลดข้อมูลได้มากขึ้น ผลการทดลองแสดงให้เห็นว่าข้อมูลที่ลดด้วยวิธีที่เสนอนี้ ให้ความถูกต้องในการจำแนกประเภทโดยเฉลี่ยดีกว่าการลดด้วยวิธีการสุ่มแบบทั่วไป

สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมคอมพิวเตอร์ลายมือชื่อนิสิต..... ๗
 สาขาวิชา วิศวกรรมคอมพิวเตอร์ลายมือชื่ออาจารย์ที่ปรึกษา..... ๘/๗/๗
 ปีการศึกษา..... 2549.....

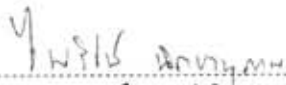

4770395121 : MAJOR COMPUTER ENGINEERING

KEYWORD : ARTIFICIAL NEURAL NETWORK / DATA REDUCTION

PAIRIT NITTAYANUPARP : A DATA REDUCTION METHOD FOR ARTIFICIAL
NEURAL NETWORK LEARNING. THESIS ADVISOR : ASSOC. PROF.
BOONSERM KIJSIRIKUL, Ph.D. 51 pp.

It has been known that the artificial neural network is a very accurate classifier which is practically used in various fields. Unfortunately, in case of a huge training dataset, we found that some neural network software crashed in the learning process. Even though the software succeeds in learning, it may take a very long time in the learning process. In this paper, we propose a data reduction method, called density-based random data reduction, which uses the concept of competence preservation. We also propose a data reduction method, called Chain-Explosion (CE) data reduction, which keeps only surrounding data of each class. Moreover, by using ReliefF which is an algorithm for dimension reduction, CE data reduction is further enhanced and is able to reduce more data. The experimental results show that the data reduced by the proposed methods give the average classification accuracy better than the data reduced by random reduction.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Department... Computer Engineering... Student's signature... 
Field of study... Computer Engineering... Advisor's signature... 
Academic year... 2006.....

กิตติกรรมประกาศ

การวิจัยและวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ ก็ด้วยความช่วยเหลืออย่างดียิ่งของรองศาสตราจารย์ ดร.บุญเสริม กิจศิริกุล อาจารย์ที่ปรึกษาที่ให้คำปรึกษาและคำแนะนำที่ดีๆ เสมอมา รวมไปถึงคณะกรรมการคุมสอบวิทยานิพนธ์ทุกท่านที่ให้ความเมตตา แนะนำแนวคิดที่เป็นประโยชน์อย่างยิ่งต่อการพัฒนางานวิจัยชิ้นนี้

นอกจากนี้บุคคลที่จะลืมเสียมิได้คือพี่ๆ น้องๆ ทุกคนในห้องปฏิบัติการอัจฉริยภาพเครื่องจักรและการค้นพบความรู้ “ແ້ບມາດ” เพื่อนๆ พี่ๆ ใน “ແ້ບหุ่ยนนต์” และผู้อื่นที่มีอาจกล่าวถึงได้หมด ที่ให้คำแนะนำอันเปรียบเสมือนแสงประกายในยามมืดมิด ซึ่งช่วยให้ผู้วิจัยฝ่าทางตันไปได้ทุกครั้งที่เกิดปัญหาใดๆ

สุดท้ายนี้ผู้วิจัยต้องขอกราบสำนึกในพระคุณของคุณพ่อ คุณแม่ และสมาชิกในครอบครัวที่เฝ้าอบรมเลี้ยงดู และให้กำลังใจเสมอมา



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ญ
สารบัญภาพ.....	ฎ
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 ขอบเขตของการวิจัย.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.5 ขั้นตอนและวิธีดำเนินการวิจัย.....	3
1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์.....	3
1.7 ผลงานตีพิมพ์จากงานวิจัย.....	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	4
2.1 ข่ายงานประสาทเทียม.....	4
2.1.1 เซลล์ประสาท (Neuron).....	5
2.1.2 การเรียนรู้ของข่ายงานประสาทเทียม (Neural Network Learning).....	6
2.1.3 เพอร์เซปตรอน (Perceptrons).....	7
2.1.4 การเรียนรู้ของเพอร์เซปตรอน (Perceptron Learning).....	9
2.1.5 การฝึกเพอร์เซปตรอนด้วยกฎเดลต้า (Delta Rule).....	10
2.1.6 ข่ายงานประสาทเทียมหลายชั้น (Multi Layer Perceptron).....	10
2.1.7 การเรียนรู้ด้วยวิธีการแพร่ย้อนกลับ (Back propagation).....	11
2.2 k -เนียร์เรสเนเบอร์ (k -Nearest Neighbor – k NN).....	12
2.3 งานวิจัยที่เกี่ยวข้อง.....	13
2.3.1 งานวิจัยเกี่ยวกับการเลือกคุณสมบัติของข้อมูล (Feature Selection).....	13
2.3.2 งานวิจัยเกี่ยวกับการลดตัวข้อมูล.....	15
บทที่ 3 วิธีการลดข้อมูล.....	16
3.1 การลดข้อมูลแบบสุ่มตามความหนาแน่น (Density-Based Random Data Reduction).....	16

	3.1.1	แรงจูงใจ.....	16
	3.1.2	การลดข้อมูลแบบสุ่มตามความหนาแน่น.....	16
	3.2	การลดข้อมูลแบบระเบิดลูกโซ่ (Chain-Explosion Data Reduction – CE Data Reduction).....	18
	3.2.1	แรงจูงใจ.....	18
	3.2.2	การลดข้อมูลแบบง่าย.....	19
	3.2.3	การลดข้อมูลแบบ CE	20
	3.2.4	ทดสอบวิธีการลดข้อมูลแบบ CE บนข้อมูลทดสอบ P2.....	23
	3.3	การลดข้อมูลแบบระเบิดลูกโซ่ขั้นสูง (Advanced Chain-Explosion Data Reduction – ACE Data Reduction).....	25
	3.4	การใช้อัลกอริทึม ReliefF ช่วยในการลดข้อมูลแบบ CE	27
	3.4.1	แรงจูงใจ.....	27
บทที่ 4		การทดลอง	29
	4.1	ข้อมูลที่ใช้ในการทดลอง	29
	4.2	การจัดเตรียมข้อมูลเพื่อฝึกข่ายงานประสาทเทียม.....	29
	4.2.1	การลดแบบสุ่ม	29
	4.2.2	การลดแบบสุ่มตามความหนาแน่น	30
	4.2.3	การลดแบบ CE	31
	4.2.4	การลดแบบ CE ที่ลดคุณสมบัติด้วยวิธี ReliefF.....	31
	4.3	การฝึกข่ายงานประสาท.....	32
	4.4	การเปรียบเทียบประสิทธิภาพของการลดข้อมูล	33
	4.4.1	การลดแบบสุ่ม และการลดแบบสุ่มตามความหนาแน่น.....	33
	4.4.2	การลดแบบ CE และการลดแบบ CE ที่ลดคุณสมบัติด้วย ReliefF	34
บทที่ 5		ผลการทดลอง.....	35
	5.1	ประสิทธิภาพของการลดข้อมูลแบบสุ่ม แบบสุ่มตามความหนาแน่น และแบบ CE.....	35
	5.2	ความเร็วในการลดข้อมูลแบบ CE และ ACE.....	40
	5.3	เปรียบเทียบผลการทดลองลดข้อมูลแบบ CE กับแบบ CE ที่ลดคุณสมบัติด้วย ReliefF.....	41
บทที่ 6		สรุปผลการวิจัยและข้อเสนอแนะ.....	47
	6.1	สรุปผลการวิจัย.....	47
	6.2	ข้อเสนอแนะ.....	48

รายการอ้างอิง.....49

ประวัติผู้เขียนวิทยานิพนธ์.....51



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

ตารางที่ 1	ความถูกต้องในการจำแนกประเภทชุดข้อมูล P2	23
ตารางที่ 2	ชุดข้อมูลที่ใช้ในการทดลอง	29
ตารางที่ 3	จำนวนคุณสมบัติที่เลือกเก็บไว้ของชุดข้อมูลต่างๆ	32
ตารางที่ 4	ค่าอัตราการเรียนรู้ และค่าโมเมนตัมที่ใช้ในการฝึก	32
ตารางที่ 5	เปรียบเทียบเวลาในการลดข้อผิดพลาดระหว่าง CE กับ ACE (วินาที).....	41



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

รูปที่ 1 แผนภูมิแสดงเซลล์ประสาทของสิ่งมีชีวิต.....5

รูปที่ 2 เพอร์เซปตรอนหนึ่งหน่วย.....8

รูปที่ 3 พื้นผิวตัดสินใจที่ใช้แบ่งแยกข้อมูล.....8

รูปที่ 4 ฟังก์ชัน AND, แบ่งแยกได้แบบเชิงเส้น.....9

รูปที่ 5 ฟังก์ชัน XOR, แบ่งแยกไม่ได้แบบเชิงเส้น.....9

รูปที่ 6 โครงสร้างของข่ายงานประสาทเทียมหลายชั้น.....11

รูปที่ 7 ฟังก์ชันซิกมอยด์.....11

รูปที่ 8 รหัสเทียมอัลกอริทึม Relief.....14

รูปที่ 9 รหัสเทียมอัลกอริทึม ReliefF.....15

รูปที่ 10 รหัสเทียมอัลกอริทึมลดข้อมูลตามความหนาแน่น.....17

รูปที่ 11 พื้นผิวตัดสินใจที่ฝึกโดยใช้ (a) ข้อมูลฝึกทั้งหมด, (b) ข้อมูลฝึกที่อยู่บริเวณขอบ.....18

รูปที่ 12 ข้อมูลที่อยู่บนขอบข้างระหว่างแนวที่พื้นผิวตัดสินใจพาดผ่าน.....18

รูปที่ 13 รหัสเทียมอัลกอริทึมลดข้อมูลแบบง่าย.....19

รูปที่ 14 ข้อมูลในกลุ่ม T_j ทุกตัวเลือกเก็บข้อมูลในกลุ่ม T_i20

รูปที่ 15 ข้อมูลบางกลุ่มย่อยของ T_i อยู่ห่างเกินไป ทำให้ถูกละเลย.....20

รูปที่ 16 ข้อมูล y เลือกข้อมูลที่ใกล้ที่สุดจาก $P(T_i)$ แล้วระเบิดข้อมูลที่อยู่ภายใต้รัศมี r ที่.....21

รูปที่ 17 ข้อมูล y เลือกข้อมูลที่ใกล้ที่สุดจาก $P(T_i)$ ตัวถัดไป แต่ไม่มีข้อมูลตัวอื่นอยู่ภายใต้รัศมี จึงตัดข้อมูลตัวใดไม่ได้เลยในรอบนี้.....21

รูปที่ 18 ข้อมูลสุดท้ายที่เหลือหลังจากลดข้อมูลแบบ CE.....21

รูปที่ 19 รหัสเทียมอัลกอริทึมลดข้อมูลแบบ CE.....22

รูปที่ 20 (a) เส้นกราฟที่ใช้แบ่งพื้นที่เพื่อสร้าง P_2 , (b) ลักษณะของชุดข้อมูล P_2 โดยมี 0 และ + เป็นสัญลักษณ์แทนข้อมูลในแต่ละประเภท, (c1-c3) พื้นผิวตัดสินใจที่ฝึกด้วยข้อมูลที่ลดด้วยวิธีสุ่ม, (d1-d3) พื้นผิวตัดสินใจที่ฝึกด้วยข้อมูลที่ลดด้วยวิธี CE.....24

รูปที่ 21 r -Clusters ของ T_i25

รูปที่ 22 รหัสเทียมอัลกอริทึมลดข้อมูลแบบ ACE.....26

รูปที่ 23 ข้อมูลที่มีคุณสมบัติ a_1 และ a_227

รูปที่ 24 ข้อมูลที่ตัดคุณสมบัติ a_2 ทิ้งแล้ว.....28

รูปที่ 25 ขั้นตอนการจัดเตรียมข้อมูลที่ลดด้วยวิธีสุ่ม และสุ่มตามความหนาแน่น.....30

รูปที่ 41 กราฟความถูกต้องในการจำแนกประเภทข้อมูลที่ลดด้วยการลดข้อมูลแบบ CE และแบบ CE ที่ลดคุณสมบัติด้วย ReliefF ของชุดข้อมูล new-thyroid ณ อัตราการลดข้อมูลต่างๆ43

รูปที่ 42 กราฟความถูกต้องในการจำแนกประเภทข้อมูลที่ลดด้วยการลดข้อมูลแบบ CE และแบบ CE ที่ลดคุณสมบัติด้วย ReliefF ของชุดข้อมูล ionosphere ณ อัตราการลดข้อมูลต่างๆ44

รูปที่ 43 กราฟความถูกต้องในการจำแนกประเภทข้อมูลที่ลดด้วยการลดข้อมูลแบบ CE และแบบ CE ที่ลดคุณสมบัติด้วย ReliefF ของชุดข้อมูล kr-vs-kp ณ อัตราการลดข้อมูลต่างๆ44

รูปที่ 44 กราฟความถูกต้องในการจำแนกประเภทข้อมูลที่ลดด้วยการลดข้อมูลแบบ CE และแบบ CE ที่ลดคุณสมบัติด้วย ReliefF ของชุดข้อมูล vehicle ณ อัตราการลดข้อมูลต่างๆ45

รูปที่ 45 กราฟความถูกต้องในการจำแนกประเภทข้อมูลที่ลดด้วยการลดข้อมูลแบบ CE และแบบ CE ที่ลดคุณสมบัติด้วย ReliefF ของชุดข้อมูล vowel ณ อัตราการลดข้อมูลต่างๆ45

รูปที่ 46 กราฟความถูกต้องในการจำแนกประเภทข้อมูลที่ลดด้วยการลดข้อมูลแบบ CE และแบบ CE ที่ลดคุณสมบัติด้วย ReliefF ของชุดข้อมูล segmentation ณ อัตราการลดข้อมูลต่างๆ.....46

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

เครื่องมือสำคัญตัวหนึ่งของศาสตร์ทางการเรียนรู้ของเครื่อง (machine learning) ที่ใช้ในการจำแนกประเภท (classification) ซึ่งได้รับความนิยมอย่างมากในปัจจุบัน คือข่ายงานประสาทเทียม (ANN – Artificial Neural Networks) [1] [2] [3] ปัจจุบันมีนักวิจัยมากมายได้นำข่ายงานประสาทเทียมไปประยุกต์ใช้ให้เข้ากับสายงานของตน ประสิทธิภาพที่ได้รับจากข่ายงานประสาทเทียมนั้นค่อนข้างดีมาก แต่ก่อนที่จะนำข่ายงานประสาทไปใช้งานได้นั้น ขั้นแรกจะต้องนำข้อมูลเบื้องต้นจำนวนหนึ่งที่เรียกว่าข้อมูลฝึก (training data) ป้อนให้ข่ายงานประสาทเรียนรู้ เพื่อสร้างพื้นผิวตัดสินใจ (decision surface) สำหรับไปใช้งานต่อไป แต่ทว่าในยุคที่สื่อบันทึกข้อมูลขนาดหลายร้อยหลายพันกิกะไบต์สามารถหาซื้อได้โดยง่าย ทำให้การเก็บข้อมูลจำนวนมากทำได้ง่ายแทบไม่มีขีดจำกัด เมื่อนำข้อมูลจำนวนมหาศาลที่มีอยู่ทั้งหมดมาฝึกข่ายงานประสาท โดยหวังว่าจะได้ข่ายงานประสาทที่มีความถูกต้องสูงในการจำแนกประเภทข้อมูล ก็กลับพบปัญหาบางอย่างตามมา

ปัญหาประการแรกคือปัญหาระบบล่มในระหว่างการฝึก โดยเราพบในการทดลองฝึกข่ายงานประสาทสำหรับทำกรู้อักขระโอซีอาร์ (Optical Character Recognition – OCR) ซึ่งมีข้อมูลที่ต้องเรียนรู้ 200,000 ตัว ข้อมูลแต่ละตัวมีคุณสมบัติ 250 อย่าง (attribute) และมีประเภทข้อมูล 178 ประเภท (class) คุณสมบัติแต่ละอย่างเก็บด้วยเลขทศนิยม ต้องใช้เนื้อที่ 6 ไบต์ ต่อ 1 คุณสมบัติ ดังนั้นจะต้องใช้เนื้อที่เก็บข้อมูลในหน่วยความจำ อย่างน้อยๆ $(250+178)*200,000*6 \text{ byte} = 490\text{MB}$ ด้วยปริมาณข้อมูลมากมายเช่นนี้ พบว่าโปรแกรมข่ายงานประสาทเทียมหลายๆ ตัวไม่สามารถทำงานกับชุดข้อมูลนี้ได้ เนื่องจากจะเกิดปัญหาระบบล่มเสียก่อน ซึ่งการที่ระบบล่มในระหว่างการฝึกข่ายงานประสาทนี้ น่าจะเกิดจากความไม่มีเสถียรภาพของระบบปฏิบัติการในการจัดการหน่วยความจำเสมือน¹ (virtual memory) การจะแก้ปัญหานี้ต้องไปแก้ที่ระบบปฏิบัติการ ซึ่งผู้ใช้งานไม่สามารถแก้ไขได้โดยง่าย หรืออาจเกิดจากความไม่มีเสถียรภาพของตัวโปรแกรมข่ายงานประสาทเทียมเอง นอกจากนี้ผลกระทบอื่นๆ ที่ตามมาก็คือ ประสิทธิภาพของเครื่องจะลดลงอย่างมาก เนื่องจากความเร็วของระบบโดยรวม ถูกความเร็วของฮาร์ดดิสก์ซึ่งช้ากว่าหน่วยความจำหลัก

¹ หน่วยความจำบนฮาร์ดดิสก์ที่ใช้ทดแทนหน่วยความจำหลักของเครื่อง ในกรณีที่หน่วยความจำหลักไม่เพียงพอต่อการใช้งาน โดยเมื่อใดที่หน่วยความจำหลักไม่เพียงพอ ระบบก็จะนำข้อมูลในหน่วยความจำหลักที่ยังไม่รับใช้งาน ไปพักเก็บไว้บนฮาร์ดดิสก์ก่อน จนเมื่อใดที่ต้องนำกลับมาใช้งาน ก็ค่อยดึงกลับขึ้นมาเมื่อนั้น

เป็นพันๆ เท่า เป็นขีดจำกัดที่กรอบไว้ นี่เป็นปัญหาของระบบคอมพิวเตอร์ในปัจจุบัน

ปัญหาประการที่สองคือความล่าช้าในการฝึกข้อมูล ถึงแม้การนำข้อมูลฝึกจำนวนมากมามหาศาลไปฝึกข่ายงานประสาทจะไม่เกิดปัญหาระบบล่ม แต่ก็ยังเกิดปัญหาอื่นตามมา นั่นก็คือต้องใช้เวลาในการฝึกข่ายงานประสาทนานมาก กว่าที่ฝึกจะสิ้นสุดลง จนทำให้เกิดคำถามขึ้นในใจว่า “จำเป็นหรือไม่ ที่จะต้องใช้ข้อมูลที่มีอยู่ทุกตัว มาฝึกข่ายงานประสาท?” หรืออาจจะถามให้ชัดลงไปว่า “มีข้อมูลที่เกินความจำเป็นในการฝึกหรือไม่?” ซึ่งถ้าหากมีข้อมูลที่เกินความจำเป็นอยู่จริง คำถามที่งานวิจัยนี้จะตอบก็คือ “ข้อมูลตัวใดที่เกินความจำเป็น และสามารถตัดทิ้งได้?”

การจะตัดข้อมูลทิ้ง หรือที่เรียกว่าการลดข้อมูล (data reduction) อาจจะทำได้ง่ายๆ ด้วยการสุ่มเลือกข้อมูลออกมาตามสัดส่วนที่ต้องการ ซึ่งเราจะเรียกวิธีนี้ว่าวิธีลดข้อมูลแบบสุ่ม (random data reduction) แต่ถ้าพิจารณาพฤติกรรมการเรียนรู้ข้อมูลของข่ายงานประสาท และนำความรู้จากการพิจารณานี้มาใช้ ก็น่าจะมีวิธีลดข้อมูลที่ดีกว่าการสุ่มเช่นนี้

1.2 วัตถุประสงค์ของการวิจัย

เพื่อหาวิธีลดจำนวนข้อมูลที่ใช้ในการเรียนรู้ข่ายงานประสาทเทียม โดยให้ความถูกต้องในการจำแนกประเภทข้อมูลไม่น้อยกว่าการลดข้อมูลแบบสุ่ม

1.3 ขอบเขตของการวิจัย

1. เนื่องจากเครื่องมือที่ใช้จำแนกประเภทข้อมูลมีมากมายให้เลือกใช้ ซึ่งแต่ละตัวก็มีวิธีการเรียนรู้ข้อมูล วิธีการจำแนกข้อมูลแตกต่างกัน การจะลดจำนวนข้อมูลที่ใช้เรียนรู้นั้น ก็มีความแตกต่างกันตามแต่พฤติกรรมการเรียนรู้ข้อมูลของเครื่องมือแต่ละตัว งานวิจัยจะมุ่งศึกษาเฉพาะวิธีการลดข้อมูลที่ใช้เรียนรู้ของข่ายงานประสาทเทียมเท่านั้น
2. เกณฑ์ที่ใช้วัดประสิทธิภาพคือความถูกต้องในการจำแนกประเภทข้อมูล เปรียบเทียบระหว่างวิธีลดข้อมูลที่นำเสนอ และวิธีลดข้อมูลแบบสุ่ม
3. การทดลองจะใช้ข้อมูลจาก UCI Repository of Machine Learning Database [4] อย่างน้อยจำนวน 8 ชุดข้อมูล และใช้วิธีการตรวจสอบไขว้ 5 พับ (5-fold cross validation)

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้วิธีลดข้อมูลฝึกทำงานประสาทที่ให้ความแม่นยำในการจำแนกประเภทข้อมูล สูงกว่าการลดข้อมูลแบบสุ่ม
2. เข้าใจพฤติกรรมของทำงานประสาทเทียมมากขึ้น

1.5 ขั้นตอนและวิธีดำเนินการวิจัย

1. ศึกษาการทำงานของทำงานประสาทเทียม และทฤษฎีต่างๆ ที่ต้องใช้ในงานวิจัย
2. ทดสอบความเป็นไปได้ของแนวคิดเบื้องต้น
3. ศึกษารูปแบบ และเลือกฐานข้อมูลเพื่อใช้สำหรับนำมาฝึกและทดสอบ
4. ทำการทดลองกับข้อมูลจริง

1.6 ลำดับการจัดเรียงเนื้อหาในวิทยานิพนธ์

วิทยานิพนธ์นี้แบ่งเนื้อหาออกเป็น 8 บทดังนี้ บทที่ 1 เป็นบทนำซึ่งกล่าวถึงที่มาและความสำคัญของปัญหา รวมทั้งวัตถุประสงค์ของงานวิจัยชิ้นนี้ บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้องในงานวิจัยนี้ บทที่ 3 กล่าวถึงแนวคิดและรายละเอียดการลดข้อมูลแบบสุ่มตามความหนาแน่น บทที่ 4 กล่าวถึงแนวคิดการลดข้อมูลแบบระเบิดลูกโซ่ (Chain-Explosion Data Reduction – CE) บทที่ 5 กล่าวถึงแนวคิดการใช้อัลกอริทึม ReliefF มาช่วยวิธี CE บทที่ 6 แสดงรายละเอียดของการทดลอง บทที่ 7 แสดงผลการทดลอง และบทที่ 8 จะเป็นข้อสรุปและข้อเสนอแนะจากการวิจัย

1.7 ผลงานตีพิมพ์จากงานวิจัย

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นบทความทางวิชาการและนำเสนอในงานประชุมวิชาการ มีรายละเอียดดังนี้

Pairit Nittayanuparp and Boonserm Kijisirikul. 2006. A Chain-Explosion Data Reduction Method for Artificial Neural Network Learning. The 10th National Computer Science and Engineering Conference (NCSEC2006), October, Khonkaen, Thailand.

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ข่ายงานประสาทเทียม

ดร. โรเบิร์ต เฮชต์-นิลเซน (Dr. Robert Hecht-Nielsen) ซึ่งเป็นผู้ประดิษฐ์คอมพิวเตอร์เชิงปัญญา เครื่องแรกๆ ของโลก ได้ให้คำจำกัดความข่ายงานประสาทเทียมไว้ว่า “ระบบการคำนวณชนิดหนึ่งซึ่งสร้างขึ้นจากหน่วยประมวลผลอย่างง่ายจำนวนมากที่เชื่อมต่อเข้าด้วยกันอย่างหนาแน่น และประมวลผลสารสนเทศโดยการตอบสนองต่อข้อมูลจากภายนอก ในสถานะที่ไม่คงตัว” [5]

ข่ายงานประสาทเทียม คือ โครงสร้างของหน่วยประมวลผลจำนวนมาก ที่ถูกจำลองขึ้นมาอย่างคร่าวๆ ตามอย่างโครงสร้างของระบบประสาทของสมองส่วนเซอรีบรัลคอร์เทกซ์ (cerebral cortex) ของมนุษย์ แต่จะมีขนาดที่เล็กกว่ามาก ข่ายงานประสาทเทียมขนาดใหญ่อาจมีจำนวนหน่วยประมวลผลได้เป็น 10^4 หน่วย ในขณะที่สมองของมนุษย์มีเซลล์ประสาทประมาณ 10^{11} เซลล์ และเซลล์แต่ละเซลล์จะสื่อสารกับเซลล์ประสาทอื่นๆ อีกประมาณ 10^4 เซลล์ [3]

สมองของมนุษย์ประกอบด้วยข่ายงานประสาท ซึ่งมีเซลล์ประสาทหลายแสนล้านเซลล์ที่สื่อสารกัน และทำงานขนานกันไป การสร้างข่ายงานประสาทเทียมสำหรับคอมพิวเตอร์ ไม่ใช่การสร้างชิ้นส่วนของคอมพิวเตอร์นับแสนล้านชิ้นที่รูปร่างเหมือนเซลล์สมองของมนุษย์ แต่เป็นการสร้างแบบจำลองเชิงคณิตศาสตร์ (mathematical model) ที่จำลองเอาวิธีการทำงานของสมองมาใช้ หรืออาจเป็นโปรแกรมคอมพิวเตอร์ที่ทำให้คอมพิวเตอร์รู้จักคิด และจดจำในแนวเดียวกับข่ายงานประสาทของมนุษย์

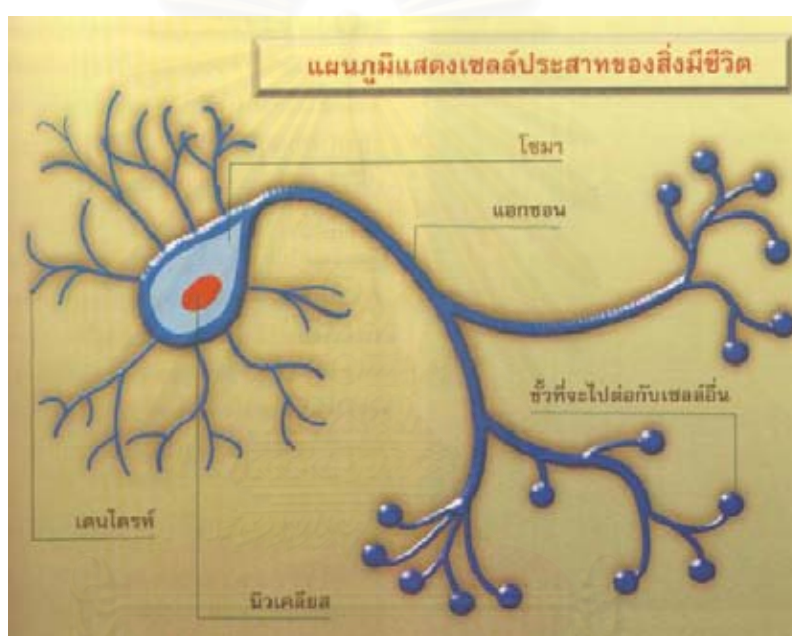
สิ่งที่น่าสนใจคือ เซลล์ประสาทแต่ละเซลล์ของสมองของมนุษย์นั้นทำงานได้ช้ากว่าหน่วยประมวลผลของดิจิทัลคอมพิวเตอร์เป็นล้านๆ เท่า [3] แต่สมองของมนุษย์ก็ยังสามารถในการทำงานที่ซับซ้อนมากๆ ได้อย่างมีประสิทธิภาพสูงกว่าคอมพิวเตอร์ที่เร็วที่สุดในโลก ความสามารถที่แตกต่างกันนี้มีรากฐานมาจากความจริงที่ว่า ลักษณะของการประมวลผลในระบบประสาทนั้นเป็นไปในลักษณะขนาน (parallel processing) แต่การทำงานของระบบคอมพิวเตอร์ในปัจจุบัน เป็นไปในลักษณะเป็นลำดับ (sequential processing) แต่ถึงกระนั้นก็ตาม แนวคิดการทำงานของระบบเซลล์ประสาทของมนุษย์ก็ยังคงมีความน่าสนใจ ที่จะนำมาสร้างเป็นแบบจำลองเพื่อให้คอมพิวเตอร์สามารถนำมาประยุกต์ใช้ในงานคำนวณที่ซับซ้อนต่างๆ ได้มีประสิทธิภาพ

การสร้างแบบจำลองของข่ายงานประสาทขึ้นมาเป็นข่ายงานประสาทเทียมนั้น ทำได้โดยการพยายามทำความเข้าใจกับกระบวนการทำงานของสมอง แล้วพยายามอธิบายการทำงานนั้นด้วยแบบจำลองเชิงคณิตศาสตร์ จากนั้นจึงออกแบบระบบคอมพิวเตอร์ หรือเขียนโปรแกรมคอมพิวเตอร์ที่จะทำงานตามแบบจำลองเชิงคณิตศาสตร์ที่ได้ [5]

2.1.1 เซลล์ประสาท (Neuron)

การพัฒนากระบวนการประมวลผลแบบข่ายงานประสาทเทียมนั้น จะอิงกับแนวทางการประมวลผลของสมองของสิ่งมีชีวิต ดังนั้นการทำความเข้าใจในคุณลักษณะเชิงกายภาพ และเชิงพฤติกรรมขององค์ประกอบต่างๆ ในสมองของสิ่งมีชีวิตจึงเป็นสิ่งจำเป็น

หน่วยรากฐานของสมองคือ เซลล์ประสาท (Neuron) สมองของมนุษย์ประกอบไปด้วย เซลล์ประสาทจำนวนมากอย่างน้อยในระดับแสนๆ ล้านเซลล์ ในแง่ของการทำงานนั้น เซลล์ประสาทแต่ละเซลล์คือ หน่วยประมวลผลอย่างง่ายๆ ซึ่งรับสัญญาณและรวมสัญญาณที่ถูกส่งมาจากเซลล์ประสาทอื่นๆ



รูปที่ 1 แผนภูมิแสดงเซลล์ประสาทของสิ่งมีชีวิต

จากรูปที่ 1 จะเห็นได้ว่าเซลล์ประสาทแต่ละเซลล์จะมีส่วนหลักๆ อยู่ 3 ส่วน คือ

1. ตัวเซลล์ซึ่งเรียกว่า โซมา (Soma) มีลักษณะเป็นรูปทรงพีระมิด หรือทรงกระบอก
2. เดนไดรต์ (Dendrite) คือ เส้นใยบางๆ ที่เซลล์ประสาทใช้รับสัญญาณไฟฟ้าเข้าสู่เซลล์ เซลล์ประสาทแต่ละเซลล์จะมีเดนไดรต์จำนวนมากเรียงตัวเป็นลักษณะเหมือนกิ่งไม้
3. แอกซอน (Axon) คือ สายส่งผ่านสัญญาณทรงกระบอกขนาดยาวและใหญ่ ที่เซลล์ประสาทใช้เป็นทางส่งสัญญาณไปยังเซลล์ประสาทอื่นๆ ส่วนปลายของแอกซอนจะแตกออกเป็นกิ่งก้านย่อยๆ โดยที่ส่วนปลายของแต่ละกิ่งก้านเหล่านี้ลักษณะเป็นปม และจะไปจ่ออยู่จนเกือบสัมผัสกับปลายของเดนไดรต์หนึ่งของเซลล์ประสาทเซลล์อื่นๆ

บริเวณที่เป็นรอยต่อระหว่างปลายของแอกซอนกับปลายของเดนไดรต์เรียกว่า ไชแนปส์ (Synapse) สัญญาณไฟฟ้าที่ถูกส่งมาถึงปลายของแอกซอน จะกระตุ้นให้เกิดการส่งผ่านสัญญาณในเชิงเคมีผ่าน ไชแนปส์ สัญญาณเชิงเคมีดังกล่าว จะถูกเดนไดรต์ตีความเป็นสัญญาณไฟฟ้าวิ่งเข้าสู่ เซลล์ประสาทต่อไป

คุณลักษณะสำคัญของ ไชแนปส์คือความแรงของสัญญาณที่ถูกส่งผ่านจะขึ้นอยู่กับความเหนียวแน่นของการเชื่อมต่อ และสัญญาณที่ถูกส่งผ่าน ไชแนปส์อาจถูกทำให้มีสภาพเป็นสัญญาณกระตุ้น (excitatory) หรือสัญญาณกด (inhibitory) ก็ได้ ขึ้นอยู่กับชนิดของสัญญาณเชิงเคมีที่ถูกกระตุ้นให้เคลื่อนผ่านรอยต่อ ซึ่งแต่ละประสาทอาจรับสัญญาณมาจาก 10,000 ไชแนปส์ หรือมากกว่า

สัญญาณจากเดนไดรต์ต่างๆ จะรวมกันวิ่งเข้าสู่ตัวเซลล์ประสาท และหากสัญญาณรวมมีความแรงเกินค่าระดับ (threshold) ของเซลล์ประสาทรุนั้นๆ เซลล์ประสาทก็จะยิงสัญญาณออกทางแอกซอนต่อไป กระบวนการเรียนรู้ในสิ่งมีชีวิต จะมีผลให้เกิดการสร้าง ไชแนปส์ระหว่างเซลล์ประสาทขึ้นมาใหม่ หรือไม่ก็ทำให้เกิดการปรับเปลี่ยนสภาพและความเหนียวแน่นของ ไชแนปส์ต่างๆ ที่มีอยู่ นั่นคือความรู้ได้ถูกเก็บไว้ในลักษณะที่กระจายไปตาม ไชแนปส์ต่างๆ ในข่ายงานของเซลล์ประสาทรุนั้นเอง อาจกล่าวได้ว่า ข่ายงานประสาทของสิ่งมีชีวิตทำงานตามโปรแกรมที่มีลักษณะกระจายไปทั่วข่ายงานของเซลล์ประสาท และข่ายงานประสาทของสิ่งมีชีวิตไม่ได้ทำงานแบบเป็นลำดับขั้นตอน

2.1.2 การเรียนรู้ของข่ายงานประสาทเทียม (Neural Network Learning)

แนวคิดของกระบวนการเรียนรู้ของข่ายงานประสาทเทียมนั้น จะประยุกต์มาจากผลการศึกษาทางพฤติกรรมศาสตร์ อาจกล่าวโดยทั่วไปได้ว่าการเรียนรู้คือ กระบวนการซึ่งระบบประสาทปรับตัวเองไปตามสิ่งเร้า จนกระทั่งสามารถให้ผลตอบได้ตามต้องการ โดยใช้การปรับตัวแปรที่ควบคุมสภาพของตัวระบบเอง ซึ่งได้มีการจำกัดความขั้นตอนของกระบวนการเรียนรู้ไว้ดังต่อไปนี้

ขั้นที่ 1 ข่ายงานประสาทถูกกระตุ้นด้วยสิ่งแวดล้อม

ขั้นที่ 2 ข่ายงานประสาทเกิดการเปลี่ยนแปลง อันเป็นผลมาจากการกระตุ้นดังกล่าว

ขั้นที่ 3 ข่ายงานประสาทตอบสนองต่อสิ่งแวดล้อมในแนวทางใหม่ อันเป็นผลมาจากการเปลี่ยนแปลงที่เกิดขึ้นในโครงสร้างภายในข่ายงาน

การเรียนรู้ยังสามารถถูกมองได้ว่า เป็นกระบวนการจัดชนิดของสิ่งเร้าทั้งหลายที่เข้ามาอย่างต่อเนื่องด้วย นั่นคือ เมื่อได้รับสิ่งเร้า หากระบบประสาทรู้จักสิ่งเร้า นั้น ก็จะให้ผลตอบได้ตามที่เคย

เข้าใจไว้ แต่หากไม่รู้จักร ก็พยายามปรับความเข้าใจในการจัดชนิดขึ้นใหม่ ในทางปฏิบัติ ระบบประสาทของสิ่งมีชีวิตจะปรับความเหนียวแน่นของการเชื่อมต่อที่ไซแนปส์ จนสร้างผลตอบต่อสิ่งเร้าได้ตามที่ต้องการ สถานะที่กระบวนการของการเรียนรู้ก็จะสิ้นสุดลง เป็นสถานะที่ถือว่าระบบประสาทได้รับความรู้ไปแล้ว

2.1.3 เพอร์เซปตรอน (Perceptrons)

หน่วยประมวลผลย่อยที่สุด ในข่ายงานประสาทเทียม เรียกว่าเพอร์เซปตรอน ซึ่งโดยตัวของเพอร์เซปตรอนเองแล้ว ไม่สามารถใช้เป็นแบบในการอธิบายการทำงานของเซลล์ประสาทของสิ่งมีชีวิตได้ถูกต้อง แต่เป็นการนำเอาแนวคิดที่ได้จากความเข้าใจการทำงานของเซลล์ประสาทของสิ่งมีชีวิตมาประยุกต์ใช้ แบบจำลองพื้นฐานของเซลล์ประสาทเทียมถูกนำเสนอโดยแม็กคัลลอคและพิตส์ [6] ตั้งแต่ปี พ.ศ. 2486 โดยมีการทำงานคร่าวๆ แบบเซลล์ประสาทของสิ่งมีชีวิตคือ ทำหน้าที่รวมสัญญาณที่เข้ามายังเพอร์เซปตรอน (ซึ่งเสมือนว่าเป็นสัญญาณที่เข้ามาตามเดนไดรต์ของเซลล์ประสาทของสิ่งมีชีวิต) ซึ่งหากผลรวมของสัญญาณเข้านั้นมีค่าสูงเกินค่าระดับ (threshold) เพอร์เซปตรอนก็จะยิงสัญญาณกระตุ้นออกไป ก็เสมือนการยิงสัญญาณไฟฟ้าออกทางแอกซอนจากเซลล์ประสาทของสิ่งมีชีวิตนั่นเอง

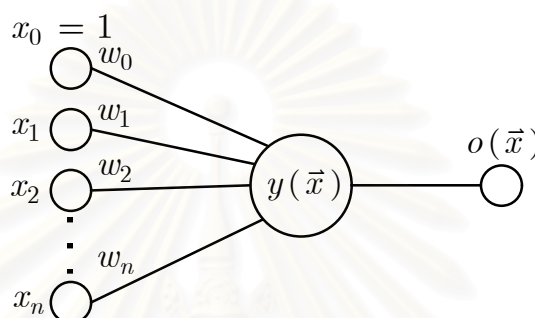
อย่างไรก็ตาม สิ่งที่สำคัญในการจำลองเซลล์ประสาทคือ การจำลองไซแนปส์ทั้งหลายในข่ายงานประสาทเทียม ซึ่งเปรียบเสมือนแหล่งสะสมความรู้ของสมอง การจำลองไซแนปส์นั้นใช้หลักการที่ว่า แต่ละไซแนปส์ทำหน้าที่เป็นตัวปรับเปลี่ยนสภาพสัญญาณไฟฟ้าที่ส่งมาจากเซลล์ประสาทตัวอื่นๆ ก่อนส่งสัญญาณนั้นผ่านเดนไดรต์เข้าสู่ตัวเซลล์ประสาท และการปรับเปลี่ยนสัญญาณดังกล่าว จะขึ้นอยู่กับความเหนียวแน่นของการเชื่อมต่อบริเวณรอยต่อไซแนปส์ โดยความแข็งแรงนี้จะเปลี่ยนไปตามความรู้ที่เซลล์ได้เรียนเข้าไป แม็กคัลลอคและพิตส์เสนอให้ใช้ตัวแปรตัวหนึ่งเรียกว่า “ค่าน้ำหนัก” (weight) ในการจำลองไซแนปส์ หากค่าน้ำหนักนี้มีขนาดใหญ่ก็จะหมายความว่า ความเหนียวแน่นของรอยต่อไซแนปส์มีค่าสูง นั่นคือส่งผ่านสัญญาณได้มาก หากค่าน้ำหนักนี้มีขนาดเล็กก็หมายความว่าสัญญาณจะส่งผ่านรอยต่อไซแนปส์ได้น้อย นอกจากนั้น ความเป็นบวกหรือลบของค่าน้ำหนักก็มีความหมายเช่นกัน หากค่าน้ำหนักมีค่าเป็นบวกจะหมายความว่าสัญญาณที่วิ่งผ่านรอยต่อไซแนปส์เข้าสู่เซลล์ประสาทจะเป็นสัญญาณกระตุ้น แต่หากค่าน้ำหนักมีค่าเป็นลบ จะหมายความว่าสัญญาณที่ผ่านรอยต่อไซแนปส์เข้าสู่เซลล์ประสาทจะมีผลเป็นสัญญาณกด

จากการทำงานของเซลล์ประสาทที่กล่าวมาในข้างต้น นำมาสร้างแบบจำลองซึ่งเป็นหน่วยย่อยที่สุดของข่ายงานประสาทเทียม เรียกว่าเพอร์เซปตรอนได้ดังรูปที่ 2

สมการที่ (1) และ (2) แสดงตัวแบบเชิงคณิตศาสตร์ของเพอร์เซปตรอน

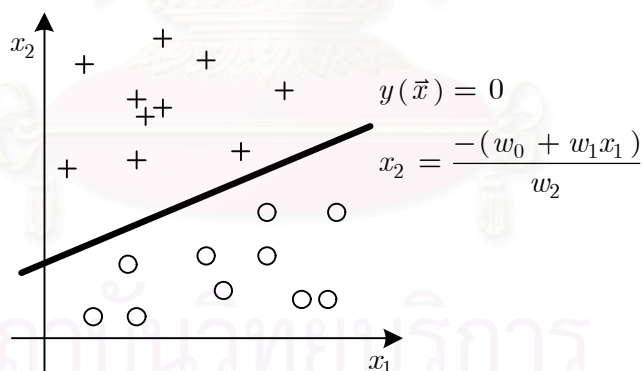
$$y(\vec{x}) = \sum_{i=0}^n w_i x_i = \vec{w} \cdot \vec{x} \quad ; x_0 = 1 \quad (1)$$

$$o(\vec{x}) = \begin{cases} 1 & \text{if } y(\vec{x}) \geq 0 \\ -1 & \text{if } y(\vec{x}) < 0 \end{cases} \quad (2)$$



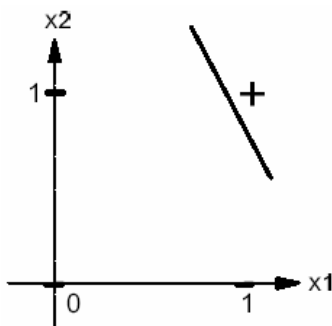
รูปที่ 2 เพอร์เซปตรอนหนึ่งหน่วย

เพอร์เซปตรอนจะนำเวกเตอร์น้ำหนักไปคูณแบบสเกลาร์กับเวกเตอร์อินพุต แล้วนำผลลัพธ์ที่ได้ไปเปรียบเทียบกับค่าขีดจำกัด (threshold) เพื่อตัดสินใจว่าคำตอบควรจะเป็น 1 หรือ -1



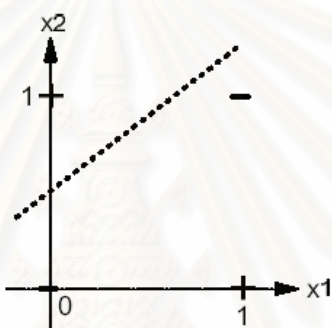
รูปที่ 3 พื้นผิวตัดสินใจที่ใช้แบ่งแยกข้อมูล

เราอาจจะแทนเพอร์เซปตรอนแต่ละตัวด้วยเวกเตอร์ \vec{w} ซึ่งเป็นพื้นผิวตัดสินใจ (decision surface) ที่มี n -มิติ โดยเพอร์เซปตรอนจะให้คำตอบเป็น 1 ถ้าข้อมูลเข้าอยู่ฝั่งหนึ่งของพื้นผิวตัดสินใจ และให้คำตอบเป็น -1 ถ้าข้อมูลเข้าอยู่อีกฝั่งของพื้นผิวตัดสินใจ ดังที่แสดงในรูปที่ 3 ซึ่งการจะได้มาซึ่ง \vec{w} นั้น กระทำได้โดยให้เพอร์เซปตรอนเรียนรู้และปรับค่าเพื่อหา \vec{w} ที่เหมาะสม



รูปที่ 4 ฟังก์ชัน AND, แบ่งแยกได้แบบเชิงเส้น

กรณีที่เพอร์เซปตรอนสามารถแบ่งข้อมูลออกเป็นสองกลุ่มได้ถูกต้องทุกข้อมูล ด้วยสมการขีดจำกัด $g(\vec{x}) = 0$ เราจะเรียกข้อมูลเข้าทั้งหมดนี้ว่า ข้อมูลที่แบ่งแยกได้แบบเชิงเส้น (linear separable data) ดังรูปที่ 4



รูปที่ 5 ฟังก์ชัน XOR, แบ่งแยกไม่ได้แบบเชิงเส้น

แต่ในกรณีรูปที่ 5 แสดงให้เห็นถึงข้อมูลที่แบ่งแยกไม่ได้แบบเชิงเส้น (non-linear separable data) ในกรณีเช่นนี้จำเป็นต้องใช้สิ่งที่เรียกว่าข่ายงานประสาทเทียมหลายชั้น (multi layer perceptron)

2.1.4 การเรียนรู้ของเพอร์เซปตรอน (Perceptron Learning)

เป้าหมายของการเรียนรู้ของเพอร์เซปตรอนคือ หาค่าน้ำหนักของเพอร์เซปตรอนแต่ละตัวที่ทำให้ได้พื้นผิวตัดสินใจที่จำแนกประเภทข้อมูลได้ถูกต้องทั้งหมด (แต่ในกรณีที่ข้อมูลไม่สามารถแบ่งแยกได้แบบเชิงเส้น ค่าน้ำหนักของเพอร์เซปตรอนจะไม่ลู่เข้า)

ขั้นตอนเริ่มจากสุ่มค่า w ให้มีค่าเล็กๆ ขึ้นมาก่อน แล้วปรับค่าน้ำหนักของเพอร์เซปตรอนไปเรื่อยๆ ตามสมการที่ (3) จนกว่าเพอร์เซปตรอนจะแบ่งแยกข้อมูลได้ถูกต้องทุกตัว

$$\begin{aligned}\Delta w_i &= \eta(t - o)x_i \\ w_i &\leftarrow w_i + \Delta w_i\end{aligned}\quad (3)$$

- x_i คือค่าข้อมูลอินพุต
 t คือค่าตอบจริง (target)
 o คือค่าที่คำนวณได้จากเพอร์เซปตรอน (output)
 η คืออัตราการเรียนรู้ (learning rate)

2.1.5 การฝึกเพอร์เซปตรอนด้วยกฎเดลต้า (Delta Rule)

ในกรณีที่ข้อมูลไม่สามารถแบ่งแยกได้แบบเชิงเส้น ถ้าใช้วิธีที่กล่าวไว้ในหัวข้อ 2.1.4 พบว่าค่าน้ำหนักจะแกว่งไปมา ไม่ลู่เข้า ดังนั้นจึงคิดวิธีการแก้ปัญหาขึ้น โดยเปลี่ยนเป้าหมายเป็นการหาค่าน้ำหนักที่ทำให้ฟังก์ชันค่าคลาดเคลื่อนของการสอนเพอร์เซปตรอน (Perceptron training error function) มีค่าน้อยที่สุดแทน

นิยามฟังก์ชันค่าคลาดเคลื่อนของการสอนตามสมการที่ (4)

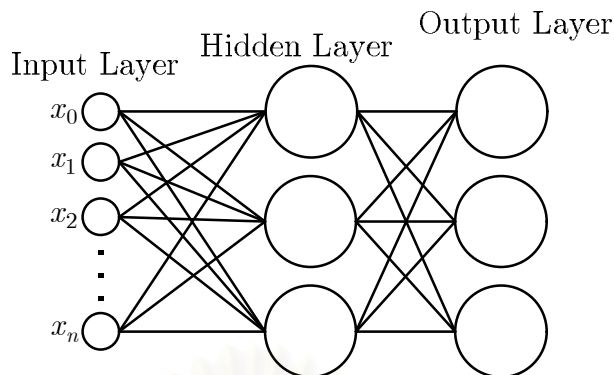
$$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2 \quad (4)$$

ขั้นตอนเริ่มจากสุ่มค่า \vec{w} ให้มีค่าเล็กๆ ขึ้นมาก่อน แล้วปรับค่าน้ำหนักของเพอร์เซปตรอนไปเรื่อยๆ ตามสมการที่ (5) จนกว่าค่าน้ำหนักไม่เปลี่ยนแปลงอีก

$$\begin{aligned}\Delta \vec{w} &= -\eta \nabla E(\vec{w}) \\ \Delta w_i &= -\eta \frac{\partial E}{\partial w_i} \\ \Delta w_i &= \eta \sum_{d \in D} (t_d - o_d) x_{id} \\ w_i &\leftarrow w_i + \Delta w_i\end{aligned}\quad (5)$$

2.1.6 ข่ายงานประสาทเทียมหลายชั้น (Multi Layer Perceptron)

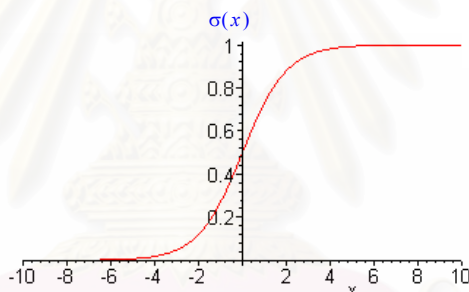
เป็นการนำเพอร์เซปตรอนหลายๆ ตัวมาต่อกันเป็นข่ายงาน โดยทั่วไปข่ายงานประสาทหลายชั้นจะประกอบไปด้วยชั้นอย่างน้อย 2 ชั้น (layer) ได้แก่ ชั้นซ่อน (hidden layer) และชั้นเอาต์พุต (output layer) ดังรูปที่ 6



รูปที่ 6 โครงสร้างของข่ายงานประสาทเทียมหลายชั้น

เปลี่ยนฟังก์ชันกระตุ้น โดยฟังก์ชันซิกมอยด์แทน (สมการที่ (6) ภาพแสดงในรูปที่ 7) จะทำให้คำตอบจากเพอร์เซปตรอนมีค่าต่อเนื่องระหว่าง 0 ถึง 1 และจะทำให้การหาอนุพันธ์ของฟังก์ชันเอาต์พุตทำได้ง่ายขึ้น

$$\sigma = \sigma(y(x)) = \frac{1}{1 + e^{-y(x)}} \quad (6)$$



รูปที่ 7 ฟังก์ชันซิกมอยด์

2.1.7 การเรียนรู้ด้วยวิธีการแพร่ย้อนกลับ (Back propagation)

เริ่มจากการป้อนอินพุตไปข้างหน้า (feed forward) จนได้เอาต์พุต กำหนดความคลาดเคลื่อนของเอาต์พุตแล้วนำกลับไปปรับค่าน้ำหนัก ซึ่งในขั้นตอนนี้เองที่เรียกว่าการแพร่ย้อนกลับ (back propagation)

นิยามฟังก์ชันค่าคลาดเคลื่อนของการสอนในกรณีข่ายงานประสาทเทียมหลายชั้นดังนี้

$$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} \sum_{k \in \text{outputs}} (t_{kd} - o_{kd})^2 \quad (7)$$

ขั้นตอนเริ่มจากสุ่มค่า w ให้มีค่าเล็กๆ ขึ้นมาก่อน แล้วปรับค่านำหนักของเพอร์เซปตรอนไปเรื่อยๆ จนกว่าค่านำหนักไม่เปลี่ยนแปลงอีก

$$\text{สำหรับชั้นเอาต์พุท} \quad \delta_k = o_k(1 - o_k)(t_k - o_k) \quad (8)$$

$$\text{สำหรับชั้นซ่อน} \quad \delta_h = o_h(1 - o_h) \sum_{k \in \text{outputs}} w_{kh} \delta_k \quad (9)$$

สำหรับการปรับค่านำหนักอินพุทจากชั้น i ไปยังชั้น j

$$\begin{aligned} \Delta w_{ji} &= \eta \delta_j x_{ji} \\ w_{ji} &\leftarrow w_{ji} + \Delta w_{ji} \end{aligned} \quad (10)$$

ถ้ามีการใช้โมเมนตัม (α) เพื่อช่วยหลีกเลี่ยงการเกิดค่าต่ำสุดเฉพาะจุด (local minima) โดย $0 \leq \alpha \leq 1$ จะต้องปรับสมการเป็น

$$\begin{aligned} \Delta w_{ji} &\leftarrow \eta \delta_j x_{ji} + \alpha \Delta w_{ji} \\ w_{ji} &\leftarrow w_{ji} + \Delta w_{ji} \end{aligned} \quad (11)$$

2.2 k -เนียร์เรสเนเบอร์ (k -Nearest Neighbor – k NN)

เป็นวิธีการเรียนรู้ข้อมูลแบบใช้ตัวอย่าง (instance-based learning methods) [3] โดยใช้สมมติฐานว่า การจะจำแนกประเภทข้อมูลตัวใดที่ไม่รู้ประเภท ทำได้โดยพิจารณาจากข้อมูลรอบข้างว่าส่วนใหญ่อยู่ประเภทใด ทั้งนี้จะใช้การพิจารณาข้อมูลรอบข้างทั้งหมด k ตัว

การจะนำ k NN ไปใช้งานได้นั้นจะต้องป้อนข้อมูลเข้าสู่ระบบก่อน โดยระบบจะนำข้อมูลฝึกไปเก็บไว้ในหน่วยความจำ เมื่อใดที่ผู้ใช้งานนำข้อมูลมาให้ k NN จำแนกประเภท ระบบก็จะนำข้อมูลตัวนั้นไปค้นหาในหน่วยความจำเพื่อหาข้อมูลฝึก k ตัวที่อยู่ใกล้ที่สุดออกมา ความซับซ้อนในการทำงานของ k NN จะอยู่ในช่วงการจำแนกประเภท ซึ่งมีประเด็นให้ต้องพิจารณาว่าจะใช้การวัดระยะห่างของข้อมูลด้วยวิธีใด

เช่นใช้ระยะห่างแบบแมนฮัตตัน (Manhattan distance) ตามสมการ (12)

$$d(x, y) = \sum_{i=1}^{n_a} |x_i - y_i| \quad (12)$$

หรือระยะห่างแบบยูคลิเดียน (Euclidean distance) ซึ่งเป็นที่นิยมทั่วไป ตามสมการ (13)

$$d(x, y) = \sqrt{\sum_{i=1}^{n_a} (x_i - y_i)^2} \quad (13)$$

นอกจากนี้ประเด็นที่ยังต้องตัดสินใจเลือกใช้สำหรับ k NN คือวิธีการลงคะแนนว่า (vote) ว่าควรจะตัดสินใจให้ข้อมูลที่ต้องการจำแนกประเภทเป็นประเภทใด

กำหนดให้ k คือจำนวนเนียร์เรียวสเนเบอร์

$NN_i(x)$ คือเนียร์เรียวสเนเบอร์ลำดับที่ i ของข้อมูล x

เราอาจจะใช้วิธีง่ายๆ เพื่อจำแนกประเภทของ x คือหนึ่งข้อมูลลงคะแนนได้หนึ่งคะแนน (one man one vote) ตามสมการที่ (14) และใช้ฟังก์ชัน $score(a, b)$ ตามสมการที่ (15) ประเภทข้อมูลใดได้คะแนนมากที่สุดจากการลงคะแนน ก็จะตัดสินใจให้ข้อมูล x เป็นประเภทนั้น

$$class(x) = \arg \max_{c \in CLASS} \sum_{i=1}^k score(c, class(NN_i(x))) \quad (14)$$

$$score(a, b) = \begin{cases} 1 & ; a = b \\ 0 & ; otherwise \end{cases} \quad (15)$$

หรืออาจจะให้น้ำหนักคะแนนตามระยะห่างของข้อมูล (distance-weighted NN) โดยใช้สมการที่ (16)

$$class(x) = \arg \max_{c \in CLASS} \sum_{i=1}^k \frac{1}{d(x, NN_i(x))} \cdot score(c, class(NN_i(x))) \quad (16)$$

2.3 งานวิจัยที่เกี่ยวข้อง

การลดข้อมูลสำหรับอัลกอริทึมเรียนรู้ (learning algorithm) สามารถแบ่งตามเป้าหมายในการลดได้เป็นสามแนวทาง [7] ได้แก่

1. การลดเพื่อกำจัดข้อมูลที่เกินความจำเป็น
2. การลดเพื่อกำจัดข้อมูลสัญญาณรบกวน
3. แบบลูกผสมระหว่างสองวิธีแรก

ได้มีงานวิจัยอื่นๆ ที่แสดงถึงความพยายามที่จะลดขนาดข้อมูลลงในแง่มุมใดแง่มุมหนึ่ง ซึ่งงานวิจัยเหล่านี้มีอิทธิพลต่อแนวความคิดในการสร้างอัลกอริทึมสำหรับงานวิจัยนี้

2.3.1 งานวิจัยเกี่ยวกับการเลือกคุณสมบัติของข้อมูล (Feature Selection)

Kira และ Rendell [8] ทำการวิจัยเรื่องการเลือกคุณสมบัติของข้อมูล (feature selection) โดยนำเสนออัลกอริทึม Relief เพื่อประเมินความสามารถในการจำแนกประเภทข้อมูลของ

คุณสมบัติแต่ละตัว โดยเริ่มจากสุ่มหยิบข้อมูลขึ้นมาทีละหนึ่งตัว แล้วหาข้อมูลตัวอื่นที่อยู่ใกล้มันมากที่สุดที่เป็นประเภทเดียวกัน เรียกว่าตัวใกล้สุดถูก (nearest hit) และหาข้อมูลตัวอื่นที่อยู่ใกล้มันมากที่สุดแต่เป็นคนละประเภท เรียกว่าตัวใกล้สุดผิด (nearest miss) มาเป็นตัววัดว่าคุณสมบัติใดใช้จำแนกประเภทข้อมูลได้ดีที่สุด สุ่มหยิบข้อมูลขึ้นมาทั้งหมด m ตัว อัลกอริทึมแสดงในรูปที่ 8

$W[A]$ เป็นคะแนนของคุณสมบัติ โดยถ้ามีค่ามากแสดงว่าเป็นคุณสมบัติที่มีความสามารถในการจำแนกประเภทข้อมูลได้ดีจึงสมควรเก็บไว้ ทั้งนี้ค่าที่ได้จะถูกทำให้ไม่เกิน 1

ฟังก์ชัน $diff(A, I_1, I_2)$ จำคำนวณความแตกต่างระหว่างค่าคุณสมบัติ A สำหรับข้อมูล I_1 และ I_2 ดังแสดงในสมการที่ (17)

$$diff(A, I_1, I_2) = \frac{|value(A, I_1) - value(A, I_2)|}{\max(A) - \min(A)} \quad (17)$$

```

procedure RELIEF
  set all weights  $W[A] \leftarrow 0$ 
  for  $i \leftarrow 1$  to  $m$  do
    randomly select an instance  $x$  from training data  $T$ 
    find nearest hit  $H$  and nearest miss  $M$ 
    for  $A \leftarrow 1$  to  $n_a$  do
       $W[A] \leftarrow W[A] - diff(A, x, H)/m + diff(A, x, M)/m$ 
    end for
  end for
end procedure

```

รูปที่ 8 รหัสเทียมอัลกอริทึม Relief

แต่เนื่องจาก Relief สามารถทำงานได้เฉพาะกับข้อมูลที่มีสองประเภทเท่านั้น Kononenko [9] จึงนำอัลกอริทึม Relief มาพัฒนาต่อ กลายเป็นอัลกอริทึม ReliefF ซึ่งอัลกอริทึมที่พัฒนาขึ้นมาี้สามารถทำงานกับข้อมูลที่มีประเภทมากกว่า 2 ประเภทได้ และใช้ตัวใกล้สุดถูก และตัวใกล้สุดผิดอย่างละ k ตัว ทำให้ทนทานต่อสัญญาณรบกวนได้ดียิ่งขึ้น โดย $P(C)$ เป็นความน่าจะเป็นของประเภทข้อมูล C ซึ่งคำนวณได้จากข้อมูลฝึกที่มีอยู่ อัลกอริทึมแสดงในรูปที่ 9

หลังจากนั้น Sikonja และ Kononenko [10] ได้นำ ReliefF มาพัฒนาต่อเป็น RReliefF เพื่อให้อัลกอริทึมทำงานกับข้อมูลที่มีประเภทเป็นค่าต่อเนื่องได้ อัลกอริทึมนี้ไม่เพียงแต่จะใช้จำแนกประเภทได้ดีเท่านั้น แต่ยังสามารถนำไปใช้ในงานทำนายค่าได้อีกด้วย (regression)

```

procedure RELIEFF
  set all weights  $W[A] \leftarrow 0$ 
  for  $i \leftarrow 1$  to  $m$  do
    randomly select an instance  $x$  from training data  $T$ 
    find  $k$  nearest hits  $H_{L,k}$ 
    for each class  $C \neq \text{class}(x)$  do
      from class  $C$  find  $k$  nearest misses  $M_{L,k}(C)$ 
      for  $A \leftarrow 1$  to  $a$  do

$$W[A] \leftarrow W[A] - \sum_{j=1}^k \text{diff}(A, x, H_j) / (m \cdot k) +$$


$$\sum_{C \neq \text{class}(x)} \left[ \frac{P(C)}{1 - P(\text{class}(x))} \sum_{j=1}^k \text{diff}(A, x, M_j(C)) \right] (m \cdot k)$$

      end for
    end for
  end for
end procedure

```

รูปที่ 9 รหัสเทียมอัลกอริทึม ReliefF

2.3.2 งานวิจัยเกี่ยวกับการลดตัวข้อมูล

แนวความคิดการลดข้อมูลถูกนำเสนอครั้งแรกโดย Hart [11] ใช้การมองการลดข้อมูลในมุมมองว่าเป็นการเลือกเซตย่อยที่สอดคล้องของเซตข้อมูลฝึก (training set consistence subset) โดยใช้ชื่ออัลกอริทึมลดข้อมูลนี้ว่า CNN Rule (Condensed Nearest Neighbor) ซึ่งเป็นการลดข้อมูลสำหรับอัลกอริทึมเนียร์เรสท์เนเบอร์ (Nearest Neighbor) การทำงานเริ่มจากเลือกข้อมูลฝึกขึ้นมาตัวหนึ่งจากข้อมูลฝึกที่มีทั้งหมด แล้วค่อยๆ เลือกข้อมูลฝึกตัวอื่นๆ ที่ยังถูกจำแนกประเภทผิด (misclassify) เข้ามาเป็นส่วนหนึ่งในข้อมูลฝึก จนกว่าข้อมูลฝึกที่เลือกมา จะเพียงพอที่จะจำแนกประเภทข้อมูลฝึกที่มีอยู่ทั้งหมดได้ถูกต้อง

หลังจากนั้นก็มีผู้เสนอวิธีลดข้อมูลสำหรับอัลกอริทึมเรียนรู้แบบใช้ตัวอย่างตามมาอีกมากมายเช่น RNN [12], SNN [13], MCS [14], MCNN [15] เป็นต้น แต่กลับไม่พบงานวิจัยด้านการลดข้อมูลสำหรับข่ายงานประสาท ทั้งนี้เนื่องมาจากข่ายงานประสาทจะสร้างพื้นผิวตัดสินใจถ้าป้อนข้อมูลฝึกให้มากขึ้น มิใช่ลดให้น้อยลง แต่เนื่องจากปัญหาที่พบในการทำงานกับข้อมูลฝึกจำนวนมากสาธการกับข่ายงานประสาท ทำให้มีความจำเป็นอย่างยิ่งที่จะต้องหาวิธีการลดข้อมูลฝึกลง เพื่อให้การฝึกข่ายงานประสาททำได้สำเร็จ

บทที่ 3

วิธีการลดข้อมูล

3.1 การลดข้อมูลแบบสุ่มตามความหนาแน่น (Density-Based Random Data Reduction)

3.1.1 แรงจูงใจ

ด้วยแนวคิดลดข้อมูลโดยใช้หลักการลดข้อมูลที่เกินความจำเป็น ทำให้การพิจารณา มีแนวทางชัดเจนว่า ข้อมูลใดที่ไม่จำเป็นต้องใช้ในการฝึกก็อาจจะตัดทิ้งได้ การจะตัดสินว่าข้อมูลใด เกินความจำเป็น หรือไม่จำเป็นต้องใช้ในการฝึก อาจจะพิจารณาง่ายๆ จากแนวคิดที่ว่า ถ้ามีข้อมูลตัว อื่นสามารถทดแทนตัวมันเองได้ ย่อมแสดงว่าตัวมันเองน่าจะมีความสำคัญลดลงจนถึงขั้นที่ตัดทิ้งก็ ไม่มีผลต่อความสามารถในการจำแนกประเภทข้อมูลของข่ายงานประสาท

เราอาจจะใช้สมมติฐานที่ว่าข้อมูลที่อยู่ใกล้ๆ กันย่อมมีคุณสมบัติต่างกัน ไม่มากนัก ย่อม สามารถใช้ทดแทนกันได้ ดังนั้นบริเวณใดก็ตามในปริภูมิข้อมูล (data space) ที่มีข้อมูลอยู่กันอย่าง หนาแน่นก็น่าจะตัดข้อมูลทิ้งได้มาก แต่บริเวณใดที่มีความหนาแน่นของข้อมูลเบาบาง การตัด ข้อมูลทิ้งเพียงเล็กน้อยก็อาจมีผลต่อรูปร่างลักษณะของข้อมูลได้มาก ซึ่งจะมียังผลมากต่อการฝึก ดังนั้นการลดข้อมูลในบริเวณที่ข้อมูลหนาแน่นเบาบางจึงควรทำในสัดส่วนที่น้อยกว่าบริเวณที่ ข้อมูลอยู่รวมกันหนาแน่นมาก

3.1.2 การลดข้อมูลแบบสุ่มตามความหนาแน่น

การลดข้อมูลวิธีที่ง่ายที่สุดคือการสุ่มลด หรือเรียกว่าการสุ่มเลือก (sampling) เป็นการสุ่ม ลดข้อมูลที่ทำได้โดยไม่ต้องใช้ความรู้ใดๆ เข้ามาช่วยพิจารณาในการลด ใช้แค่เพียงการสุ่มแล้วก็ลด เท่านั้น ซึ่งถ้าต้องการให้การสุ่มลดเป็นไปอย่างมีชั้นเชิง ก็อาจใช้การศึกษาสำนึก (heuristic) เพื่อ ชี้นำการสุ่มลด ผู้วิจัยเสนอให้ใช้ความหนาแน่นของข้อมูลเป็นแนวทางชี้นำการสุ่มลดได้โดยใช้ แนวคิดที่ว่าบริเวณที่หนาแน่นมากจะมีโอกาสถูกลดข้อมูลมาก บริเวณที่หนาแน่นน้อยมีโอกาสดูถูก ลดข้อมูลน้อย

การคำนวณความหนาแน่นของข้อมูลในบริเวณต่างๆ อาจจะทำได้โดยการตีตารางแบ่ง ข้อมูลเป็นช่องๆ แล้วนับจำนวนข้อมูลที่อยู่ในแต่ละช่อง ซึ่งการทำเช่นนี้ในทางปฏิบัติอาจจะเป็นไปได้ยาก โดยเฉพาะอย่างยิ่งเมื่อข้อมูลมีจำนวนมิติ (dimension) สูงมาก ดังนั้นเราจึงใช้อีกวิธีหนึ่ง คือ ใช้ k NN ช่วยคำนวณความหนาแน่นของข้อมูล

สำหรับข้อมูล x ใดๆ เราจะหาความหนาแน่นของบริเวณนั้นได้ด้วยการหาเนียร์เรียง-
เนเบอร์ที่อยู่ใกล้มันมากที่สุด k ตัว แล้วดูว่ามีกี่ตัวที่เป็นประเภทเดียวกับมัน โอกาสที่ข้อมูล x ตัว
นั้นๆ จะถูกตัดทิ้งก็ขึ้นอยู่กับความหนาแน่นของข้อมูลในบริเวณที่มันอยู่

กำหนดให้

k คือจำนวนเนียร์เรียงเนเบอร์

$NN_i(x)$ คือเนียร์เรียงเนเบอร์ลำดับที่ i ของข้อมูล x

ความน่าจะเป็นในการที่จะตัดข้อมูล x คือค่า $p(x)$ คำนวณได้ตามสมการที่ (18)

$$p(x) = \frac{\sum_{i=1}^k \text{score}(x, NN_i(x))}{k} \quad (18)$$

โดยกำหนดให้ใช้ $\text{score}(x, y)$ ตามสมการที่ (19) ตามลำดับ

$$\text{score}(x, y) = \begin{cases} 1 & ; \text{class}(x) = \text{class}(y) \\ 0 & ; \text{otherwise} \end{cases} \quad (19)$$

เราอาจจะเปลี่ยนวิธีการคำนวณความน่าจะเป็นในการตัดข้อมูลให้นำหนักขึ้นอยู่กับ
ระยะห่างของข้อมูลด้วย โดยเปลี่ยน $p(x)$ เป็นดังสมการที่ (20) และใช้ $\omega_i = \frac{1}{d(x, NN_i(x))}$

$$p(x) = \frac{\sum_{i=1}^k \omega_i \cdot \text{score}(x, NN_i(x))}{\sum_{i=1}^k \omega_i} \quad (20)$$

procedure DENSITY-BASED_REDUCE($T, rate$)

$target \leftarrow N(T) \cdot rate$

while $target \neq 0$ **do**

$x \leftarrow$ select an existing data from T

if $random \leq p(x)$ **then**

$T \leftarrow T - \{x\}$

$target \leftarrow target - 1$

end if

end while

return T

end procedure

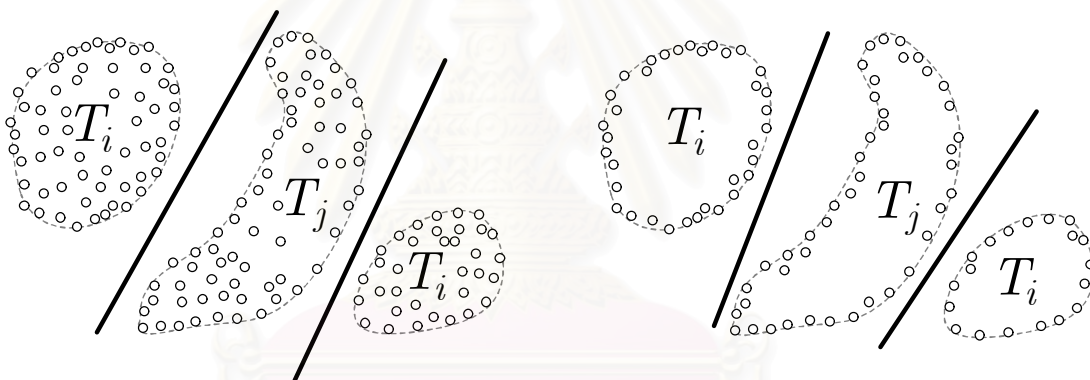
รูปที่ 10 รหัสเทียมอัลกอริทึมลดข้อมูลตามความหนาแน่น

อัลกอริทึมลดข้อมูลตามความหนาแน่นแสดงไว้ในรูปที่ 10 โดยจะต้องกำหนดอัตราการลดข้อมูลที่ต้องการ ($0 \leq \text{rate} \leq 1$) อัลกอริทึมจะวนซ้ำไปจนกว่าจะตัดข้อมูลที่ทิ้งได้จำนวนตัวเท่ากับค่า *target* โดยในแต่ละรอบจะเลือกข้อมูล x ขึ้นมาตัวหนึ่งจากข้อมูลฝึกที่ยังไม่ถูกตัดใน T แล้วคำนวณค่าความหนาแน่นเพื่อหาความน่าจะเป็นที่ข้อมูล x จะถูกตัดโดยใช้ฟังก์ชัน $p(x)$

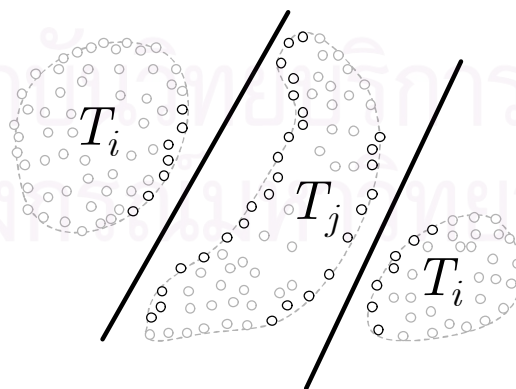
3.2 การลดข้อมูลแบบระเบิดลูกโซ่ (Chain-Explosion Data Reduction – CE Data Reduction)

3.2.1 แรงจูงใจ

สมมติว่าข้อมูล T ซึ่งมี 2 ประเภทคือ T_i กับ T_j เมื่อนำข้อมูลนี้ไปสอนข่ายงานประสาทจะได้พื้นผิวตัดสินใจออกมามีลักษณะดังรูปที่ 11a เมื่อพิจารณาให้ดีพบว่าเราไม่จำเป็นต้องใช้ข้อมูลที่มีอยู่ทั้งหมดในการสอนข่ายงานประสาท ถ้าเราเก็บเฉพาะข้อมูลที่อยู่บริเวณขอบของข้อมูลแต่ละประเภทมาสอน ก็ยังคงฝึกได้พื้นผิวตัดสินใจออกมาในลักษณะเดิม ดังรูปที่ 11b



รูปที่ 11 พื้นผิวตัดสินใจที่ฝึกโดยใช้ (a) ข้อมูลฝึกทั้งหมด, (b) ข้อมูลฝึกที่อยู่บริเวณขอบ



รูปที่ 12 ข้อมูลที่อยู่ขอบข้างระหว่างแนวที่พื้นผิวตัดสินใจขาดผ่าน

แต่อันที่จริงเราไม่จำเป็นต้องเก็บข้อมูลบริเวณขอบทั้งหมดมาใช้ในการฝึก เพราะจะเป็นการเก็บข้อมูลมากเกินไป เราจะได้เก็บเฉพาะข้อมูลสำคัญที่เพียงพอต่อการสร้างพื้นผิว

ตัดสินใจ ซึ่งก็คือข้อมูลที่อยู่บนข้างระหว่างแนวที่พื้นผิวตัดสินใจจะพาดผ่านเท่านั้น (รูปที่ 12) ข้อมูลพวกนี้มีคุณสมบัติอย่างหนึ่ง คือในบรรดาข้อมูลที่เป็นประเภทเดียวกันกับมัน จะมีข้อมูลประเภทอื่นอย่างน้อยหนึ่งตัวอยู่ใกล้มันมากที่สุด ซึ่งสามารถเขียนนิยามของ T_i

3.2.2 การลดข้อมูลแบบง่าย

สมมติข้อมูลฝึก T มี m ประเภท กำหนดให้ $T_i = \{x \in T \mid \text{class}(x) = i\}$ การลดแบบง่ายนี้จะพยายามใช้หลักการที่ได้กล่าวไปในหัวข้อที่ 3.2.1 เพื่อเลือกเก็บเฉพาะข้อมูลที่อยู่ตรงขอบ ซึ่งสามารถนิยาม T' ได้ด้วยสมการ (21)

$$T' = \{z \in T \mid (\forall x \in T_{\text{class}(z)}) (\exists y \in T - T_{\text{class}(z)}) d(y, z) \leq d(y, x)\} \quad (21)$$

ขั้นตอนการเลือกเก็บข้อมูลจาก T_i เริ่มจากให้ข้อมูลทุกตัวใน T_j แต่ละตัวมาเลือกข้อมูลใน T_i ที่อยู่ใกล้มันมากที่สุดมาหนึ่งตัว เรียกกระบวนการนี้ว่า “ T_j เลือก T_i ” เราจะเก็บข้อมูลจาก T_i ที่เคยถูกเลือกเอาไว้ และให้ทำกระบวนการเลือกข้อมูลนี้กับข้อมูลประเภทอื่นๆ ทั้งหมดจนหมดทุกคู่

อัลกอริทึมนี้มีการทำงานเป็นแบบไม่ขึ้นกับลำดับ (order-independent) หมายความว่าสำหรับชุดข้อมูลใดๆ ผลลัพธ์จากการลดข้อมูลจะเหมือนกันทุกครั้ง ไม่ว่าจะสลับลำดับการนำข้อมูลไปประมวลผลอย่างไรก็ตาม

อัลกอริทึมแสดงในรูปที่ 13 ภาพประกอบรูปที่ 14 โดยลูกศรแสดงการเลือกข้อมูลตัวที่อยู่ใกล้มากที่สุด

```
procedure SIMPLE_REDUCE( $T_{1..m}$ )
```

```
   $T' \leftarrow \emptyset$ 
```

```
  for  $i \leftarrow 1$  to  $m$  do
```

```
    for  $j \leftarrow 1$  to  $m$ ,  $i \neq j$  do
```

```
      for all  $y \in T_j$  do
```

```
         $z \leftarrow \underset{x \in T_i}{\text{argmin}} d(y, x)$ 
```

```
         $T' \leftarrow T' \cup \{z\}$ 
```

```
      end for
```

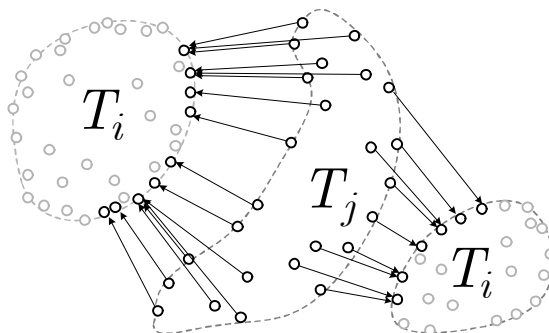
```
    end for
```

```
  end for
```

```
  return  $T'$ 
```

```
end procedure
```

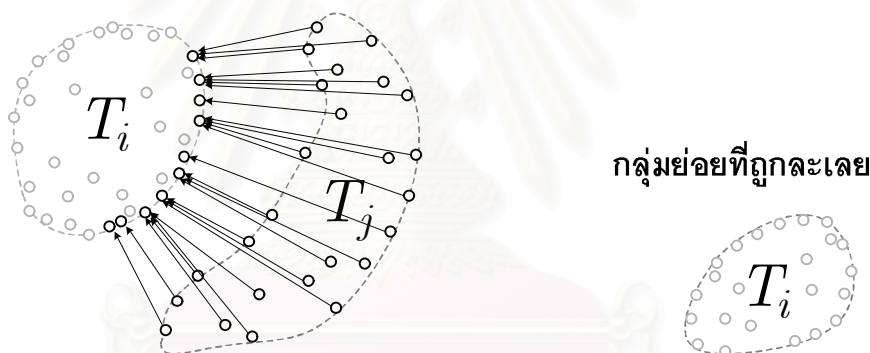
รูปที่ 13 รหัสเทียมอัลกอริทึมลดข้อมูลแบบง่าย



รูปที่ 14 ข้อมูลในกลุ่ม T_j ทุกตัวเลือกเก็บข้อมูลในกลุ่ม T_i

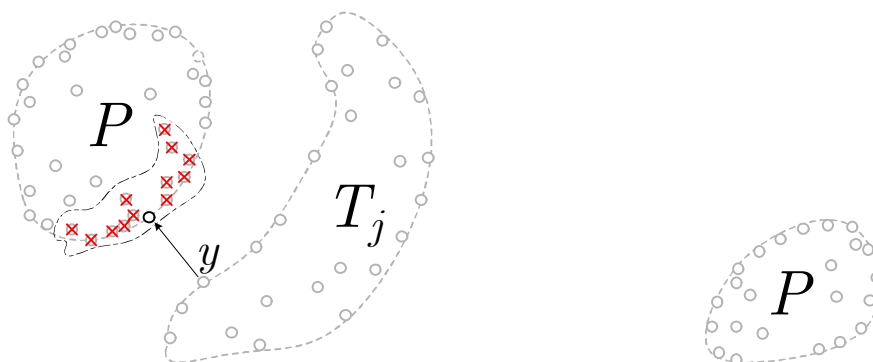
3.2.3 การลดข้อมูลแบบ CE

ในกรณีข้อมูลบางประเภท เช่น T_i มีกลุ่มย่อยซึ่งวางตัวอยู่ห่างมากเกินไป ถ้าใช้วิธีลดข้อมูลแบบง่ายดังที่กล่าวไว้แล้วในหัวข้อ 3.2.2 จะทำให้ข้อมูลในกลุ่มย่อยนั้น ไม่ถูกเลือกเก็บไว้เลย อันเป็นผลให้ข้อมูลที่ลดเสร็จแล้วไม่เป็นตัวแทนที่ดีของข้อมูลก่อนลด (รูปที่ 15)



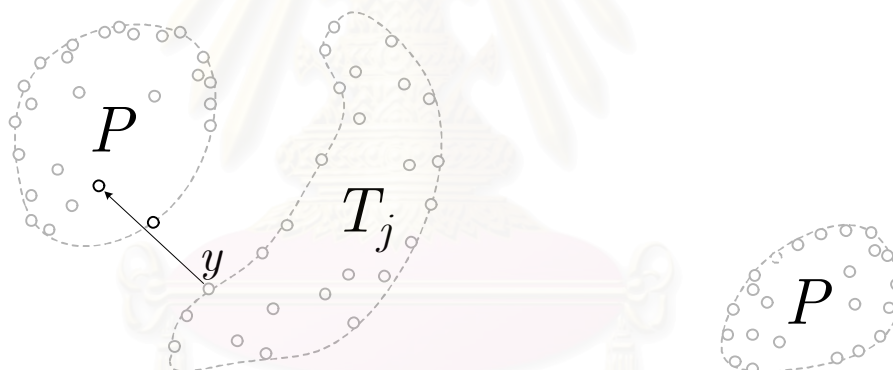
รูปที่ 15 ข้อมูลบางกลุ่มย่อยของ T_i อยู่ห่างเกินไป ทำให้ถูกละเลย

เพื่อแก้ปัญหาข้างต้นจึงได้พัฒนาอัลกอริทึมลดข้อมูลแบบ CE ขึ้น โดยกำหนดให้มีรัศมีระเบิดเท่ากับ r หน่วย เราจะไม่ทำงานบน T_i โดยตรง แต่เราจะทำสำเนา T_i ลงไปเก็บใน P แล้วให้ T_j เลือกข้อมูลจาก P ซึ่งเปรียบเสมือนตัวแทนของ T_i โดยเริ่มจากกำหนดให้ y เป็นข้อมูลตัวแรกใน T_j เลือกเก็บข้อมูลใน P ตัวที่อยู่ใกล้ y มากที่สุดมาหนึ่งตัว (รูปที่ 16) แต่กระบวนการเลือกจะยังไม่จบเพียงเท่านี้ เราจะต้องตัดข้อมูลใน P ทิ้งไปบางส่วน แล้วเลือกข้อมูลตัวที่อยู่ใกล้ y ที่สุดตัวต่อไปอีก ทั้งนี้การจะตัดข้อมูลตัวใดที่นั่นทำตามเงื่อนไขว่าข้อมูลใน P ตัวใดอยู่ห่างจากข้อมูลตัวที่เพิ่งถูกเลือกไม่เกินระยะ r จะถูกตัดทิ้งจาก P ทั้งหมด และข้อมูลทุกตัวที่ถูกตัดออกไปนี้ก็จะทำการระเบิดต่อเนื่องเป็นระเบิดลูกโซ่ เพื่อตัดข้อมูลที่อยู่ห่างมันไม่เกินระยะ r ทิ้งไปเรื่อยๆ จนกว่าจะไม่มีข้อมูลให้ตัดได้อีก (เครื่องหมาย x ในรูปที่ 16 แสดงการตัดข้อมูล)

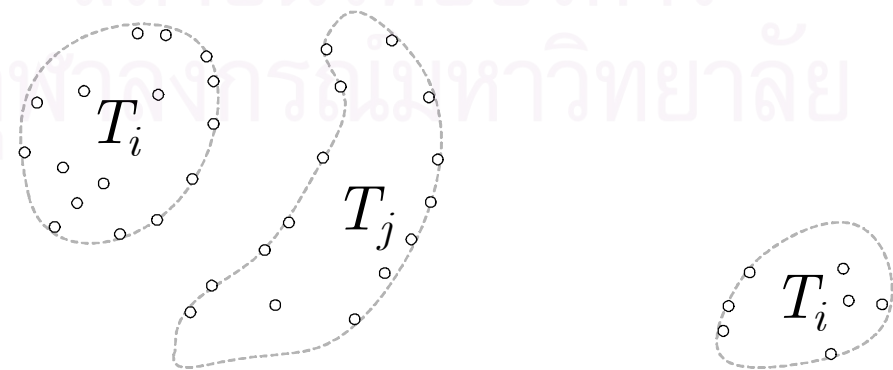


รูปที่ 16 ข้อมูล y เลือกข้อมูลที่ใกล้ที่สุดจาก P (T_j) แล้วระเบิดข้อมูลที่อยู่ภายใต้รัศมี r ที่

จะเห็นได้ว่าการระเบิดข้อมูลในแต่ละครั้งย่อมจะไม่กินอาณาบริเวณเข้าไปยังกลุ่มย่อยที่อยู่ห่างมากกว่ารัศมี r จากนั้นให้ y เลือกข้อมูลที่ใกล้ที่สุดใน P ต่อไป (รูปที่ 17) สลับไปกับการระเบิดข้อมูลทั้งจนกว่าจะไม่เหลือข้อมูลใน P อีก การระเบิดข้อมูลทั้งและเลือกเก็บตัวใกล้ที่สุดสลับกันไปนี้จะทำให้ข้อมูลกลุ่มย่อยที่อยู่ห่างไกลมีโอกาสถูกเลือกเก็บได้ และหลังจากนั้นจึงให้ข้อมูลตัวอื่นๆ ใน T_j ทุกตัวมาเลือกข้อมูลใน P ที่ถูกสร้างขึ้นมาใหม่ และทำเช่นนี้กับข้อมูลประเภทอื่นๆ จนครบทุกประเภท (รูปที่ 18) อัลกอริทึมแสดงในรูปที่ 19



รูปที่ 17 ข้อมูล y เลือกข้อมูลที่ใกล้ที่สุดจาก P (T_j) ตัวถัดไป แต่ไม่มีข้อมูลตัวอื่นอยู่ภายใต้รัศมี จึงตัดข้อมูลตัวใดไม่ได้เลยในรอบนี้



รูปที่ 18 ข้อมูลสุดท้ายที่เหลือหลังจากลดข้อมูลแบบ CE

เพื่อจะวิเคราะห์ความซับซ้อนเชิงเวลาของอัลกอริทึม CE กำหนดให้ n คือจำนวนข้อมูลเฉลี่ยในแต่ละประเภท การทำงานใน Loop A เริ่มจากเลือกข้อมูลตัวใกล้สุดมาเก็บไว้ และเข้าสู่การระเบิดข้อมูลทิ้งด้วย Loop B พบว่าเวลาที่ใช้ในการทำงานส่วนใหญ่จะถูกใช้ไปในการคำนวณระยะห่างระหว่างข้อมูลที่เป็นประเภทเดียวกันเพื่อระเบิดข้อมูลที่อยู่ห่างไม่เกินรัศมี r ทั้งกระบวนการระเบิดข้อมูลทิ้งนี้จะต้องทำไปจนกว่าข้อมูลประเภทนั้นจะหมด กล่าวคือทำงานกว่า P จะกลายเป็นเซตว่าง ซึ่งจะต้องคำนวณระยะห่างระหว่างข้อมูลในประเภทเดียวกันไม่เกิน $n(n-1)/2$ ครั้ง (ข้อมูลที่ถูกต้องออกจาก P แล้วไม่ต้องคำนวณระยะห่างเทียบกับอีกจึงทำให้ไม่ต้องคำนวณ $n(n-1)/2$ ครั้งเสมอไป) ซึ่งคิดเป็น $O(n^2)$ และต้องทำเช่นนี้กับข้อมูลทุกประเภทและทุกตัว $O(m^2n)$ ดังนั้นอัลกอริทึมนี้จึงใช้เวลาทำงานเป็น $O(m^2n^3)$

แต่ในกรณีเร็วสุดคือ r ใหญ่มากจนข้อมูลถูกตัดทิ้งหมดในการระเบิดเพียงครั้งเดียว Loop A จะคำนวณระยะห่างระหว่างข้อมูลเพียง n ครั้งเท่านั้น ความซับซ้อนเชิงเวลาในกรณีนี้จะเหลือเพียง $O(m^2n^2)$

```

procedure CE_REDUCE( $T_{1..m}, r$ )
   $T' \leftarrow \emptyset$ 
  for  $i \leftarrow 1$  to  $m$  do
    for  $j \leftarrow 1$  to  $m, i \neq j$  do
      for all  $y \in T_j$  do
         $P \leftarrow T_i$ 
        while  $P \neq \emptyset$  do /* Loop A */
           $z \leftarrow \underset{p \in P}{\operatorname{arg\,min}} d(y, p)$ 
           $T' \leftarrow T' \cup \{z\}$ 
          /* begin chain explosion */
           $S \leftarrow \{z\}$ 
          while  $S \neq \emptyset$  do /* Loop B */
             $P \leftarrow P - S$ 
             $S \leftarrow \{p \in P \mid (\exists s \in S) d(p, s) \leq r\}$ 
          end while
        end while
      end for
    end for
  end for
  return  $T'$ 
end procedure

```

3.2.4 ทดสอบวิธีการลดข้อมูลแบบ CE บนข้อมูลทดสอบ P2

เราทำการทดลองเบื้องต้นบนข้อมูลทดสอบ P2 [16] ซึ่งเป็นชุดข้อมูลสังเคราะห์ที่มีคุณสมบัติ 2 ตัว และมี 2 ประเภท (รูปที่ 20b) โดยข้อมูล P2 นี้สร้างขึ้นโดยใช้การตัดกันของเส้นกราฟ 4 เส้น (รูปที่ 20a) มีข้อมูลฝึก 800 ตัว ข้อมูลทดสอบ 200 ตัว

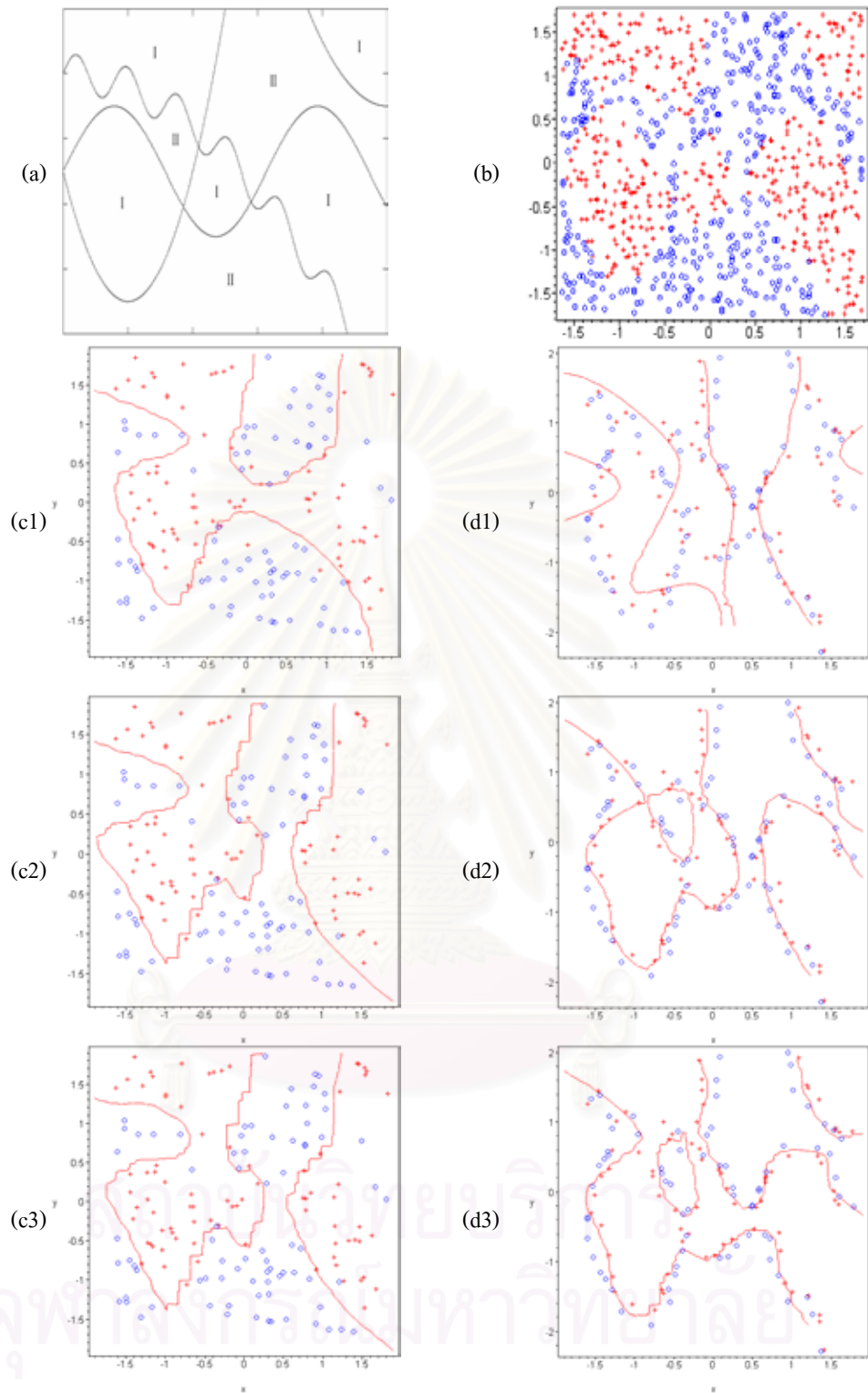
เรานำข้อมูลฝึก P2 ไปลดด้วยวิธี CE โดยใช้ค่า $r = 20$ ซึ่งลดแล้วเหลือข้อมูล 150 ตัว เปรียบเทียบกับวิธีลดข้อมูลแบบสุ่มที่ลดแล้วให้เหลือข้อมูล 150 ตัวเช่นกัน หลังจากนั้นนำข้อมูลที่ลดแล้วด้วยสองวิธีนี้ปรับให้อยู่ในช่วง 0 ถึง 1 แล้วนำไปฝึกข่ายงานประสาทโดยใช้โหนดซ่อนจำนวน 10, 50 และ 100 โหนด

รูปที่ 20 (c1-c3) และรูปที่ 20 (d1-d3) เป็นพื้นผิวตัดสินใจที่ได้หลังจากนำข้อมูลไปฝึกข่ายงานประสาทที่มีโหนดซ่อนจำนวน 10, 50 และ 100 โหนด ตามลำดับ โดยรูปที่ 20c เป็นพื้นผิวตัดสินใจที่ฝึกด้วยข้อมูลที่ลดด้วยวิธีสุ่ม ส่วนรูปที่ 20d เป็นพื้นผิวตัดสินใจที่ฝึกด้วยข้อมูลที่ลดด้วยวิธี CE ซึ่งจากรูปจะเห็นได้ว่าการลดข้อมูลด้วยวิธี CE จะเหลือเฉพาะข้อมูลตรงขอบของแต่ละประเภท ซึ่งถ้านำไปฝึกให้ได้พื้นผิวตัดสินใจที่แทรกตัวเข้ามาแบ่งระหว่างขอบนี้ ก็น่าจะให้ความถูกต้องอย่างดีเยี่ยม

ตารางที่ 1 ความถูกต้องในการจำแนกประเภทชุดข้อมูล P2

	จำนวนโหนดซ่อน		
	10	50	100
ข้อมูลฝึกที่ลดแบบสุ่ม	83.0	83.5	85.5
ข้อมูลฝึกที่ลดแบบ CE	72.5	86.5	90.5

ตารางที่ 1 แสดงให้เห็นว่าเมื่อนำข้อมูลที่ลดด้วยวิธีสุ่มไปฝึกข่ายงานประสาทที่แม้จะมีโหนดซ่อนน้อยก็ให้ความถูกต้องที่ดีแล้ว แต่กระนั้นการเพิ่มโหนดซ่อนให้มากขึ้นก็ไม่ได้ช่วยทำให้ความถูกต้องสูงเพิ่มขึ้นอย่างชัดเจนนัก เนื่องจากการเพิ่มโหนดซ่อนให้มากขึ้น ไม่ได้ทำให้พื้นผิวตัดสินใจมีการพัฒนาแตกต่างจากเดิมเท่าไรนัก (รูปที่ 20c) ในขณะที่เมื่อนำข้อมูลที่ลดแบบ CE ไปฝึกข่ายงานประสาทที่มีโหนดซ่อนน้อย จะยังไม่ให้ความถูกต้องสูง แต่เมื่อเพิ่มจำนวนโหนดซ่อนให้มากขึ้นๆ ความถูกต้องก็เพิ่มขึ้นเป็นอย่างมาก (รูปที่ 20d) เนื่องจากการเพิ่มโหนดซ่อนให้มากขึ้นจะทำให้พื้นผิวตัดสินใจมีการพัฒนารูปร่างให้สอดแทรกไประหว่างช่องว่างระหว่างประเภทข้อมูลได้ดีขึ้นอย่างชัดเจน ซึ่งทำให้ได้พื้นผิวตัดสินใจที่มีความถูกต้องสูงตามไปด้วย



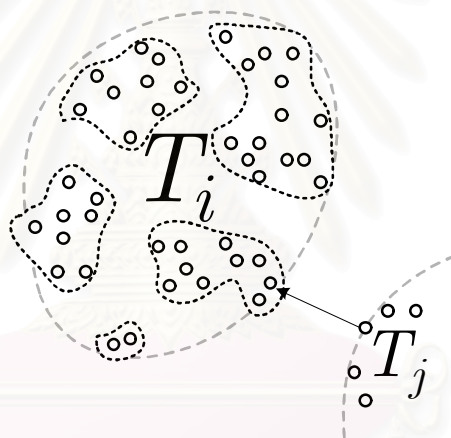
รูปที่ 20 (a) เส้นกราฟที่ใช้แบ่งพื้นที่เพื่อสร้าง P2, (b) ลักษณะของชุดข้อมูล P2 โดยมี o และ + เป็นสัญลักษณ์แทนข้อมูลในแต่ละประเภท, (c1-c3) พื้นผิวตัดสินใจที่ฝึกด้วยข้อมูลที่ลดด้วยวิธีสุ่ม, (d1-d3) พื้นผิวตัดสินใจที่ฝึกด้วยข้อมูลที่ลดด้วยวิธี CE

3.3 การลดข้อมูลแบบระเบิดลูกโซ่ขั้นสูง (Advanced Chain-Explosion Data Reduction – ACE Data Reduction)

วิธีลดข้อมูลแบบ CE พบว่า y แต่ละตัวจะใช้เวลาในการเลือกข้อมูลจาก T_i เป็น $O(n^2)$ ซึ่งจริงๆ แล้วเราสามารถลดความซับซ้อนเชิงเวลาลงได้อีกโดยใช้แนวคิดของ $rCluster$ ซึ่งสามารถบรรยายได้ดังสมการที่ (22)

$$rCluster(x) = \{x\} \cup \left\{ y \in T_{class(x)} \mid (\exists s \in rCluster(x)) d(y,s) \leq r \right\} \quad (22)$$

$rCluster(x)$ เป็นข้อมูลทุกตัวที่จะถูกระเบิดไปพร้อมๆ กันถ้าเริ่มระเบิดลูกโซ่ที่ x ด้วยรัศมีการระเบิด r (รูปที่ 21) และไม่ว่าจะเริ่มระเบิดจากข้อมูลตัวใดก็ตามใน $rCluster(x)$ ก็จะทำให้ข้อมูลตัวอื่นๆ ใน $rCluster(x)$ ถูกระเบิดทิ้งไปพร้อมๆ กันด้วย ดังนั้นถ้าเราทำการประมวลผลไว้ล่วงหน้า (preprocess) กับข้อมูลทุกตัว ว่าข้อมูลแต่ละตัวเป็นสมาชิกของ $rCluster$ ใด ก็จะทำให้กระบวนการระเบิดข้อมูลทำงานได้ในเวลา $O(n)$



รูปที่ 21 $rClusters$ ของ T_i

อัลกอริทึมนี้เรียกว่าการลดข้อมูลแบบ ACE รหัสเทียมอยู่ในรูปที่ 22 การทำงานเริ่มจากการประมวลผลล่วงหน้าเพื่อหา $rCluster$ ของข้อมูลแต่ละประเภท ซึ่งการทำงานในส่วนนี้สามารถทำได้โดยใช้ดิสจอยท์เซต (disjoint sets) การจะสร้างดิสจอยท์เซตของ $rCluster$ ทั้งหมดใช้เวลา $O(mn^2)$

หลังจากนั้นอัลกอริทึมจะทำการลดข้อมูลโดยการให้ y เลือกข้อมูล x ตัวที่ใหญ่ที่สุดจาก T_i ขึ้นมาทีละตัว ซึ่งจะตอบได้ในเวลา $O(1)$ ว่าข้อมูล x ตัวนี้เป็นสมาชิกใน $rCluster$ ใด และเพียงแค่เลือกข้อมูลหนึ่งตัวที่อยู่ใกล้ y มากที่สุดจากแต่ละ $rCluster$ ก็จะเทียบเท่ากับภาระงานของการให้ y เลือกข้อมูลจาก T_i ซึ่ง y แต่ละตัวจะทำงานนี้ได้เสร็จในเวลา $O(n)$ เป็นผลให้วิธีการลดข้อมูลแบบ ACE ใช้เวลาโดยรวมเป็น $O(m^2n^2)$ เร็วกว่าการลดแบบ CE ซึ่งใช้เวลา $O(m^2n^3)$


```

procedure ACE_REDUCE( $T_{1..m}, r$ )
    /* construct disjoint sets */
    ( $\forall x \in T_i$ )  $rCluster(x) \leftarrow \{x\}$ 
    for  $i \leftarrow 1$  to  $m$  do
        for all  $x \in T_i$  do
            for all  $y \in T_i, x \neq y$  do
                if  $d(x,y) \leq r$  then
                     $merge(rCluster(x), rCluster(y))$ 
                end if
            end for
        end for
    end for
     $T' \leftarrow \emptyset$ 
    for  $i \leftarrow 1$  to  $m$  do
        for  $j \leftarrow 1$  to  $m, i \neq j$  do
            for all  $y \in T_j$  do
                /* Loop A */
                for all  $C \leftarrow rClusters\ of\ T_i$  do
                     $T' \leftarrow T' \cup \underset{x \in C}{\arg\ min}\ d(x, y)$ 
                end for
            end for
        end for
    end for
    return  $T'$ 
end procedure

```

รูปที่ 22 รหัสเทียมอัลกอริทึมลดข้อมูลแบบ ACE

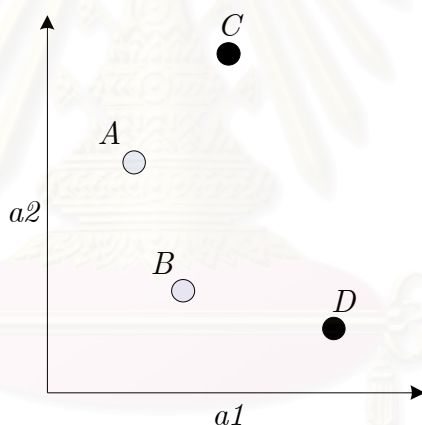
สำหรับการลดข้อมูลแบบ CE และ ACE ผู้ใช้จะควบคุมอัตราการลดข้อมูลไม่ได้โดยตรงว่าจะให้เหลือข้อมูลกี่เปอร์เซ็นต์ เพราะต้องใช้ในการกำหนดค่ารัศมีการระเบิด r แต่ทั้งนี้ทั้งนั้นในเบื้องต้นผู้วิจัยแนะนำว่าค่า r ที่ใช้ควรจะอยู่ในช่วง $0 < r < \sqrt{\sum_{i=1}^{n_a} (\max x_i - \min x_i)^2}$

3.4 การใช้อัลกอริทึม ReliefF ช่วยในการลดข้อมูลแบบ CE

หลังจากผู้วิจัยนำอัลกอริทึม CE ไปลดข้อมูลแล้วนำไปฝึกข่ายงานประสาทแล้ว พบว่า ความถูกต้องในการจำแนกประเภทสูงกว่าการลดแบบสุ่มเกือบๆ ทุกอัตราการลดข้อมูล (ผลการทดลองแสดงไว้ในบทที่ 5) แต่พบว่าอัลกอริทึม CE นี้มีข้อจำกัดในประเด็นที่ไม่สามารถควบคุมอัตราการลดข้อมูลได้โดยตรง เนื่องจากต้องใช้การปรับค่ารัศมีการระเบิด r และในบางชุดข้อมูล การลดแบบ CE ก็ลดข้อมูลได้ไม่มากนัก ผู้วิจัยจึงได้พยายามหาวิธีปรับปรุงอัลกอริทึมนี้ให้ลดข้อมูลได้ในปริมาณมากขึ้น โดยใช้อัลกอริทึม ReliefF เข้าช่วย

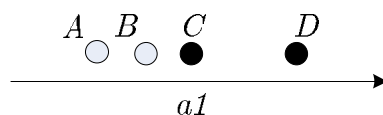
3.4.1 แรงจูงใจ

สาเหตุที่ชุดข้อมูลบางชุดลดจำนวนข้อมูลลงได้ไม่มากนัก เป็นเพราะมีจำนวนประเภทข้อมูลมากทำให้ต้องเลือกเก็บข้อมูลตรงขอบไว้มาก เนื่องจากต้องพิจารณาหาขอบของข้อมูลแต่ละประเภทเทียบกันทั้งหมด หรือเป็นเพราะมีจำนวนคุณสมบัติมาก การที่มีจำนวนคุณสมบัติมากจะทำให้อัลกอริทึมลดข้อมูลแบบ CE เลือกเก็บข้อมูลไว้มากเกินความจำเป็น



รูปที่ 23 ข้อมูลที่มีคุณสมบัติ $a1$ และ $a2$

รูปที่ 23 แสดงให้เห็นว่า A จะเลือกเก็บ C ซึ่งเป็นข้อมูลที่อยู่ใกล้กันมากที่สุด และ B จะเลือกเก็บ D ซึ่งเป็นข้อมูลที่อยู่ใกล้กันมากที่สุดเช่นกัน แต่หากเราเชื่อว่าคุณสมบัติที่มีอยู่ทั้งหมด อาจจะไม่ใช่ว่าคุณสมบัติที่ควรนำมาฝึกทั้งหมดก็เป็นได้ กล่าวอีกนัยหนึ่งว่าบางคุณสมบัติอาจจะมีอิทธิพลในการจำแนกประเภทต่ำมาก ซึ่งสมควรตัดทิ้งได้ หากเป็นเช่นนี้ถ้าเราตัดคุณสมบัติที่ไม่จำเป็นหรือมีอิทธิพลน้อยต่อการจำแนกประเภททิ้งไป แล้วจึงค่อยนำข้อมูลที่ได้มาลดด้วย CE ก็น่าจะทำให้ลดจำนวนข้อมูลลงได้อีกมาก ในกรณีนี้สมมติว่าถ้าคุณสมบัติ $a2$ ไม่ใช่คุณสมบัติที่ดีต่อการจำแนกประเภท เมื่อเราตัดทิ้งให้เหลือแต่คุณสมบัติ $a1$ ดังแสดงในรูปที่ 24 ซึ่งจะทำให้ทั้ง A และ B เลือกเก็บ C เพียงตัวเดียว จึงทำให้ลดข้อมูลได้มากขึ้นนั่นเอง



รูปที่ 24 ข้อมูลที่ตัดคุณสมบัติ a_2 ทิ้งแล้ว

ในการทดลองลดข้อมูลแบบ CE ที่ลดจำนวนคุณสมบัติลงด้วย ReliefF นี้ จะเริ่มจากนำข้อมูลฝึกมาลดจำนวนคุณสมบัติลงด้วยวิธี ReliefF ให้เหลือจำนวนคุณสมบัติที่ดีที่สุดประมาณ 80% ของจำนวนคุณสมบัติที่มีอยู่ทั้งหมด โดยใช้สมมติฐานว่าคะแนนของคุณสมบัติมีการกระจายแบบปกติ ดังนั้นเมื่อนำคะแนนของคุณสมบัติแต่ละค่าที่ ReliefF คำนวณได้มาทำให้เป็นค่าปกติ เราจะได้เลือกเก็บเฉพาะคุณสมบัติที่มีค่าปกติของคะแนนเกิน -0.85 เท่านั้น (พื้นที่ใต้กราฟโค้งปกติตั้งแต่ค่า -0.85 ขึ้นไปถึงอนันต์จะมีพื้นที่ประมาณ 80% ของพื้นที่ใต้กราฟทั้งหมด)

หลังจากลดจำนวนคุณสมบัติด้วย ReliefF แล้ว ก็นำข้อมูลที่ได้ไปลดด้วยวิธี CE แต่เราจะไม่นำข้อมูลสุดท้ายที่ได้จากการลดด้วยวิธี CE ไปฝึกข่ายงานประสาทโดยตรง เราจะฝึกข่ายงานประสาทโดยใช้ข้อมูลตัวเดิมก่อนที่จะลดคุณสมบัติ ดังนั้นในหลักการสรุปได้ว่าข้อมูลที่ลดคุณสมบัติแล้วเมื่อนำไปลดข้อมูลลงด้วยวิธี CE สิ่งที่ได้ก็คือหมายเลขของข้อมูลที่จะนำไปใช้ฝึกจริงเท่านั้น ซึ่งเราจะต้องไปดึงข้อมูลจากข้อมูลฝึกที่ยังไม่ได้ลดคุณสมบัติ โดยดึงมาตามลำดับหมายเลขที่ได้จากการลดแบบ CE นี้

บทที่ 4

การทดลอง

เราจะทำการทดลองเพื่อเปรียบเทียบประสิทธิภาพในสองประเด็นคือ

1. ความแม่นยำในการจำแนกประเภทข้อมูลของข่ายงานประสาท ที่ฝึกมาจากข้อมูลที่ลดด้วยวิธีการต่างๆ อันได้แก่ การลดข้อมูลแบบสุ่ม, แบบสุ่มตามความหนาแน่นแบบ CE และแบบ CE ที่ใช้อัลกอริทึม ReliefF ช่วยลดคุณสมบัติก่อน
2. ความเร็วในการลดข้อมูลแบบ CE กับ ACE

4.1 ข้อมูลที่ใช้ในการทดลอง

ข้อมูลที่ใช้ในการทดลองนำมาจาก UCI Repository of Machine Learning Database [4] และใช้วิธีการตรวจสอบไขว้ 5 พับ (5-fold cross validation) รายละเอียดแสดงไว้ในตารางที่ 2

ตารางที่ 2 ชุดข้อมูลที่ใช้ในการทดลอง

ชุดข้อมูล	#ข้อมูล	#คุณสมบัติ	#ประเภท
iris	150	4	3
wine	178	13	3
glass	214	9	6
new-thyroid	215	5	3
ionosphere	351	33	2
kr-vs-kp	500	39	2
vehicle	846	18	4
vowel	990	10	11
segmentation	2,310	19	7

4.2 การจัดเตรียมข้อมูลเพื่อฝึกข่ายงานประสาทเทียม

ข้อมูลที่ใช้ในการทดลองจะถูกแบ่งเป็น 5 พับ (fold) เพื่อแบ่งข้อมูลเป็นกลุ่มฝึกและกลุ่มทดสอบ ตามวิธีการตรวจสอบไขว้ 5 พับ ข้อมูลที่จะเอาไปลดด้วยวิธีการต่างๆ จะถูกนอร์มอลไลซ์ (normalize) ให้ค่าคุณสมบัติแต่ละค่ามีการกระจายตัวแบบปกติ (normal distribution)

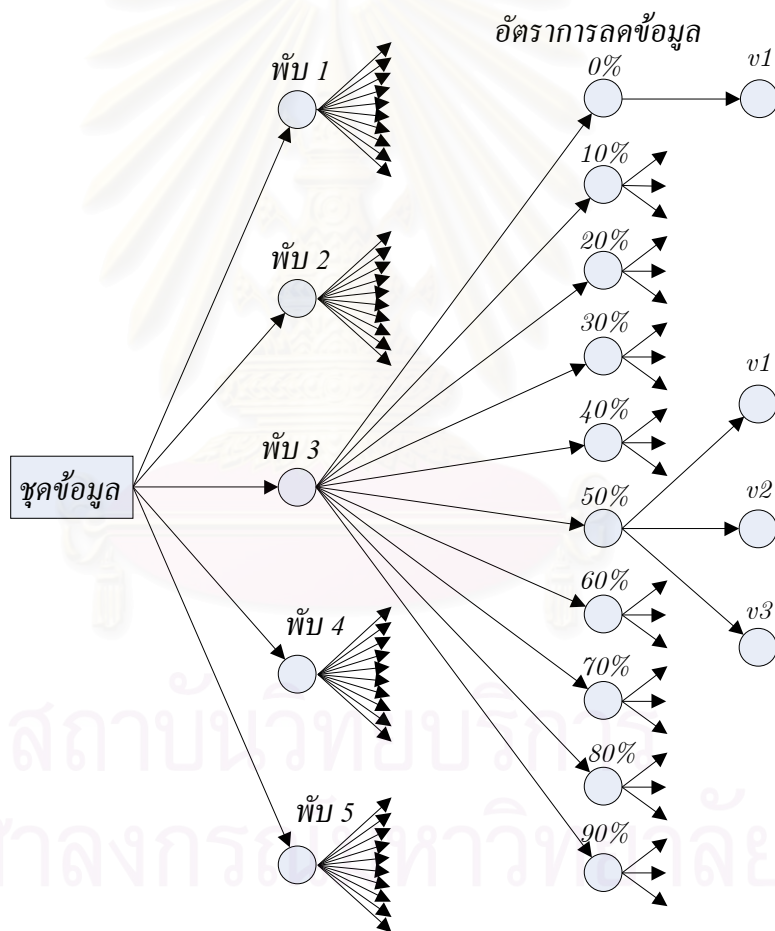
4.2.1 การลดแบบสุ่ม

เราเตรียมข้อมูลฝึกที่ลดด้วยวิธีสุ่มในหลายๆ อัตราการลด (reduction rate) ตั้งแต่ 0% 10% 20% ไปจนถึง 90% แต่เนื่องจากการลดข้อมูล จะให้ผลการลดข้อมูลแต่ละครั้งไม่เหมือนกัน ดังนั้น

เราจึงลดข้อมูลสำหรับทุกๆ อัตราการลดซ้ำอีก 3 ครั้ง จะได้ข้อมูลที่ลดแล้วเป็น 3 ชุดข้อมูลย่อย v_1, v_2 และ v_3 แล้วสุดท้ายจะใช้ค่าเฉลี่ยความถูกต้องในการจำแนกประเภทเป็นตัวแทนเพื่อเปรียบเทียบประสิทธิภาพต่อไป

4.2.2 การลดแบบสุ่มตามความหนาแน่น

เราเตรียมข้อมูลฝึกที่ลดด้วยวิธีสุ่มตามความหนาแน่นด้วยอัตราการลดตั้งแต่ 0% 10% ไปจนถึง 90% เช่นกัน และทุกๆ อัตราการลดก็ทำซ้ำอีก 3 ครั้งแยกเป็น 3 ชุดข้อมูลย่อย ด้วยเหตุผลเดียวกันกับการเตรียมข้อมูลฝึกที่ลดด้วยวิธีสุ่ม แต่สิ่งที่เพิ่มขึ้นมาคือต้องกำหนดค่า k ที่จะใช้ด้วย ซึ่งในการทดลองนี้เรากำหนดให้ $k = 10$ สำหรับทุกๆ ชุดข้อมูล ใช้ระยะห่างแบบยูคลิดเลียนตามสมการที่ (13) และให้คะแนนตามระยะห่างของข้อมูล ตามสมการที่ (16)



รูปที่ 25 ขั้นตอนการจัดเตรียมข้อมูลที่ลดด้วยวิธีสุ่ม และสุ่มตามความหนาแน่น

สำหรับการลดข้อมูลแบบสุ่ม และสุ่มตามความหนาแน่นมีขั้นตอนการจัดเตรียมข้อมูลดังที่แสดงไว้ในรูปที่ 25 กล่าวโดยสรุปคือเริ่มจากแบ่งข้อมูลออกเป็น 5 พับ ลดข้อมูลทุกพับด้วยอัตราการลดที่ 10% ไปจนถึง 90% โดยทำการลดสำหรับทุกๆ อัตราการลดซ้ำ 3 ครั้ง จะได้ชุดข้อมูลย่อย v_1, v_2 และ v_3 เพื่อนำไปฝึกข่ายงานประสาทต่อไป

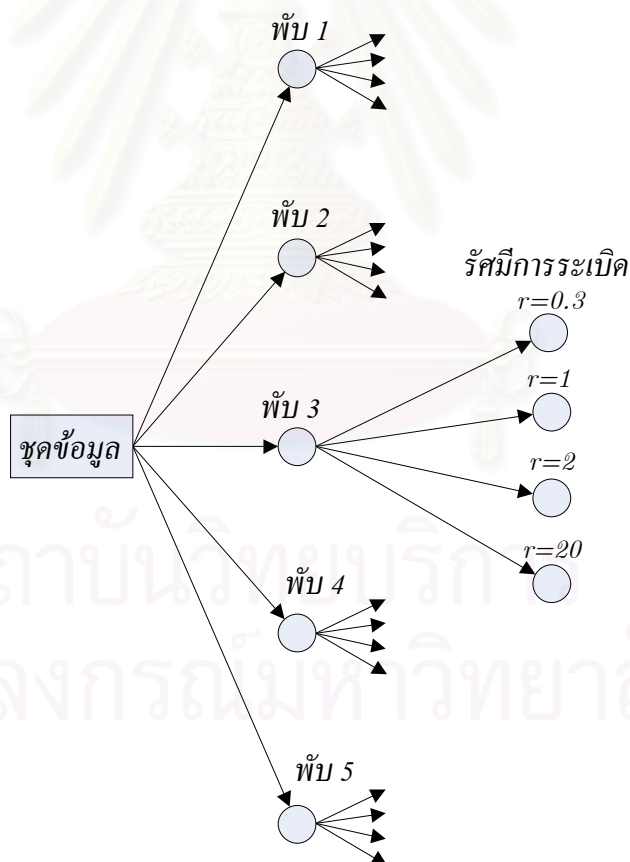
4.2.3 การลดแบบ CE

เราเตรียมข้อมูลฝึกโดยใช้รัศมีระเบิด 4 ค่า ได้แก่ $r = \{0.3, 1, 2, 20\}$ ยกเว้นบางชุดข้อมูลจะใช้ r ค่าอื่นด้วย เพื่อให้กราฟแสดงการเปรียบเทียบความถูกต้องในการจำแนกประเภทที่มีความชัดเจนและครอบคลุมมากที่สุด

4.2.4 การลดแบบ CE ที่ลดคุณสมบัติด้วยวิธี ReliefF

ขั้นตอนหลักๆ ก็ เราจะนำข้อมูลฝึกมาลดจำนวนคุณสมบัติลงก่อน ด้วยวิธี ReliefF แล้วนำข้อมูลที่ลดคุณสมบัติแล้วนี้ไปลดด้วยวิธี CE โดยใช้ค่า $r = \{0.3, 1, 2, 20\}$ ยกเว้นบางชุดข้อมูลจะใช้ r ค่าอื่นด้วย

การลดคุณสมบัติด้วยวิธี ReliefF มีพารามิเตอร์ที่ต้องกำหนดคือ จำนวนข้อมูลที่จะสุ่มมาเพื่อวิเคราะห์ค่าคุณสมบัติ ในการทดลองนี้เราจะไม่ใช้การสุ่ม แต่จะใช้ข้อมูลฝึกที่มีอยู่ทุกตัวเพื่อวิเคราะห์คุณสมบัติ และใช้ค่า $k = 10$ ในการเลือกตัวใกล้สุดถูกและตัวใกล้สุดผิด



รูปที่ 26 แสดงขั้นตอนการจัดเตรียมข้อมูลที่ลดด้วยวิธี CE และ CE ที่ลดคุณสมบัติด้วย ReliefF

สำหรับการลดข้อมูลแบบ CE และ CE ที่ลดคุณสมบัติลงด้วย ReliefF แสดงไว้ในรูปที่ 26 ซึ่งแต่ละค่ารัศมีการระเบิดที่ใช้ไม่ต้องทำซ้ำ 3 ครั้ง ดังนั้นจึงมีชุดข้อมูลย่อยที่ต้องนำไปฝึกข่ายงานประสาทน้อยกว่าการทดลองแบบสุ่ม

จากการทดลองใช้ ReliefF กับชุดข้อมูลทั้งหมดแล้ว ชุดข้อมูลแต่ละชุดจะเหลือจำนวนคุณสมบัติดังแสดงในตารางที่ 3

ตารางที่ 3 จำนวนคุณสมบัติที่เลือกเก็บไว้ของชุดข้อมูลต่างๆ

ชุดข้อมูล	#คุณสมบัติ	#คุณสมบัติที่เลือกไว้	%คุณสมบัติที่เลือกไว้
iris	4	3	75.0%
wine	13	10	76.9%
glass	9	8	88.9%
new-thyroid	5	4	80.0%
ionosphere	33	26	78.8%
kr-vs-kp	39	29	74.4%
vehicle	18	13	72.2%
vowel	10	7	70.0%
segmentation	19	12	63.2%

4.3 การฝึกข่ายงานประสาท

ตารางที่ 4 ค่าอัตราการเรียนรู้และค่าโมเมนตัมที่ใช้ในการฝึก

ชุดข้อมูล	อัตราการเรียนรู้	ค่าโมเมนตัม
iris	0.05	0.3
wine	0.1	0.3
glass	0.05	0.3
new-thyroid	0.05	0.3
ionosphere	0.01	0.1
kr-vs-kp	0.05	0.3
vehicle	0.01	0.7
vowel	0.05	0.7
segmentation	0.05	0.3

เมื่อลดข้อมูลเสร็จแล้วก็นำข้อมูลที่ลดแล้วนี้ไปฝึกข่ายงานประสาท โดยจะปรับค่าจำนวนโหนดซ่อนหลายๆ ค่า ($h_1 \dots h_4$) ให้แปรไปตามจำนวนคุณสมบัติของข้อมูล (a) อันได้แก่ $h = \{a/2, a, 2a, 4a\}$ ส่วนค่าอัตราการเรียนรู้ (learning rate) และค่าโมเมนตัมจะใช้ค่าตามที่

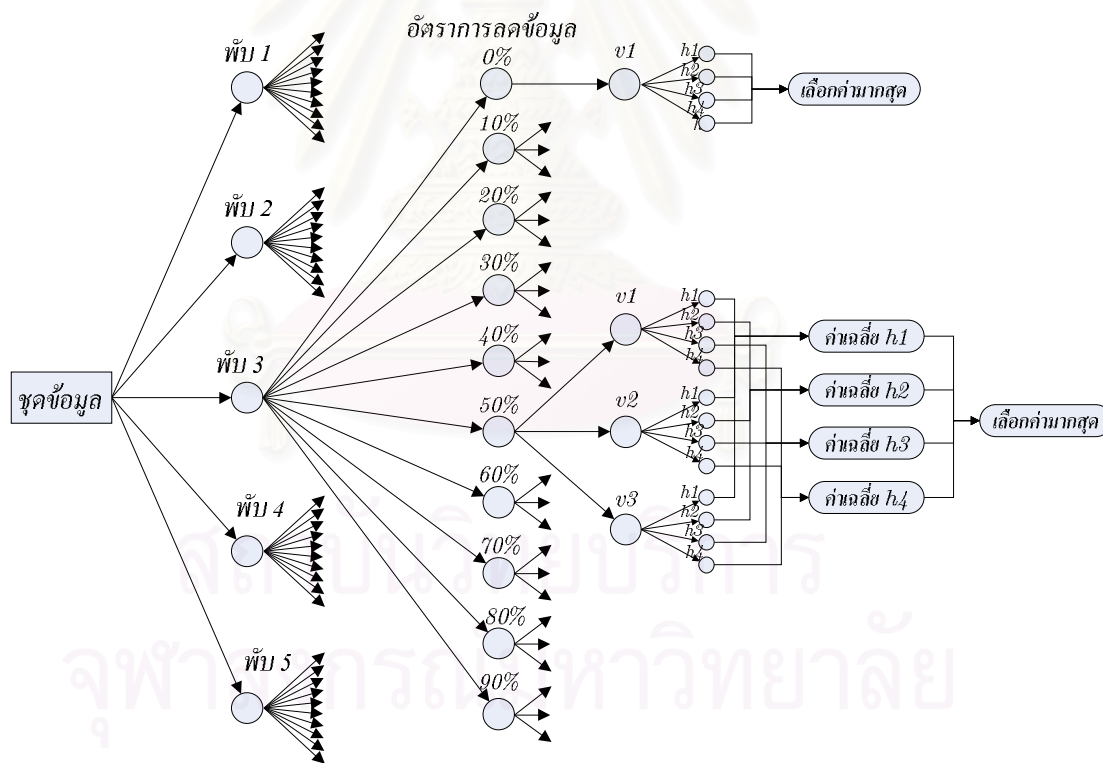
แสดงไว้ในตารางที่ 4 ซึ่งเป็นค่าที่ได้มาจากการทดลองเบื้องต้นเพื่อหาค่าพารามิเตอร์ที่ให้ผลดีสำหรับฝึกชุดข้อมูลแต่ละชุด

4.4 การเปรียบเทียบประสิทธิภาพของการลดข้อมูล

เราจะเปรียบเทียบประสิทธิภาพของวิธีการลดข้อมูลแต่ละวิธีโดยพิจารณาว่า วิธีการลดข้อมูลแบบใดให้ข้อมูลที่นำไปฝึกข่ายงานประสาทแล้ว ได้พื้นผิวตัดสินใจที่ให้ความถูกต้องในการจำแนกประเภทข้อมูลสูงที่สุด ทั้งนี้การเปรียบเทียบจะต้องกระทำบนเงื่อนไขที่ว่าจำนวนข้อมูลที่นำไปฝึกข่ายงานประสาทจะต้องมีเท่าๆ กัน

สิ่งที่ต้องกำหนดให้ชัดเจนคือ ค่าตัวแทนของความถูกต้องในการจำแนกประเภทของวิธีการลดข้อมูลแต่ละวิธี ที่จะนำมาเขียนกราฟเปรียบเทียบ จะใช้วิธีการคำนวณอย่างไรเพื่อให้ยุติธรรมในการเปรียบเทียบ

4.4.1 การลดแบบสุ่ม และการลดแบบสุ่มตามความหนาแน่น



รูปที่ 27 ขั้นตอนการฝึกและการคำนวณค่าความถูกต้องในการจำแนกประเภทของข้อมูลฝึกที่ลดด้วยวิธีสุ่ม และวิธีสุ่มตามความหนาแน่น

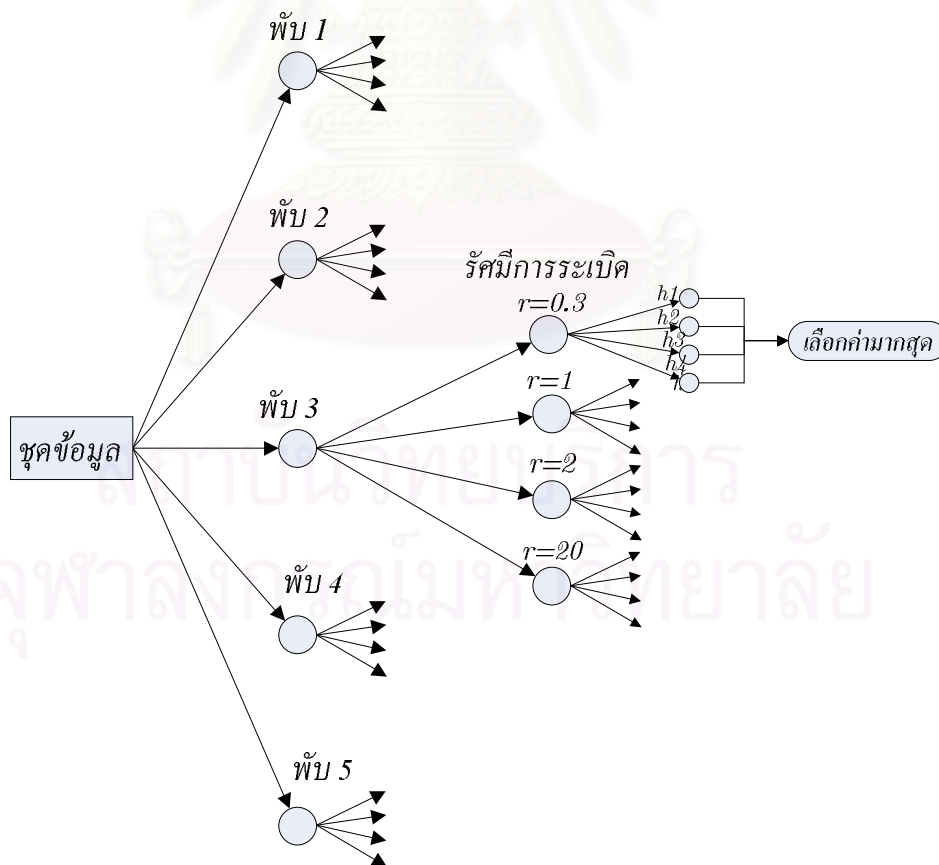
ขั้นตอนการฝึกและการคำนวณความถูกต้องในการจำแนกประเภทแสดงไว้ในรูปที่ 27 เมื่อฝึกข่ายงานประสาทด้วยชุดข้อมูลย่อย v_1, v_2 และ v_3 บนข่ายงานประสาทที่มีจำนวนโหนดซ่อน h_1 ถึง h_4 นอกจากนี้สำหรับค่าของจำนวนโหนดซ่อนแต่ละค่า เราจะทดลองซ้ำ 8 ครั้ง แล้วนำ

ความถูกต้องทั้งหมดที่ได้จากชุดข้อมูลย่อยทุกชุดมาเฉลี่ยกัน แล้วจึงเลือกค่าสูงสุดจากค่าเฉลี่ยที่ได้ เพื่อเป็นตัวแทนความถูกต้องในการจำแนกประเภทของข้อมูลแต่ละพับที่อัตราการลดข้อมูลหนึ่งๆ และสุดท้าย เราจะนำค่าความถูกต้องของข้อมูลทุกพับมาเฉลี่ยกัน เพื่อเป็นค่าความถูกต้องในการ จำแนกประเภท ณ อัตราการลดข้อมูลหนึ่งๆ

4.4.2 การลดแบบ CE และการลดแบบ CE ที่ลดคุณสมบัติด้วย ReliefF

เช่นเดียวกับการฝึกข้อมูลที่ลดด้วยวิธีสุ่ม สำหรับค่าของจำนวนโนนคซ่อนแต่ละค่า เราจะ ทดลองซ้ำ 8 ครั้งแล้วนำค่าความถูกต้องที่ได้มาเฉลี่ยกัน จากนั้นเลือกค่าความถูกต้องที่ดีที่สุดเพื่อ เป็นตัวแทนความถูกต้องของแต่ละพับ ขั้นตอนแสดงในรูปที่ 28

เนื่องจากการลดข้อมูลแบบ CE จะต้องใช้การกำหนดค่ารัศมีระเบิด r จึงทำให้เราไม่ สามารถควบคุมได้โดยตรงว่าจะให้เหลือข้อมูลกี่เปอร์เซ็นต์หลังการลด ดังนั้นถึงแม้ข้อมูลแต่ละพับ จะใช้ r ค่าเดียวกัน แต่ภายหลังการลดอาจจะเหลือข้อมูลจำนวนไม่เท่ากันก็เป็น ได้ ดังนั้นเราจะใช้ ค่าเฉลี่ยของจำนวนข้อมูลที่เหลือหลังการลดแล้วมาคำนวณเปอร์เซ็นต์ เพื่อให้เป็นตัวแทนอัตราการ ลดข้อมูล



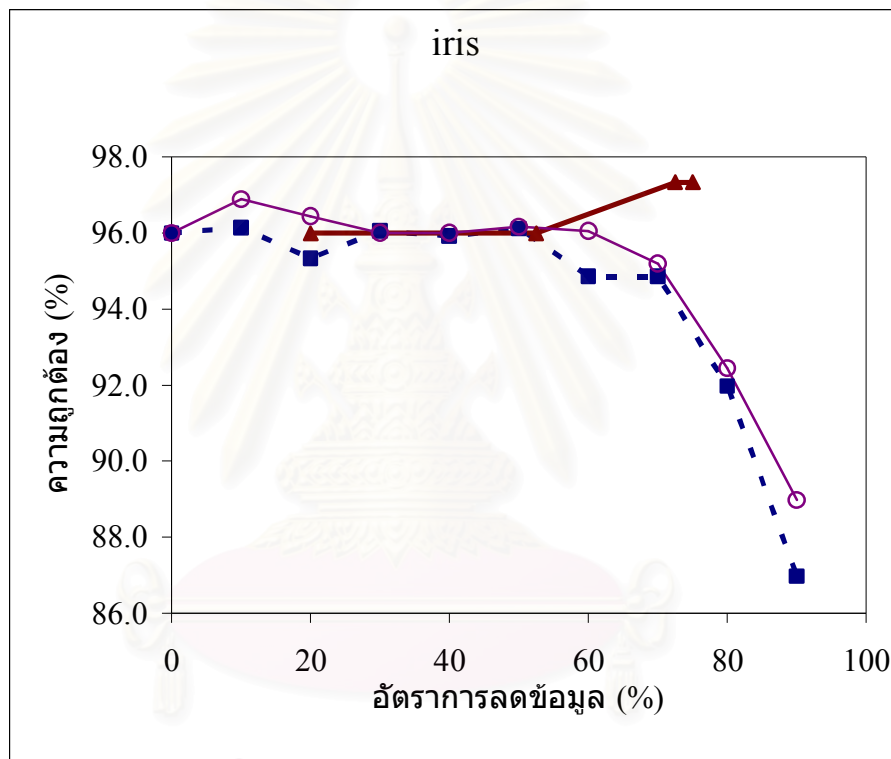
รูปที่ 28 ขั้นตอนการฝึกและการคำนวณค่าความถูกต้องในการจำแนกประเภทของข้อมูลฝึกที่ ลดด้วยวิธี CE และ CE ที่ลดคุณสมบัติด้วย ReliefF

บทที่ 5

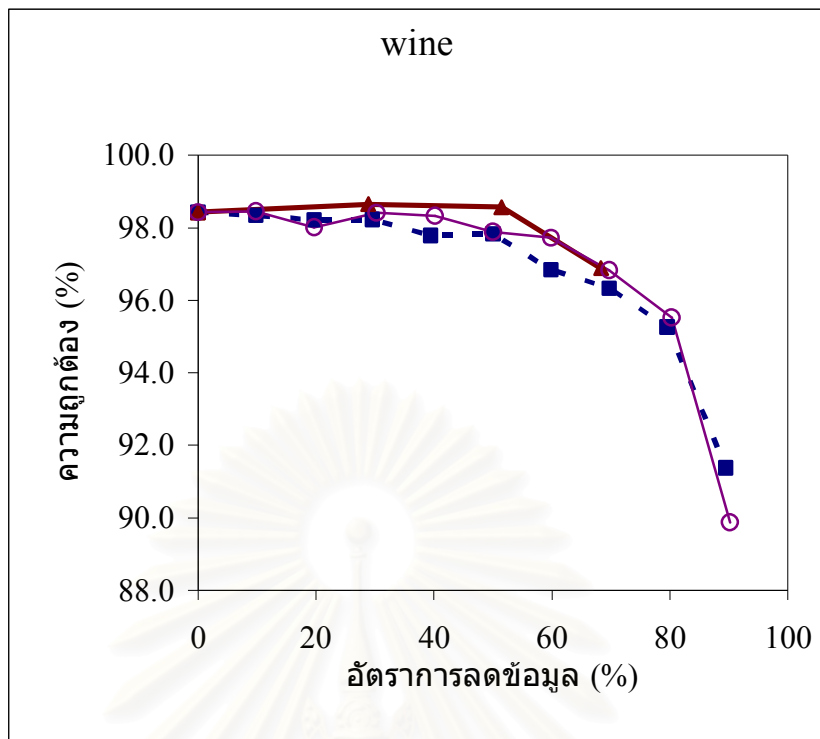
ผลการทดลอง

5.1 ประสิทธิภาพของการลดข้อมูลแบบสุ่ม แบบสุ่มตามความหนาแน่น และแบบ CE

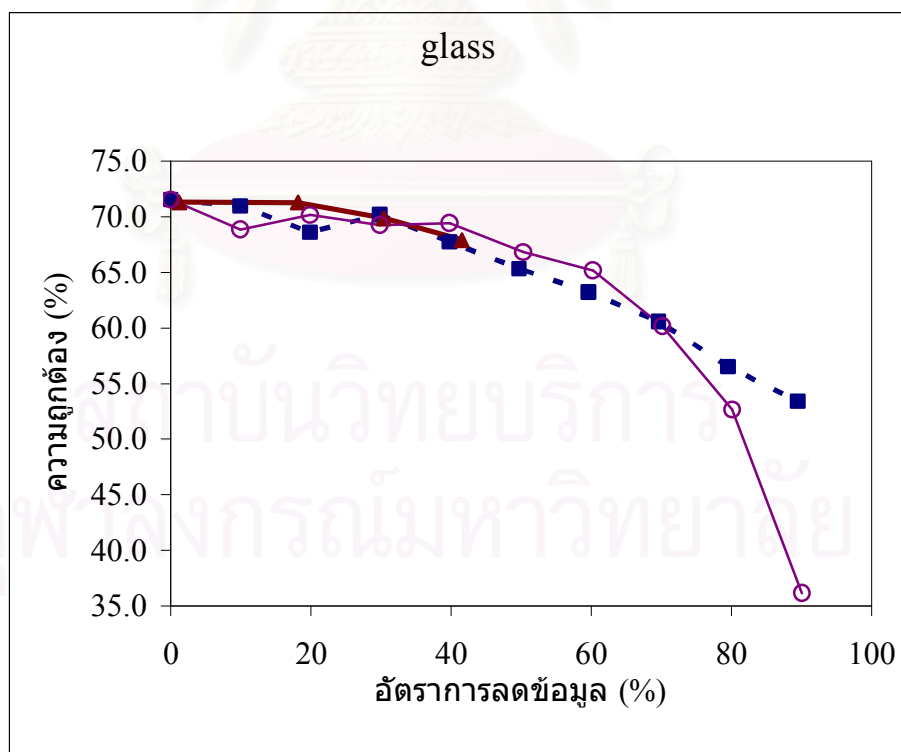
ผลการทดลองแสดงในรูปที่ 29 ถึงรูปที่ 37 โดยกราฟแสดงความถูกต้องในการจำแนกประเภทข้อมูลของข่ายงานประสาท ที่ถูกฝึกด้วยข้อมูลที่ลดจากวิธีการลดข้อมูลทั้ง 3 วิธี ได้แก่ การลดข้อมูลแบบสุ่ม (แสดงด้วยเส้นประ) การลดข้อมูลแบบสุ่มตามความหนาแน่น (แสดงด้วยเส้นบางกับวงกลมโปร่ง) และการลดข้อมูลแบบ CE (แสดงด้วยเส้นทึบกับสามเหลี่ยมทึบ)



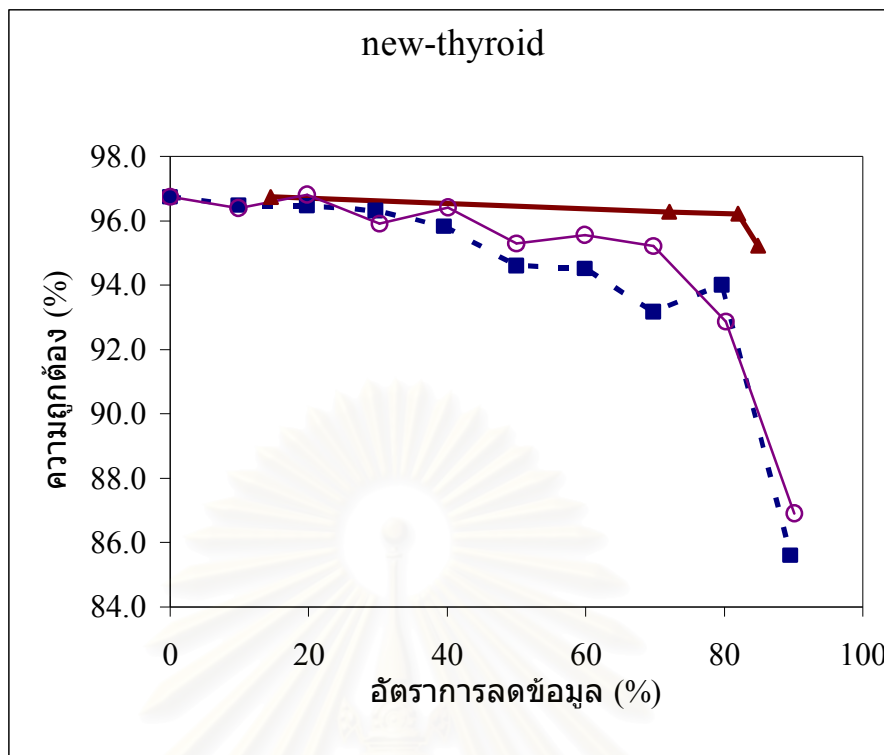
รูปที่ 29 กราฟความถูกต้องในการจำแนกประเภทข้อมูลของการลดข้อมูลแบบสุ่ม แบบสุ่มตามความหนาแน่น และแบบ CE ของชุดข้อมูล iris ณ อัตราการลดข้อมูลต่างๆ



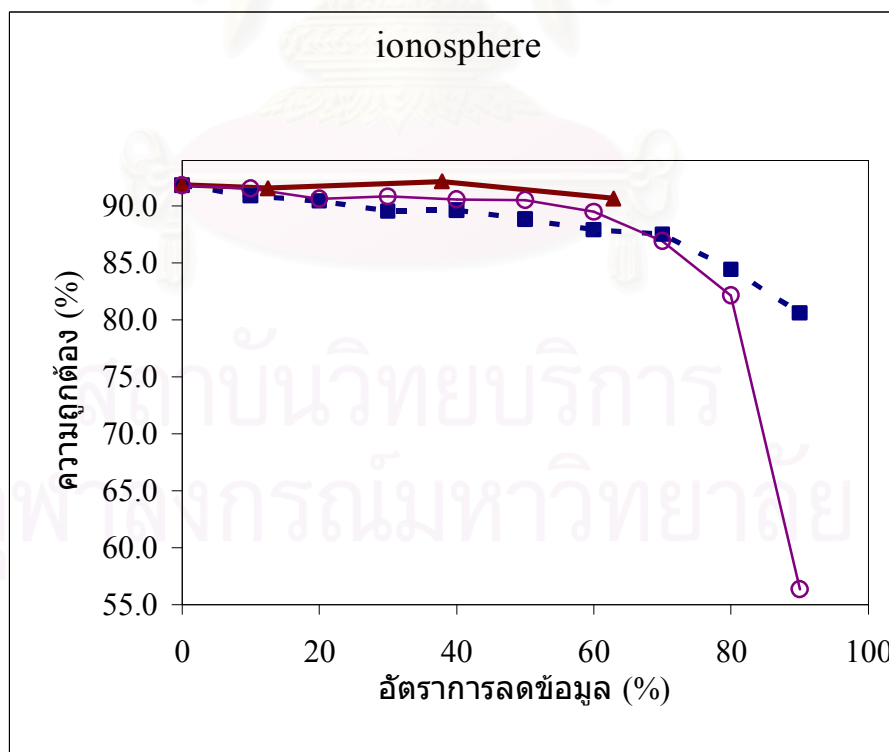
รูปที่ 30 กราฟความถูกต้องในการจำแนกประเภทข้อมูลของการลดข้อมูลแบบสุ่ม แบบสุ่มตามความหนาแน่น และแบบ CE ของชุดข้อมูล wine ณ อัตราการลดข้อมูลต่างๆ



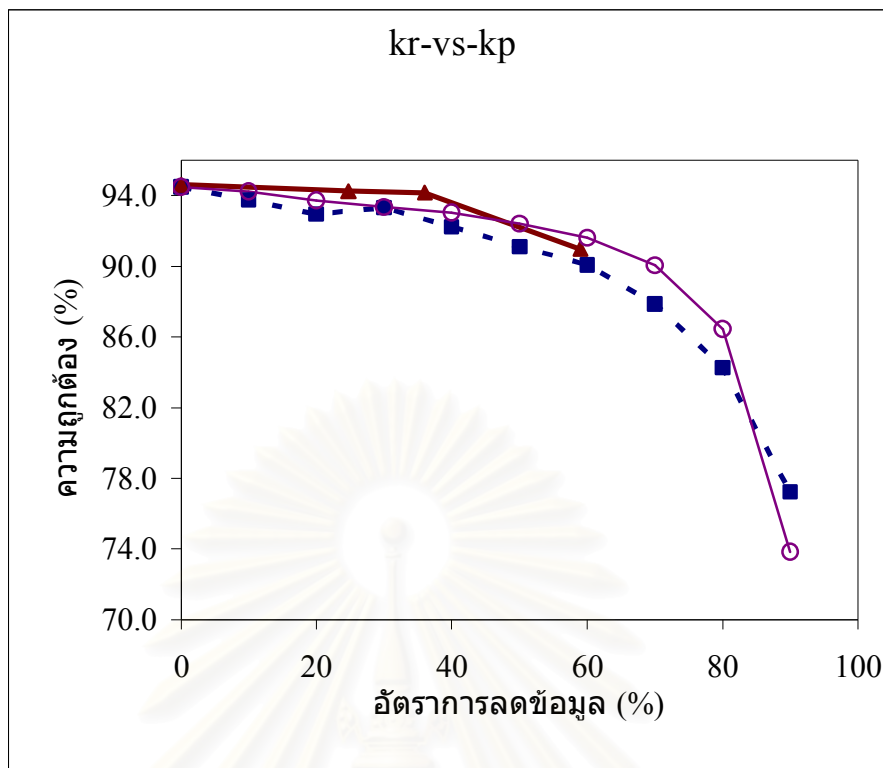
รูปที่ 31 กราฟความถูกต้องในการจำแนกประเภทข้อมูลของการลดข้อมูลแบบสุ่ม แบบสุ่มตามความหนาแน่น และแบบ CE ของชุดข้อมูล glass ณ อัตราการลดข้อมูลต่างๆ



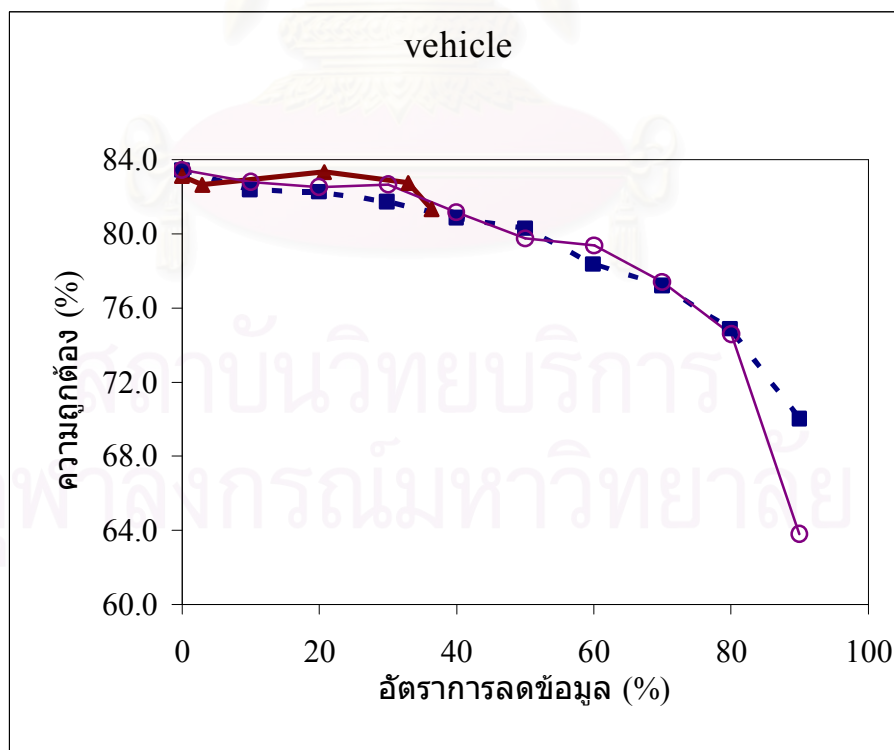
รูปที่ 32 กราฟความถูกต้องในการจำแนกประเภทข้อมูลของการลดข้อมูลแบบสุ่ม แบบสุ่มตามความหนาแน่น และแบบ CE ของชุดข้อมูล new-thyroid ณ อัตราการลดข้อมูลต่างๆ



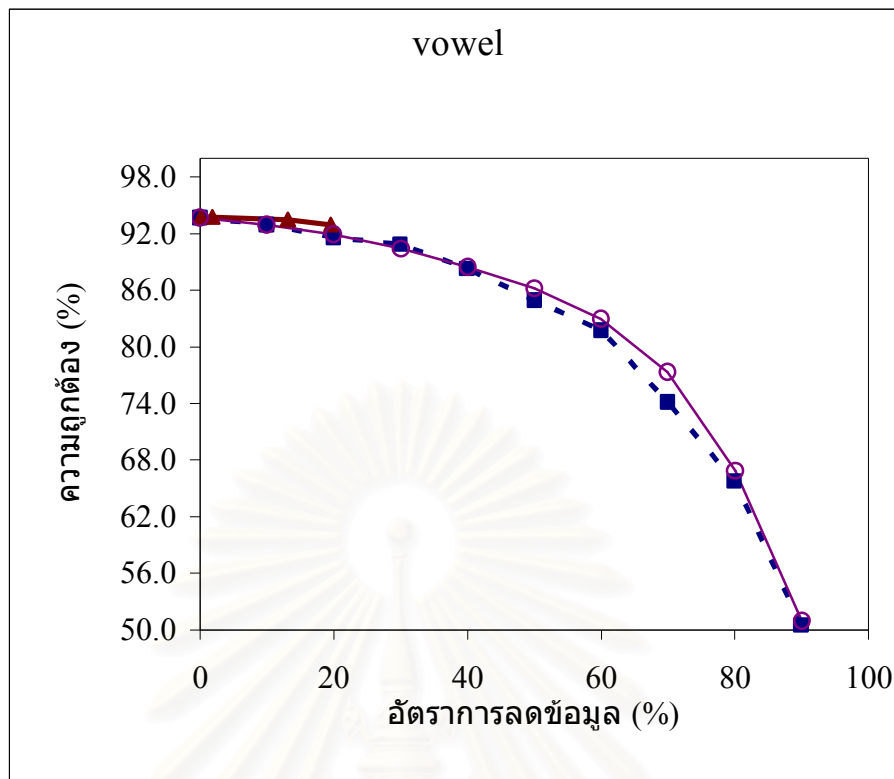
รูปที่ 33 กราฟความถูกต้องในการจำแนกประเภทข้อมูลของการลดข้อมูลแบบสุ่ม แบบสุ่มตามความหนาแน่น และแบบ CE ของชุดข้อมูล ionosphere ณ อัตราการลดข้อมูลต่างๆ



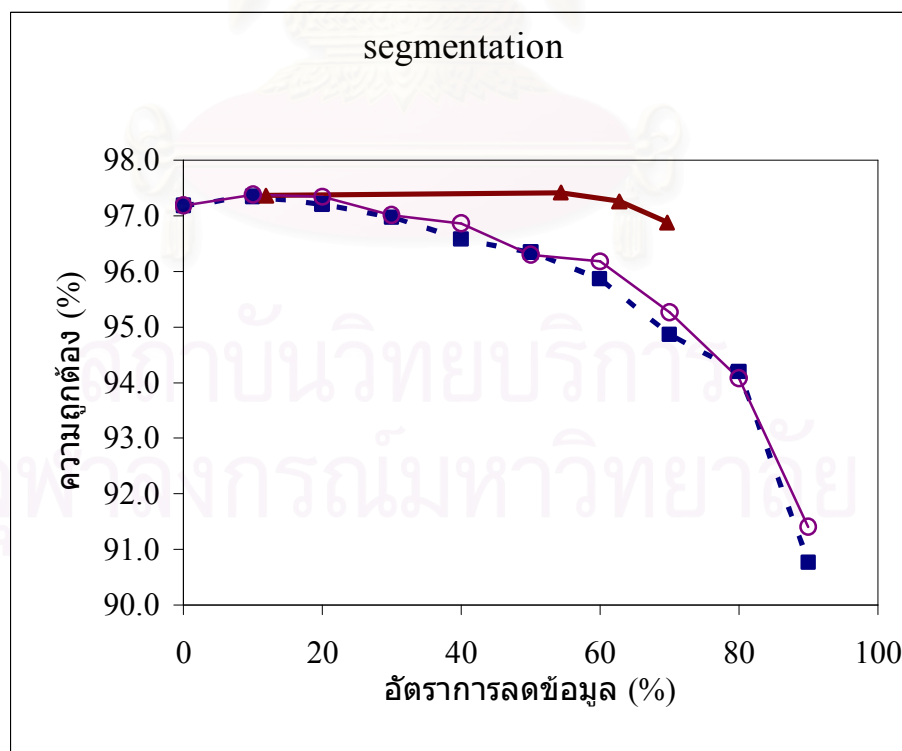
รูปที่ 34 กราฟความถูกต้องในการจำแนกประเภทข้อมูลของการลดข้อมูลแบบสุ่ม แบบสุ่มตามความหนาแน่น และแบบ CE ของชุดข้อมูล kr-vs-kp ณ อัตราการลดข้อมูลต่างๆ



รูปที่ 35 กราฟความถูกต้องในการจำแนกประเภทข้อมูลของการลดข้อมูลแบบสุ่ม แบบสุ่มตามความหนาแน่น และแบบ CE ของชุดข้อมูล vehicle ณ อัตราการลดข้อมูลต่างๆ



รูปที่ 36 กราฟความถูกต้องในการจำแนกประเภทข้อมูลของการลดข้อมูลแบบสุ่ม แบบสุ่มตามความหนาแน่น และแบบ CE ของชุดข้อมูล vowel ณ อัตราการลดข้อมูลต่างๆ



รูปที่ 37 กราฟความถูกต้องในการจำแนกประเภทข้อมูลของการลดข้อมูลแบบสุ่ม แบบสุ่มตามความหนาแน่น และแบบ CE ของชุดข้อมูล segmentation ณ อัตราการลดข้อมูลต่างๆ

โดยภาพรวมจะเห็นได้ว่าทั้งการลดข้อมูลแบบสุ่มตามความหนาแน่น และการลดข้อมูลแบบ CE จะให้ความแม่นยำในการจำแนกประเภทข้อมูลสูงกว่าการลดด้วยวิธีสุ่มแบบปกติ เกือบจะทุกๆ อัตราการลดข้อมูล แต่สำหรับการลดข้อมูลแบบสุ่มตามความหนาแน่น ณ อัตราการลดข้อมูลประมาณ 70%-80% ขึ้นไปดูเหมือนว่าสำหรับหลายๆ ชุดข้อมูลจะให้ผลไม่ดีนัก คือให้ผลที่แย่กว่าการลดข้อมูลแบบสุ่ม

สำหรับการลดข้อมูลแบบ CE ในการทดลองนี้ได้กำหนดค่า r ไว้สูงสุดที่ 20 ซึ่งเป็นค่า r ที่ทำให้ CE ลดข้อมูลได้มากที่สุดเท่าที่จะทำได้ จากผลการทดลองพบว่าส่วนใหญ่การลดแบบ CE จะลดข้อมูลได้มากถึง 60% ยกเว้นชุดข้อมูล glass, vehicle และ vowel เนื่องจากชุดข้อมูลมีจำนวนประเภทข้อมูลมาก โดยเฉพาะ vowel ซึ่งมีประเภทข้อมูลมากถึง 11 ประเภท การที่ชุดข้อมูลมีประเภทข้อมูลมาก จะทำให้ชุดข้อมูลนั้นถูกเลือกเก็บข้อมูลไว้มาก ซึ่งเป็นผลทำให้มีอัตราการลดข้อมูลที่น้อย

เป็นที่น่าสังเกตว่าโดยทั่วไปว่าเมื่อลดข้อมูลออกไปด้วยอัตราการลดที่มากขึ้น สำหรับวิธีการลดต่างๆ ความถูกต้องในการจำแนกจะลดลง แต่สำหรับชุดข้อมูล iris กลับพบว่าเมื่อยังลดข้อมูลด้วยวิธี CE มากขึ้นกลับทำให้ความถูกต้องในการจำแนกสูงขึ้น และยังพบอีกว่าชุดข้อมูลหลายชุด เช่น new-thyroid, ionosphere, segmentation เมื่อลดข้อมูลมากขึ้นด้วยวิธี CE ก็ไม่ได้ทำให้ความถูกต้องลดลงเท่าใดนัก ในขณะที่การลดแบบสุ่มให้ความถูกต้องตกลงมาอย่างเห็นได้ชัด

5.2 ความเร็วในการลดข้อมูลแบบ CE และ ACE

เวลาที่ใช้ในการทำงานของ CE และ ACE ที่ค่า r ต่างๆ แสดงในตารางที่ 5 ผลการทดลองแสดงให้เห็นว่าวิธีการลดข้อมูลแบบ ACE ซึ่งปรับปรุงมาจากวิธีการแบบ CE ด้วยการประมวลผลข้อมูล $rCluster$ ไว้วางหน้า ทำให้ใช้เวลาในการลดข้อมูลได้เร็วขึ้นเป็นอย่างมาก เช่นกรณีชุดข้อมูล segmentation วิธีการลดข้อมูลแบบ ACE ทำเวลาได้เร็วกว่าวิธีเดิมได้ถึง 8.96 เท่า

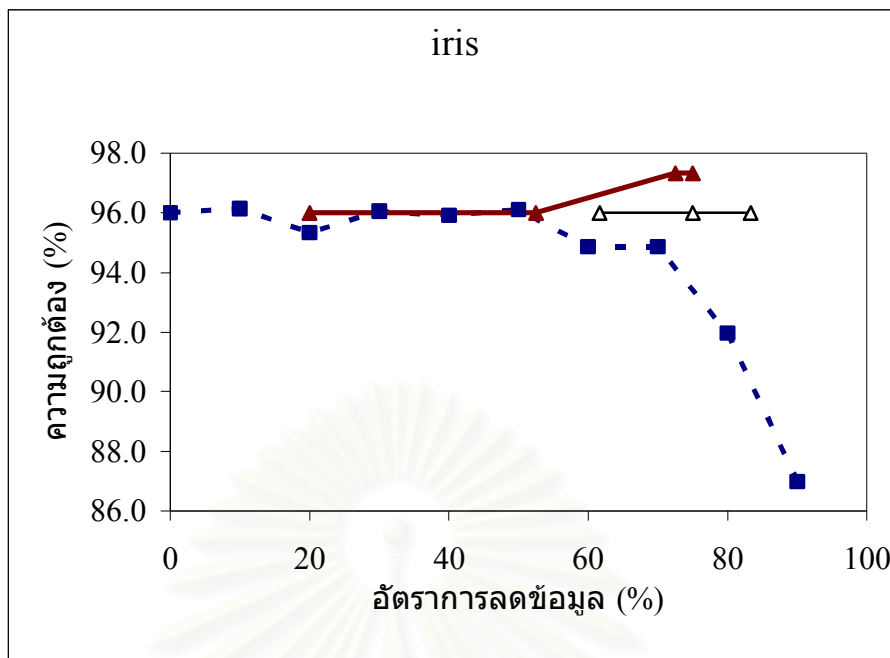
เป็นที่น่าสังเกตว่าวิธีการแบบ CE จะมีเวลาการทำงานขึ้นอยู่กับค่า r ที่ใช้ โดยยิ่ง r มีค่าใหญ่เวลาที่ใช้จะยิ่งน้อยลง ในขณะที่วิธีการแบบ ACE จะมีเวลาการทำงานค่อนข้างคงที่ ไม่ขึ้นอยู่กับ r เนื่องจากในกรณีการลดแบบ CE อัลกอริทึมถ้าเลือกใช้ r ขนาดใหญ่จะทำให้ตัดข้อมูลทิ้งได้มากในแต่ละรอบการทำงาน โดยเฉพาะในการทำงานรอบแรกๆ ทำให้รอบถัดๆ ไป ไม่ต้องเสียเวลาไปคำนวณระยะห่างเทียบกับข้อมูลที่ถูกลดทิ้งไปแล้ว ในขณะที่ถ้าใช้ r ขนาดเล็กๆ จะทำให้ตัดข้อมูลในแต่ละรอบได้น้อย เป็นผลทำให้แต่ละรอบการทำงานจะต้องคำนวณระยะห่างระหว่างข้อมูลหลายคู่มาก เป็นผลให้ต้องใช้เวลาตามไปด้วย

ตารางที่ 5 เปรียบเทียบเวลาในการลดข้อมูลระหว่าง CE กับ ACE (วินาที)

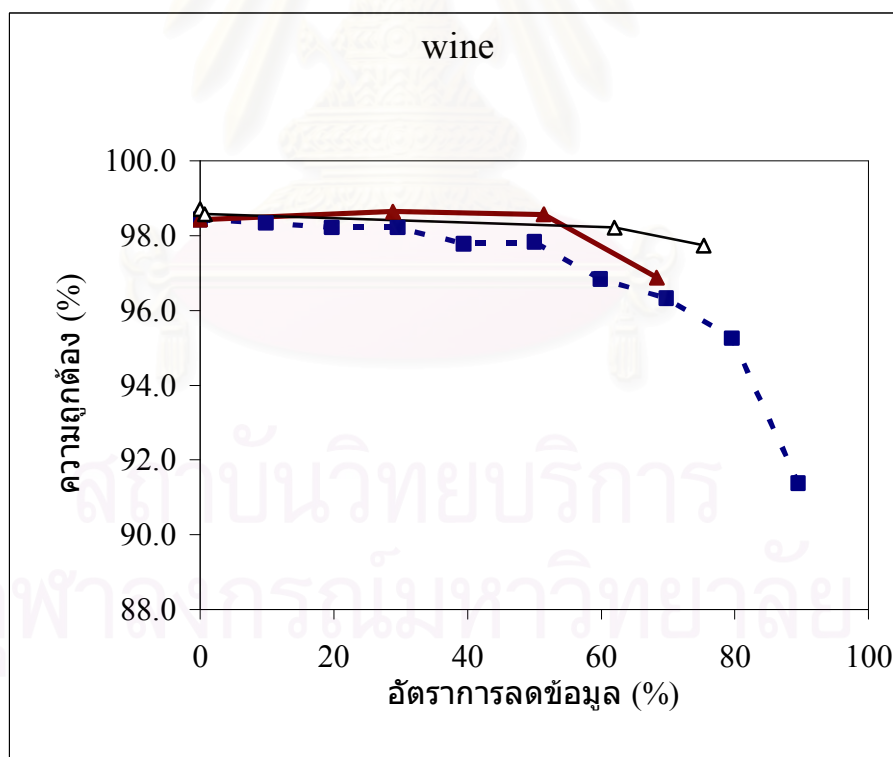
ชุดข้อมูล	วิธีลด	รัศมีการระเบิด (r)				
		0.3	1	2	20	avg.
iris	CE	0.95	0.70	0.55	0.52	0.68
	ACE	0.33	0.31	0.36	0.33	0.33
wine	CE	1.61	1.64	1.56	0.89	1.43
	ACE	0.56	0.58	0.56	0.59	0.57
glass	CE	2.19	2.11	1.66	1.22	1.79
	ACE	0.72	0.72	0.72	0.75	0.73
new-thyroid	CE	1.94	1.28	1.06	0.89	1.29
	ACE	0.50	0.53	0.58	0.59	0.55
ionosphere	CE	10.73	10.88	10.41	4.56	9.14
	ACE	3.09	3.09	3.13	3.34	3.16
kr-vs-kp	CE	30.66	31.77	31.84	13.33	26.90
	ACE	7.55	7.59	7.58	8.05	7.69
vehicle	CE	102.11	106.73	90.17	39.44	84.61
	ACE	13.39	13.42	13.44	13.55	13.45
vowel	CE	80.84	82.67	73.30	37.33	68.54
	ACE	15.39	14.92	15.02	14.55	14.97
segmentation	CE	1331.61	1199.20	883.55	442.55	964.23
	ACE	109.58	108.20	105.44	107.20	107.61

5.3 เปรียบเทียบผลการทดลองลดข้อมูลแบบ CE กับแบบ CE ที่ลดคุณสมบัติด้วย ReliefF

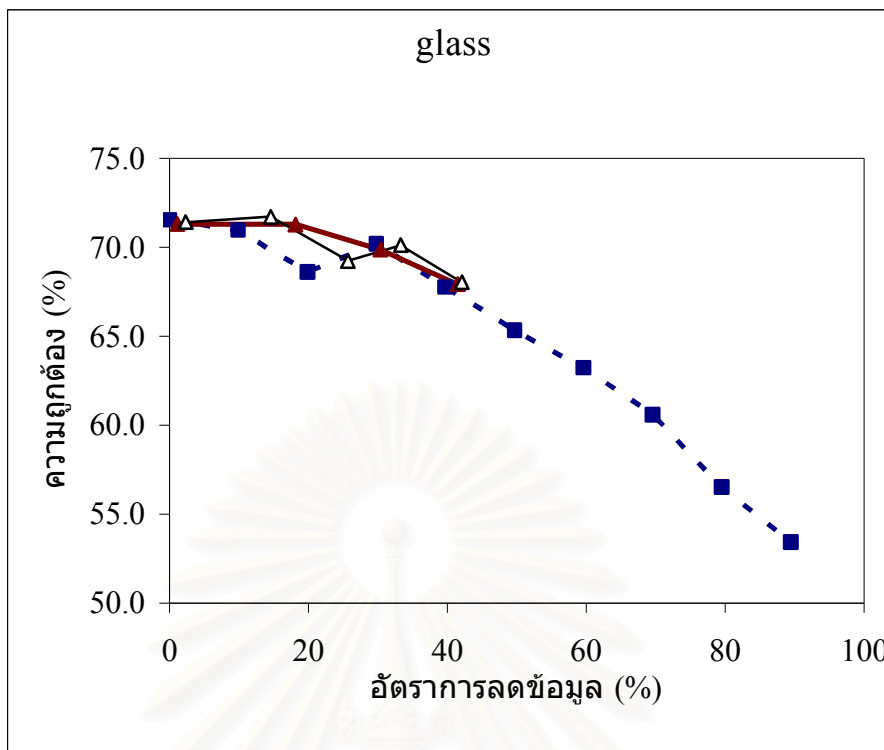
ผลการทดลองแสดงในรูปที่ 38 ถึงรูปที่ 46 โดยกราฟแสดงความถูกต้องในการจำแนกประเภทข้อมูลของช่างานประสาท ที่ถูกฝึกด้วยข้อมูลที่ลดจากวิธีการลดข้อมูลแบบสุ่ม (แสดงด้วยเส้นประ) การลดข้อมูลแบบ CE (แสดงด้วยเส้นทึบกับสามเหลี่ยมทึบ) และการลดข้อมูลแบบ CE ที่ลดคุณสมบัติมาก่อนด้วย ReliefF (แสดงด้วยเส้นบางกับสามเหลี่ยมโปร่ง)



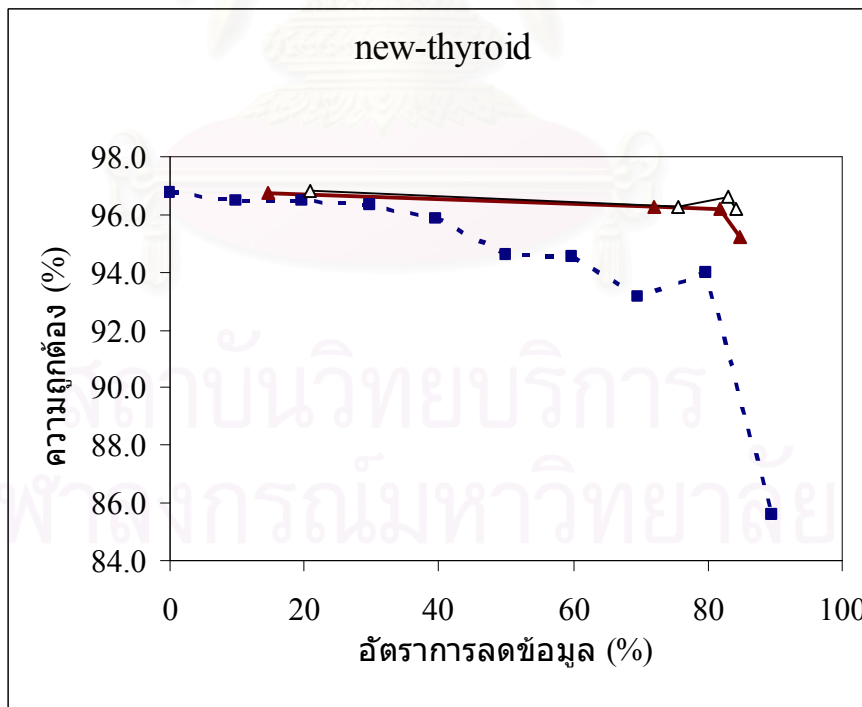
รูปที่ 38 กราฟความถูกต้องในการจำแนกประเภทข้อมูลที่เกิดด้วยการลดข้อมูลแบบ CE และแบบ CE ที่ลดคุณสมบัติด้วย ReliefF ของชุดข้อมูล iris ณ อัตราการลดข้อมูลต่างๆ



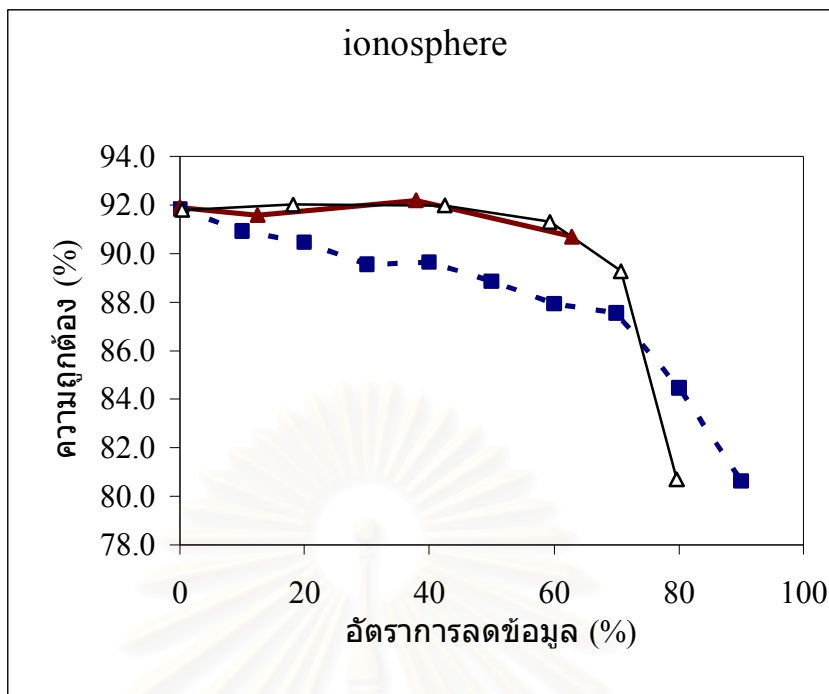
รูปที่ 39 กราฟความถูกต้องในการจำแนกประเภทข้อมูลที่เกิดด้วยการลดข้อมูลแบบ CE และแบบ CE ที่ลดคุณสมบัติด้วย ReliefF ของชุดข้อมูล wine ณ อัตราการลดข้อมูลต่างๆ



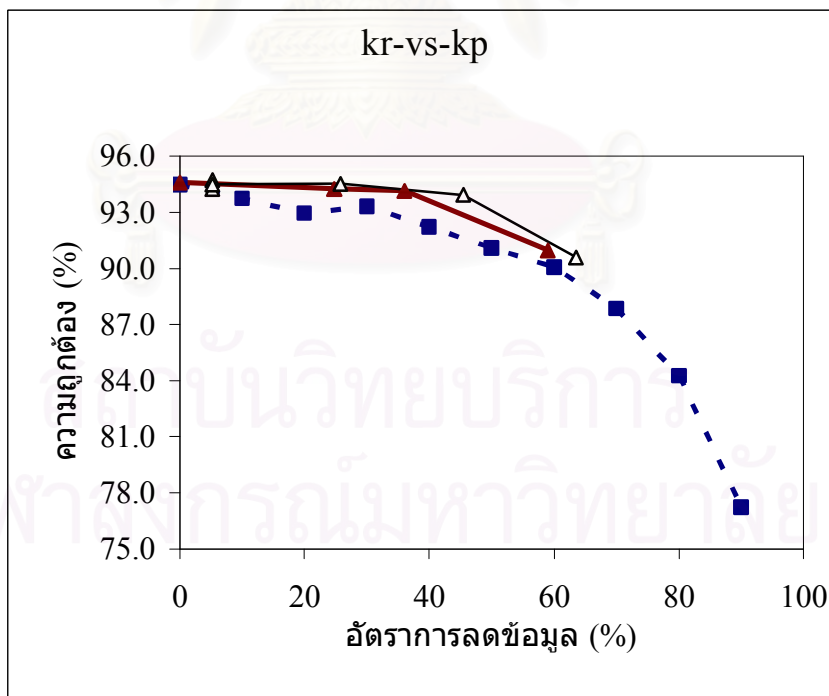
รูปที่ 40 กราฟความถูกต้องในการจำแนกประเภทข้อมูลที่ลดด้วยการลดข้อมูลแบบ CE และแบบ CE ที่ลดคุณสมบัติด้วย ReliefF ของชุดข้อมูล glass ณ อัตราการลดข้อมูลต่างๆ



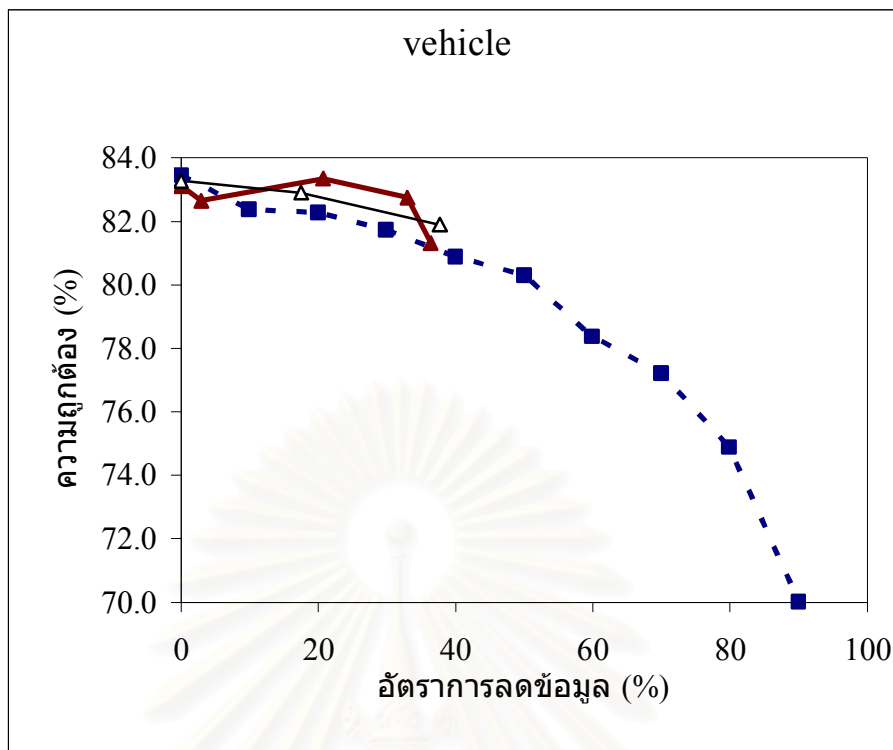
รูปที่ 41 กราฟความถูกต้องในการจำแนกประเภทข้อมูลที่ลดด้วยการลดข้อมูลแบบ CE และแบบ CE ที่ลดคุณสมบัติด้วย ReliefF ของชุดข้อมูล new-thyroid ณ อัตราการลดข้อมูลต่างๆ



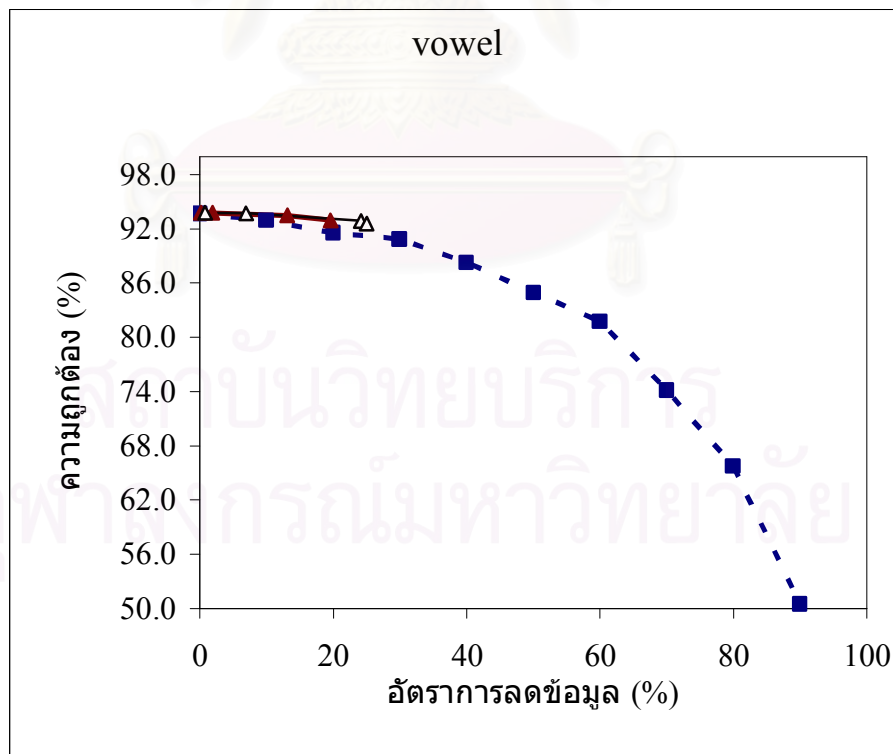
รูปที่ 42 กราฟความถูกต้องในการจำแนกประเภทข้อมูลที่ลดด้วยการลดข้อมูลแบบ CE และแบบ CE ที่ลดคุณสมบัติด้วย ReliefF ของชุดข้อมูล ionosphere ณ อัตราการลดข้อมูลต่างๆ



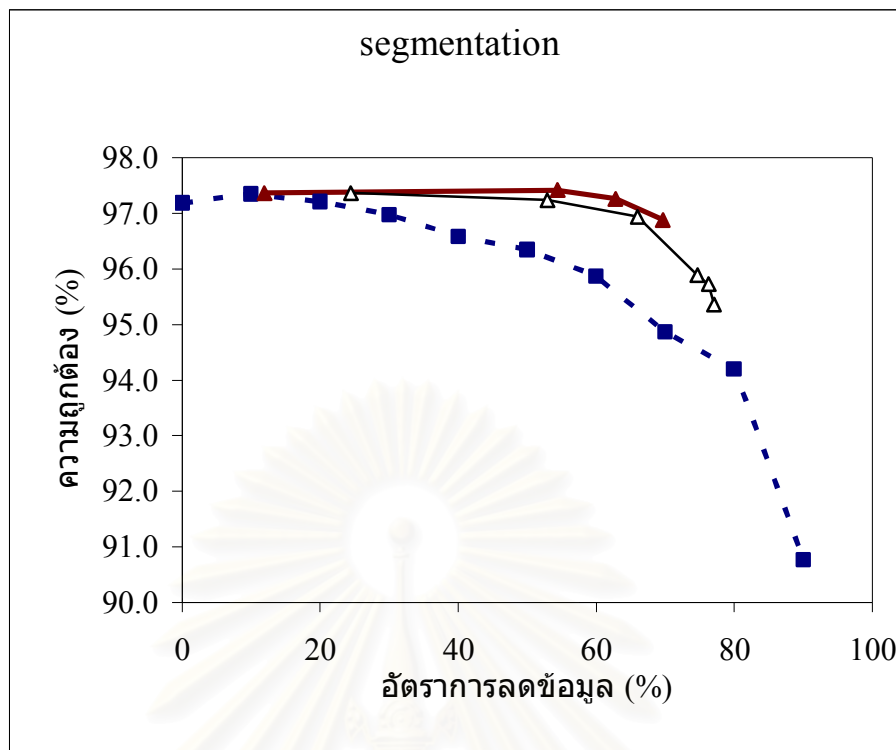
รูปที่ 43 กราฟความถูกต้องในการจำแนกประเภทข้อมูลที่ลดด้วยการลดข้อมูลแบบ CE และแบบ CE ที่ลดคุณสมบัติด้วย ReliefF ของชุดข้อมูล kr-vs-kp ณ อัตราการลดข้อมูลต่างๆ



รูปที่ 44 กราฟความถูกต้องในการจำแนกประเภทข้อมูลที่ลดด้วยการลดข้อมูลแบบ CE และแบบ CE ที่ลดคุณสมบัติด้วย ReliefF ของชุดข้อมูล vehicle ณ อัตราการลดข้อมูลต่างๆ



รูปที่ 45 กราฟความถูกต้องในการจำแนกประเภทข้อมูลที่ลดด้วยการลดข้อมูลแบบ CE และแบบ CE ที่ลดคุณสมบัติด้วย ReliefF ของชุดข้อมูล vowel ณ อัตราการลดข้อมูลต่างๆ



รูปที่ 46 กราฟความถูกต้องในการจำแนกประเภทข้อมูลที่ลดด้วยการลดข้อมูลแบบ CE และแบบ CE ที่ลดคุณสมบัติด้วย ReliefF ของชุดข้อมูล segmentation ณ อัตราการลดข้อมูลต่างๆ

จากกราฟผลการทดลองจะเห็นได้ว่าการใช้อัลกอริทึม ReliefF ช่วยการลดข้อมูลแบบ CE ณ ตำแหน่งที่ใช้รัศมีระดับค่าต่างๆ จะสามารถลดข้อมูลได้มากกว่าการลดข้อมูลแบบ CE ที่ไม่ได้ลดจำนวนคุณสมบัติมาก่อน สำหรับบางชุดข้อมูลเช่น ionosphere สามารถลดข้อมูลได้มากขึ้นเกือบจะ 20% แต่ความถูกต้องในการจำแนกประเภทก็ตกลงมามากด้วยเช่นกัน

ในภาพรวมพบว่าการใช้อัลกอริทึม ReliefF มาช่วย ก็ไม่ได้ทำให้ความถูกต้องในการจำแนกแย่ง กล่าวคือความถูกต้องในการจำแนกก็ยังคงทำได้ดีพอๆ กับการลดแบบ CE ที่ไม่ใช้ ReliefF มาช่วย แต่ในบางชุดข้อมูลเช่น wine และ kr-vs-kp ดูเหมือนว่าเมื่อใช้ ReliefF มาช่วยแล้ว จะทำให้ได้ความถูกต้องในอัตราลดช่วงปลายๆ สูงขึ้น และลดข้อมูลได้มากขึ้นอีกเล็กน้อยด้วย

บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการวิจัย

วิทยานิพนธ์ฉบับนี้เสนอวิธีการต่างๆ เพื่อใช้ลดข้อมูลสำหรับฝึกข่ายงานประสาทเทียม โดยได้เสนอวิธีการลดที่เกี่ยวข้องกับกระบวนการสุ่ม คือวิธีการลดข้อมูลแบบสุ่มตามความหนาแน่น และวิธีการลดที่ไม่เกี่ยวข้องกับกระบวนการสุ่ม คือวิธีการลดข้อมูลแบบ CE (ระเบิดลูกโซ่) และวิธีการลดข้อมูลแบบ ACE (ระเบิดลูกโซ่ขั้นสูง) นอกจากนี้ยังได้เสนอวิธีปรับปรุงวิธีการลดข้อมูลแบบระเบิดลูกโซ่ให้สามารถลดข้อมูลได้มากขึ้น โดยใช้อัลกอริทึม ReliefF มาช่วย

จากผลการทดลองพบว่าทุกวิธีการลดข้อมูลที่เสนอมานี้ให้ความถูกต้องในการจำแนกประเภทสูงกว่าการลดด้วยวิธีสุ่มแบบทั่วไปเกือบๆ จะทุกอัตราการลดข้อมูล แสดงให้เห็นว่า

1. แนวคิดเบื้องต้นที่ว่าบริเวณใดที่มีข้อมูลอยู่มาก เกิดความซ้ำซ้อนของข้อมูล เราอาจจะตัดข้อมูลทิ้งได้มากกว่าบริเวณที่มีข้อมูลอยู่เบาบาง เป็นแนวคิดที่ใช้ได้จริง ซึ่งวิทยานิพนธ์ฉบับนี้ได้แสดงให้เห็นแล้วจากวิธีการลดข้อมูลแบบสุ่มตามความหนาแน่น ที่ให้ความถูกต้องในการจำแนกประเภทสูงกว่าการลดด้วยวิธีสุ่มแบบทั่วไป
2. แนวคิดที่จะเลือกเก็บเฉพาะข้อมูลตรงขอบมาใช้ฝึกข่ายงานประสาท ซึ่งเป็นการละเลยต่อข้อมูลที่อยู่ตรงใจกลางของกลุ่มข้อมูลแต่ละประเภท เป็นแนวคิดที่ใช้งานได้จริง ซึ่งวิทยานิพนธ์ฉบับนี้ได้แสดงให้เห็นแล้วจากวิธีการลดข้อมูลแบบ CE ว่าให้ความถูกต้องในการจำแนกประเภทไม่ต่ำกว่าการลดข้อมูลด้วยวิธีสุ่มทั้งแบบทั่วไป และสุ่มลดตามความหนาแน่น
3. การตัดคุณสมบัติที่มีอิทธิพลน้อยต่อการจำแนกประเภท ทำให้ลดข้อมูลได้มากยิ่งขึ้น ซึ่งวิทยานิพนธ์ฉบับนี้ได้แสดงให้เห็นแล้วจากการใช้อัลกอริทึม ReliefF ลดคุณสมบัติก่อน แล้วจึงค่อยลดข้อมูลด้วยวิธี CE

นอกจากนี้ผู้วิจัยยังได้เสนอวิธีการปรับปรุงความเร็วในการทำงานของการลดข้อมูลแบบ CE เป็น ACE ซึ่งให้ผลการลดข้อมูลเหมือนกันทุกประการ โดยใช้แนวคิดของ $rCluster$ และมีการประมวลผลล่วงหน้าโดยใช้คิสจอยท์เซท ทำให้อัลกอริทึมมีการทำงานเร็วเพิ่มขึ้นอย่างเห็นได้ชัด โดยความซับซ้อนเชิงเวลาลดลงจาก $O(m^2n^3)$ เหลือเพียง $O(m^2n^2)$

ข้อมูลที่ได้จากการลดข้อมูลแบบ CE จะมีโครงสร้างที่ซับซ้อนต่อการเรียนรู้ เนื่องจากอัลกอริทึม CE จะพยายามเลือกเก็บเฉพาะข้อมูลตรงขอบของแต่ละประเภทข้อมูลเท่านั้น การจะ

เรียนรู้ข้อมูลที่มีความเฉพาะเจาะจงมากๆ เช่นนี้ได้ จำเป็นอย่างยิ่งที่จะต้องใช้จำนวนโหนดซ่อน และจำนวนรอบในการเรียนที่เพียงพอ จึงจะให้พื้นผิวดัดสินใจที่มีความถูกต้องในการจำแนกประเภทข้อมูลสูงๆ ได้

การลดจำนวนคุณสมบัติที่มีผลน้อยต่อการจำแนกประเภทก่อน โดยใช้อัลกอริทึม Relief แล้วค่อยนำข้อมูลที่ได้ไปลดด้วยวิธี CE ทำให้สามารถลดข้อมูลได้มากยิ่งขึ้น แต่ทั้งนี้การจากทดลองพบว่า ในบางชุดข้อมูลจะให้ความถูกต้องในการจำแนกประเภทลดลงในช่วงอัตราการลดข้อมูลสูงๆ แต่ถ้าเป็นอัตราการลดข้อมูลช่วงต้นๆ พบว่าการใช้ Relief มาช่วยก็ให้ผลความถูกต้องไม่ต่างกันเท่าใดนัก

6.2 ข้อเสนอแนะ

การลดข้อมูลแบบ CE จะให้ข้อมูลที่มีโครงสร้างที่ซับซ้อนต่อการเรียนรู้ เนื่องจากอัลกอริทึม CE จะพยายามเลือกเก็บเฉพาะข้อมูลตรงขอบของแต่ละประเภทข้อมูลเท่านั้น การจะเรียนรู้ข้อมูลที่มีความเฉพาะเจาะจงมากๆ เช่นนี้ได้ จำเป็นอย่างยิ่งที่จะต้องใช้จำนวนโหนดซ่อน และจำนวนรอบในการเรียนที่เพียงพอ ซึ่งจากการทดลองก็พบว่าเป็นอย่างนั้น กล่าวคือเมื่อนำข้อมูลที่ลดด้วยวิธี CE ไปฝึกข่ายงานประสาท พบว่าความถูกต้องในการจำแนกประเภทจะเริ่มสูงกว่าการลดด้วยวิธีสุ่ม เมื่อใช้จำนวนโหนดในชั้นซ่อนที่ $h = 4a$ ซึ่งถ้าหากมีโอกาสได้ทำการทดลองต่อไป ก็น่าจะทดลองเพิ่มจำนวนโหนดซ่อนให้เป็น $h = 8a$ หรือมากกว่านี้ และต้องใช้จำนวนรอบการฝึกให้เพียงพอ ก็น่าจะได้ความถูกต้องในการจำแนกประเภทสูงเพิ่มขึ้นอีก

นอกจากนี้การนำอัลกอริทึม Relief มาช่วยการลดข้อมูลแบบ CE มีประเด็นที่ต้องขบคิด ตั้งแต่จะเลือกคุณสมบัติมาจำนวนมากน้อยเท่าใดจึงจะเหมาะสม ค่า k ที่ควรใช้ จำนวนรอบที่ควรใช้ ซึ่งจำเป็นจะต้องศึกษาในประเด็นย่อยเหล่านี้เพิ่มเติม

รายการอ้างอิง

- [1] C. M. Bishop. Neural Networks for Pattern Recognition. Oxford University Press, 1995.
- [2] S. Haykin. Neural Networks: a comprehensive foundation. 2 ed. Prentice Hall, 1999.
- [3] T. M. Mitchell. Machine Learning. McGraw-Hill, 1997.
- [4] C. Blake, E. Keogh, and C. Merz. UCI Repository of Machine Learning Databases. Irvine: University of California, 1998.
- [5] M. Caudill. Neural networks primer, part I. AI Expert. 2 (1987): 46-52.
- [6] W. S. McCulloch and W. H. Pitts. A Logical Calculus for Ideas Imminent in Nervous Activity. Bulletin of Mathematical Biophysics. 5 (1943): 115-133.
- [7] D. R. Wilson and T. R. Martinez. Reduction Techniques for Exemplar-Based Learning Algorithms. Machine Learning: ECML-94. 38 (2000): 257-286.
- [8] K. Kira and L. A. Rendell. A practical approach to feature selection. Machine Learning: Proceedings of International Conference (ICML'92). (1992): 249-256.
- [9] I. Kononenko. Estimating attributes: Analysis and extensions of Relief. Machine Learning: ECML-94. (1994): 171-182.
- [10] R. Sikonja and I. Kononenko. An adaptation of relief for attribute estimation in regression. Morgan Kaufmann, 1997.
- [11] P. Hart. The condensed nearest neighbor rule. IEEE Trans. on Information Theory. 14 (1968): 515-516.
- [12] W. Gates. The reduced nearest neighbor rule. IEEE Trans. on Information Theory. 18 (1972): 431-433.
- [13] G. Ritter, H. Woodruff, S. Lowry, and T. Isenhour. An algorithm for a selective nearest neighbor decision rule. IEEE Trans. on Information Theory. 21 (1975): 665-669.
- [14] C. E. Brodley. Addressing the Selective Superiority Problem: Automatic Algorithm/Model Class Selection. The 10th International Machine Learning Conference. (1993): 17-24.

- [15] F. Devi and M. Murty. An incremental prototype set building technique. Pattern Recognition. 35 (2002): 505-513.
- [16] G. Valentini and T. G. Dietterich. Bias-Variance Analysis of Support Vector Machines for the Development of SVM-Based Ensemble Methods. Journal of Machine Learning Research. 5 (2004): 725-775.



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายไพรัช นิตยานุภาพ เกิดเมื่อวันอังคารที่ 31 มีนาคม พ.ศ. 2524 ที่จังหวัด กรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต (เกียรตินิยมอันดับ 1) สาขาวิชา วิศวกรรมคอมพิวเตอร์ จากภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2547 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2547



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย