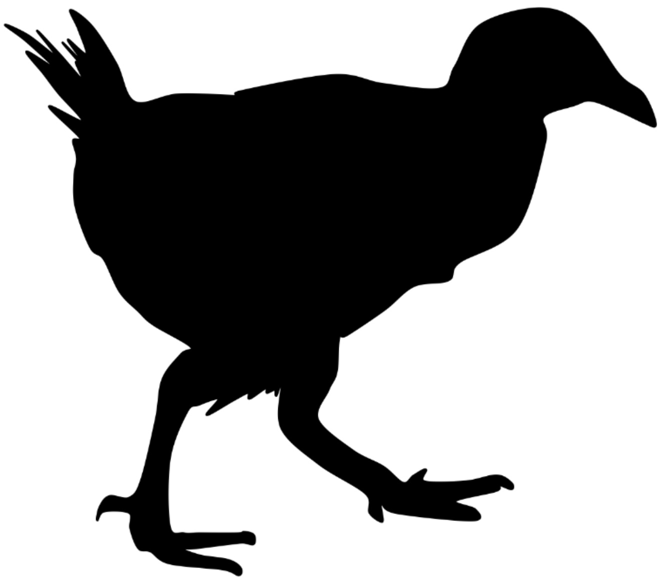
การรู้จำแบบ

PATTERN RECOGNITION



WEKA

ผู้ช่วยศาสตราจารย์ ดร.ชุมพล บุญคุ้มพรภัทร

ภาควิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

มหาวิทยาลัยเชียงใหม่

การรู้จำแบบ

PATTERN RECOGNITION

บทนำ การแทนแบบ

ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุด ตัวจำแนกเบส์ ต้นไม้การตัดสินใจ

โครงข่ายประสาทและเครื่องกลเวกเตอร์สนับสนุน ปัญหาชั้นข้อมูลอสมดุล

ผู้ช่วยศาสตราจารย์ ดร.ชุมพล บุญคุ้มพรภัทร

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่

๒๕๖๓

คำนำ

ตำราวิชาการรู้จำแบบ (Pattern Recognition) รหัสวิชา 204453 ได้เรียบเรียงขึ้นอย่างเป็นระบบ ครอบคลุมเนื้อหาสาระตามคำอธิบายของรายวิชาในสาขาวิชาวิทยาการคอมพิวเตอร์ เพื่อใช้ในการเรียนการสอนให้แก่นักศึกษาหรือผู้ที่สนใจในวิชาดังกล่าว สามารถอ่านและทำความเข้าใจในเนื้อหาได้ด้วยตนเอง ตำราเล่มนี้ได้เพิ่มเติมเนื้อหาให้ละเอียดขึ้น มีการแทรกผลงานวิจัยของผู้แต่งเข้าไปในบทสุดท้าย เพื่อเป็นประโยชน์ต่อการเรียนการสอน ผู้แต่งหวังว่าตำรานี้คงอำนวยประโยชน์ต่อการเรียนการสอนตามสมควร หากท่านที่นำไปใช้มีข้อเสนอแนะหรือคำติชม ผู้แต่งยินดีรับฟังความคิดเห็น และขอขอบคุณมา ณ โอกาสนี้ด้วย

ชุมพล บุญคุ้มพรภัทร
13 มิถุนายน 2563

กิตติกรรมประกาศ

ตำราการรู้จำแบบฉบับนี้จะสำเร็จลุล่วงไม่ได้ ถ้าไม่ได้รับความความอนุเคราะห์จากผู้มีพระคุณดังต่อไปนี้ ผู้มีพระคุณท่านแรกที่ผู้แต่งใคร่ขอกราบขอบพระคุณคือ ผู้ช่วยศาสตราจารย์ ดร.กรุง สินอภิรมย์สราญ อาจารย์ที่ปรึกษาในระหว่างที่ผู้แต่งกำลังศึกษาในระดับปริญญาเอก ที่ถ่ายทอดวิชาความรู้ทางด้านการรู้จำแบบ และได้แนะนำหัวข้อวิจัยทางด้านปัญหาชั้นข้อมูลอสมดุล ผู้มีพระคุณท่านต่อไปคือ รองศาสตราจารย์ ดร.จีรยุทธ ไชยจารุวณิช อดีตหัวหน้าภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเชียงใหม่ ต้นสังกัดของผู้แต่ง ที่ได้โปรดมอบทุนสนับสนุนเพื่อผลิตผลงานทางวิชาการสำหรับเป็นค่าใช้จ่ายในการผลิตตำราฉบับนี้ ผู้มีพระคุณท่านสุดท้ายคือ ผู้ช่วยศาสตราจารย์ ดร.วัฒนา จินดาหลวง เพื่อนร่วมงานของผู้แต่ง ที่สละเวลามันมีค่าในการตรวจทานและแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่ทุกขั้นตอน เพื่อให้ตำราฉบับนี้สมบูรณ์ที่สุด นอกจากนี้ผู้แต่งขอขอบคุณนางสาวจินตนา ตาคำ นักศึกษาปริญญาโทที่ผู้แต่งเป็นอาจารย์ที่ปรึกษา ที่ได้ช่วยงานในตำแหน่งผู้ช่วยนักวิจัย เหนือสิ่งอื่นใด ผู้แต่งขอขอบพระคุณบุคคลทุกท่าน ซึ่งไม่อาจจะนำมากล่าวได้ทั้งหมด ที่ได้ให้ความช่วยเหลือสนับสนุนและให้กำลังใจด้วยดีตลอดมา

ชุมพล บุญคุ้มพรภัทร

อภิธานศัพท์

ตำราการรู้จำแบบฉบับปรับปรุงใหม่นี้ แต่งและเรียบเรียงขึ้นจากหนังสือและงานวิจัยจากต่างประเทศจำนวนหลายฉบับ เนื่องจากภาษาที่ใช้ในสิ่งพิมพ์เหล่านี้เป็นภาษาอังกฤษ ผู้แต่งจึงจำเป็นต้องแปลเนื้อหาเป็นภาษาไทย แต่คำศัพท์บางคำมีความไม่เหมาะสมเมื่อต้องแปลเป็นภาษาไทย ดังนั้น คำศัพท์คำใดแปลแล้วผู้อ่านสามารถทำความเข้าใจได้ ผู้แต่งจะทำการแปลเป็นภาษาไทย ส่วนคำศัพท์คำใดแปลแล้วอาจทำให้ผู้อ่านสับสน ผู้อ่านจำเป็นต้องใช้คำทับศัพท์แทน โดยรายละเอียดของคำศัพท์แปลที่พบบ่อยในตำราฉบับนี้ แสดงตามตารางด้านล่างนี้

ภาษาไทย	**ภาษาอังกฤษ**
ตัวจำแนก ตัวทำนาย	Classifier
ชั้นข้อมูล	Class
ป้ายชื่อชั้นข้อมูล	Class Label
เซตฝึกฝน แบบฝึกฝน	Training Set, Training Pattern
เซตทดสอบ แบบทดสอบ	Test Set Test Pattern
แบบใหม่	Unseen Pattern
ฟีเจอร์ ตัวแปร	Feature, Attribute, Variable
แบบ	Pattern, Record
โหนด	Node
เส้นเชื่อม	Edge
การกระชับมากเกินไป	Over-fitting
อคติ	Bias
ขีดแบ่ง	Threshold
ปริภูมิ	Space
การเจือปน ความไม่บริสุทธิ์	Impurity
สปริท	Split
ชั้นข้อมูลอสมดุล	Class Imbalance
ชั้นข้อมูลหลัก	Majority Class, Negative Class
ชั้นข้อมูลรอง	Minority Class, Positive Class
ขั้นตอนวิธี	Algorithm

สารบัญ

หน้า

บทที่ 1 บทนำ

วัตถุประสงค์การเรียนรู้ 1

1. ความหมายของการรู้จำแบบ 1

2. โปรแกรมประยุกต์และเซตข้อมูลสำหรับการรู้จำแบบ 3

3. กรณีศึกษา บทบาทของการรู้จำแบบในเกมส์คอมพิวเตอร์ 8

4. กระบวนการเคดีดี 18

5. คู่มือการใช้งานโปรแกรมเวก้า 19

6. กระบวนทัศน์สำหรับการรู้จำแบบ 26

อภิปราย 27

สรุป 27

แบบฝึกหัด 29

บทที่ 2 การแทนแบบ

วัตถุประสงค์การเรียนรู้ 31

1. ปัญหาการจำแนกแบบ 31

2. โครงสร้างข้อมูลสำหรับการแทนแบบ 33

3. หน่วยวัดความใกล้เคียง 37

4. การจัดกลุ่มข้อมูล 46

5. การกำหนดสาระสำคัญของเซตข้อมูล 56

6. การประเมินค่าตัวจำแนกแบบ 57

7. การสกัดฟีเจอร์ 61

8. การคัดเลือกฟีเจอร์ 70

อภิปราย 80

สรุป 80

แบบฝึกหัด 83

สารบัญ (ต่อ)

หน้า

บทที่ 3 ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุด

วัตถุประสงค์การเรียนรู้	85
1. ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุด	85
2. ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเคตัว	88
3. ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเคตัวที่ถูกดัดแปร	91
4. เพื่อนบ้านใกล้รัศมีอาร์	93
5. ตัวจำแนกระยะทางสั้นที่สุด	94
6. ตัวอย่างเพิ่มเติม	96
7. การปรับปรุงประสิทธิภาพการค้นหาเพื่อนบ้านใกล้ที่สุด	103
8. ขั้นตอนวิธีการย่อ	116
9. ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดคลุมเครือ	126
อภิปราย	128
สรุป	129
แบบฝึกหัด	130

บทที่ 4 ตัวจำแนกเบส์

วัตถุประสงค์การเรียนรู้	133
1. ทฤษฎีเบส์	133
2. การประมาณค่าพารามิเตอร์	137
3. ตัวจำแนกนาอีฟเบส์	138
4. เครือข่ายความเชื่อเบร์เซี่ยน	147
อภิปราย	153
สรุป	155
แบบฝึกหัด	156

สารบัญ (ต่อ)

หน้า

บทที่ 5 ต้นไม้การตัดสินใจ

วัตถุประสงค์การเรียนรู้ 159

1. ต้นไม้การตัดสินใจสำหรับการจำแนกแบบ 159
2. ระนาบเกิน 168
3. หน่วยวัดการเจือปน 172
4. สปลิท 177
5. เกนข่าวสาร 182
6. การเล็มกิ่ง 187
7. การอุปนัยต้นไม้การตัดสินใจ 190

อภิปราย 203

สรุป 204

แบบฝึกหัด 206

บทที่ 6 โครงข่ายประสาทและเครื่องกลเวกเตอร์สนับสนุน

วัตถุประสงค์การเรียนรู้ 209

1. ฟังก์ชันการจำแนกเชิงเส้น 209
2. การทำให้อยู่ในรูปปกติ 219
3. การเรียนรู้เพอร์เซ็ปตรอน 222
4. ปัญหาหลายชั้นข้อมูล 233
5. โครงข่ายประสาท 243
6. เครื่องกลเวกเตอร์สนับสนุน 255

อภิปราย 270

สรุป 271

แบบฝึกหัด 273

สารบัญ (ต่อ)

หน้า

บทที่ 7 ปัญหาชั้นข้อมูลอสมดุล

วัตถุประสงค์การเรียนรู้	275
1. นิยามของปัญหา	275
2. หน่วยวัดสมรรถนะ	276
3. การสุ่มเพิ่มและการสุ่มลด	279
4. สโมท	280
5. มิวท์	290
6. เซฟเลเวลกราฟ	293
7. คอร์	295
อภิปราย	297
สรุป	300
แบบฝึกหัด	301
บรรณานุกรม	302
ประวัติผู้แต่ง	304

บทที่ 1

บทนำ

บทแรกของตำราฉบับนี้เริ่มต้นด้วยการแนะนำให้รู้จักกับการรู้จำแบบ โดยมีรายละเอียดเบื้องต้นได้แก่ ความหมายของการรู้จำแบบ โปรแกรมประยุกต์ทางด้านการรู้จำแบบ เซตข้อมูลสำหรับการรู้จำแบบ ซอฟต์แวร์ทางด้านการรู้จำแบบ กรณีศึกษาของการรู้จำแบบในเกมส์คอมพิวเตอร์ กระบวนการเคดีดี และ กระบวนทัศน์สำหรับการรู้จำแบบ

วัตถุประสงค์การเรียนรู้

- สามารถนิยามความหมายของคำว่าการรู้จำแบบได้
- เข้าใจความสำคัญของการรู้จำแบบในเซตข้อมูลและโปรแกรมประยุกต์ต่าง ๆ ได้
- มองเห็นบทบาทของการรู้จำแบบผ่านกรณีศึกษาได้
- สามารถอธิบายขั้นตอนการทำงานเคดีดีได้
- สามารถแยกความแตกต่างกระบวนทัศน์หลักทั้งสองทางด้านการรู้จำแบบได้

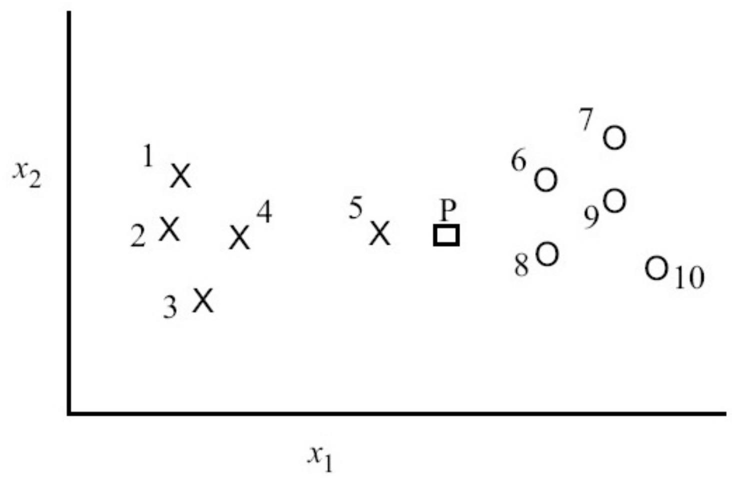
1. ความหมายของการรู้จำแบบ

การรู้จำแบบ (Pattern Recognition) หรือ การจำแนกประเภทข้อมูล (Classification) เป็นศาสตร์ที่ถูกใช้ใน การทำเหมืองข้อมูล (Data Mining) ที่สังเกตและจดจำรูปแบบเหตุการณ์ในอดีตที่เกิดขึ้นซ้ำแล้วซ้ำเล่าเป็นประจำสม่ำเสมอ เพื่อทำนายและแจ้งเตือนเหตุการณ์ในอนาคตที่อาจเกิดขึ้นซ้ำได้อีก โดยเหตุการณ์อาจถูกบรรจุอยู่ในรูปของ ระเบียน (Record) ในฐานข้อมูลหรือไฟล์เป็นต้น หรืออาจกล่าวได้ว่า เหตุการณ์ 1 เหตุการณ์ คือ การแทน (Representation) ของ ระเบียน 1 ระเบียน หรือแบบ (Pattern) 1 แบบ ตำราฉบับนี้จะใช้คำว่าแบบแทนคำว่าระเบียนตลอดเนื้อหาทั้งฉบับ

ตัวอย่างสำคัญของ โปรแกรมประยุกต์ (Application) ทางด้านการรู้จำแบบ คือ การรู้จำเอกสารสื่อประสม (MDR: Multimedia Document Recognition) [13] ที่จัดการกับข้อมูลผสมของ ข้อความ เสียง และ วีดิทัศน์ โดยที่ ข้อความถูกสร้างขึ้นจากอักขระหรือตัวเลขของภาษาใดภาษาหนึ่งหรือหลายภาษา เสียงอยู่ในรูปของบทสนทนาหรือเพลง และ วีดิทัศน์อาจเป็นภาพเดี่ยวหรือภาพเคลื่อนไหว สำหรับตัวอย่างการใช้งานการรู้จำเอกสารสื่อประสมได้แก่การบันทึก ใบหน้า ลายนิ้วมือ และ ลายมือชื่ออาชญากร เป็นรูปภาพเดี่ยว สื่อประสมนี้ถูกนำไปใช้ให้เกิดประโยชน์สำหรับการจับกุมอาชญากรที่แฝงตัวมาในคราบผู้โดยสารในสนามบิน โดยการนำภาพถ่ายอาชญากรไปตรวจสอบกับภาพเคลื่อนไหวจากกล้อง

วงจรปิดสำหรับบันทึกภาพผู้โดยสารในสนามบิน นอกจากนี้ โปรแกรมประยุกต์ทางด้านการรู้จำแบบทั่วไป จำเป็นต้องแปลง ข้อมูลดิบ (Raw Data) ให้อยู่ในรูปแบบที่สามารถประมวลผลได้ เช่น แปลงสื่อประสมเป็น เวกเตอร์ (Vector) ของ ฟีเจอร์ (Feature) เป็นต้น กระบวนการนี้เรียกว่า การเตรียมก่อนการประมวลผล (Preprocessing) โดยที่ฟีเจอร์คือค่าต่าง ๆ ในเวกเตอร์ อาจเป็น จำนวนเต็ม จำนวนจริง นิพจน์บูลีน อักขระ ข้อความ หรือ ชนิดข้อมูลอื่นก็ได้ เช่น ฟีเจอร์ของเวกเตอร์นักศึกษาอาจประกอบไปด้วย ชื่อ นามสกุล เพศ เกรดเฉลี่ย คณะ ภาควิชา สถานภาพ เป็นต้น

ในกระบวนการการรู้จำแบบ แบบแต่ละตัวต้องถูกกำหนด ป้ายชื่อ (Label) หรือ ชั้นข้อมูล (Class) โดยทั่วไปแบบที่อยู่ในชั้นข้อมูลเดียวกันจะมีลักษณะคล้ายกันมากกว่าแบบจากต่างชั้นข้อมูลกัน ในรูปที่ 1.1 แบบทั้งหมดถูกแบ่งเป็น 2 ชั้นข้อมูล ได้แก่ ชั้นข้อมูล X และ ชั้นข้อมูล O โดยที่ แบบอยู่ในเซตข้อมูล 2 มิติ ที่ประกอบไปด้วย แกน x_1, x_2 และ P คือ แบบใหม่ที่ยังไม่เคยถูกกำหนดชั้นข้อมูลมาก่อน วัตถุประสงค์ของการรู้จำแบบคือ การกำหนดป้ายชื่อที่เหมาะสมให้กับแบบใหม่ ในตัวอย่างนี้ X หรือ O จะถูกกำหนดให้เป็นป้ายชื่อของ P ผ่านการวิเคราะห์โดยใช้วิธีการต่าง ๆ เช่น เปรียบเทียบความคล้ายกันของแบบ คำนวณความน่าจะเป็นในการเกิดขึ้นของแบบ หรือ หาเงื่อนไขการเกิดขึ้นของแบบจากชั้นข้อมูลต่าง ๆ เป็นต้น ยกตัวอย่างเช่นถ้าใช้ความคล้ายกันของแบบเป็นตัวตัดสินแล้ว P ควรถูกกำหนดชั้นข้อมูลเป็น X เนื่องจาก แบบที่ใกล้ P มากสุดที่สุดคือแบบ 5 ที่มีชั้นข้อมูลคือ X เป็นต้น



รูปที่ 1.1 เซตตัวอย่างของแบบ [13]

ในระบบที่ถูกสร้างขึ้นเพื่อจำแนกประเภทความสูงของบุคคล จากการพิจารณาฟีเจอร์ น้ำหนัก (Weight) โดยที่ความสูงถูกแบ่งออกเป็น 3 ชั้นข้อมูล ได้แก่ คนที่มีรูปร่าง สูง (Tall) สันทัด (Medium) และ เตี้ย (Short) ตารางที่ 1.1 แสดงข้อมูลที่ถูกเก็บรวบรวมในระบบดังกล่าว ถ้าบุคคลใหม่ที่เพิ่มเข้ามาในระบบมีน้ำหนัก 46 กิโลกรัม เขาอาจถูกกำหนดป้ายชื่อชั้นข้อมูลเป็น “Short” เพราะ 46 มีค่าใกล้เคียง 50 มากที่สุดเมื่อเปรียบเทียบกับค่าอื่น อย่างไรก็ตาม การกำหนดป้ายชื่อโดยอาศัยข้อมูลในตารางนี้มี

ความไม่เหมาะสม เนื่องจากน้ำหนักและส่วนสูงไม่สามารถ เทียบสัมพันธ์ (Correlate) กันได้ สำหรับตัวอย่างนี้ ส่วนสูงเป็นฟีเจอร์ที่มีความเหมาะสมมากกว่าน้ำหนัก หรืออาจกล่าวได้ว่า การสร้างระบบการรู้จำแบบ ข้อมูลที่เก็บรวบรวมมาต้องมีความสัมพันธ์กับชั้นข้อมูล เพื่อให้ประสิทธิภาพการทำนายมีค่าสูงสุด

ตารางที่ 1.1 การจำแนกประเภทมนุษย์

น้ำหนัก (กิโลกรัม)	ชั้นข้อมูล
40	Tall
50	Short
60	Tall
70	Short

2. โปรแกรมประยุกต์และเซตข้อมูลสำหรับการรู้จำแบบ

การรู้จำแบบถูกนำไปใช้ประโยชน์อย่างกว้างขวางในสาขาต่าง ๆ ได้แก่ การเกษตร การศึกษา การรักษาความปลอดภัย การขนส่ง การเงิน การแพทย์ และ ความบันเทิง ผ่านโปรแกรมประยุกต์ต่าง ๆ ได้แก่ เทคโนโลยีชีวภาพ (Biometrics) ชีวสารสนเทศศาสตร์ (Bioinformatics) การวิเคราะห์ข้อมูลสื่อประสม (Multimedia Data Analytic) การรู้จำเอกสาร (Document Recognition) การวินิจฉัยข้อผิดพลาด (Fault Diagnostic) ระบบผู้เชี่ยวชาญ (Expert Systems) การรู้จำอักขระ (Character Recognition) การรู้จำคำพูด (Speech Recognition) และ การรู้จำวัตถุ (Object Recognition) ในรูปภาพ เป็นต้น

เว็บไซต์ UC Irvine Machine Learning Repository [1] เป็นเว็บไซต์ยอดนิยมที่บรรจุเซตข้อมูลประเภทต่าง ๆ สำหรับนำมาใช้เป็น เกณฑ์เปรียบเทียบสมรรถนะ (Benchmark) ในการทดลองเพื่อวัดค่า ความแม่นยำ (Accuracy) ของขั้นตอนวิธีการรู้จำแบบและการทำเหมืองข้อมูล นอกจากนี้นักวิจัยยังสามารถหาเซตข้อมูลที่มีขนาดใหญ่เพิ่มเติมได้จากเว็บไซต์ KDnuggets (http://www.kdnuggets.com) และ เว็บไซต์ KDD (http://www.kdd.ics.uci.edu/index.php) ที่บรรจุเซตข้อมูลจากการแข่งขัน KDD Cup ในแต่ละปี

ตัวอย่างของเซตข้อมูลจากเว็บไซต์ข้างต้น ได้แก่ Iris ที่ปรากฏอยู่ในงานวิจัยจำนวนหลายฉบับ เซตข้อมูลนี้เป็นเซตข้อมูลดอกไม้ไอริสจำนวน 3 สายพันธุ์ (ชั้นข้อมูล) ได้แก่ Iris Setosa, Iris Versicolour และ Iris Virginica ที่มีการเก็บตัวอย่างจากดอกไม้ไอริสสายพันธุ์ละ 50 ดอก รวมทั้งหมดมีจำนวน 150 ดอก โดยฟีเจอร์ในเซตข้อมูลไอริส มีดังต่อไปนี้

- ความยาวกลีบเลี้ยง (Sepal Length)
- ความกว้างกลีบเลี้ยง (Sepal Width)
- ความยาวกลีบดอก (Petal Length)
- ความกว้างกลีบดอก (Petal Width)
- สายพันธุ์ (Class)

แบบจำนวนหนึ่งของเซตข้อมูล Iris แสดงตามตารางที่ 1.2 ความกว้างและความยาวของกลีบเลี้ยงและกลีบดอกมีหน่วยวัดเป็น เซนติเมตร (cm) โดยมีชนิดข้อมูลเป็นจำนวนจริง จุดประสงค์ของการรู้จำแบบคือการสร้างตัวแบบที่สามารถทำนายสายพันธุ์ของดอกไม้ที่ไม่ทราบสายพันธ์ได้ว่าควรมีสายพันธุ์อะไร จากการวิเคราะห์ค่าต่าง ๆ ของกลีบเลี้ยงและกลีบดอก

ตารางที่ 1.2 ตัวอย่างเซตข้อมูล Iris

Sepal Length (cm)	Sepal Width (cm)	Petal Length (cm)	Petal Width (cm)	Class
5.1	3.5	1.4	0.2	Iris Setosa
4.9	3.0	1.4	0.2	Iris Setosa
4.7	3.2	1.3	0.2	Iris Setosa
7.0	3.2	4.7	1.4	Iris Versicolour
6.4	3.2	4.5	1.5	Iris Versicolour
6.9	3.1	4.9	1.5	Iris Versicolour
6.3	3.3	6.0	2.5	Iris Virginica
5.8	2.7	5.1	1.9	Iris Virginica
7.1	3.0	5.9	2.1	Iris Virginica

อีกตัวอย่างหนึ่งที่น่าสนใจได้แก่ Car Evaluation เซตข้อมูลนี้เป็นการประเมินคุณภาพรถยนต์ทั้งหมดจำนวน 1,728 คัน จากการพิจารณาองค์ประกอบต่าง ๆ ของรถยนต์ ซึ่งแจกแจงผลการประเมินออกเป็นคุณภาพรถยนต์จำนวน 4 ระดับ (ชั้นข้อมูล) โดยฟีเจอร์ในเซตข้อมูลการประเมินรถยนต์ มีดังต่อไปนี้

- ราคาที่ซื้อมา (Buying) มีการเก็บค่าเป็น สูงมาก (VHigh) สูง (High) ปานกลาง (Med) และ ต่ำ (Low)
- ค่าบำรุงรักษา (Maint) มีการเก็บค่าเหมือน ราคาที่ซื้อมา
- จำนวนประตู (Doors) มีการเก็บค่าเป็น 2, 3, 4 และ มากกว่า 4 ประตูขึ้นไป (More)

- จำนวนผู้โดยสาร (Persons) มีการเก็บค่าเป็น 2, 3, 4 และ มากกว่า 4 คนขึ้นไป (More)
- ความจุกระโปรงท้ายรถ (Lug_Boot) มีการเก็บค่าเป็น น้อย (Small) ปานกลาง (Med) และ ใหญ่ (Big)
- ความปลอดภัยโดยรวม (Safety) มีการเก็บค่าเป็น ต่ำ (Low) ปานกลาง (Med) และ สูง (High)
- คุณภาพรถยนต์ (Class) แบ่งเป็น 4 ชั้นข้อมูล ได้แก่ ยอมรับไม่ได้ (UnAcc) ยอมรับได้ (Acc) ดี (Good) และ ดีมาก (VGood)

แบบจำนวนหนึ่งของเซตข้อมูล Car Evaluation แสดงตามตารางที่ 1.3 เซตข้อมูลนี้มีความแตกต่างจากเซตข้อมูลไอริสเนื่องจากมีชนิดข้อมูลเป็น ประเภท (Category) จะเห็นได้ว่าเซตข้อมูลสำหรับการรู้จำแบบอาจมีชนิดข้อมูลอะไรก็ได้ หรือมีหลายชนิดข้อมูลในเซตข้อมูลเดียวกันก็ได้

ตารางที่ 1.3 ตัวอย่างเซตข้อมูล Car Evaluation

Buying	Maint	Doors	Persons	Lug_Boot	Safety	Class
VHigh	VHigh	2	2	Small	Low	UnAcc
VHigh	VHigh	2	2	Small	Med	UnAcc
VHigh	Low	2	4	Big	Med	Acc
VHigh	Low	2	4	Big	High	Acc
Low	Low	More	4	Big	Med	Good
Low	Low	4	More	Med	Med	Good
Low	Low	4	More	Med	High	VGood
Low	Low	More	4	Big	High	Vgood

ตัวอย่างเซตข้อมูลอื่น ๆ ที่น่าสนใจ มีดังต่อไปนี้

- Pima Indians Diabetes คือ เซตข้อมูลโรคเบาหวานของชนเผ่าอินเดียนแดง เซตข้อมูลนี้มีการเก็บค่า จำนวนครั้งการตั้งครรภ์ ความเข้มข้นของน้ำตาลในเลือด ความดันเลือด ความหนาของไขมันที่ต้นแขนด้านหลัง เซรั่มอินซูลิน ดัชนีมวลกาย ฟังก์ชันพงศาวลีโรคเบาหวาน และ อายุ เพื่อวินิจฉัยโรคว่าผู้ป่วยมีโอกาสเป็นโรคเบาหวานหรือไม่
- Landsat Satellite (Satimage) คือ เซตข้อมูลภาพถ่ายจากดาวเทียมที่ถูกบันทึกในหลายช่วงคลื่นที่แตกต่างกัน เซตข้อมูลนี้มีการเก็บค่าสีของพิกเซลในรูปของเลขฐานสองจำนวน 8 บิต เพื่อตรวจจับพื้นที่ประเภทต่าง ๆ ได้แก่ ดินสีแดง ทุ่งฝ้าย ดินสีเทา ดินสีเทาชื้น ดินที่มีตอซังพืช ดินผสม และ ดินสีเทาชื้นมาก

- Glass Identification คือ เซตข้อมูลของกระจกที่มีการเก็บค่าดรรชนีการหักเหของแสง ปริมาณร้อยละโดยน้ำหนักของธาตุโซเดียม แมกนีเซียม อลูมิเนียม ซิลิคอน โพแทสเซียม และ เหล็กของกระจก เพื่อตรวจสอบว่าเป็นกระจกจากผลิตภัณฑ์ใด กระจกอาคารที่มีความโปร่งแสง กระจกอาคารที่ไม่มีความโปร่งแสง กระจกยานพาหนะที่มีความโปร่งแสง กระจกยานพาหนะที่ไม่มีความโปร่งแสง ภาชนะแก้ว เครื่องครัวแก้ว กระจกไฟหน้ารถ
- Haberman's Survival คือ เซตข้อมูลคนไข้ที่ผ่านการศัลยกรรมมะเร็งเต้านม เซตข้อมูลนี้มีการเก็บค่าอายุของผู้ป่วย ณ วันที่ได้รับการผ่าตัด จำนวนปีที่ผ่านการผ่าตัด และ จำนวนต่อมน้ำเหลืองที่รักแร้ของผู้ป่วย เพื่อทำนายว่าภายใน 5 ปี หลังการผ่าตัด คนไข้จะมีโอกาสเสียชีวิตหรือรอดชีวิต
- Computer Network Intrusion Detection จากการแข่งขัน KDD Cup ค.ศ. 1999 คือ เซตข้อมูลระบบดักจับการบุกรุกเครือข่ายคอมพิวเตอร์ เซตข้อมูลนี้มีการเก็บค่าฟีเจอร์ต่าง ๆ ของแพ็กเกจออกเป็นกลุ่มต่าง ๆ เช่น ระยะเวลาในการติดต่อสื่อสาร โปรโตคอล เซอร์วิส จำนวนบิตที่สื่อสารระหว่างต้นทางและปลายทาง สถานะของการล็อกอินว่าสำเร็จหรือล้มเหลว จำนวนครั้งการเข้าถึงรูธ ร้อยละของการสื่อสารที่มาจากเซอร์วิสเดิม จำนวนครั้งของการสื่อสารไปยังปลายทางเดิม เพื่อดักจับการบุกรุกประเภทต่าง ๆ เช่น การบุกรุกโดยการเดาสุ่มรหัสผ่าน การส่งรีเควสท์จำนวนวนมหาศาลไปยังเซิร์ฟเวอร์ และ บัฟเฟอร์โอเวอร์โฟลว์ที่ DNS เป็นต้น

Weka (Waikato Environment for Knowledge Analysis) [15] คือ ซอฟต์แวร์สำหรับงานทางด้าน การรู้จำแบบ โดยบรรจุขั้นตอนวิธีทางด้าน การทำเหมืองข้อมูล (Data Mining) และ การเรียนรู้ของเครื่องจักร (Machine Learning) สำหรับแก้ปัญหา การจำแนกประเภท การจัดกลุ่มข้อมูล (Clustering) กฎความสัมพันธ์ (Association Rule) และ ปัญหาอื่น ๆ ที่เกี่ยวข้อง ตัวอย่างของหน้าต่างโปรแกรมแสดงตามรูปที่ 1.2

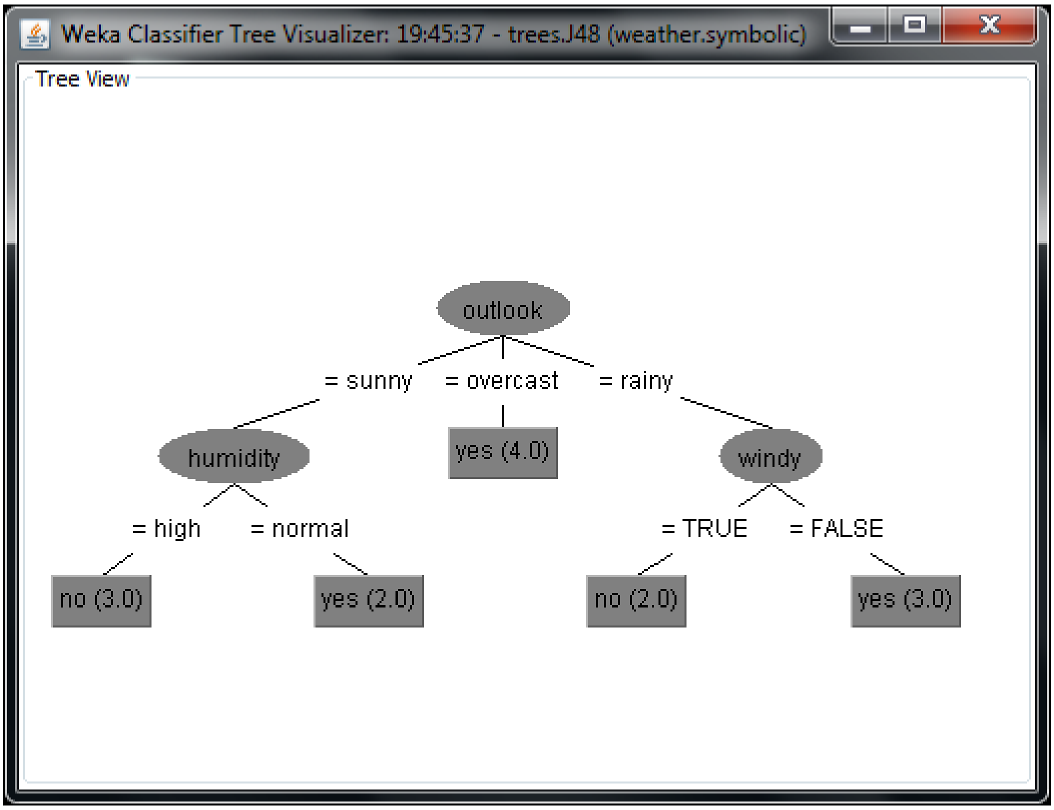
เซตข้อมูลในรูปที่ 1.2 คือ Weather ซึ่งเป็นเซตข้อมูลสภาพอากาศสำหรับการทำนายว่าสามารถออกไปเล่นนอกบ้านได้หรือไม่ จากการพิจารณาปัจจัยทางธรรมชาติต่าง ๆ โดยฟีเจอร์ทั้งหมดในเซตข้อมูลนี้ มีดังต่อไปนี้

- ทัศนียภาพ (Outlook) มีการเก็บค่าเป็น แดดออก (Sunny) มืดครึ้ม (Overcast) และ ฝนตก (Rainy)
- อุณหภูมิ (Temperature) มีการเก็บค่าเป็นจำนวนจริง

- ความชื้น (Humidity) มีการเก็บค่าเป็นจำนวนจริง
- ลมแรง (Windy) มีการเก็บค่าเป็น จริง (TRUE) หรือ เท็จ (FALSE)
- การเล่นนอกบ้านได้ (Play: Class) มีการเก็บค่าเป็น ใช่ (Yes) และ ไม่ใช่ (No)

ตัวแบบในรูปที่ 1.2 คือ ต้นไม้การตัดสินใจ ซึ่งมีองค์ประกอบต่าง ๆ ได้แก่ โหนดภายใน (สัญลักษณ์รูปวงรี) แทนฟีเจอร์ในเซตข้อมูล เส้นเชื่อมแทนค่าที่เป็นไปได้ของฟีเจอร์ และ โหนดใบ (สัญลักษณ์รูปสี่เหลี่ยม) แทนชั้นข้อมูลที่ต้องการทำนาย โดยการทำนายของต้นไม้การตัดสินใจอยู่ในรูปของกฎ "ถ้า-แล้ว" ยกตัวอย่างเช่น ถ้าทัศนียภาพมีแดดออกและมีความชื้นสูง แล้วไม่สามารถเล่นได้เป็นต้น ในตัวอย่างนี้ไม่จำเป็นต้องพิจารณาค่าอุณหภูมิและลมแรงเนื่องจากฟีเจอร์ทั้งสองนี้ไม่ได้อยู่ในเงื่อนไขของกฎ

ต้นไม้การตัดสินใจมีจุดเด่นคือสามารถเห็นโครงสร้างของตัวแบบได้ง่าย และสามารถอธิบายขั้นตอนการทำนายโดยใช้ภาษามนุษย์ได้ในรูปของเงื่อนไขถ้าแล้ว นอกจากนี้ตัวทำนายชนิดนี้ยังใช้ทำนายได้กับเซตข้อมูลที่มีชนิดข้อมูลเป็นตัวเลขหรือเป็นประเภทก็ได้ รายละเอียดของตัวแบบชนิดนี้จะกล่าวถึงอีกครั้งในบทที่ 5



รูปที่ 1.2 ซอฟต์แวร์ Weka

3. กรณีศึกษา บทบาทของการรู้จำแบบในเกมส์คอมพิวเตอร์

บทนี้ได้ยกตัวอย่างกรณีศึกษา บทบาทของการรู้จำแบบในเกมส์คอมพิวเตอร์ [12] โดยที่ เกมส์คอมพิวเตอร์จัดเป็นโปรแกรมประยุกต์เฉพาะด้าน และมีสมมติฐานว่าเหตุการณ์ทุกเหตุการณ์ที่เกิดขึ้นเป็นผลสืบเนื่องมาจากปรากฏการณ์ตั้งต้นในลักษณะความสัมพันธ์ ถ้า-แล้ว นอกจากนี้เกมส์ยังเป็นระบบพลวัต (Dynamic System) กล่าวคือผู้เล่นมีอิสระในการเล่น ด้วยการเล่นแต่และแบบที่แตกต่างกันย่อมส่งผลให้ได้ผลลัพธ์ที่แตกต่างกันตามไปด้วย ดังนั้นเกมส์จึงมีหลายเหตุการณ์ที่เป็นไปได้ในการเล่นแต่ละครั้งไม่ตายตัว ขึ้นกับว่าผู้เล่นมีการตัดสินใจอย่างไรระหว่างเล่นเกมส์

การรู้จำแบบถูกนำมาใช้สำหรับสร้าง ปัญญาประดิษฐ์ หรือ เอไอ (AI) เพื่อเรียนรู้พฤติกรรมการเล่นที่หลากหลายของมนุษย์แล้วหาวิธีตอบโต้ที่เหมาะสม เช่น ในเกมส์ต่อสู้อาจแบ่งผู้เล่นทั้งหมดออกเป็นประเภทต่าง ๆ ได้แก่ แบบบุก แบบตั้งรับ และ แบบสมดุล ดังนั้นเอไอไม่ควรใช้วิธีการเดียวในการรับมือคู่ต่อสู้ทุกประเภทเนื่องจากเมื่อผู้เล่นคุ้นเคยกับการเล่นของเอไอ แล้วจะสามารถหาวิธีในการเอาชนะได้ ในทางกลับกัน เอไอที่ดีควรมีวิธีที่แตกต่างกันในการจัดการกับผู้เล่นทุกแบบและปรับตัวตามพฤติกรรมที่เปลี่ยนไปของผู้เล่นอยู่เสมอ เพื่อให้ผู้เล่นพบกับความยากลำบากในการเล่น

จุดประสงค์ของการรู้จำแบบที่นำมาประยุกต์ใช้ในเกมส์คอมพิวเตอร์คือ การพัฒนาให้เอไอมีความสามารถในการรู้จำพฤติกรรมของผู้เล่นที่เป็นมนุษย์ได้โดยใช้ ระบบการตัดสินใจ (Decision-making System) ซึ่งแสดงตามแผนภาพในรูปที่ 1.3 ระบบนี้ทำหน้าที่สร้าง แนวคิด (Concept) จากการกำหนดสาระสำคัญของแบบต่าง ๆ ภายในเกมส์ โดยที่ แบบหนึ่งแบบแทนเหตุการณ์ที่เป็นไปได้ 1 เหตุการณ์ซึ่งเป็นผลลัพธ์มาจากการตัดสินใจแต่ละครั้งของผู้เล่น องค์ประกอบแต่ละส่วนในแผนภาพมีรายละเอียด ดังต่อไปนี้

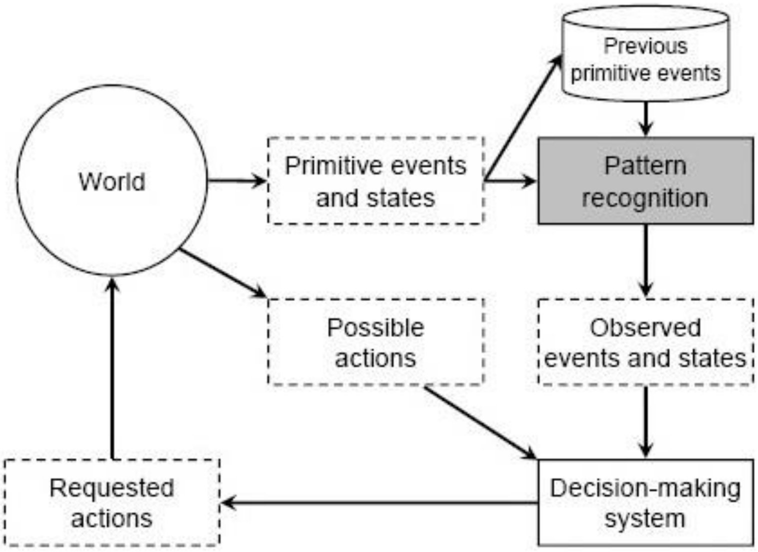
- **World** คือทุกสิ่งทุกอย่างภายในเกมส์ประกอบไปด้วย ผู้เล่น ศัตรู สิ่งของ แผนที่ รวมถึงระบบการเล่นและกฎเกณฑ์ต่าง ๆ ภายในเกมส์ หรืออาจกล่าวได้ว่าองค์ประกอบนี้คือ รหัสต้นฉบับ (Source Code) ทั้งหมดที่บรรจุอยู่ในสื่อบันทึกข้อมูลนั่นเอง
- **Primitive Events and States** คือเหตุการณ์และสถานะในอดีตทั้งหมดของผู้เล่น รวมถึงพฤติกรรมทุกสิ่งทุกอย่างตั้งแต่เริ่มเล่นจนถึงจบเกมส์
- **Previous Primitive Events** คือเหตุการณ์ในอดีตเพียงบางส่วนที่ถูกเลือกบันทึกลงในฐานข้อมูล เพื่อเตรียมส่งให้การรู้จำแบบประมวลผลต่อไป สาเหตุที่เก็บเหตุการณ์เพียงบางส่วนเนื่องจาก เหตุการณ์ทุกเหตุการณ์ในเกมส์ไม่ได้มีความสำคัญเท่ากันหมด ระบบสนับสนุนการตัดสินใจจะเลือกเก็บเหตุการณ์ที่น่าสนใจสูง หรือมีความถี่ในการเกิดขึ้นบ่อยครั้ง นอกจากนี้ ยัง

เพื่อประหยัดพื้นที่การจัดเก็บข้อมูล โดยส่งผลให้คอมพิวเตอร์ประมวลผลได้เร็วเพราะฐานข้อมูลมีขนาดกระชับ

- **Pattern Recognition** คือเทคนิคการรู้จำแบบที่ใช้สำหรับวิเคราะห์พฤติกรรมของผู้เล่น โดยส่วนมากจะเป็นเทคนิคทางด้านการทำเหมืองข้อมูล ได้แก่ การจัดกลุ่มข้อมูล และ การจำแนกประเภท
- **Observed Events and States** คือเหตุการณ์และสถานะในปัจจุบันของผู้เล่นที่ถูกการรู้จำแบบสังเกต และกำลังจะถูกประมวลผลต่อไป
- **Decision-making System** คือระบบการตัดสินใจมีหน้าที่ส่งชุดคำสั่งให้กับเอไอในการตอบโต้ผู้เล่นได้อย่างเหมาะสมในแต่ละสถานการณ์
- **Possible Actions** คือคำสั่งของศัตรูที่เป็นไปได้ทั้งหมด หมายถึงความเป็นไปได้ทั้งหมดที่ศัตรูมีโอกาสเลือกตอบโต้ผู้เล่น ซึ่งอาจมีทั้งชุดคำสั่งที่มีประสิทธิภาพหรือไม่มีประสิทธิภาพก็ได้
- **Requested Actions** คือคำสั่งที่ปัญญาประดิษฐ์สั่งให้ศัตรูตอบโต้ต่อการเล่นของผู้เล่นอย่างเหมาะสม

ยกตัวอย่างการทำงานของระบบสนับสนุนการตัดสินใจในแผนภาพนี้จากเกมส์ประเภทต่อสู้หนึ่งต่อหนึ่ง ระหว่าง ผู้เล่น (มนุษย์) และ ศัตรู (ปัญญาประดิษฐ์) ในระหว่างการเล่นเกมส์ ผู้เล่นอาจโจมตีศัตรูได้หลากหลายวิธีขึ้นอยู่กับสถานการณ์ในแต่ละช่วงเวลา เช่น โจมตีระยะประชิดตัว โจมตีระยะไกล หรือโจมตีทางอากาศ เป็นต้น อย่างไรก็ดี ในสถานการณ์ปกติ ผู้เล่นอาจบังคับตัวละครในแบบที่ตัวเองถนัดที่สุด เช่น ถ้าผู้เล่นชื่นชอบการโจมตีทางอากาศ ชุดคำสั่งในการโจมตีทางอากาศจะมีปริมาณมากกว่าชุดคำสั่งในการโจมตีภาคพื้นดิน ดังนั้นระบบสนับสนุนการตัดสินใจจะเน้นบันทึกชุดคำสั่งการโจมตีทางอากาศมากกว่าการโจมตีภาคพื้นดิน เพื่อประมวลผลและจดจำรูปแบบการโจมตีของผู้เล่น แล้วสั่งงานให้ศัตรูเน้นป้องกันและตอบโต้การโจมตีทางอากาศได้อย่างเหมาะสม เห็นได้ว่าระบบสนับสนุนการตัดสินใจสามารถเลือกรูปแบบเพื่อสั่งให้ปัญญาประดิษฐ์ตอบสนองต่อผู้เล่นได้หลากหลายวิธี ขึ้นอยู่กับพฤติกรรมของผู้เล่นแต่ละคน ซึ่งมีความฉลาดมากกว่าปัญญาประดิษฐ์ที่มีชุดคำสั่งเพียงชุดเดียวและไม่สามารถปรับตัวไปตามพฤติกรรมที่แตกต่างของผู้เล่นแต่ละคนได้

การรู้จำแบบถูกนำไปประยุกต์ใช้ในเกมส์คอมพิวเตอร์หลากหลายประเภท เช่น ต่อสู้ (Fighting) วางแผนการรบ (RTS: Real-time Strategy) และ กีฬา (Sport) เป็นต้น โดยรายละเอียดการนำการรู้จำแบบไปประยุกต์ในเกมส์แต่ละประเภท มีดังต่อไปนี้



รูปที่ 1.3 ความสัมพันธ์ระหว่าง โลก การรู้จำแบบ และ ระบบการตัดสินใจ [12]

- ***ต่อสู้*** ตัวอย่างของเกมส์ประเภทนี้ได้แก่ Mortal Kombat, Tekken, King of Fighters, Marvel vs. Capcom และ Street Fighter (รูปที่ 1.4) ซึ่งมีลักษณะการเล่นเป็นการต่อสู้หนึ่งต่อหนึ่งระหว่าง ผู้เล่น (มนุษย์) และ ศัตรู (ปัญญาประดิษฐ์) บทบาทของการรู้จำแบบถูกใช้ในการวางแผนให้ปัญญาประดิษฐ์ตอบโต้การเคลื่อนไหวที่ปรากฏขึ้นบ่อยครั้งของมนุษย์ได้อย่างมีประสิทธิภาพ (รายละเอียดเพิ่มเติมได้กล่าวไว้แล้วในหน้าที่ผ่านมา) สำหรับเกมส์แนวนี้ วัตถุหลักที่เกี่ยวข้องในเกมส์มีเพียง 2 วัตถุได้แก่ผู้เล่นและศัตรูเท่านั้น ทำให้ง่ายต่อการประยุกต์ใช้การรู้จำแบบในการเจาะจงไปยังวัตถุที่สนใจเท่านั้น

รูปที่ 1.4 เกมส์แนวต่อสู้ Street Fighter V

(**ที่มา:** https://www.mango.in.th/product/Street-Fighter-V/)

- **วางแผนการรบ** ตัวอย่างของเกมส์ประเภทนี้ได้แก่ Warcraft, StarCraft, Age of Empires, Company of Heroes และ Command and Conquer (รูปที่ 1.5) ซึ่งมีลักษณะการเล่นเป็นการควบคุมกองทัพขนาดใหญ่ที่ประกอบไปด้วยยูนิตและสิ่งก่อสร้างปริมาณมาก ผู้เล่นต้องจัดสรรทรัพยากรประเภทต่าง ๆ เช่น เงิน ท่อนไม้ แร่ธาตุ น้ำ น้ำมัน ในการสร้างหรือฝึกฝนยูนิตแต่ละประเภท เช่น ทหาร รถถัง เครื่องบิน เรือ เป็นต้น นอกจากนี้ ผู้เล่นต้องสร้างสิ่งปลูกสร้างและจัดวางตำแหน่งภายในอาณาเขตของตนเอง เช่น โรงไฟฟ้า ศูนย์ฝึกทหารใหม่ โรงประกอบรถ สนามบิน อู่ต่อเรือ เป็นต้น เมื่อยูนิตและสิ่งปลูกสร้างถูกจัดเตรียมพร้อมแล้ว ขั้นตอนต่อไปคือการจัดกองทัพในการบุกฐานทัพของข้าศึก จะเห็นได้ว่าการเล่นเกมส์ไม่ได้มีเพียงการต่อสู้เพียงอย่างเดียว แต่ต้องจัดสรรทรัพยากรที่มีอยู่อย่างจำกัดรวมถึงวางแผนในการโจมตีข้าศึกด้วย ดังนั้นระบบเกมส์โดยรวมมีความซับซ้อนมากกว่าเกมส์ต่อสู้ บทบาทของการรู้จำแบบถูกใช้ในการลดความเสียหายจากภัยคุกคาม และวางยุทธศาสตร์ในการโต้กลับผู้เล่นสำหรับเกมส์แนวนี้ วัตถุในเกมส์มีจำนวนมากจึงไม่สามารถแยกพิจารณาเป็นรายวัตถุได้ จึงจำเป็นต้องจัดกลุ่มของยูนิตแทน เช่น กลุ่มกองทัพในการโจมตีทางบก เป็นต้น และ ชุดคำสั่งที่ส่งให้ปัญญาประดิษฐ์ตัดสินใจ ก็จะสั่งการในลักษณะกลุ่มของยูนิตเช่นกัน

รูปที่ 1.5 เกมส์แนววางแผนการรบ C&C Generals Zero Hour

(**ที่มา:** https://www.appnationconference.com/article/generals)

- **กีฬา** ตัวอย่างของเกมส์ประเภทนี้ได้แก่ Punch Out, NBA Jam, Madden NFL, FIFA และ Pro Evolution Soccer (รูปที่ 1.6) ซึ่งมีลักษณะการเล่นเป็นการบังคับนักกีฬาประเภทต่าง ๆ แบ่งเป็น ประเภทเดี่ยว (ต่อยมวย มวยปล้ำ ฟันดาบ) ประเภทคู่ (เทนนิส ปิงปอง แบดมินตัน) หรือ ประเภททีม (ฟุตบอล บาสเกตบอล อเมริกันฟุตบอล) ระบบของเกมส์จึงเหมือนกับกฎกติกาของกีฬาจริง บทบาทการรู้จำแบบถูกใช้ในการอ่านเกมส์การแข่งขันเพื่อแก้เกมส์ของผู้เล่นสำหรับเกมส์แนวนี้ การแพ้ชนะไม่ได้ขึ้นอยู่กับความเชี่ยวชาญในการบังคับนักกีฬาเพียงอย่างเดียว แต่ยังขึ้นอยู่กับการวางแผนการเล่นรวมถึงการแก้เกมส์ระหว่างการแข่งขันอีกด้วย เนื่องจากแผนการเล่นแบบต่าง ๆ มักจะมีการได้เปรียบและเสียเปรียบกัน เช่น ในกีฬาฟุตบอล การป้องกันโดยวางกองหลังตัวกลางเพียง 3 คน ย่อมเสียเปรียบฝ่ายบุกที่โจมตีทางกราบของสนามเนื่องจากเป็นพื้นที่ที่แนวป้องกันหละหลวมเป็นต้น ดังนั้น การรู้จำแบบจะพิจารณาแผนการเล่นควบคู่ไปกับพฤติกรรมการเล่นของผู้เล่น เพื่อเลือกทางเลือกที่เหมาะสมให้กับปัญญาประดิษฐ์ต่อไป

รูปที่ 1.6 เกมส์แนวกีฬา PES 2017

(**ที่มา:** https://chocobonplan.com/tests/jeux-video/ps4/pes-2017-ps4-un-jeu-moitie-fini/)

มุมมองทัศมิติ (Perspective View) ของการรู้จำแบบในเกมส์คอมพิวเตอร์แบ่งออกได้เป็น ระดับการตัดสินใจ (Decision-making Levels) ทัศนคติต่อผู้เล่น (Stance towards Players) และ กราฟเกมส์ (Game Graph)

3.1 ระดับการตัดสินใจ

การสร้างการตัดสินใจแบ่งออกเป็น 3 ระดับ ได้แก่ ยุทธศาสตร์ (Strategical) ปฏิบัติการ (Operational) และ กลยุทธ์ (Tactical)

- **ยุทธศาสตร์** การรู้จำแบบถูกนำมาประยุกต์ใช้ในขณะ ออฟไลน์ (Offline) หรือ ฉากหลัง (Background) ซึ่งไม่ใช่ช่วงเวลาที่ผู้เล่นกำลังเล่นเกมส์อยู่ แต่เป็นช่วงเวลาที่เกมส์กำลังถูกพัฒนาโดยผู้ผลิตเกมส์ อาจอยู่ในช่วงกำลังทดสอบเกมส์เพื่อทดลองการทำงานของปัญญาประดิษฐ์ในกรณีต่าง ๆ ที่เป็นไปได้ทั้งหมด จุดประสงค์เพื่อหารูปแบบพื้นฐานเพื่อตอบโต้ผู้เล่นสำหรับสถานการณ์ต่าง ๆ การทดสอบเหล่านี้ต้องมีจำนวนครั้งที่มากพอเพื่อป้องกันการตัดสินใจผิดพลาดของปัญญาประดิษฐ์ เพราะอาจส่งผลถึงความเสียหายร้างแรงที่จะตามมา เช่น การที่ปัญญาประดิษฐ์เดินตกเหวตายเอง โดยที่ผู้เล่นไม่จำเป็นต้องโจมตีแม้แต่เพียงครั้งเดียว เป็นต้น การรู้จำแบบในช่วงเวลานี้ใช้เวลายาวนาน เนื่องจากเกี่ยวข้องกับข้อมูลปริมาณมหาศาล ได้แก่ ตัวละคร สิ่งของ ฉาก เหตุการณ์ และ กฎเกณฑ์ต่าง ๆ ทั้งหมดในโลกของเกมส์ ดังนั้นจึงต้องประมวลผลในช่วงที่เกมส์ยังไม่ออกวางจำหน่าย การตัดสินใจในระดับนี้ของปัญญาประดิษฐ์มีรูปแบบเป็น *“อะไรจะเกิดขึ้นถ้า”* (What-if) เช่น ถ้าผู้เล่นถนัดการต่อสู้แบบประชิดตัว แนวทางการตอบโต้ของปัญญาประดิษฐ์จะตัดสินใจเว้นระยะห่างให้ได้มากที่สุดเท่าที่จะทำได้ เพื่อควบคุมความได้เปรียบระหว่างการต่อสู้ เป็นต้น การตัดสินใจที่ได้ในระดับนี้มีลักษณะพื้นฐานคร่าว ๆ เพื่อตอบโต้ผู้เล่นปกติทั่วไป ไม่มีการเจาะจงตอบโต้แนวทางการเล่นของผู้เล่นแบบใดแบบหนึ่งเป็นพิเศษ

- **ปฏิบัติการ** การรู้จำแบบถูกนำมาประยุกต์ใช้ขณะ ออนไลน์ (Online) หรือ ในสนามการแข่งขัน (in-the-Field) ซึ่งเป็นช่วงเวลาที่ผู้เล่นกำลังเล่นเกมส์อยู่ และเป็นช่วงเวลาที่เกมส์ออกวางจำหน่ายเรียบร้อยแล้ว จุดประสงค์เพื่อค้นหารูปแบบพฤติกรรมของผู้เล่นแบบเฉพาะเจาะจงรายบุคคล ซึ่งต่างจากการรู้จำแบบในระดับยุทธศาสตร์ที่ค้นหาเพียงพฤติกรรมโดยรวมของผู้เล่นทั่วไปเท่านั้น นอกจากนี้การรู้จำแบบสนใจ สรรพสิ่งเดี่ยว (Atomatory Entity) ซึ่งหมายถึงวัตถุเพียงหน่วยเดียวเท่านั้น ยกตัวอย่างเช่นในเกมส์ที่ผู้เล่นบังคับตัวละครได้หลายตัวละครพร้อมกันเป็นกลุ่ม การรู้จำแบบจะประมวลผลพฤติกรรมเพียงยูนิตใดยูนิตหนึ่งที่กำลังถูกผู้เล่นบังคับอยู่เท่านั้น สาเหตุที่เจาะจงเพียงยูนิตเดียวเนื่องจากต้องการให้ได้เวลาอันรวดเร็วในการทำนายพฤติกรรมของผู้เล่นขณะเล่นเกมส์ใน เวลาจริง (Real-Time) ขณะเล่นเกม จึงไม่สามารถให้ผู้เล่นรอการประมวลผลยาวนานได้ ดังนั้นผลการทำนายจึงอาจไม่แม่นยำเท่าที่ควร และความ

ผิดพลาดที่เกิดขึ้นไม่สามารถกลับไปแก้ไขได้ ปัญหานี้เรียกว่า ปัญหาเปลี่ยนแปลงไม่ได้ (Irrevocable Problem)

- **กลยุทธ์** ระดับนี้ถูกผสมผสานระหว่างระดับยุทธศาสตร์และระดับปฏิบัติการ การรู้จำแบบสนใจสรรพสิ่ง (Entity) ซึ่งแทนกลุ่มของวัตถุ กล่าวคือการรู้จำแบบจะประมวลผลกลุ่มของยูนิต ซึ่งต่างจากการรู้จำแบบในอีกสองระดับที่ประมวลผลยูนิตเดียวหรือยูนิตทั้งหมด นอกจากนี้ การประมวลผลเกิดขึ้นในขณะที่เกมส์ออกวางจำหน่ายแล้ว แต่ไม่ได้ประมวลผลในเวลาจริงขณะเล่นเกมส์ การประมวลผลอาจถูกซ่อนไว้ในระหว่างฉากที่ไม่มีการบังคับใด ๆ จากผู้เล่นเกิดขึ้น เช่น อาจมีภาพยนตร์ระหว่างฉากให้รับชมระหว่างนั้นก็แอบประมวลผลพฤติกรรมผู้เล่นไปด้วย การรู้จำแบบในระดับนี้เมื่อเปรียบเทียบกับระดับยุทธศาสตร์แล้วประมวลผลได้เร็วกว่า และเมื่อเปรียบเทียบกับระดับปฏิบัติการแล้วทำนายได้แม่นยำกว่า สรุปได้ว่าการรู้จำแบบในระดับนี้อยู่กึ่งกลางระหว่างอีกสองระดับนั่นเอง

3.2 ทัศนคติต่อผู้เล่น

ทัศนคติต่อผู้เล่นหมายถึงปัญญาประดิษฐ์มีท่าทีอย่างไรต่อผู้เล่น แบ่งออกเป็น ศัตรู (Enemy) พันธมิตร (Ally) และ ไม่ฝักใฝ่ฝ่ายใด (Neutral) โดยมีรายละเอียด ดังต่อไปนี้

- **ศัตรู** ทัศนคติประเภทนี้พบได้ในเกมส์ส่วนใหญ่ทั่วไปที่บทบาทของปัญญาประดิษฐ์มีหน้าที่ตัดสินใจเลือกการกระทำในลักษณะให้โทษกับผู้เล่น ยกตัวอย่างเช่น ปัญญาประดิษฐ์ในเกมส์ต่อสู้จะโจมตีผู้เล่นเพื่อก่อให้เกิดความเสียหายเป็นต้น การรู้จำแบบถูกใช้ในการเรียนรู้ โมดัสโอเปรานดิ (Modus Opeandi) หรือพฤติกรรมของมนุษย์ กล่าวคือ การเล่นที่แตกต่างกันของผู้เล่นย่อมส่งผลให้ปัญญาประดิษฐ์ตอบสนองต่อผู้เล่นแตกต่างกันไปด้วย ปัญญาประดิษฐ์ไม่ควรใช้ชุดคำสั่งเพียงชุดเดียวตอบสนองต่อผู้เล่นทุกประเภท เพราะอาจทำให้ขาดประสิทธิภาพได้
- **พันธมิตร** บทบาทของทัศนคติประเภทนี้มีหน้าที่ตัดสินใจเลือกการกระทำในลักษณะให้คุณกับผู้เล่น โดยมีท่าทีเป็นมิตรกับผู้เล่นและเป็นศัตรูกับปัญญาประดิษฐ์ ยกตัวอย่างเช่น เอ็นพีซี (NPC: Non Player Character) ในเกมส์ต่อสู้ ซึ่งเป็นตัวละครที่ผู้เล่นไม่สามารถบังคับได้และมีการตัดสินใจของตนเองโดยอัตโนมัติ มีหน้าที่คอยให้ความช่วยเหลือผู้เล่นในการต่อสู้กับศัตรูหรือช่วยฟื้นฟูความสูญเสียที่ผู้เล่นได้รับ การรู้จำแบบในเกมส์วางแผนการรบทำหน้าที่คอยรายงานการเคลื่อนไหวของศัตรูในขณะที่โจมตีผู้เล่น หรือแนะนำแนวทางการโจมตีที่มีประสิทธิภาพ เป็นต้น

- **ไม่ฝักใฝ่ฝ่ายใด** บทบาทของทัศนคติประเภทนี้ไม่ให้ทั้งคุณหรือโทษกับผู้เล่น ส่วนใหญ่พบได้ในเกมส์กีฬา ซึ่งการรู้จำแบบมีหน้าที่คอยควบคุมการเคลื่อนไหวของมุมกล้องให้เหมาะสมกับพฤติกรรมของผู้เล่น เช่นในเกมส์ฟุตบอลอาจแบ่งเกมส์รุกของผู้เล่นได้เป็น 2 ประเภท ได้แก่ การต่อบอลสั้น และ การโยนบอลยาว ในกรณีต่อบอลสั้น การรู้จำแบบจะทำหน้าที่ซูมมุมกล้องเข้าหาผู้เล่น เพื่อให้ผู้เล่นได้เห็นรายละเอียดของนักฟุตบอลในบริเวณรอบข้างทำให้ง่ายต่อการส่งบอล แต่ในกรณีโยนบอลยาวการรู้จำแบบจะทำหน้าที่ซูมมุมกล้องออกจากผู้เล่น เพื่อให้เห็นภาพรวมของสนามฟุตบอลทำให้ง่ายต่อการส่งบอลระยะไกล เป็นต้น นอกจากการปรับมุมกล้องแล้ว การรู้จำแบบอาจมีการปรับแต่ง ส่วนต่อประสาน (Interface) ตามความต้องการของผู้เล่น กล่าวคือผู้เล่นแต่ละคนอาจมีส่วนต่อประสานที่หน้าตาแตกต่างกัน ยกตัวอย่างเช่น ผู้เล่นที่มีความเชี่ยวชาญในการเล่นย่อมมีส่วนต่อประสานที่มีรายละเอียดน้อยเพราะผู้เล่นเข้าใจในเกมส์เพียงพออยู่แล้ว แต่ผู้เล่นหน้าใหม่อาจจำเป็นต้องมีส่วนต่อประสานที่มีรายละเอียดมากมายเพื่อช่วยเหลือให้ผู้เล่นสามารถเล่นเกมส์ได้ เป็นต้น

3.3 **กราฟเกมส์**

เกมส์และภาพยนตร์มีการดำเนินเรื่องที่ความแตกต่างกันคือ เรื่องราว (Story) ในภาพยนตร์ดำเนินเรื่องในลักษณะ แนวตรง (Linear) ไม่อนุญาตให้มี การเบี่ยงเบน (Diversion) ของเนื้อเรื่อง กล่าวคือการดำเนินเรื่องของบทภาพยนตร์จะเหมือนเดิมทุกครั้งไม่มีการเปลี่ยนแปลง ไม่ว่าผู้ชมจะรับชมภาพยนตร์กี่รอบก็ตาม ในขณะที่เกมส์ดำเนินเรื่องโดยมี เจตจำนงเสรี (Free Will) เป็นตัวขับเคลื่อน กล่าวคือการเล่นเกมส์แต่ละรอบของผู้เล่นอาจจะให้ผลลัพธ์ที่ตามมาแตกต่างกัน ขึ้นอยู่กับการตัดสินใจในแต่ละฉากของผู้เล่นเป็นต้น

การดำเนินเรื่องภายในเกมส์สามารถถูกแทนด้วยโครงสร้างข้อมูลประเภท กราฟ (Graph) โดยที่แต่ละองค์ประกอบของกราฟ มีรายละเอียดดังนี้

- **จุดต่อ (Node)** เป็นตัวแทน สถานะ (State) หรือสภาพของตัวละคร ยกตัวอย่างสถานะของผู้เล่นเช่น มีชีวิต บาดเจ็บ และ ตาย เป็นต้น นอกจากนี้จุดต่ออาจใช้แทนสถานะของเหตุการณ์หรือวัตถุภายในเกมส์ได้อีกด้วย
- **เส้นเชื่อมแบบมีทิศทาง (Direct Arch)** เป็นตัวแทน การกระทำ (Action) หรือ การใส่คำสั่งภายในเกมส์ อาจมาจากผู้เล่นเองหรือปัญญาประดิษฐ์ก็ได้ ยกตัวอย่างการกระทำของผู้เล่นเช่น โจมตี ถูกโจมตี ป้องกัน และ หลบหนี เป็นต้น

รูปที่ 1.4 แสดงกราฟเกมส์แบบต่าง ๆ โดยที่ S_i, S_j, S_k, S_l, S_m แทนสถานะต่าง ๆ ของผู้เล่น และ a_1, a_2, a_3, a_4 แทนการกระทำต่าง ๆ ของผู้เล่น ตัวอย่างของความสัมพันธ์ระหว่างสถานะและการกระทำได้แก่ ถ้าผู้เล่นอยู่สถานะบาดเจ็บและการกระทำคือถูกโจมตีแล้วสถานะจะเปลี่ยนเป็นตาย แต่ถ้าอยู่ในสถานะบาดเจ็บและการกระทำคือหนีแล้วสถานะจะเปลี่ยนเป็นรอดชีวิต เป็นต้น เงื่อนไขเหล่านี้แสดงได้จากกราฟตามรูปที่ 1.4 (b) โดยที่ สถานะ S_i, S_j และ S_k แทน บาดเจ็บ ตาย และ รอดชีวิต ตามลำดับ ในขณะที่ การกระทำ a_1 และ a_2 แทน ถูกโจมตี และ หนี ตามลำดับ

การที่เกมส์จะมีอิสระในการเล่นหรือไม่มี สามารถวิเคราะห์ได้จากเทอมที่เกี่ยวข้องกับจำนวนเส้นเชื่อมแบบมีทิศทางที่ออกหรือเข้าจุดต่อ โดยมีรายละเอียดดังต่อไปนี้

- **ดีกรีออก (Outdegree)** คือ จำนวนของเส้นเชื่อมแบบมีทิศทางที่ออกจากจุดต่อ ถ้าดีกรีออกมีค่าสูงหมายถึงผู้เล่นมีอิสระในการเล่น กล่าวคือ ผู้เล่นมีโอกาสในการเลือกการทำต่าง ๆ ได้หลากหลายวิธีจากแต่ละสถานะของผู้เล่นเอง หรืออาจจะเป็นเหตุการณ์และวัตถุใด ๆ ที่เกี่ยวข้องกับผู้เล่นก็ได้

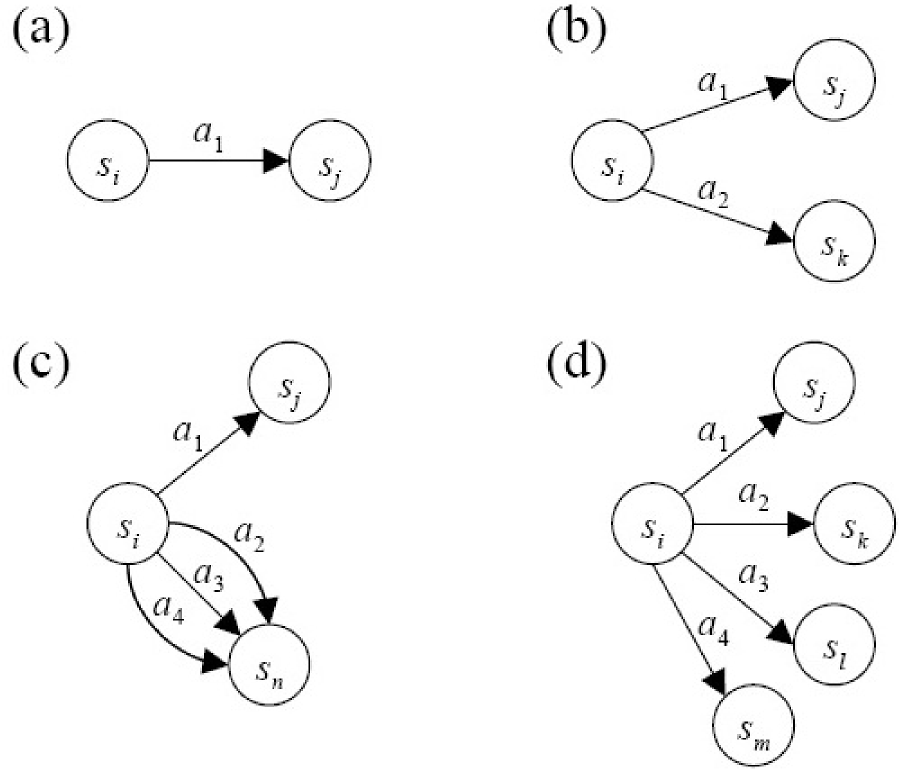
- **ดีกรีเข้า (Indegree)** คือ จำนวนของเส้นเชื่อมแบบมีทิศทางที่เข้าสู่จุดต่อ ถ้าดีกรีเข้ามีค่าสูงหมายถึง ผู้เล่นขาดอิสระในการเล่น หรือมีคุณสมบัติ ความเป็นได้อย่างเดียว (Uniqueness) กล่าวคือ อาจมีบางสถานะที่ผู้เล่นต้องเข้าสู่สถานะนี้เสมอ ไม่ว่าก่อนหน้านี้ผู้เล่นจะเลือกการกระทำใด ๆ ก็ตาม

รูปที่ 1.4 (a) แสดงเกมส์ที่มีการดำเนินเรื่องคล้ายภาพยนตร์ที่มีลักษณะเป็นเรื่องราว เนื่องจากความเป็นไปได้ของเกมส์มีเพียงเส้นทางเดียวเท่านั้น กล่าวคือ ถ้าสถานะของผู้เล่นคือ S_i และผู้เล่นเลือกการกระทำ a_1 แล้วสถานะของผู้เล่นจะเปลี่ยนไปเป็น S_j

รูปที่ 1.4 (b) แสดงเกมส์ที่มีอิสระในการเล่นมากขึ้นเมื่อเปรียบเทียบกับ รูปที่1.4 (a) เนื่องจากสถานะของผู้เล่นสามารถเปลี่ยนจาก S_i ไปเป็น S_j หรือ S_K ก็ได้ โดยขึ้นอยู่กับการกระทำของผู้เล่นว่าเลือก a_1 หรือ a_2 ในตัวอย่างนี้ ดีกรีออกของจุดต่อ S_i มีค่าเท่ากับ 2

รูปที่ 1.4 (c) สถานะ S_j ถูกบังคับว่า ต้องมีการกระทำ a_1 จากสถานะ S_i เท่านั้น ในขณะที่สถานะ S_n สามารถมาจากการกระทำ a_2, a_3 หรือ a_4 ในตัวอย่างนี้ ดีกรีเข้าของจุดต่อ S_n มีค่าเท่ากับ 3 และ S_n มีคุณสมบัติความเป็นไปได้อย่างเดียว กล่าวคือ ผู้เล่นขาดอิสระในการเล่น เพราะมีถึง 3 การกระทำที่ไม่ว่าผู้เล่นจะเลือกอะไรก็จะมุ่งเข้าสู่จุดต่อนี้เสมอ

รูปที่ 1.4 (d) มีจำนวนของการกระทำที่เป็นไปได้ทั้งหมดเท่ากับ รูปที่ 1.4 (c) แต่การกระทำมีการตอบสนองเฉพาะตัวมากกว่า เนื่องจากความเป็นไปได้มี 4 สถานะ ในขณะที่ รูปที่ 1.4 (c) มีเพียง 2 สถานะเท่านั้น อาจกล่าวได้ว่า S_i มีอิสระในการเล่นสูง เพราะในแต่ละการกระทำที่แตกต่างกัน จะมุ่งไปสู่แต่ละสถานะที่แตกต่างกัน



รูปที่ 1.7 กราฟเกมส์ [13]

การรู้จำพฤติกรรมผู้เล่นเกมส์สามารถวิเคราะห์ได้จากขั้นตอนวิธีของกราฟได้แก่ การวนซ้ำในวัฏจักร (Cycle) กล่าวคือ เมื่อการดำเนินเรื่องภายในเกมส์ถูกแทนด้วยกราฟแล้ว พฤติกรรมของผู้เล่นก็ถูกแสดงออกด้วยการเลือกกลุ่มการกระทำที่เกิดขึ้นบ่อยครั้ง ยกตัวอย่างเช่น ในเกมส์ต่อสู้ ผู้เล่นที่ถนัดการโจมตีระยะไกลทางอากาศมักจะเลือกกลุ่มการกระทำเป็น ถอยหนี กระโดด และ โจมตี ตามลำดับ เมื่อวัฏจักรดังกล่าวปรากฏมากพอที่ระบบสนับสนุนการตัดสินใจตรวจจับได้ ก็จะส่งแนวทางการตอบโต้ที่เหมาะสมให้กับปัญญาประดิษฐ์ต่อไป

4. กระบวนการเคดีดี

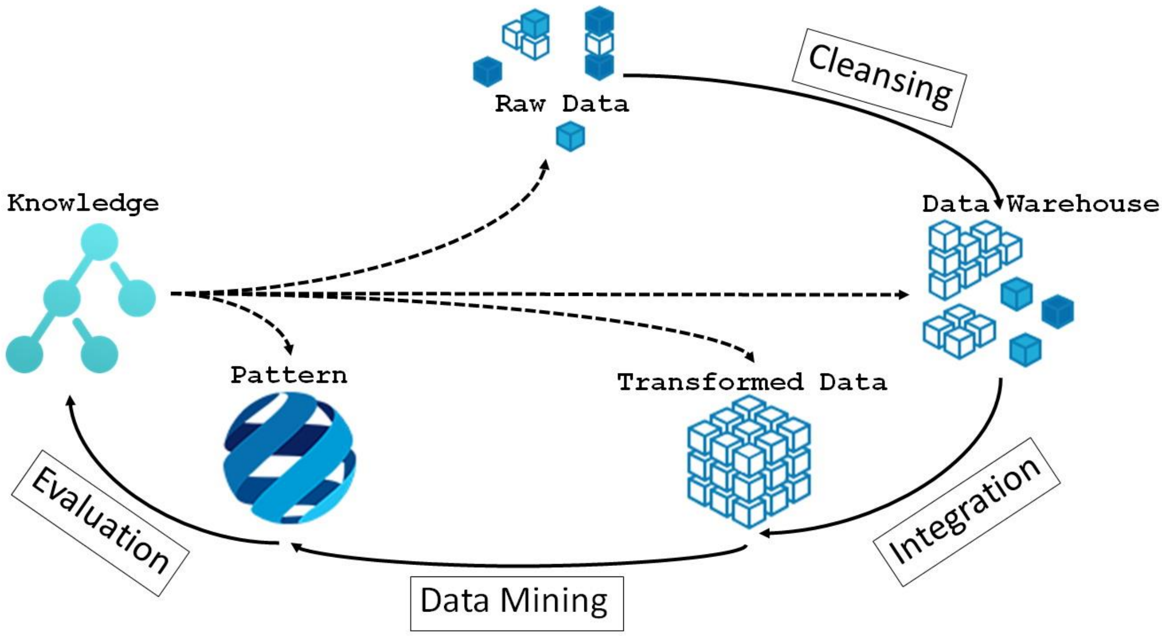
กระบวนการเคดีดี (KDD Process: Knowledge Discovery in Database Process:) ในรูปที่ 1.8 ถูกใช้ในการสกัดหารูปแบบต่าง ๆ ที่ปรากฏขึ้นบ่อยครั้งในข้อมูลดิบ เพื่อสร้างตัวแบบการรู้จำแบบ กระบวนการนี้มี 4 ขั้นตอนหลัก (แสดงการไหลของกระบวนการตามเส้นทึบ) ดังต่อไปนี้

ขั้นตอนที่ 1 การทำความสะอาดข้อมูล (Cleansing) เป็นการปรับปรุง ข้อมูลดิบ (Raw Data) ให้มีคุณภาพที่ดีขึ้น และมั่นใจได้ว่าจะสามารถนำไปวิเคราะห์ในขั้นตอนต่อไปได้อย่างมีประสิทธิภาพ โดยทั่วไปข้อมูลดิบที่ได้มาอาจมี ค่าซ้ำซ้อน (Redundant) คือเป็นข้อมูลเดิมที่ปรากฏในหลายส่วนของแหล่งข้อมูลเดียวกัน หรืออาจเป็น ข้อมูลรบกวน (Noise) ที่มีค่าแตกต่างจากข้อมูลส่วนใหญ่ นอกจากนี้ข้อมูลดิบที่ได้มาอาจมีข้อมูลไม่ครบถ้วน กล่าวคือมี ค่าที่ขาดหายไป (Missing Value) ลักษณะผิดปกติต่าง ๆ เหล่านี้ส่งผลให้คุณภาพของข้อมูลดิบมีค่าต่ำ จึงต้องหาวิธีการจัดการให้ข้อมูลดิบมีคุณภาพสูงที่สุดเท่าที่จำทำได้ก่อน ข้อมูลดิบที่ผ่านการทำความสะอาดข้อมูลแล้วเรียกว่า คลังข้อมูล (Data Warehouse)

ขั้นตอนที่ 2 การบูรณการ (Integration) เป็นการผสมผสาน คลังข้อมูล (Data Warehouse) จากแหล่งข้อมูลต้นทางหลายแหล่ง รวมเป็นคลังข้อมูลเดียวกัน เนื่องจากข้อมูลที่นำมารวมกันนี้อาจมีฟีเจอร์จำนวนมาก ทั้งที่เกี่ยวข้องหรือไม่เกี่ยวข้องกับชั้นข้อมูลที่ต้องการทำนาย ดังนั้นอาจมีการประยุกต์ใช้การคัดเลือกฟีเจอร์ (Feature Selection) เพื่อคัดเลือกฟีเจอร์ที่เป็นปัจจัยบ่งชี้ชั้นข้อมูลปลายทางได้ โดยเก็บแต่ฟีเจอร์ดังกล่าวไว้ในคลังข้อมูล และทำการกำจัดฟีเจอร์ที่ไม่สามารถบ่งชี้ชั้นข้อมูลได้ทิ้งไป คลังข้อมูลที่ผ่านการบูรณการแล้วเรียกว่า ข้อมูลแปลง (Transformed Data)

ขั้นตอนที่ 3 การทำเหมืองข้อมูล (Data Mining) เป็นการส่งข้อมูลที่ผ่านการแปลงแล้ว (Transformed Data) ให้คอมพิวเตอร์รู้จำเพื่อให้ได้มาซึ่งตัวจำแนกแบบ ซึ่งเป็นตัวแบบที่ใช้ในการทำนายแบบใหม่ว่าควรมีชั้นข้อมูลที่เหมาะสมเป็นชั้นข้อมูลอะไร ข้อมูลแปลงที่ผ่านการทำเหมืองข้อมูลแล้วเรียกว่า แบบ (Pattern) เนื้อหาส่วนใหญ่ของตำราเล่มนี้จะเน้นหนักในขั้นตอนนี้ เพื่อเสนอแนวทางในการสร้างตัวจำแนกแบบประเภทต่าง ๆ

ขั้นตอนที่ 4 การประเมิน (Evaluation) เป็นการวัดค่าประสิทธิภาพการทำนายของตัวจำแนกแบบ ถ้าประสิทธิภาพที่ได้ไม่เป็นที่พอใจ ตัวจำแนกแบบจะถูกทิ้งไป และจำเป็นต้องย้อนกลับไปทำขั้นตอนก่อนหน้าซ้ำใหม่อีกครั้ง (แสดงการไหลของกระบวนการตามแนวเส้นประ) การย้อนกลับไปประมวลผลขั้นตอนเดิมอาจเกิดขึ้นได้บ่อยครั้งเท่าที่ต้องการ จนกระทั่งได้ผลการทดลองเป็นที่น่าพอใจ ผลลัพธ์ในขั้นตอนนี้จัดเป็น องค์ความรู้ (Knowledge) หรือตัวจำแนกแบบที่สามารถนำไปประยุกต์ใช้ได้จริงนั่นเอง



รูปที่ 1.8 กระบวนการเคดีดี

5. การใช้งานโปรแกรมเวก้า

โปรแกรมเวก้า (WEKA: Waikato Environment for Knowledge Analysis) ถูกพัฒนาขึ้นโดยมหาวิทยาลัยไวกาโต ประเทศนิวซีแลนด์ เวก้าเป็นซอฟต์แวร์ฟรีทางด้าน การเรียนรู้ของเครื่อง การทำเหมืองข้อมูล การรู้จำแบบ เป็นต้น ผู้ใช้งานสามารถดาวน์โหลดได้ที่ URL ด้านล่างนี้

https://www.cs.waikato.ac.nz/ml/weka/

ในหัวนี้จะนำเสนอการใช้งานพื้นฐานของโปรแกรมนี้ เมื่อเริ่มการทำงานหน้าต่างของโปรแกรมเวก้าจะแสดงดังรูปที่ 1.9 ให้ผู้ใช้กดปุ่ม Explorer เพื่อเข้าสู่ขั้นตอนถัดไป

ชนิดของไฟล์ที่โปรแกรมเวก้ายอมรับ ได้แก่

- CSV (Comma Separated Values) คือไฟล์นามสกุล *.csv ที่ค่าแต่ละค่า (ข้อความ ตัวเลข หรือ ตัวอักษร) ถูกแบ่งด้วย จุลภาค (,) ไฟล์ชนิดนี้สามารถสร้าง บันทึก หรือแก้ไขได้ภายในโปรแกรม เอกซ์เซล (Excel)

- ARFF (Attribute Relation File Format) ซึ่งพัฒนาโดยมหาวิทยาลัยไวกาโต เป็นไฟล์นามสกุล *.arff ที่มีรูปแบบเดียวกันกับไฟล์ CSV ซึ่งสามารถใช้งานได้โดยโปรแกรมเอ็กซ์เซลเช่นกัน

นอกจากนี้ โปรแกรมเวก้ายังรองรับไฟล์ประเภทอื่น ๆ อีก ได้แก่ JSON Instances Files (*.json) และ XRFF data files (*.xrff) เป็นต้น

รูปที่ 1.9 หน้าต่างแรกของโปรแกรมเวก้า

เมื่อเข้าสู่โปรแกรมเวก้าจากแทบ Explorer แล้ว หน้าตาของโปรแกรมจะแสดงดังรูปที่ 1.10 โดยอยู่ที่แทบแรก Preeprocess แทบนี้มีรายละเอียด ดังต่อไปนี้

หน้าต่างของแทบ Preprocess

หมายเลข 1 คือ แทบ Open file… ใช้สำหรับเปิดไฟล์ที่โปรแกรมเวก้ารองรับ เช่น *.csv *.arff ตามที่ได้กล่าวไปแล้วข้างต้น ณ จุดนี้ ผู้ใช้สามารถทดลองเปิดไฟล์ iris.arff จากโฟลเดอร์เวก้าในเครื่องของคอมพิวเตอร์ของผู้ใช้ได้ที่ตำแหน่ง *C:\Program Files\Weka-3-9-4\data* ตำแหน่งที่บรรจุไฟล์นี้อาจแตกต่างกันไปในแต่ละเครื่องคอมพิวเตอร์ของผู้ใช้

หมายเลข 2 คือ พื้นที่ Atributes แสดงฟีเจอร์ทั้งหมดของไฟล์ที่เปิด ณ จุดนี้ คือฟีเจอร์ของไฟล์ Iris Dataset จากพื้นที่ดังกล่าว ฟีเจอร์ทั้งหมด ได้แก่ sepallength, sepalwidth, petal length, petalwidth และ class โดยส่วนใหญ่แล้วฟีเจอร์สุดท้ายจะแสดงชั้นข้อมูลของแต่ละแบบ

หมายเลข 3 คือ พื้นที่ Selected attributes สำหรับแสดงค่าทางสถิติของฟีเจอร์ที่เลือกจากหมายเลข 2 ได้แก่ ค่าน้อยที่สุด (Minimum) ค่ามากที่สุด (Maximum) ค่าเฉลี่ย (Mean) และ ค่าเบี่ยงเบนมาตรฐาน (StdDev)

หมายเลข 4 คือ พื้นที่ Class สำหรับแสดงการกระจายตัวของค่าในฟีเจอร์ที่เลือกจากหมายเลข 2 จากรูป เป็นของฟีเจอร์แรก แกนนอนแทนค่าของฟีเจอร์จากค่าน้อยที่สุดไปค่ามากที่สุด แกนตั้งแทนจำนวนแบบที่สัมพันธ์กับค่าในแกนนอน สีแทนชั้นข้อมูล (ดูรายละเอียได้ที่ หมายเลข 4 หน้าต่างของแทบ Visualize) จากตัวอย่างนี้ ค่าของฟีเจอร์ sepallength ถ้ามีค่าน้อย มีโอกาสที่จะอยู่ในชั้นข้อมูล Iris-setosa ถ้ามีค่าปานกลาง มีโอกาสที่จะอยู่ในชั้นข้อมูล Iris-versicolor ถ้ามีค่าสูง มีโอกาสที่จะอยู่ในชั้นข้อมูล Iris-virginica

แทบต่อไปคือ Classify ใช้สำหรับจำแนกแบบโดยตัวแบบชนิดต่าง ๆ โดยสามารถเลือกวิธีการทดลองได้ รายละเอียดในการใช้งานแทบนี้ มีดังต่อไปนี้

หน้าต่างของแทบ Classify

หมายเลข 1 เมื่อกดแล้วจะเป็นการเริ่มต้นการทำงานของแทบ Classifier

หมายเลข 2 คือ ปุ่ม Choose ใช้สำหรับเลือกตัวจำแนกแบบชนิดต่าง ๆ ซึ่งมีมากมายหลายประเภท สำหรับหัวข้อนี้ จะให้รายละเอียเฉพาะตัวแบบที่อยู่ในตำรานี้ ได้แก่

- ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุด อยู่ในโฟลเดอร์ bayes ชื่อตัวแบบ NaiveBayes รายละเอียดของตัวจำแนกแบบชนิดนี้อยู่ในบทที่ 3
- ตัวจำแนกเบส์ อยู่ในโฟลเดอร์ lazy ชื่อตัวแบบ Ibk รายละเอียดของตัวจำแนกแบบชนิดนี้อยู่ในบทที่ 4
- ต้นไม้การตัดสินใจ อยู่ในโฟลเดอร์ trees ชื่อตัวแบบ รายละเอียดของตัวจำแนกแบบชนิดนี้อยู่ในบทที่ 5

- โครงข่ายประสาท อยู่ในโฟลเดอร์ function ชื่อตัวแบบ MultilayerPerceptron รายละเอียดของตัวจำแนกแบบชนิดนี้อยู่ในบทที่ 6
- เครื่องกลเวกเตอร์สนับสนุน อยู่ในโฟลเดอร์ function ชื่อตัวแบบ SOM รายละเอียดของตัวจำแนกแบบชนิดนี้อยู่ในบทที่ 6

หมายเลข 3 คือ พื้นที่ Test options สำหรับเลือกวิธีการประเมินค่าตัวจำแนกแบบ ดังต่อไปนี้

- Use training set คือ การใช้ทั้งเซตข้อมูลเป็นทั้งเซตฝึกฝนและเซตทดสอบ
- Supplied test set คือ การเลือกเซตทดสอบที่แตกต่างจากเซตฝึกฝน
- Cross-validation คือ การพับเคครั้ง
- Percentage Split คือ การแบ่งเซตข้อมูลออกเป็นเซตฝึกฝนและเซตทดสอบ

สำหรับรายละเอียดของแต่ละวิธีการ จะอธิบายต่อไปในบทที่ 2

หมายเลข 4 คือ ปุ่ม Start กดเมื่อต้องการเริ่มต้นการทำงานของตัวจำแนกแบบ

หมายเลข 5 คือ พื้นที่ Classifer Output แสดงผลที่ได้จากการจำแนกแบบ รายละเอียดต่าง ๆ จะกล่าวถึงในบทที่ 7

แทบสุดท้ายคือ Visualize สำหรับแสดงผลกราฟการกระจายตัวของแบบจากแต่ละชั้นข้อมูล รายละเอียดในการใช้งานแทบนี้ มีดังต่อไปนี้

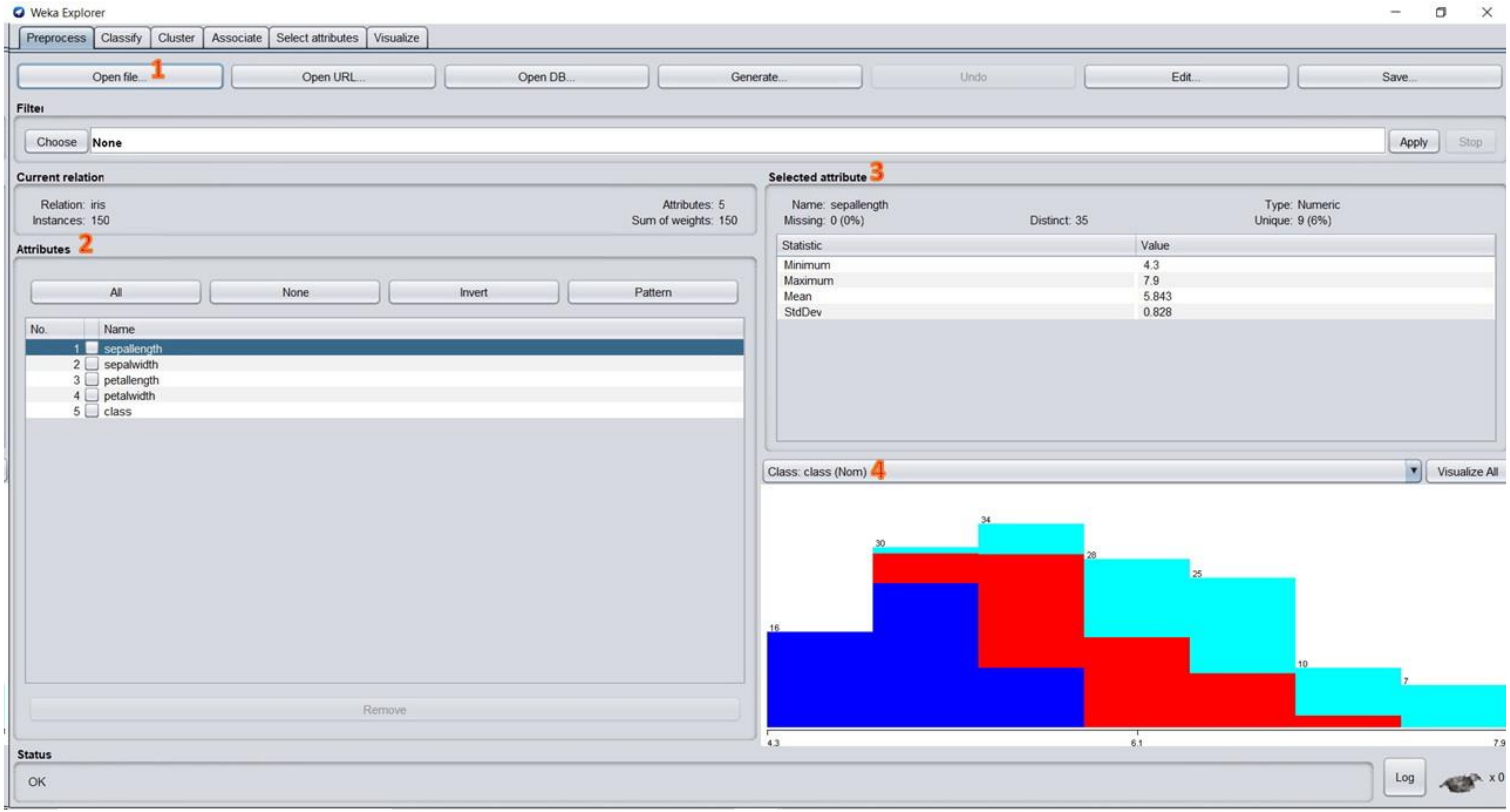
หน้าต่างของแทบ Visualize

หมายเลข 1 เมื่อกดแล้วจะเป็นการเริ่มต้นการทำงานของแทบ Visualize

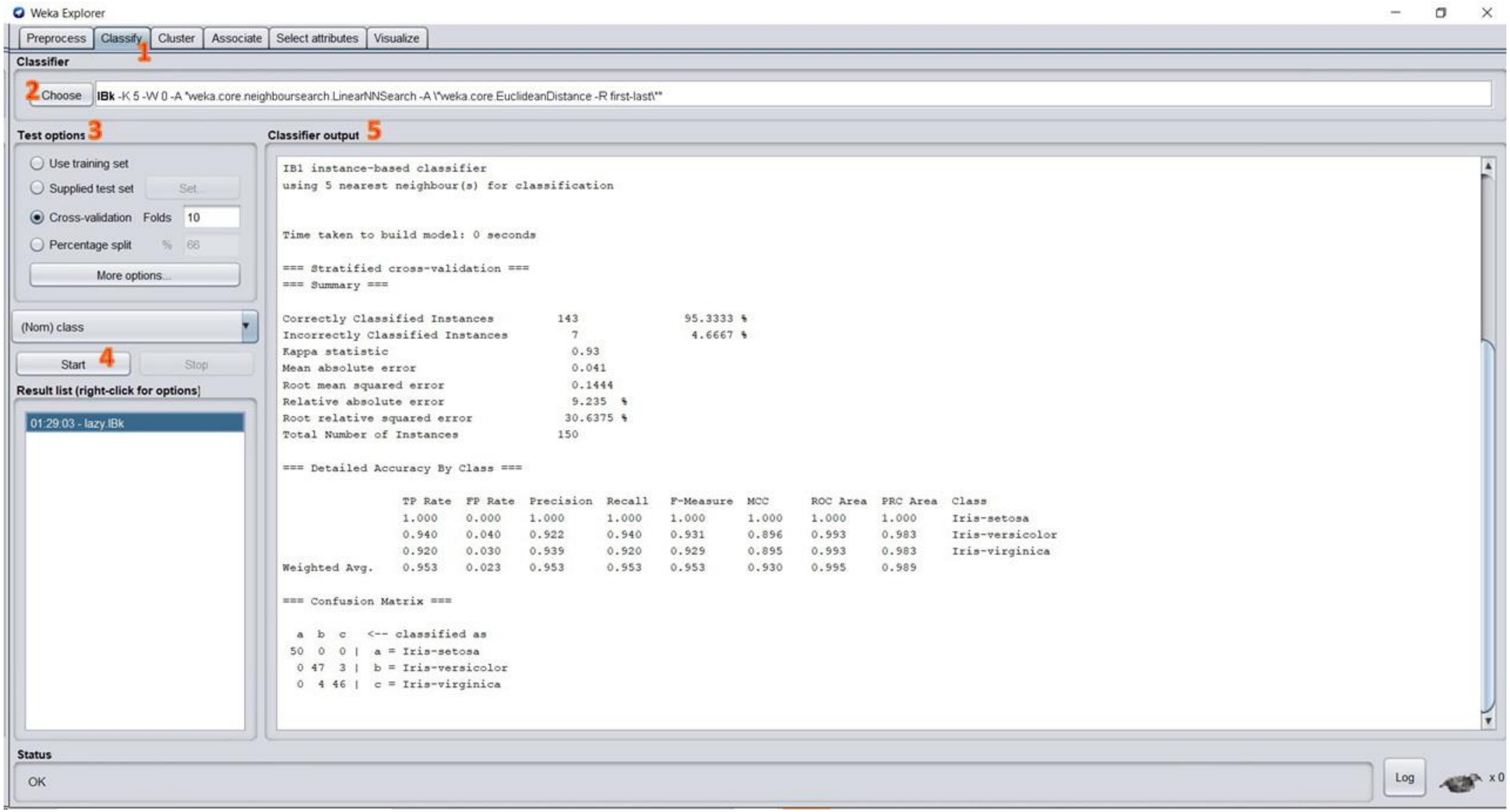
หมายเลข 2 คือ การเลือกคู่ของฟีเจอร์ที่ต้องการให้แสดงผลใน 2 แกน

หมายเลข 3 คือ หน้าต่าง Visualizing สำหรับแสดงค่าฟีเจอร์ในแกนนอนและแกนตั้ง

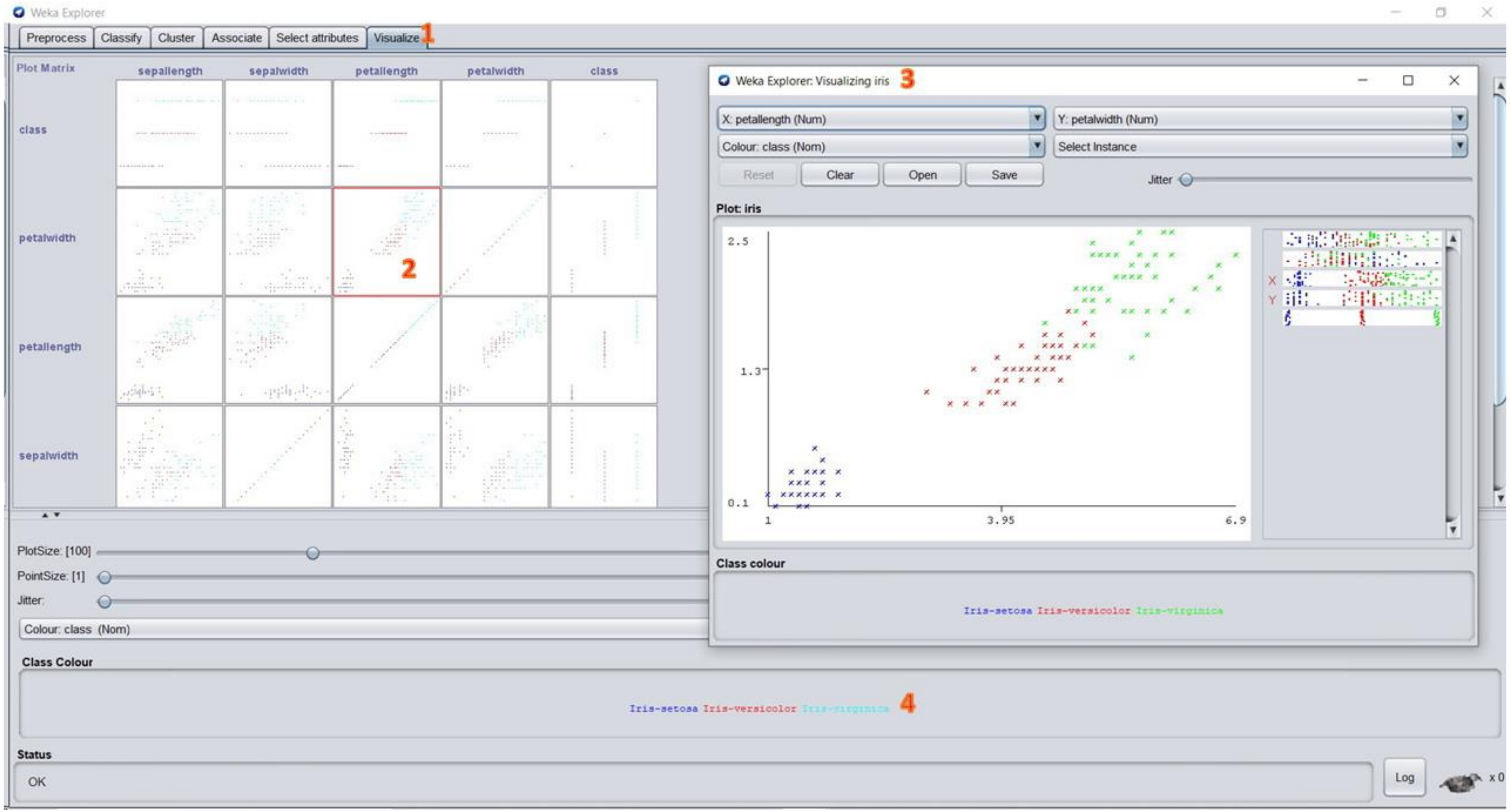
หมายเลข 4 คือ สีสัญลักษณ์ที่ระบุว่าแบบในสีใดอยู่ในชั้นข้อมูลไหน



รูปที่ 1.10 หน้าต่างของแทบ Preprocess



รูปที่ 1.11 หน้าต่างของแทบ Classify



รูปที่ 1.12 หน้าต่างของแทบ Visualize

6. กระบวนทัศน์สำหรับการรู้จำแบบ

การแก้ปัญหาการรู้จำแบบสามารถแบ่งออกได้เป็น 2 กระบวนทัศน์หลัก ได้แก่ การรู้จำแบบทางสถิติ (Statistical Pattern Recognition) และ การรู้จำแบบทางไวยากรณ์ (Syntactic Pattern Recognition) การรู้จำแบบทางสถิติได้รับความนิยมและความสนใจมากกว่าและพบได้เป็นส่วนใหญ่จากงานวิจัยทางด้านการรู้จำแบบ เนื่องจากปัญหาในทางปฏิบัติส่วนใหญ่ต้องจัดการกับ ข้อมูลรบกวน (Noisy Data) และ ความไม่แน่นอน (Uncertainty) ดังนั้น สถิติและความน่าจะเป็นคือเครื่องมือที่ดีในการแก้ปัญหาดังกล่าว การรู้จำแบบทางไวยากรณ์ใช้ภูมิหลังของ ทฤษฎีภาษาทางการ (Formal Language Theory) ที่มีพื้นฐานบน เครื่องมือภาษามนุษย์ (Linguistic Tool) การรู้จำแบบชนิดนี้ไม่เหมาะสมกับสภาพแวดล้อมที่มีสิ่งรบกวน

ตำราฉบับนี้มุ่งเน้นไปที่การรู้จำแบบทางสถิติที่ใช้ ปริภูมิเวกเตอร์ (Vector Space) เป็นตัวแทนของแบบและชั้นข้อมูล โดยทั่วไป การกำหนดสาระสำคัญ (Abstraction) ต้องจัดการกับ ความน่าจะเป็นของ ความหนาแน่น (Density) หรือ การแจกแจง (Distribution) ของจุดใน ปริภูมิหลายมิติ (Multi-dimensional Space) เนื่องจาก การใช้ตัวแทนปริภูมิเวกเตอร์สามารถนำไปคำนวณเป็น ปริภูมิย่อย (Sub-space) การทอดเงา (Projection) และ ความคล้าย (Similarity) ระหว่างจุดในเทอมของ การวัดระยะทาง (Distance Measure) ได้

การจัดกลุ่มข้อมูล (Clustering) ถูกใช้กับเซตข้อมูลที่มีขนาดมหึมา เทคนิคนี้มีจุดประสงค์เพื่อแบ่งกลุ่มแบบทั้งหมดออกจากกัน โดยแบบในกลุ่มเดียวกันมีความคล้ายกันมากกว่าแบบจากต่างกลุ่มกัน หลังจากแบบทั้งหมดถูกแบ่งออกเป็นกลุ่มข้อมูลต่าง ๆ แล้ว เป็นไปได้ว่าในเซตข้อมูลเดียวกันอาจมีตัวจำแนกแบบหลายตัว ซึ่งตัวจำแนกแบบแต่ละตัวจะถูกใช้ในแต่ละกลุ่มข้อมูล รายละเอียดการแบ่งกลุ่มข้อมูลจะถูกกล่าวถึงในบทที่ 2

ตัวจำแนกแบบที่ได้รับความนิยมมากที่สุดและมีแนวคิดพื้นฐานที่สุดคือ ตัวจำแนกเพื่อนบ้านใกล้ที่สุด (Nearest Neighbour Classifier) สำหรับเทคนิคนี้ แบบใหม่จะถูกทำนายให้มีชั้นข้อมูลเดียวกันกับแบบที่อยู่ใกล้ที่สุดโดยไม่จำเป็นต้องใช้เซตข้อมูลทดสอบ รายละเอียดตัวจำแนกแบบเพื่อนบ้านใกล้ที่สุดจะถูกกล่าวถึงในบทที่ 3

ตัวจำแนกเบส์ (Bayes Classifier) มีพื้นฐานอยู่บนทฤษฎีเบส์ที่ประยุกต์ใช้ความน่าจะเป็นในการทำนาย สำหรับเทคนิคนี้ แบบใหม่จะถูกทำนายให้มีชั้นข้อมูลเดียวกันกับแบบที่มีค่าฟีเจอร์โดยรวมคล้ายกันมากที่สุดและมีความถี่ในการเกิดขึ้นบ่อยครั้งที่สุด รายละเอียดของตัวจำแนกเบส์จะถูกกล่าวถึงในบทที่ 4

โครงข่ายประสาท (Neural Networks) มีการทำงานโดยเลียนแบบการทำงานของสมองมนุษย์ ตัวจำแนกแบบชนิดหนึ่งของโครงข่ายประสาทเทียม ได้แก่ เพอร์เซ็ปตรอน (Perceptron) ที่ทำนายแบบใหม่โดยการหาเส้นแบ่งปริภูมิหลายมิติที่แบ่งแบบจากแต่ละชั้นข้อมูลออกจากกัน นอกจากนี้ เอสวีเอ็ม (SVM) ก็ใช้พื้นฐานนี้ในการทำนายแบบใหม่เช่นกัน รายละเอียดของตัวแบบทั้งหมดทางด้านนี้จะถูกอธิบายในบทที่ 6

ปัญหาชั้นข้อมูลอสมดุล (Class Imbalance Problem) เป็นปัญหาที่พบบ่อยในเซตข้อมูลจริง ซึ่งเกิดขึ้นกับเซตข้อมูลที่สัดส่วนของแบบในแต่ละชั้นข้อมูลมีความแตกต่างกันสูงมาก ส่งผลให้ประสิทธิภาพการทำนายของตัวจำแนกมีคุณภาพต่ำ เนื่องจากตัวจำแนกแบบส่วนใหญ่จะให้ความสำคัญกับแบบจากชั้นข้อมูลที่มีปริมาณสูง

อภิปราย

อุปสรรคที่จัดว่ามีความท้าทายสำหรับการรู้จำแบบเกิดขึ้นในกรณีที่จำนวนของแบบในแต่ละชั้นข้อมูลมีสัดส่วนที่แตกต่างกันสูงมาก สำหรับกรณีนี้ แบบจากชั้นข้อมูลที่มีสัดส่วนต่ำจะมีปริมาณไม่มากพอให้ตัวจำแนกแบบเรียนรู้ได้อย่างมีประสิทธิภาพ ส่งผลให้การทำนายแบบจากชั้นข้อมูลนี้ขาดความแม่นยำจนไม่สามารถใช้งานได้ ยกตัวอย่างเช่นในเกมส์คอมพิวเตอร์ ผู้เล่นที่มีพฤติกรรมแปลกประหลาดซึ่งแตกต่างจากผู้เล่นโดยส่วนใหญ่ทั่วไป ย่อมมีจำนวนแบบปริมาณน้อยมากเมื่อเทียบกับแบบจากผู้เล่นปกติ และอาจไม่เพียงพอที่จะทำให้ตัวจำแนกแบบเรียนรู้เพื่อดักจับพฤติกรรมจากแบบส่วนน้อยนี้ได้ สำหรับปัญหานี้ ตัวจำแนกแบบส่วนใหญ่จะให้ความสำคัญกับแบบจากชั้นข้อมูลที่มีปริมาณสูง ดังนั้นแนวทางการแก้ปัญหาจำเป็นต้องประยุกต์เทคนิคสำหรับแก้ปัญหาชั้นข้อมูลอสมดุล ในการปรับสัดส่วนของแบบทั้งสองประเภทให้สมดุลขึ้น เพื่อให้การทำนายของตัวจำแนกแบบมีความแม่นยำขึ้น นิยามของปัญหาและวิธีการแก้ปัญหาจากกงานวิจัยของผู้แต่งและงานวิจัยใหม่ ๆ จะถูกอธิบายในบทที่ 7

สรุป

การรู้จำแบบเกี่ยวข้องกับการจำแนกประเภทของแบบว่าควรอยู่ชั้นข้อมูลใด โดยสังเกตจากฟีเจอร์ที่เกิดขึ้นบ่อยครั้งของแบบในแต่ละชั้นข้อมูลนั้น ความหมายของแบบคือฟีเจอร์ทั้งหมดในแต่ละแถวของเซตข้อมูล แนวคิดพื้นฐานของการรู้จำแบบถูกประยุกต์ใช้ในหลากหลายสาขา เช่น การวินิจฉัยโรคภัยไข้เจ็บ การดักจับแพ็กเกจผู้บุกรุกระบบเครือข่าย การทำนายพฤติกรรมลูกค้า เป็นต้น

กระบวนการเคดีดีเป็นกระบวนการที่ใช้ในการแปลงข้อมูลดิบเป็นองค์ความรู้ที่ใช้ในการจำแนกแบบ กระบวนการนี้เริ่มต้นจากการทำความสะอาดข้อมูลเพื่อให้คุณภาพของข้อมูลเป็นที่น่าพอใจ รวบรวมข้อมูลดิบจากหลายแหล่งต้นทางเพื่อหลอมรวมเป็นคลังข้อมูลเดียว เรียนรู้แบบที่ได้จากคลังข้อมูล รวมถึงการทดลองวัดค่าประสิทธิภาพการทำนายจนกว่าจะเป็นที่น่าพอใจและได้ตัวจำแนกแบบออกมาใช้จริง

การรู้จำแบบสามารถกระทำได้ทั้งทางสถิติและทางไวยากรณ์ โดยที่การรู้จำแบบทางสถิติได้รับความนิยมมากกว่าเพราะเกี่ยวข้องกับเซตข้อมูลส่วนใหญ่ที่มีลักษณะเป็นเวกเตอร์ สำหรับกรณีตัวอย่างบทบาทของการรู้จำแบบในเกมส์คอมพิวเตอร์ การรู้จำแบบถูกประยุกต์ใช้ในการคาดเดาพฤติกรรมของผู้เล่นที่กำลังจะตัดสินใจในอนาคต เพื่อวางแผนให้ปัญญาประดิษฐ์ตอบโต้ผู้เล่นได้อย่างเหมาะสมและมีประสิทธิภาพ

เนื้อหาในบทต่อไปกล่าวถึง โครงสร้างข้อมูลชนิดต่าง ๆ สำหรับใช้เป็นตัวแทนของแบบเพื่อเตรียมบรรจุแบบทั้งหมดลงในเซตข้อมูล สำหรับการประมวลผลการรู้จำแบบในขั้นตอนต่อไป รวมถึงหน่วยวัดความใกล้เคียงวิธีต่าง ๆ ที่ใช้สำหรับเปรียบเทียบว่าแบบสองแบบมีความคล้ายกันมากน้อยเพียงใด

แบบฝึกหัด

1. จากรูปที่ 1.1 จงวิเคราะห์ว่าแบบ P ควรมีป้ายชื่อชั้นข้อมูลเป็น X หรือ O โดยให้เหตุผลสนับสนุน
2. จากตารางที่ 1.1 ป้ายชื่อชั้นข้อมูลที่เหมาะสมกับฟีเจอร์น้ำหนัก ควรประกอบด้วยอะไรบ้าง
3. จงยกตัวอย่างโปรแกรมประยุกต์ทางด้านการรู้จำแบบในสาขา การเกษตร การศึกษา การรักษาความปลอดภัย การขนส่ง การเงิน การแพทย์ และ ความบันเทิง นอกเหนือจากที่อาจารย์ยกตัวอย่างในชั้นเรียน
4. พิจารณางานในการรู้จำตัวเลข 0 ถึง 9 ที่เขียนด้วยลายมือ ปัญหานี้ประกอบด้วยฟีเจอร์อะไรบ้าง และ ฟีเจอร์ดังกล่าวจัดเป็นกระบวนทัศน์ทางไวยากรณ์หรือสถิติ
5. จงออกแบบการรู้จำแบบในเกมส์หมากรุก ให้อธิบายการทำงานของปัญญาประดิษฐ์ในมุมมองของศัตรู พันธมิตร และ ไม่ฝักใฝ่ฝ่ายใด โดยให้ยกตัวอย่างประกอบ
6. จากข้อมูลดิบในตารางที่ 1.4 จงระบุว่าข้อมูลใดมีลักษณะเป็น ข้อมูลรบกวน (Noise) ด้วยเหตุผลใด
7. จากข้อมูลดิบในตารางที่ 1.5 จงหาวิธีการเติม ค่าที่ขาดหายไป (Missing Value) ว่าควรมีค่าเท่าไหร่

ตารางที่ 1.4 ข้อมูลดิบ

A1	A2	A3	A4	Class
19	10	7	1	A
13	20	5	2	A
18	48	13	14	A
8	7	9	19	A
6	19	6	15	B
10	20	18	16	B
2	12	-9	13	B
22	15	12	14	B
12	132	17	20	C
0	7	98	11	C
16	17	15	1	C

ตารางที่ 1.5 ข้อมูลดิบ

A1	A2	A3	A4	Class
19	10	7	1	A
13	20	5	2	A
18	48	13	14	A
	7		19	A
6	19	6	15	B
10	20	18	16	B
2	12	9	13	B
	15	12	14	B
12	13	17	20	C
10	7	98	11	C
16	17	15	1	C

8. จงออกแบบฐานข้อมูลลูกค้าของธนาคารแห่งหนึ่งสำหรับใช้กับการรู้จำแบบ เพื่อนำไปแก้ปัญหากลุ่มมิจฉาชีพหลอกลวงให้โอนเงินทางเอทีเอ็ม กำหนดให้ 1 ระเบียน แทนการทำธุรกรรมทางอิเล็กทรอนิกส์จำนวน 1 ครั้ง ให้ระบุสดมภ์ที่จำเป็นว่าควรเก็บข้อมูลอะไรบ้าง

บทที่ 2

การแทนแบบ

จากเนื้อหาในบทแรก ที่ได้เกริ่นนำถึงความหมายของการรู้จำแบบ ตัวจำแนกแบบ และ แบบ ในบทนี้จะมีการเชื่อมโยงมาสู่นิยามของปัญหาการจำแนกแบบ การแทนแบบด้วยโครงสร้างข้อมูลเพื่อจัดเก็บลงในหน่วยความจำของเครื่องคอมพิวเตอร์ นอกจากนี้ยังปูทางนำไปสู่เครื่องมือที่ถูกใช้ในการวัดความแตกต่างระหว่างแบบสองแบบ ในกรณีที่แบบในเซตข้อมูลมีจำนวนมาก การจัดกลุ่มข้อมูลถูกนำเสนอในการแบ่งแบบที่มีความคล้ายกันให้อยู่กลุ่มข้อมูลเดียวกัน ก่อนการกำหนดสาระสำคัญของข้อมูล ในส่วนท้ายของบทนี้ เป็นการประเมินค่าประสิทธิภาพของตัวแบบจากปัจจัยต่าง ๆ ที่เกี่ยวข้องกับการทำนาย

วัตถุประสงค์การเรียนรู้

- ทราบการแทนแบบเป็น เวกเตอร์ ข้อความ นิพจน์ตรรกะ และ เซตคลุมเครือ
- ได้ศึกษาการจำแนกแบบโดยใช้หน่วยวัดความใกล้เคียงชนิดเมตริกและนอนเมตริก
- สามารถจัดกลุ่มข้อมูลโดยใช้ขั้นตอนวิธีเคมีนส์ได้
- อธิบายการกำหนดสาระสำคัญของเซตข้อมูลได้
- ระบุปัจจัยที่มีผลกระทบต่อการประเมินค่าตัวจำแนกแบบได้

1. ปัญหาการจำแนกแบบ

แบบ (Pattern) อาจเป็น วัตถุรูปธรรม (Physical Object) หรือ ความนึกคิดนามธรรม (Abstract Notion) ก็ได้ โดยที่วัตถุรูปธรรมคือสิ่งที่จับต้องได้ เช่น ลูกบอลกีฬา สัตว์ สิ่งของ พนักงาน และนักศึกษา เป็นต้น ในขณะที่ความคิดนึกคิดนามธรรมคือสิ่งที่จับต้องไม่ได้ เช่น ทัศนคติทางด้านการเมือง รีวิวร้านอาหาร และ สภาพอากาศ เป็นต้น แบบต้องประกอบไปด้วยกลุ่มของคำอธิบายแบบ ยกตัวอย่างเช่น คำอธิบายของลูกบอลกีฬาประเภทต่าง ๆ อาจประกอบไปด้วย ขนาด น้ำหนัก และ สีของลูกบอลซึ่งมีรายละเอียดแสดงตามตารางที่ 2.1 เป็นต้น ส่วนคำอธิบายของสัตว์อาจประกอบไปด้วย ขนาด จำนวนขา และ ประเภทของสัตว์ (สัตว์บก สัตว์น้ำ หรือ สัตว์ครึ่งบกครึ่งน้ำ) เป็นต้น โดยที่ คำอธิบายเหล่านี้จัดเป็น คุณสมบัติ (Attribute) ของแบบ อาจกล่าวโดยสรุปเกี่ยวกับนิยามของแบบได้ว่า แบบคือ

ตัวแทนของวัตถุใด ๆ ที่อาจเป็นสิ่งที่จับต้องได้หรือจับต้องไม่ได้ก็ได้ และสามารถถูกอธิบายได้ด้วยเซตของคุณสมบัติ

ตารางที่ 2.1 ตัวอย่างเซตข้อมูลของลูกบอลกีฬา

ขนาด	น้ำหนัก	สี	ลูกบอลกีฬา
ปานกลาง	เบา	เขียว	ลูกเทนนิส
ใหญ่	หนัก	ขาว	ลูกฟุตบอล
เล็ก	เบา	ส้ม	ลูกปิงปอง
ปานกลาง	ปานกลาง	แดง	ลูกคริกเก็ต
ปานกลาง	หนัก	เงิน	ลูกเปตอง
ใหญ่	หนัก	ส้ม	ลูกบาสเกตบอล
ปานกลาง	ปานกลาง	ขาว	ลูกเบสบอล
เล็ก	เบา	ขาว	ลูกกอล์ฟ
ใหญ่	หนัก	น้ำตาล	ลูกรักบี้
ใหญ่	หนัก	ดำ	ลูกโบว์ลิ่ง
ใหญ่	ปานกลาง	ขาว	ลูกวอลเล่ย์บอล
เล็ก	เบา	ขาว	ลูกแบดมินตัน
ปานกลาง	ปานกลาง	เหลือง	ลูกตะกร้อ
เล็ก	ปานกลาง	ดำ	ลูกฮอกกี้น้ำแข็ง
เล็ก	ปานกลาง	หลายสี	ลูกสนุกเกอร์

ปัญหาการจำแนกแบบ (Classification Problem) คือ ปัญหาที่ว่าด้วยการกำหนดชั้นข้อมูลให้แบบใหม่ที่ไม่ทราบชั้นข้อมูลมาก่อน โดยพิจารณาองค์ความรู้จากแบบเก่าที่ระบุชั้นข้อมูลครบถ้วน เซตข้อมูลสำหรับการรู้จำแบบที่บรรจุแบบเก่านั้น แบบทุกแบบต้องรู้ค่าคุณสมบัติครบถ้วน และ แบบต้องเป็นประเภทใดประเภทหนึ่งในชั้นข้อมูลเท่านั้น แบบหนึ่งแบบไม่สามารถอยู่ในหลายชั้นข้อมูลได้ ยกตัวอย่างเช่น แบบของลูกบอลอาจมีชั้นข้อมูลเป็น ลูกฟุตบอล ลูกคริกเก็ต หรือ ลูกปิงปอง อย่างใดอย่างหนึ่งเท่านั้น ลูกบอลหนึ่งชนิดไม่สามารถเป็นทั้งลูกฟุตบอลและลูกปิงปองพร้อมกันได้สำหรับปัญหาการจำแนกแบบ

แบบที่ดีสำหรับปัญหาการจำแนกแบบคือ แบบจากต่างชั้นข้อมูลกันต้องมีความแตกต่างกัน หรืออาจกล่าวได้ว่าแบบที่ดีต้องสามารถระบุชั้นข้อมูลที่เหมาะสมหรือถูกต้องได้ จากการพิจารณาค่าคุณสมบัติในเซตข้อมูล จากตารางที่ 2.1 สามารถยกตัวอย่างการจำแนกแบบจากการพิจารณาค่าคุณสมบัติได้ ดังต่อไปนี้

- ถ้าลูกบอลใหม่มีสีเหลืองแล้วจะถูกจำแนกให้อยู่ในชั้นข้อมูลลูกตระกร้อ เนื่องจากในเซตข้อมูลมีเพียงลูกตระกร้อเพียงชนิดเดียวเท่านั้นที่มีสีเหลือง
- ถ้าลูกบอลใหม่มีสีส้มแล้วยังไม่อาจจำแนกแบบได้ เนื่องจากทั้งลูกปิงปองและลูกบาสเกตบอลต่างมีสีส้มเหมือนกัน แต่เมื่อพิจารณาจากลูกบอลใหม่ที่มีขนาดเบาแล้ว จึงสามารถจำแนกได้ว่าลูกบอลใหม่เป็นลูกปิงปอง
- ถ้าลูกบอลใหม่มีสีขาวและมีขนาดปานกลางแล้ว ลูกบอลที่เกี่ยวข้องมี 2 ชนิดได้แก่ ลูกเบสบอล และ ลูกวอลเล่ย์บอล ดังนั้นจึงยังไม่อาจตัดสินได้ แต่ลูกบอลใหม่มีขนาดใหญ่จากข้อมูลตรงนี้จึงสามารถระบุได้ว่า แบบใหม่มีชั้นข้อมูลเป็นลูกวอลเล่ย์บอล

เมื่อแบบในเซตข้อมูลมีค่าคุณสมบัติที่แตกต่างกันแล้วระหว่างแต่ละชั้นข้อมูล การทำนายชั้นข้อมูลจะทำได้ง่ายเมื่อพิจารณาจากค่าคุณสมบัติ นอกจากนี้การจำแนกแบบยังสามารถลดเวลาการประมวลผลได้อีกด้วย

2. โครงสร้างข้อมูลสำหรับการแทนแบบ

การแทนแบบ (Pattern Representative) คือ การใช้โครงสร้างข้อมูลประเภทต่าง ๆ เช่น เวกเตอร์ (Vector) ข้อความ (String) นิพจน์ตรรกะ (Logical Expression) หรือ เซตคลุมเครือ (Fuzzy Set) ในการจัดเก็บแบบต่าง ๆ ลงในคอมพิวเตอร์ เพื่อจุดประสงค์สำหรับการประมวลผลในขั้นตอนการจำแนกแบบ โครงสร้างข้อมูลประเภทต่าง ๆ มีรายละเอียดดังหัวข้อย่อยต่อไปนี้

2.1 เวกเตอร์

เวกเตอร์เป็นโครงสร้างข้อมูลที่เห็นภาพชัดเจนที่สุดสำหรับการแทนแบบ โดยที่องค์ประกอบ (Element) ของเวกเตอร์สามารถเก็บชนิดข้อมูลได้ทั้งจำนวนและประเภท เวกเตอร์และแบบมีความสัมพันธ์กันคือ องค์ประกอบแต่ละส่วนของเวกเตอร์แทนคุณสมบัติแต่ละค่าของแบบ ยกตัวอย่างเช่น เวกเตอร์ลูกเทนนิสจากตารางที่ 2.1 ได้แก่ (ปานกลาง, เบา, เขียว) โดยที่ องค์ประกอบแรกถึงองค์ประกอบสุดท้ายแทน ขนาด น้ำหนัก และ สี ของลูกเทนนิสตามลำดับ เป็นต้น หรือในอีกกรณีหนึ่งอาจแทนวัตถุรูปทรงกลมด้วยจำนวน 2 จำนวนได้แก่ น้ำหนัก และ ความยาวเส้นผ่านศูนย์กลาง ตัวอย่างเช่น เวกเตอร์ (30, 1) แทน วัตถุรูปทรงกลมที่มีน้ำหนัก 30 หน่วย และเส้นผ่านศูนย์กลางมีความยาว 1 หน่วย เป็นต้น นอกจากนี้ชั้นข้อมูลสามารถเพิ่มเป็นส่วนหนึ่งของเวกเตอร์ได้ ถ้าวัตถุทรงกลมนี้อยู่ในชั้นข้อมูลลำดับที่ 1 แล้ว เวกเตอร์นี้สามารถแสดงได้เป็น (30, 1, 1) โดยที่ตัวเลขตัวสุดท้ายแสดงป้ายชื่อชั้นข้อมูลของแบบเป็นต้น

สำหรับตัวอย่างที่ 1 ด้านล่างนี้ เวกเตอร์แต่ละตัวประกอบไปด้วย 3 องค์ประกอบ โดยที่สององค์ประกอบแรกของเวกเตอร์แทนฟีเจอร์แรกและฟีเจอร์ที่สองของแบบ ส่วนองค์ประกอบสุดท้ายของเวกเตอร์แทนป้ายชื่อชั้นข้อมูลของแบบ เซตข้อมูลนี้ประกอบไปด้วยเวกเตอร์ทั้งหมดจำนวน 16 เวกเตอร์ แสดงการกระจายตัวของเวกเตอร์ได้จากกราฟในรูปที่ 2.1 ในกราฟนี้ประกอบไปด้วยแกน 2 แกนได้แก่ แกน x_1 และ แกน x_2 โดยที่ แกน x_1 แทนองค์ประกอบแรกของเวกเตอร์ และ แกน x_2 แทนองค์ประกอบที่สองของเวกเตอร์ สัญลักษณ์ต่าง ๆ ในกราฟแทนชั้นข้อมูลต่าง ๆ (องค์ประกอบสุดท้ายของเวกเตอร์) โดยมีรายละเอียดดังต่อไปนี้

- สัญลักษณ์ + แทนแบบที่มีป้ายชื่อชั้นข้อมูลที่ 1
- สัญลักษณ์ x แทนแบบที่มีป้ายชื่อชั้นข้อมูลที่ 2
- สัญลักษณ์สี่เหลี่ยมจัตุรัสมีจุดตรงกลาง แทนแบบที่ไม่เคยเห็นมาก่อน (Unseen Pattern) คือแบบที่ต้องการจำแนกแบบหรือทำนาย แบบชนิดนี้จะยังไม่มีป้ายชื่อชั้นข้อมูล จุดประสงค์ของการรู้จำแบบในปัญหาการจำแนกแบบคือการระบุป้ายชื่อชั้นข้อมูลที่เหมาะสมให้กับแบบที่ไม่เคยเห็นมาก่อนหรือแบบใหม่นั่นเอง

สำหรับตัวอย่างนี้ แบบที่ไม่เคยเห็นมาก่อนต้องถูกทำนายว่าควรอยู่ในชั้นข้อมูลใด และการทำนายต้องมีเพียงคำตอบเดียวเท่านั้นคือ แบบใหม่ควรอยู่ในชั้นข้อมูลที่ 1 หรือ ชั้นข้อมูลที่ 2 เพียงชั้นข้อมูลเดียวเท่านั้น ตัวจำแนกแบบแต่ละประเภทจะใช้แนวคิดที่แตกต่างกันในการทำนาย บางตัวใช้การวัดระยะทาง บางตัวใช้ความน่าจะเป็น บางตัวใช้ความสัมพันธ์ ถ้า-แล้ว บางตัวใช้คณิตศาสตร์ชั้นสูงในการสร้างเส้นแบ่งหรือระนาบแบ่ง ทฤษฎีต่าง ๆ เหล่านี้จะถูกกล่าวถึงในบทต่อ ๆ ไป

ตัวอย่างที่ 1 การใช้เวกเตอร์เป็นตัวแทนเซตของแบบ

1.0, 1.0, 1 ; 1.0, 2.0, 1

2.0, 1.0, 1 ; 2.0, 2.0, 1

4.0, 1.0, 2 ; 5.0, 1.0, 2

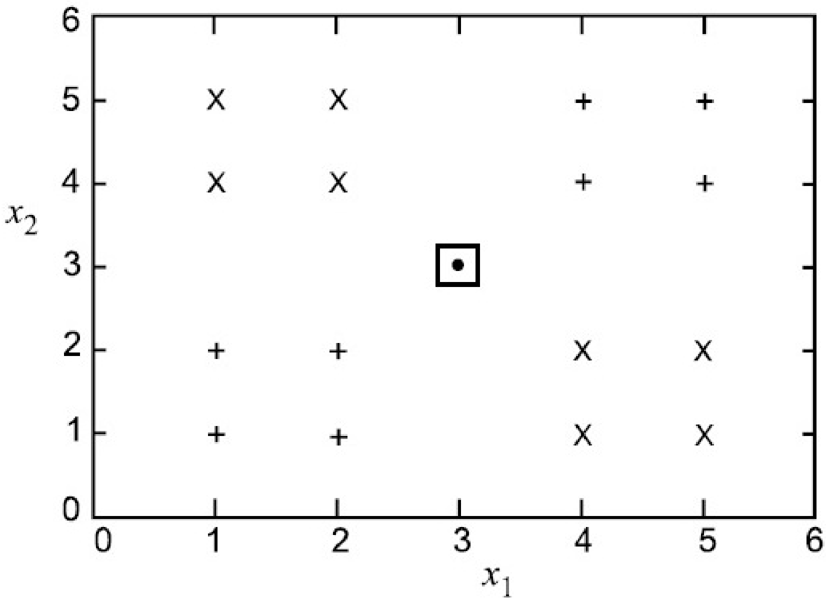
4.0, 2.0, 2 ; 5.0, 2.0, 2

1.0, 4.0, 2 ; 1.0, 5.0, 2

2.0, 4.0, 2 ; 2.0, 5.0, 2

4.0, 4.0, 1 ; 5.0, 5.0, 1

4.0, 5.0, 1 ; 5.0, 4.0, 1



รูปที่ 2.1 ตัวอย่างเซตข้อมูล

2.2 ข้อความ

นอกจากเวกเตอร์แล้ว แบบอาจถูกแทนด้วยข้อความก็ได้ นิยามของข้อความคืออักขระเดี่ยวหรือกลุ่มของอักขระ ข้อความมีรูปแบบเป็นคำศัพท์หรือประโยคของภาษาใดภาษาหนึ่งในโลก ข้อความอาจบรรจุอักขระปริมาณมหาศาลจนเป็นบทความขนาดใหญ่ นอกจากนี้ข้อความอาจไม่เกี่ยวข้องกับภาษาของมนุษย์เลยก็ได้ เช่น โครงสร้างชีวภาพอย่าง ลำดับดีเอ็นเอ (DNA Sequence) หรือ ลำดับโปรตีน (Protein Sequence) เป็นต้น ตัวอย่างของข้อความด้านล่าง ได้แก่ ยีน (Gene) ที่ถูกกำหนดด้วยแถบของรูปแบบ ดีเอ็นเอของโครโมโซม (Chromosomal DNA) ซึ่งประกอบไปด้วยไนโตรเจนพื้นฐาน 4 ชนิด ได้แก่ อะดีนีน (Adenine) กัวนีน (Guanine) ไซโตซีน (Cytosine) และ ไทมีน (Thymine) โดยใช้สัญลักษณ์ A, G, C และ T ตามลำดับ

GAAGTCCAG …

2.3 นิพจน์ตรรกะ

แบบสามารถแสดงด้วย คำอธิบายเชิงตรรกะ (Logical Description) ที่อธิบายโดยใช้การเชื่อม (Conjunction) ระหว่าง และ ($\wedge$) กับ หรือ ($\vee$) ตามรูปแบบด้านล่าง โดยที่ x_1, x_2 แทนชื่อคุณสมบัติของแบบ และ a_i, b_i แทนค่าคุณสมบัติของแบบ เมื่อ i แทน ลำดับที่ของค่าคุณสมบัตินั้น

$$(x_1 = a_1 \vee a_2 \vee \ldots) \wedge (x_2 = b_1 \vee b_2 \vee \ldots) \wedge \ldots$$

ยกตัวอย่างจากตารางที่ 2.1 ตัวแปรต่าง ๆ แทนค่า ดังต่อไปนี้

- x_1 แทน คุณสมบัติขนาดของลูกบอลกีฬา
- x_2 แทน คุณสมบัติน้ำหนักของลูกบอลกีฬา
- x_3 แทน คุณสมบัติสีของลูกบอลกีฬา
- a_1 แทน ค่าเล็ก ปานกลาง ใหญ่
- a_2 แทน ค่าเบา ปานกลาง หนัก
- a_3 แทน ค่าสีต่าง ๆ เป็นต้น

คำอธิบายของนิพจน์ตรรกะประกอบไปด้วยการเชื่อมของ การเลือกเชิงตรรกะ (Logical Disjunction) ยกตัวอย่างเช่น คำอธิบายเชิงตรรกะของลูกคริกเก็ตด้านล่างนี้ สามารถอธิบายเป็นภาษามนุษย์ได้คือ “ลูกคริกเก็ตมีสีแดงหรือขาวและทำจากหนังและมีรูปทรงกลม” การเลือกเชิงตรรกะปรากฏในคุณสมบัติแรก กล่าวคือสีของลูกคริกเก็ตสามารถถูกเลือกได้ว่าอาจเป็นสีแดงหรือสีขาวก็ได้เพื่อให้ตรรกะเป็นจริง เป็นต้น

$$(\text{Colour} = \text{Red} \vee \text{White}) \wedge (\text{Make} = \text{Leather}) \wedge (\text{Shape} = \text{Sphere})$$

2.4 เซตคลุมเครือ

ความคลุมเครือ (Fuzziness) หรือ ความไม่แน่นอน (Uncertainty) หมายถึงการที่ไม่สามารถระบุค่าที่แท้จริงของข้อมูลได้อย่างชัดเจน ซึ่งอาจมาจากข้อมูลไม่ถูกระบุค่ามาตั้งแต่ต้น ถูกระบุค่ามาเป็นช่วง หรือเป็นค่าที่ต้องใช้ความรู้สึกในการตีความเป็นต้น แบบที่มีลักษณะเช่นนี้จะถูกเรียกว่า แบบคลุมเครือ (Fuzzy Pattern) ยกตัวอย่างเช่นแบบ X ด้านล่างที่บรรจุ ค่าภาษา (Linguistic Value)

$$X = (\text{Small}, \text{Large})$$

คำอธิบายที่ได้คือ ถ้าคุณสมบัติแรกมีค่าเป็น Small และ คุณสมบัติหลังมีค่าเป็น Large แล้ว แบบที่ได้คือแบบ X ค่าภาษาจัดว่ามีความคลุมเครือเนื่องจากเป็นค่าที่ขึ้นอยู่กับความรู้สึกของผู้สังเกต เช่น ขนาดเล็กของแต่ละผู้สังเกตย่อมตีความไม่เหมือนกัน ขนาดเล็กของคนหนึ่งอาจถูกตีความเป็นขนาดใหญ่ของอีกคนหนึ่งก็ได้ ลักษณะอื่น ๆ ที่จัดว่ามีความคลุมเครืออีกได้แก่

- ข้อมูลที่ขาดความแม่นยำ (Imprecise Data) คือข้อมูลที่บรรจุ จำนวนคลุมเครือ (Fuzzy Number) ที่ค่าสมาชิกของวัตถุในเซตมีค่าเป็นช่วง (Interval) ตัวอย่างเช่น แบบ Y ด้านล่างนี้ จะ

เห็นได้ว่าคุณสมบัติแรกมีค่าเป็นช่วงตั้งแต่ 0 ถึง 1 ซึ่งยากต่อการประมาณค่า เพราะจำนวนจริงระหว่างสองค่านี้มีมากมายนับไม่ถ้วนเป็นอนันต์

$$Y = ([0, 1], 6.2, 7)$$

- ข้อมูลที่ไม่สมบูรณ์ (Incomplete Data) คือข้อมูลที่บรรจุ ค่าที่ขาดหาย (Missing Value) ตามแบบ Z ด้านล่าง จะเห็นได้ว่าคุณสมบัติแรกไม่ทราบค่าเลย อาจมาจากการเกิดข้อผิดพลาดระหว่างการเก็บข้อมูล ข้อมูลชนิดนี้จัดเป็นกรณีที่ระบุค่าได้ยากที่สุด เพราะไม่มีช่วงหรือจุดสังเกตใด ๆ ให้คาดเดาได้เลย

$$Z = (?, 6.2, 7)$$

จากคุณสมบัติต่าง ๆ ของความไม่แน่นอนที่กล่าวถึงข้างต้น ฟีเจอร์ของแบบอาจผสมระหว่าง จำนวนจริง ค่าภาษา จำนวนคลุมเครือ และ ค่าที่ขาดหาย ก็ได้ ถ้ามีลักษณะของข้อมูลแบบนี้ปรากฏใน เซตข้อมูลควรต้องประมาณค่าให้เสร็จสิ้นก่อนเข้าสู่ขั้นตอนการรู้จำแบบต่อไป

3. หน่วยวัดความใกล้เคียง

หน่วยวัดความใกล้ (Proximity Measure) ถูกใช้ในการเปรียบเทียบ ความต่าง (Dissimilarity) หรือ ระยะทาง (Distance) ระหว่างแบบสองแบบ ตัวอย่างของการนำไปใช้งานในการเปรียบเทียบว่านักศึกษามีความคล้ายกับเพื่อนคนใดมากที่สุดโดยพิจารณาจากฟีเจอร์ต่าง ๆ ที่เกี่ยวข้องกับนักศึกษา เช่น ส่วนสูง หรือ น้ำหนัก เป็นต้น หน่วยวัดความใกล้เปรียบเสมือนไม้บรรทัดหรือตลับเมตรนั่นเอง ในการจำแนกแบบ แบบฝึกฝน (Training Pattern) คือ แบบที่ทราบชั้นข้อมูลมาก่อน ในขณะที่แบบทดสอบ (Test Pattern) คือ แบบใหม่ที่ยังไม่มีการระบุชั้นข้อมูล หน่วยวัดความใกล้อาจถูกใช้ในการระบุชั้นข้อมูลที่เหมาะสมสำหรับแบบทดสอบ กล่าวคือ แบบทดสอบจะถูกกำหนดให้มีป้ายชื่อชั้นข้อมูลเดียวกับแบบฝึกฝนที่มีความต่างต่ำที่สุดหรือระยะทางใกล้ที่สุด รายละเอียดของการจำแนกแบบด้วยวิธีการนี้จะถูกกล่าวถึงในบทต่อไป

ฟังก์ชันระยะทาง (Distance Function) หรือ ฟังก์ชันความคล้าย (Similarity Function) สามารถแบ่งออกได้เป็น 2 ประเภท ได้แก่ เมตริก (Metric) และ นอนเมตริก (Non-metric) โดยที่ ฟังก์ชันเมตริกต้องมีคุณสมบัติ สะท้อนทางบวก (Positive Reflexivity) สมมาตร (Symmetry) และ อสมการสามเหลี่ยม (Triangular Inequality) ในขณะที่ฟังก์ชันนอนเมตริกมีคุณสมบัติเหล่านี้ไม่ครบทั้งหมด คำอธิบายของ

แต่ละคุณสมบัติ แสดงตามหัวข้อด้านล่างนี้ โดยที่ตัวแปร x, y, z แทนแบบใด ๆ และ d แทน ฟังก์ชันระยะทาง

- **สะท้อนทางบวก** $d(x, x) = 0$ ไม่มีระยะทางเกิดขึ้นระหว่างแบบใดแบบหนึ่งถึงตัวมันเอง หรืออาจกล่าวได้ว่าระยะทางมีค่าเป็นศูนย์
- **สมมาตร** $d(x, y) = d(y, x)$ ระยะทางระหว่างสองแบบใด ๆ ย่อมมีค่าเท่ากัน ไม่ว่าแบบใดจะเป็นต้นทางหรือปลายทางก็ตามในการวัดค่า กล่าวคือการวัดระยะทางจากแบบ x ไปยังแบบ y ย่อมมีค่าเท่ากับการวัดระยะทางจากแบบ y ไปยังแบบ x
- **อสมการสามเหลี่ยม** $d(x, y) \leq d(x, z) + d(z, y)$ ระยะทางโดยตรงระหว่างแบบต้นทางและแบบปลายทางมีค่าสั้นที่สุด เมื่อเปรียบเทียบกับระยะทางโดยอ้อมของ แบบต้นทาง (x) ไปยังแบบใดแบบหนึ่ง (z) และจากแบบนั้นไปยัง แบบปลายทาง (y)

ตัวอย่างฟังก์ชันเมตริกได้แก่ หน่วยวัดระยะทาง (Distance Measure) และ หน่วยวัดระยะทางถ่วงน้ำหนัก (Weighted Distance Measure) เป็นต้น ในขณะที่ ตัวอย่างฟังก์ชันนอนเมตริกได้แก่ ระยะทางมัธยฐานเค (k-median Distance) และ ระยะทางเพื่อนบ้านร่วมกัน (MND: Mutual Neighbourhood Distance) เป็นต้น รายละเอียดของฟังก์ชันระยะทางที่กล่าวถึงข้างต้นมีรายละเอียดตามหัวข้อย่อยต่อไปนี้

3.1 หน่วยวัดระยะทาง

หน่วยวัดระยะทางที่นิยมใช้กันอย่างแพร่หลาย คือ ตัววัดมินคอฟสกี (Minkowski Metric) ตามรูปแบบด้านล่างนี้

$$d^m(X,Y) = \left(\sum_{k=1}^{n} | x_k - y_k |^m \right)^{\frac{1}{m}}$$

ตัวแปรต่าง ๆ ที่เกี่ยวข้องกับสมการข้างต้น มีคำอธิบายดังต่อไปนี้

- d คือ ฟังก์ชันระยะทาง
- m คือ ลำดับของฟังก์ชันระยะทาง
- X, Y คือ เวกเตอร์ของแบบ
- k คือ ลำดับที่ของแบบ X หรือ แบบ Y
- n คือ จำนวนคุณสมบัติของแบบ X หรือ แบบ Y

- x_k คือ คุณสมบัติลำดับที่ k ของแบบ X
- y_k คือ คุณสมบัติลำดับที่ k ของแบบ Y

รูปแบบทั่วไปของตัววัดมินคอฟสกีสามารถเขียนโดยใช้สัญลักษณ์ L_m ในกรณีที่ m มีค่าเป็น 1 ตัววัดนี้จะเรียกว่า ระยะทาง L_1 หรือ ระยะทางแมนฮัตตัน (Manhattan Distance) แสดงตามสมการด้านล่างนี้ ตัวอย่างการคำนวณค่าระยะทางแมนฮัตตันระหว่างแบบสองแบบ แสดงตามตัวอย่างที่ 2

$$d(X,Y) = \sum_{k=1}^{n} | x_k - y_k |$$

ตัวอย่างที่ 2 ถ้า X = (4, 1, 3) และ Y = (2, 5, 1) แล้ว ระยะทางแมนฮัตตันสามารถคำนวณได้ ดังนี้

$X = (x_1, x_2, x_3) = (4, 1, 3)$

$Y = (y_1, y_2, y_3) = (2, 5, 1)$

d(X, Y) = |4 - 2| + |1 - 5| + |3 - 1| = 8

สำหรับตัวอย่างนี้ ค่าตัวแปรต่าง ๆ มีค่าดังนี้ m = 1, n = 3 และ $k \in \{1, 2, 3\}$

ตัววัดที่ได้รับความนิยมมากที่สุด ได้แก่ ระยะทางยุคลิด (Euclidean Distance) หรือ L_2 เมื่อกำหนดให้ m มีค่าเป็น 2 แสดงตามสมการด้านล่างนี้ ตัวอย่างการคำนวณค่าระยะทางยุคลิดระหว่างแบบสองแบบ แสดงตามตัวอย่างที่ 3

$$\mathrm{d}^2(\mathrm{X},\mathrm{Y}) = \sqrt{(\mathrm{x}_1 - \mathrm{y}_1)^2 + (\mathrm{x}_2 - \mathrm{y}_2)^2 + \ldots + (\mathrm{x}_\mathrm{d} - \mathrm{y}_\mathrm{d})^2}$$

ตัวอย่างที่ 3 ถ้า X = (4, 1, 3) และ Y = (2, 5, 1) แล้ว ระยะทางยุคลิดสามารถคำนวณได้ ดังนี้

$$d^2(X,Y) = \sqrt{(4-2)^2 + (1-5)^2 + (3-1)^2} = 4.9$$

หน่วยวัดระยะทางมีความเหมาะสมที่จะใช้กับเซตข้อมูล ที่ฟีเจอร์ต่าง ๆ มีค่าอยู่ในช่วงที่ใกล้เคียงกัน ถ้าช่วงของค่าคุณสมบัติมีความแตกต่างกันมากเกินไป อาจจะประสบความล้มเหลวในการจำแนกแบบได้ เนื่องจากในการคำนวณหน่วยวัดระยะทาง ช่วงที่มีขนาดกว้างย่อมมี นัยสำคัญ (Significant) มากกว่าช่วงที่มีขนาดแคบ สังเกตแบบ M และ N ด้านล่างนี้ คุณสมบัติลำดับสุดท้ายมีความกว้างของช่วงมากกว่าคุณสมบัติสี่ลำดับแรก ดังนั้นเมื่อคำนวณตัววัดมินคอฟสกีแล้ว ทำให้ความสำคัญของการคำนวณขึ้นอยู่กับคุณสมบัติสุดท้ายมากที่สุด และละเลยการให้ความสำคัญกับ

คุณสมบัติที่เหลือทั้งหมด จึงเกิดปัญหาขึ้นว่า คุณสมบัติสี่ลำดับแรกแทบไม่มีผลกับการคำนวณความคล้ายเลย ทั้ง ๆ ที่ การคำนวณความคล้ายควรให้ความสำคัญกับทุกคุณสมบัติอย่างเท่าเทียมกัน

M = (8, 4, 2, 16, 1024)

N = (9, 3, 11, 23, 1)

อย่างไรก็ตาม ปัญหานี้สามารถแก้ไขได้โดยใช้ การทำให้อยู่ในรูปปกติ (Normalisation) เพื่อให้ฟีเจอร์ทั้งหมดมีความสำคัญเทียบเท่ากัน กล่าวคือคุณสมบัติทั้งหมดของแบบจะมีค่าอยู่ในช่วงเดียวกัน วิธีการทำให้อยู่ในรูปปกติมีวิธีการแสดงตามสองขั้นตอนด้านล่างนี้ ผลลัพธ์ที่ได้คือ ช่วงทั้งหมดของแบบจะมีความกว้างเท่ากันหมด คือตั้งแต่ 0 ถึง 1 ตัวอย่างของการทำให้อยู่ในรูปปกติแสดงตาม ตัวอย่างที่ 4

การทำให้อยู่ในรูปปกติ

1. หาค่าสูงสุดในแต่ละฟีเจอร์ของแบบทั้งหมด
2. หารแต่ละค่าฟีเจอร์ด้วยค่าสูงสุดในสดมภ์เดียวกัน

ตัวอย่างที่ 4 พิจารณาเซตข้อมูลของแบบที่มี 2 ฟีเจอร์ ดังนี้

$X_1 = (2, 120)$

$X_2 = (8, 533)$

$X_3 = (1, 987)$

$X_4 = (15, 1121)$

$X_5 = (18, 1023)$

ค่าสูงสุดของฟีเจอร์ที่ 1 คือ 18

ค่าสูงสุดของฟีเจอร์ที่ 2 คือ 1,121

หลังจากทำให้อยู่ในรูปปกติแล้ว ฟีเจอร์มีค่าดังนี้

$X'_1 = (2/18, 120/1{,}121) = (0.11, 0.11)$

$X'_2 = (8/18, 533/1{,}121) = (0.44, 0.48)$

$X'_3 = (1/18, 987/1{,}121) = (0.06, 0.88)$

$X'_4 = (15/18, 1{,}121/1{,}121) = (0.83, 1.0)$

$X'_5 = (18/18, 1{,}023/1{,}121) = (1.0, 0.91)$

3.2 หน่วยวัดระยะทางถ่วงน้ำหนัก

การทำให้อยู่ในรูปปกติมีข้อเสียคือ ทำให้แต่ละฟีเจอร์มีขอบเขตของช่วงที่ผิดเพี้ยนไปจากเซตข้อมูลดั้งเดิม อย่างไรก็ตามยังมีหน่วยวัดอีกประเภทหนึ่งได้แก่ หน่วยวัดระยะทางถ่วงน้ำหนักที่ไม่ปรับเปลี่ยนความกว้างของช่วง แต่ถูกใช้ในกรณีที่ต้องการให้น้ำหนักความสำคัญกับฟีเจอร์ใดเป็นพิเศษมากกว่าฟีเจอร์อื่น กล่าวคือค่าที่ขอบเขตของช่วงกว้างอาจจะให้ความสำคัญน้อย ในทางกลับกันค่าที่มีขอบเขตของช่วงแคบอาจจะให้ความสำคัญมากกว่า ทั้งนี้เพื่อจุดประสงค์ในการสร้างความสมดุลระหว่างคุณสมบัติที่มีช่วงกว้างและคุณสมบัติที่มีช่วงแคบนั่นเอง หน่วยวัดระยะทางถ่วงน้ำหนัก มีสูตรการคำนวณแสดงตามสมการด้านล่างนี้

$$d^{m}(X,Y)=\left(\sum_{k=1}^{d} w_k \times |x_k - y_k|^{m}\right)^{\frac{1}{m}}$$

ตัวแปรต่าง ๆ ที่เกี่ยวข้องกับสมการข้างต้น มีคำอธิบายเหมือนตัววัดมินคอฟสกีในหัวข้อ 3.1 ในส่วนของตัวแปรที่เพิ่มมา 1 ตัว มีคำอธิบายดังนี้

- w_k คือ น้ำหนัก (Weight) ของฟีเจอร์ที่ k

วิธีการการคำนวณหน่วยวัดระยะทางถ่วงน้ำหนักแสดงตามตัวอย่างที่ 5 ฟีเจอร์ที่สองมีน้ำหนักสูงสุด ผลที่เกิดขึ้นตามมาคือฟีเจอร์นี้จะมีผลต่อการคำนวณระยะทางมากที่สุด ในทางกลับกันคุณสมบัติที่สามที่มีค่าน้ำหนักต่ำสุดจะมีผลต่อการคำนวณระยะทางน้อยที่สุด อย่างไรก็ตาม ข้อเสียของหน่วยวัดระยะทางถ่วงน้ำหนักคือการกำหนดค่าน้ำหนักนั่นเอง เนื่องจากเป็นการยากที่เลือกค่าน้ำหนักที่เหมาะสมในแต่ละเซตข้อมูล ยิ่งไปกว่านั้น ถ้าการกำหนดค่าน้ำหนักเกิดขึ้นโดยให้มนุษย์ตัดสินแล้วมนุษย์แต่ละคนย่อมกำหนดค่าน้ำหนักแตกต่างกัน ส่งผลให้ค่าระยะทางที่ได้จากการคำนวณเซตข้อมูลเดียวกัน มีค่าไม่เท่ากันอีกด้วย

ตัวอย่างที่ 5 ถ้า X = (4, 1, 3), Y = (2, 5, 1) และ $w_1 = 0.3$, $w_2 = 0.6$, $w_3 = 0.1$ แล้ว

$$d^{2}(X,Y)=\sqrt{0.3\times(4-2)^{2}+0.6\times(1-5)^{2}+0.1\times(3-1)^{2}}=3.35$$

3.3 ระยะทางมัธยฐานเค

ระยะทางประเภทนี้ใช้ตัวดำเนินการ k-median ในการคืนค่าความแตกต่างที่น้อยเป็นลำดับที่ k จากความแตกต่างของฟีเจอร์ทั้งหมด สูตรการคำนวณระยะทางมัธยฐานเคแสดงตามสมการด้านล่างนี้

โดยที่ตัวแปรต่าง ๆ ที่เกี่ยวข้องมีคำอธิบายเหมือนตัววัดมินคอฟสกีในหัวข้อ 3.1 และ ลำดับขั้นตอนการคำนวณมี 3 แสดงต่อจากสมการด้านล่างนี้

$$d(X, Y) = \text{k-median}\{|x_1 - y_1|, \ldots, |x_n - y_n|\}$$

ระยะทางมัธยฐานเค

1. คำนวณ เวกเตอร์ความแตกต่าง (Difference Vector) ดังนี้
   1.1 นำค่าแต่ละคุณสมบัติของแบบมาลบกัน
   1.2 เปลี่ยนค่าผลต่างที่ได้ให้เป็นค่าสัมบูรณ์
2. เรียงลำดับค่าในเวกเตอร์ความแตกต่างจากน้อยไปหามาก
3. นำค่าลำดับที่ k ในเวกเตอร์ระยะทางที่เรียงลำดับแล้ว มาใช้เป็นค่าระยะทาง

ตัวอย่างการคำนวณระยะทางมัธยฐานเคแสดงตามตัวอย่างที่ 6 ด้านล่างนี้ โดยทั่วไปค่า k จะถูกประมาณค่าที่ตำแหน่งกึ่งกลาง การกำหนดค่า k เป็นเท่าใดนั้น ค่าที่กำหนดต้องถูกใช้ตลอดทั้งเซตข้อมูลเพื่อให้มีมาตรฐานเดียวกัน

ตัวอย่างที่ 6 ถ้า X = (50, 3, 100, 29, 62, 140) และ Y = (55, 15, 80, 50, 70, 170) กำหนดค่าพารามิเตอร์ k = 3 แล้วขั้นตอนการคำนวณ k-median มีดังต่อไปนี้

คำนวณ เวกเตอร์ความแตกต่าง ดังนี้

{|50 – 55|, |3 - 15|, |100 - 80|, |29 - 50|, |62 - 70|, |140 -170|} =

{5, 12, 20, 21, 8, 30}

เรียงลำดับค่าในเวกเตอร์ความแตกต่าง ได้ {5, 8, 12, 20, 21, 30}

นำค่าลำดับที่ 3 ในเวกเตอร์ที่เรียงลำดับแล้ว มาใช้เป็นค่าระยะทาง

ดังนั้น d(X, Y) = 12

3.4 ระยะทางเพื่อนบ้านร่วมกัน

กำหนดให้ตัวแปร N คือจำนวนของแบบทั้งหมดในเซตข้อมูล เพื่อนบ้านของแบบใด ๆ คือแบบที่เหลือทั้งหมดที่ไม่ใช่ตัวมันเอง ดังนั้นเพื่อนบ้านของแบบใด ๆ จะมีจำนวน N – 1 แบบ นอกจากนี้เพื่อนบ้านทั้งหมดของแบบแต่ละตัวจะถูกกำหนดค่า อันดับ (Ranking) ตั้งแต่ 1 ถึง N – 1 โดยที่ 1 แทนเพื่อนบ้านใกล้ที่สุด และ 2 แทนเพื่อนบ้านใกล้เป็นลำดับที่สอง จนถึง N – 1 แทนเพื่อนบ้านไกลที่สุด ระยะทางเพื่อนบ้านใกล้ที่สุดระหว่างแบบสองแบบ แสดงตามสมการด้านล่างนี้

$$MND(u, v) = NN(u, v) + NN(v, u)$$

ตัวแปรและตัวดำเนินการต่าง ๆ ที่เกี่ยวข้องกับสมการข้างต้น มีคำอธิบายดังต่อไปนี้

- u และ v คือ แบบสองแบบใด ๆ ที่ต้องการคำนวณค่าระยะทางระหว่างกัน
- NN(u, v) คือ ตัวดำเนินการที่ใช้คืนค่าอันดับ เมื่อกำหนดให้ u คือแบบที่กำลังพิจารณา และ v คือเพื่อนบ้านของ u กล่าวคือ ใช้ในการคืนค่าเป็นจำนวนเต็มว่า v อยู่ใกล้ u เป็นลำดับที่เท่าใด เมื่อเปรียบเทียบกับเพื่อนบ้านทั้งหมดของ u
- NN(v, u) คำนวณคล้าย NN(u, v) แต่คิดในทางตรงข้ามกัน กล่าวคือ v คือแบบที่กำลังพิจารณา และ u คือเพื่อนบ้านของ v แทน
- MND(u, v) คือ ตัวดำเนินการที่ใช้ในการคืนค่าระยะทางเพื่อนบ้านร่วมกันระหว่างแบบ u และ v

ตัวอย่างการคำนวณระยะทางเพื่อนบ้านร่วมกันจากสูตรดังกล่าว แสดงตามตัวอย่างที่ 6 และ 7

ตัวอย่างที่ 6 ตารางที่ 2.1 แสดงค่าอันดับของแบบ A, B, C จากรูปที่ 2.2 รายละเอียดของค่าต่าง ๆ ในตาราง มีคำอธิบายดังต่อไปนี้ หัวสดมภ์แทนแบบที่กำลังพิจารณา ในขณะที่หัวแถวแทนอันดับของเพื่อนบ้าน และค่าในตารางแทนแบบที่กำลังถูกพิจารณาให้เป็นเพื่อนบ้าน ยกตัวอย่างเช่นค่า D ในแถวแรกและหลักแรกหมายถึง D เป็นเพื่อนบ้านใกล้ที่สุดของ A (ใกล้เป็นลำดับที่ 1) จึงเขียนสมการโดยใช้ตัวดำเนินการได้เป็น NN(A, D) = 1 อีกตัวอย่างหนึ่งได้แก่ค่า C ในแถวแรกและหลักสุดท้ายตีความได้ว่า C คือเพื่อนบ้านไกลที่สุดของ A (ใกล้เป็นลำดับที่ 5) กล่าวคือ NN(A, C) = 5 เป็นต้น

ตัวอย่างการคำนวณค่าระยะทาง MND ระหว่างแบบ A, B, C สามารถเขียนแจกแจงเป็นสมการได้ ดังต่อไปนี้

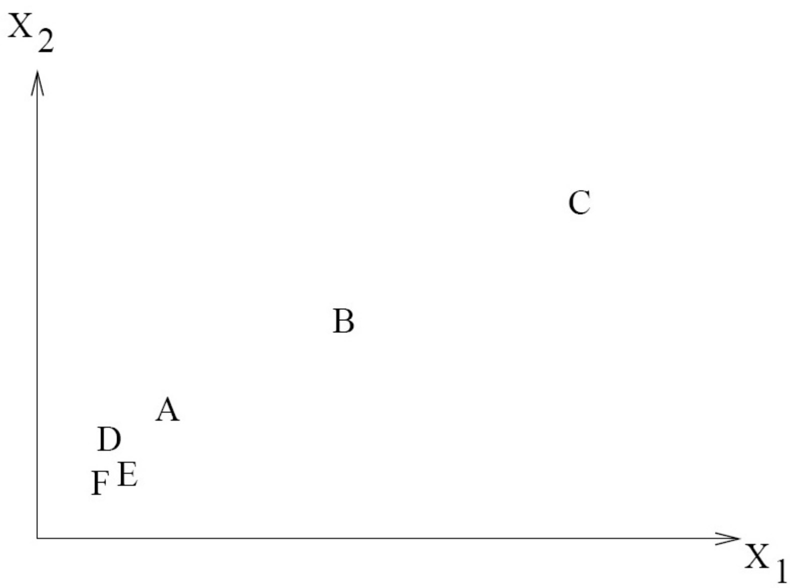
$$MND(A, B) = NN(A, B) + NN(B, A) = 4 + 1 = 5$$

$$MND(B, C) = NN(B, C) + NN(C, B) = 2 + 1 = 3$$

$$MND(A, C) = NN(A, C) + NN(C, A) = 5 + 2 = 7$$

ตารางที่ 2.2 ค่าอันดับของแบบ A, B, C

	1	2	3	4	5
A	D	E	F	B	C
B	A	C	D	E	F
C	B	A	D	E	F

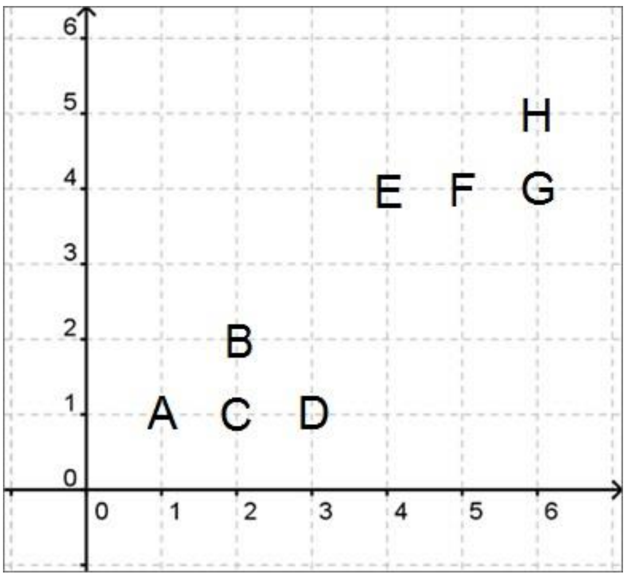


รูปที่ 2.2 กราฟของแบบ A – E

ตัวอย่างที่ 7 จงคำนวณค่าระยะทางเพื่อนบ้านร่วมกัน (MND) ระหว่างแบบ D และ E โดยใช้ระยะทางแมนแฮตตัน กำหนดค่าคุณสมบัติของแบบทั้งหมด ดังต่อไปนี้

A = (1, 1);	B = (2, 2);	C = (2, 1);	D = (3, 1)
E = (4, 4);	F = (5, 4);	G = (6, 4);	H = (6, 5)

จากแบบทั้งหมดข้างต้น สามารถวาดกราฟ 2 มิติ โดยที่แกน X แทนคุณสมบัติแรก และแกน Y แทนคุณสมบัติหลัง กราฟดังกล่าวแสดงตามด้านล่างนี้



รูปที่ 2.3 กราฟของแบบ A – H

ระยะทางแมนแฮตตันระหว่างแบบสองแบบใด ๆ แสดงตามตารางที่ 2.3 ด้านล่างนี้ โดยที่ตัวเลขในตารางคือค่าระยะทางดังกล่าว หัวแถวและหัวสดมภ์แทนแบบสองแบบที่ห่างกันด้วยระยะทางนั้น ยกตัวอย่างเช่นค่า 6 ในแถวที่ 5 และสดมภ์ที่ 1 ตีความได้ว่า ระยะทางระหว่างแบบ E และ A คือ 6 หน่วย สังเกตได้ว่ามีค่าเท่ากับ ระยะทางระหว่างแบบ A และ E ในแถวที่ 1 และสดมภ์ที่ 5 เนื่องจากระยะทางแมนแฮตตันเป็นระยะทางเมตริกซ์ ซึ่งมีคุณสมบัติสมมาตรตามสมการ d(E, A) = d(A, E) นอกจากนี้ ถ้าสังเกตค่าในตำแหน่งที่สดมภ์และแถวมีค่าเท่ากันเช่น แถวที่ 1 และสดมภ์ที่ 1 มีค่าระยะทางเป็น 0 เพราะว่าระยะทางเมตริกซ์มีคุณสมบัติสะท้อนทางบวกตามสมการ d(A, A) = 0

ตารางที่ 2.3 เมตริกซ์ระยะทางของแบบ A - H

	A	B	C	D	E	F	G	H
A	0	2	1	2	6	7	8	9
B	2	0	1	2	4	5	6	7
C	1	1	0	1	5	6	7	8
D	2	2	1	0	4	5	6	7
E	6	4	5	4	0	1	2	3
F	7	5	6	5	1	0	1	2
G	8	6	7	6	2	1	0	1
H	9	7	8	7	3	2	1	0

จากเมตริกซ์ระยะทางแมนแฮตตันข้างต้น สามารถเขียนตารางค่าอันดับของแบบทั้งหมด ตามตารางที่ 2.4 ด้านล่างนี้ บางช่องในตารางอาจมีหลายแบบบรรจุอยู่เช่น แถวที่ 3 และหลักที่ 1 มีทั้งหมด 3 แบบ จึงตีความได้ว่า ทั้งแบบ A, B, C เป็นเพื่อนบ้านที่ใกล้ที่สุดร่วมกัน (ใกล้เป็นลำดับที่ 1) ของแบบ C ดังนั้น แบบ C ไม่มีเพื่อนบ้านใกล้ที่สุดลำดับที่ 2 และ 3 แต่เพื่อนบ้านใกล้ที่สุดลำดับต่อไปของแบบ C จะเป็นแบบลำดับที่ 4 แทน ได้แก่แบบ E ในแถวที่ 3 และสดมภ์ที่ 4

จากค่าอันดับของแบบในตารางที่ 2.4 สามารถคำนวณค่าระยะทางเพื่อนบ้านใกล้ที่สุดระหว่างแบบ D และ E ได้ตามสมการด้านล่างนี้

$$
\begin{aligned}
MND(D, E) &= NN(D, E) + NN(E, D) \\
&= 4 + 4 \\
&= 8
\end{aligned}
$$

ตารางที่ 2.4 ค่าอันดับของแบบ A - H

	1	2	3	4	5	6	7
A	C	B, D	-	E	F	G	H
B	C	A, D	-	E	F	G	H
C	A, B, D	-	-	E	F	G	H
D	C	A, B	-	E	F	G	H
E	F	G	H	B, D	-	C	A
F	E, G	-	H	B, D	-	C	A
G	F, H	-	E	B, D	-	C	A
H	G	F	E	B, D	-	C	A

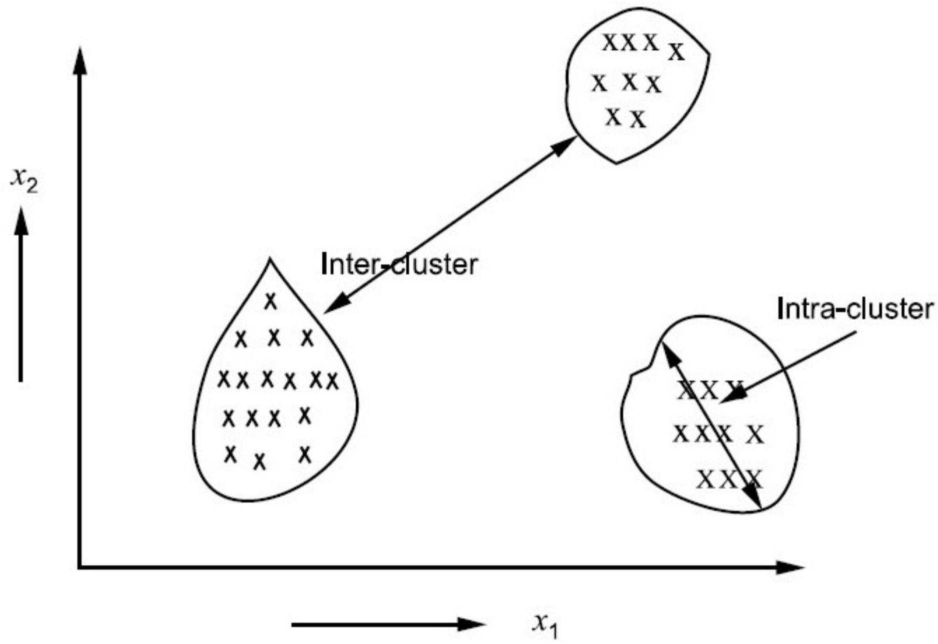
จากสองตัวอย่างข้างต้นจะเห็นได้ว่า ระยะทางเพื่อนบ้านใกล้ที่สุดมีพื้นฐานอยู่บนการเปรียบเทียบบนแบบสองแบบว่า แบบหนึ่งไกลจากอีกแบบเป็นอันดับที่เท่าไหร่จากเพื่อนบ้านทั้งหมด อย่างไรก็ตามในการคำนวณอันดับจำเป็นต้องใช้ระยะทางอื่นร่วมด้วย เพื่อเปรียบเทียบว่าเพื่อนบ้านแต่ละแบบใกล้แบบที่กำลังพิจารณาเป็นอันดับที่เท่าใด หน่วยวัดที่ใช้สามารถเป็นระยะทางประเภทใดก็ได้ ตัวอย่างที่ 6 ใช้ระยะทางยุคลิด ในขณะที่ตัวอย่างที่ 7 ใช้ระยะทางแมนแฮตตัน ความแตกต่างระหว่างระยะทางทั้ง 2 ชนิดนี้คือ ระยะทางยุคลิดคือระยะทางปกติระหว่างแบบสองแบบในแนวเส้นตรงซึ่งอาจสามารถวัดได้ด้วยไม้บรรทัด แต่ระยะทางแมนแฮตตันคือระยะทางที่เกิดขึ้นจากผลรวมของระยะทางในแนวนิ่งและระยะทางในแนวนอนระหว่างแบบสองแบบ

4. การจัดกลุ่มข้อมูล

การจัดกลุ่มข้อมูล (Clustering) คือกระบวนการทางด้านการทำเหมืองข้อมูล ที่ใช้ในการแบ่งแบบทั้งหมดในเซตข้อมูลออกเป็นกลุ่มต่าง ๆ ที่เรียกว่า กลุ่มข้อมูล (Cluster) โดยที่แบบในกลุ่มข้อมูลเดียวกันมีความคล้ายกันมากกว่าแบบจากต่างกลุ่มข้อมูลกัน ปัจจัยที่มีผลต่อคุณภาพของการจัดกลุ่มข้อมูลได้แก่ ระยะทางภายในและภายนอกกลุ่มข้อมูล แสดงตามรูปที่ 2.4 ซึ่งมีนิยามตามคำอธิบายด้านล่างนี้

- ระยะทางภายในกลุ่มข้อมูล (Intra-cluster Distance) คือระยะทางที่วัดจากแบบสองแบบในกลุ่มข้อมูลเดียวกันที่อยู่ไกลกันที่สุด
- ระยะทางภายนอกกลุ่มข้อมูล (Inter-cluster Distance) คือระยะทางที่วัดจากแบบสองแบบจากต่างกลุ่มข้อมูลกันที่อยู่ใกล้กันที่สุด

กลุ่มข้อมูลที่ดีควรมีค่าระยะทางภายในกลุ่มข้อมูลต่ำ เนื่องจากระยะทางนี้ยิ่งมีค่าต่ำหมายความว่ากลุ่มข้อมูลยิ่งมีขนาดเล็ก ผลลัพธ์ที่ได้คือแบบต่าง ๆ ที่อยู่ภายในกลุ่มข้อมูลขนาดเล็กจะอยู่ใกล้ชิดกันมาก ทำให้แบบมีความคล้ายกันมาก นอกจากนี้ ระหว่างสองกลุ่มข้อมูลใด ๆ ระยะทางภายนอกกลุ่มข้อมูลควรมีค่าสูง เพราะระยะทางนี้ยิ่งมีค่าสูงหมายถึงกลุ่มข้อมูลอยู่ไกลกันมาก ย่อมส่งผลให้แบบจากต่างกลุ่มข้อมูลกันมีความคล้ายกันน้อย



รูปที่ 2.4 ระยะทางภายในกลุ่มข้อมูล และ ระยะทางภายนอกกลุ่มข้อมูล

แนวคิดพื้นฐานของการจัดกลุ่มข้อมูลคือ ความพยายามให้แบบที่มีความคล้ายกันควรอยู่ในกลุ่มข้อมูลเดียวกัน ในทางตรงกันข้าม แบบมีความแตกต่างกันควรอยู่ต่างกลุ่มข้อมูลกัน โดยใช้หน่วยวัดระยะทางในการเปรียบเทียบความคล้ายหรือความต่างระหว่างแบบ จากแนวคิดดังกล่าว สามารถแสดงขั้นตอนการจัดกลุ่มข้อมูลได้ตามตัวอย่างที่ 8 ด้านล่างนี้

ตัวอย่างที่ 8 เซตข้อมูลเวกเตอร์ของแบบด้านล่างนี้ แสดงตามกราฟในรูปที่ 2.5 แบบกลุ่มนี้มี 2 ฟีเจอร์ โดยที่แกน X แทนฟีเจอร์แรก และแกน Y แทนฟีเจอร์หลัง

$X_1 = (1, 1)$; $X_2 = (2, 1)$; $X_3 = (1, 2)$; $X_4 = (2, 2)$; $X_5 = (6, 1)$

$X_6 = (7, 1)$; $X_7 = (6, 2)$; $X_8 = (6, 7)$; $X_9 = (7, 7)$; $X_{10} = (7, 6)$

ระยะทางระหว่าง 2 เวกเตอร์ใด ๆ แสดงตามตารางที่ 2.5 โดยวัดจาก ระยะทางยุคลิดยกกำลังสอง (Squared Euclidean Distance) ตามสมการด้านล่างนี้ ตัวแปรต่าง ๆ ที่เกี่ยวข้องมีคำอธิบาย

เหมือนตัววัดมินคอฟสกีในหัวข้อ 3.1 ระยะทางยุคลิดยกกำลังสองมีความคล้ายกับระยะทางยุคลิดแต่แตกต่างกันที่ระยะทางยุคลิดยกกำลังสองไม่มีการหาค่ารากที่สอง

$$d^2(X,Y) = (x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_d - y_d)^2$$

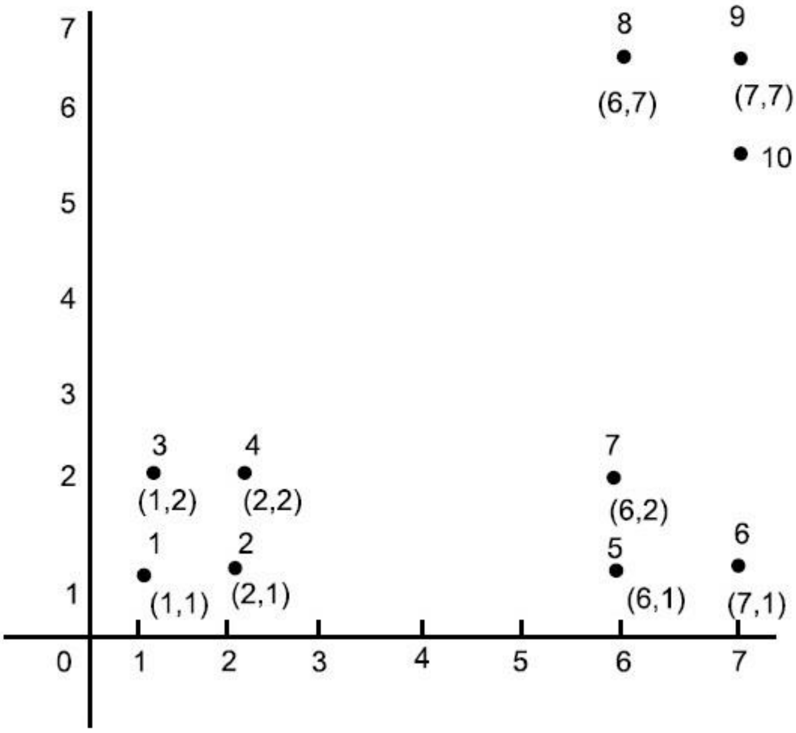
สำหรับขั้นตอนการจัดกลุ่มข้อมูล ผู้ใช้อาจมีการกำหนดค่า ขีดแบ่ง (Threshold) ซึ่งถูกใช้ในการพิจารณาว่า 2 เวกเตอร์ใด ๆ จะอยู่ในกลุ่มข้อมูลเดียวกัน ถ้าระยะทางระหว่าง 2 เวกเตอร์นั้น มีค่าน้อยกว่าขีดแบ่ง ยกตัวอย่างเช่น ถ้ากำหนดให้ขีดแบ่งมีค่าเท่ากับ 5 หน่วย แล้วจะได้กลุ่มข้อมูลทั้งหมด ดังนี้

กลุ่มข้อมูลที่ 1 ประกอบด้วย X_1, X_2, X_3, X_4 เนื่องจากทั้ง 4 เวกเตอร์นี้ มีระยะทางระหว่างกันไม่เกิน 5 หน่วย และด้วยเหตุผลเดียวกันนี้ กลุ่มข้อมูลที่เหลือได้แก่

กลุ่มข้อมูลที่ 2 ประกอบด้วย X_5, X_6, X_7

กลุ่มข้อมูลที่ 3 ประกอบด้วย X_8, X_9, X_{10}

เมื่อพิจารณากราฟในรูปที่ 2.5 จะเห็นได้ว่าเวกเตอร์ในกลุ่มข้อมูลเดียวกันจะกระจุกตัวอยู่รวมกันและห่างจากเวกเตอร์จากกลุ่มข้อมูลอื่น



รูปที่ 2.5 เซตข้อมูลของเวกเตอร์ 2 มิติ

ตารางที่ 2.5 ตำแหน่ง ij ในเมตริกซ์ แทน ระยะทางระหว่าง X_i และ X_j

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
X_1	0	1	1	2	25	36	26	61	72	61
X_2	1	0	2	1	16	25	17	52	61	50
X_3	1	2	0	1	26	37	25	50	61	52
X_4	2	1	1	0	17	26	16	41	50	41
X_5	25	16	26	17	0	1	1	36	37	26
X_6	36	25	37	26	1	0	2	37	36	25
X_7	26	17	25	16	1	2	0	25	26	17
X_8	61	52	50	41	36	37	25	0	1	2
X_9	72	61	61	50	37	36	26	1	0	1
X_{10}	61	50	52	41	26	25	17	2	1	0

เคมีนส์ (k-Means) [10] คือขั้นตอนวิธีที่ใช้ในการจัดกลุ่มข้อมูลที่มีความซับซ้อนกว่าวิธีการก่อนหน้านี้ โดยผู้ใช้สามารถกำหนดจำนวนกลุ่มข้อมูลที่ต้องการได้ กำหนดให้ n คือจำนวนแบบทั้งหมดในเซตข้อมูล และ k คือจำนวนกลุ่มข้อมูล เคมีนส์มีการประมวลผล ตามลำดับขั้นตอนด้านล่างนี้ กำหนด

ขั้นตอนวิธีเคมีนส์

1. สุ่มเลือกแบบจำนวน k แบบ เพื่อกำหนดให้เป็น จุดศูนย์กลางกลุ่มข้อมูลเริ่มต้นจำนวน k กลุ่ม
2. จากแบบที่เหลือ n – k แบบ วัดระยะทางระหว่างแต่ละแบบไปยังทุกจุดศูนย์กลางกลุ่มข้อมูล ถ้าแบบอยู่ใกล้จุดศูนย์กลางไหน กำหนดให้แบบอยู่ในกลุ่มข้อมูลเดียวกับจุดศูนย์กลางนั้น
3. คำนวณจุดศูนย์กลางของแต่ละกลุ่มข้อมูลใหม่ โดยใช้ค่าเฉลี่ยของแบบทั้งหมดในกลุ่มข้อมูล จุดศูนย์กลางใหม่ที่ได้จำนวน k จุด ไม่ใช่แบบในเซตข้อมูลเนื่องจากเป็นเพียงค่าเฉลี่ยจากทุกแบบในกลุ่มข้อมูลเดียวกัน
4. จากแบบทั้งหมด n แบบ กำหนดให้แต่ละแบบเป็นสมาชิกในกลุ่มข้อมูลเดียวกันกับ ค่าเฉลี่ยที่อยู่ใกล้ที่สุด ขั้นตอนนี้คล้ายขั้นตอนที่ 2 แต่เปลี่ยนจากแบบที่ถูกสุ่มตอนเริ่มต้นเป็นค่าเฉลี่ยของแบบแทน
5. ในกรณีที่แบบทั้งหมดไม่มีการเปลี่ยนกลุ่มข้อมูล ให้หยุดการทำงาน แต่ถ้ามีอย่างน้อย 1 แบบเกิดการเปลี่ยนกลุ่มข้อมูล ให้กลับไปยังขั้นตอนที่ 3 ใหม่

คุณภาพของกลุ่มข้อมูลที่ได้จากวิธีการเคมีนส์อยู่กับการเลือก จุดศูนย์กลางกลุ่มข้อมูลเริ่มต้น (Initial Cluster Centre) เนื่องจากจุดนี้มาจากการสุ่มในขั้นตอนที่ 1 กลุ่มข้อมูลที่ได้ในแต่ละครั้งจึงอาจไม่เหมือนกัน แนวทางการปรับปรุงวิธีการเคมีนส์ อาจเปลี่ยนจากการสุ่มเป็นการเจาะจงเลือกจุดที่ส่งผลดีแทน ตัวอย่างการทำงานของวิธีการเคมีนส์แสดงตามตัวอย่างที่ 8 และตัวอย่างที่ 9

ตัวอย่างที่ 8 กำหนดแบบด้านล่างนี้

A = (1, 1); B = (1, 2); C = (2, 2);
D = (6, 2); E = (7, 2);
F = (6, 6); G = (7, 6)

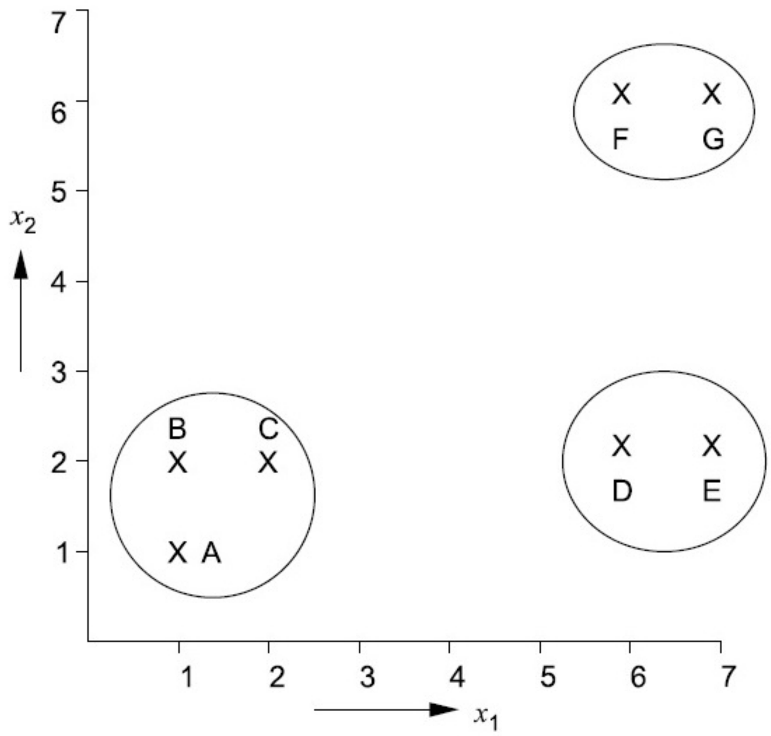
ขั้นตอนที่ 1 กำหนด จุดศูนย์กลางกลุ่มข้อมูลเริ่มต้น คือ A, D, F ซึ่งสมมติว่าได้จากการสุ่ม

ขั้นตอนที่ 2 กลุ่มข้อมูลที่ได้ คือ {A, B, C}; {D, E} และ {F, G} เนื่องจาก
B, C อยู่ใกล้ A มากกว่า D, F ดังนั้น A, B, C จึงอยู่ในกลุ่มข้อมูลเดียวกัน
E อยู่ใกล้ D มากกว่า A, F ดังนั้น E, D จึงอยู่ในกลุ่มข้อมูลเดียวกัน
G อยู่ใกล้ F มากกว่า A, D ดังนั้น F, G จึงอยู่ในกลุ่มข้อมูลเดียวกัน

ขั้นตอนที่ 3 คำนวณค่าเฉลี่ยของแต่ละกลุ่มข้อมูล ดังนี้
ค่าเฉลี่ยของ A, B, C คือ ((1 + 1 + 2) / 3, (1 + 2 + 2) / 3)) = (1.33, 1.66) = C_1
ค่าเฉลี่ยของ D, E คือ ((6 + 7) / 2, (2 + 2) / 2) = (6.5, 2) = C_2
ค่าเฉลี่ยของ F, G คือ ((6 + 7) / 2, (6 + 6) / 2) = (6.5, 6) = C_3

ขั้นตอนที่ 4 กลุ่มข้อมูลที่ได้ คือ {A, B, C}; {D, E} และ {F, G} เนื่องจาก
A, B, C อยู่ใกล้ C_1 มากกว่า C_2, C_3 ดังนั้น A, B, C จึงอยู่ในกลุ่มข้อมูลของ C_1
D, E อยู่ใกล้ C_2 มากกว่า C_1, C_3 ดังนั้น E, D จึงอยู่ในกลุ่มข้อมูลของ C_2
F, G อยู่ใกล้ C_3 มากกว่า C_1, C_2 ดังนั้น F, G จึงอยู่ในกลุ่มข้อมูลของ C_3

ขั้นตอนที่ 5 แบบในแต่ละกลุ่มข้อมูลไม่มีการเปลี่ยนกลุ่ม ดังนั้นจึงจบเงื่อนไขการทำงาน
ผลลัพธ์ของการจัดกลุ่มข้อมูลโดยวิธีเคมีนส์แสดงตามรูปที่ 2.5
เมื่อ x_1 คือ ค่าคุณสมบัติแรก และ x_2 คือค่าคุณสมบัติหลัง ของแบบ



รูปที่ 2.6 กลุ่มข้อมูลเคมีนส์ของแบบ A – G

ตัวอย่างที่ 9 กำหนดแบบ A1 ถึง A8 ด้านล่างนี้ เนื่องจากแบบทั้งหมดมี 2 ฟีเจอร์ จึงสามารถแสดงตำแหน่งของแบบในกราฟ 2 มิติได้ตามรูปที่ 2.7 โดยที่แกน x และแกน y คือคุณสมบัติแรกและคุณสมบัติหลังของแบบตามลำดับ

A1 = (2, 10)

A2 = (2, 5)

A3 = (8, 4)

A4 = (5, 8)

A5 = (7, 5)

A6 = (6, 4)

A7 = (1, 2)

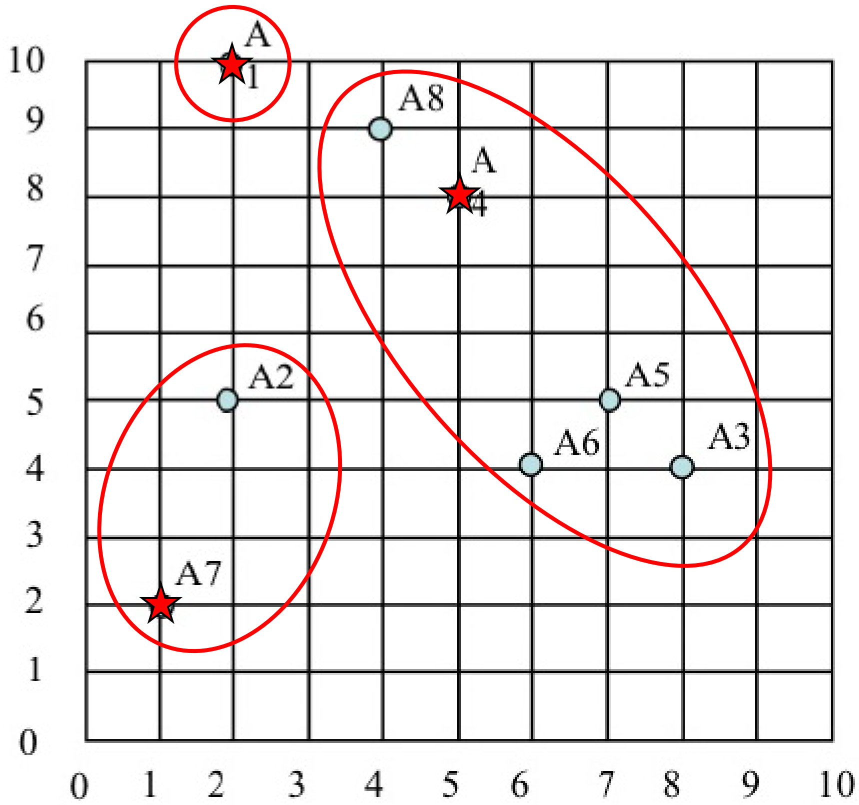
A8 = (4, 9)

ถ้าผู้ใช้กำหนดค่าพารามิเตอร์ k ให้มีค่าเท่ากับ 3 แล้ว หมายความว่าวิธีการเคมีนส์จะทำการแบ่งแบบทั้งหมดในเซตข้อมูลออกเป็น 3 กลุ่มข้อมูล สมมุติให้จุดศูนย์กลางกลุ่มข้อมูลเริ่มต้น ได้แก่ A1, A4 และ A7 เคมีนส์จะมีการประมวลผลในแต่ละรอบ ตามขั้นตอนดังต่อไปนี้

รอบที่ 1

จุดศูนย์กลางกลุ่มข้อมูลที่ 1 คือ A1 (โจทย์กำหนดมาให้ แสดงด้วยสัญลักษณ์รูปดาว)
จุดศูนย์กลางกลุ่มข้อมูลที่ 2 คือ A4 (โจทย์กำหนดมาให้ แสดงด้วยสัญลักษณ์รูปดาว)
จุดศูนย์กลางกลุ่มข้อมูลที่ 3 คือ A7 (โจทย์กำหนดมาให้ แสดงด้วยสัญลักษณ์รูปดาว)

กลุ่มข้อมูลที่ได้ คือ {A1}; {A3, A4, A5, A6, A8} และ {A2, A7} แสดงตามรูปที่ 2.7 เนื่องจาก ไม่มีแบบใดอยู่ใกล้ A1 มากกว่า A4 และ A7 เลย ดังนั้นกลุ่มข้อมูลนี้จึงมี A1 เพียงแบบเดียว A3, A5, A6, A8 อยู่ใกล้ A4 มากกว่า A1, A7 ดังนั้น A3, A4, A5, A6, A8 จึงอยู่ในกลุ่มข้อมูลเดียวกัน A2 อยู่ใกล้ A7 มากกว่า A1, A4 ดังนั้น A2, A7 จึงอยู่ในกลุ่มข้อมูลเดียวกัน



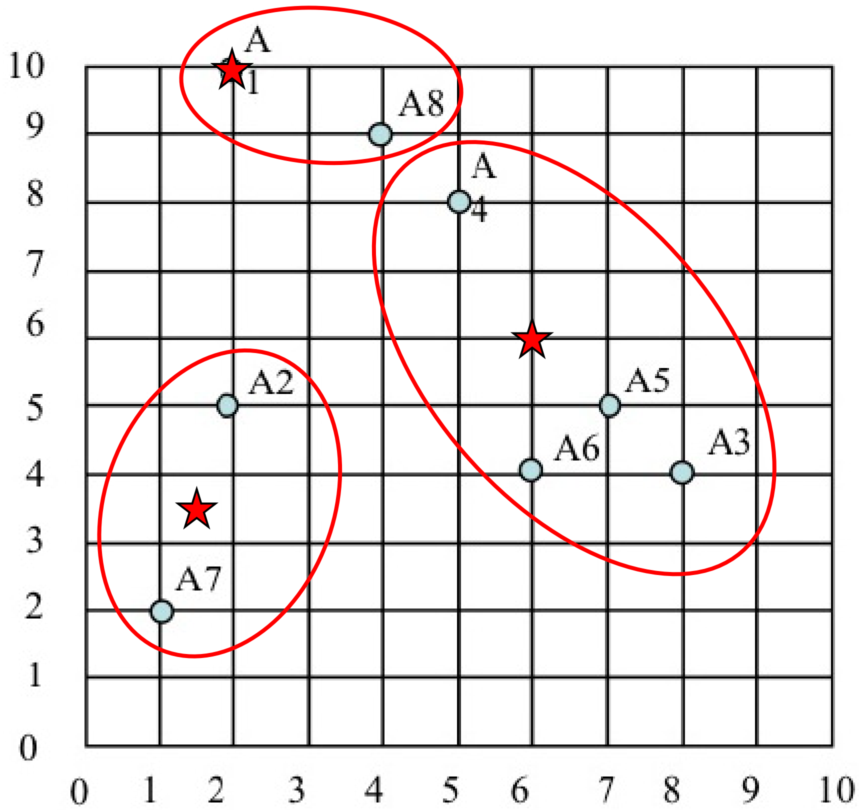
รูปที่ 2.7 ตัวแทนกลุ่มข้อมูลของแบบ A1 – A8 ประมวลผลรอบที่ 1

เนื่องจากเป็นการประมวลผลในรอบแรก จึงยังไม่ใช่เงื่อนไขหยุดการทำงาน ดังนั้นจะมีการประมวลผลใน รอบต่อไป

รอบที่ 2

คำนวณจุดศูนย์กลางของกลุ่มข้อมูลใหม่ โดยใช้ค่าเฉลี่ยของแบบทั้งหมดในกลุ่มข้อมูล ดังต่อไปนี้
จุดศูนย์กลางกลุ่มข้อมูลที่ 1 (C1) คือ A1 = (2, 10) เนื่องจากมีเพียงแบบเดียวในกลุ่มข้อมูลนี้
จุดศูนย์กลางกลุ่มข้อมูลที่ 2 (C2) คือ ((8 + 5 + 7 + 6 + 4) / 5, (4 + 8 + 5 + 4 + 9) / 5) = (6, 6)
จุดศูนย์กลางกลุ่มข้อมูลที่ 3 (C3) คือ ((2 + 1) / 2, (5 + 2) / 2)) = (1.5, 3.5)
ค่าเฉลี่ย C1, C2, C3 แสดงด้วยสัญลักษณ์รูปดาวในรูปที่ 2.8

กลุ่มข้อมูลใหม่ที่ได้ คือ {A1, A8}; {A3, A4, A5, A6} และ {A2, A7} แสดงตามรูปด้านล่างนี้ เนื่องจาก
A1, A8 อยู่ใกล้ C1 มากกว่า C2, C3 ดังนั้น A1, A8 จึงอยู่ในกลุ่มข้อมูลเดียวกัน
A3, A4, A5, A6 อยู่ใกล้ C2 มากกว่า C1, C2 ดังนั้น A3, A4, A5, A6 จึงอยู่ในกลุ่มข้อมูลเดียวกัน
A2, A7 อยู่ใกล้ C3 มากกว่า C1, C2 ดังนั้น A2, A7 จึงอยู่ในกลุ่มข้อมูลเดียวกัน



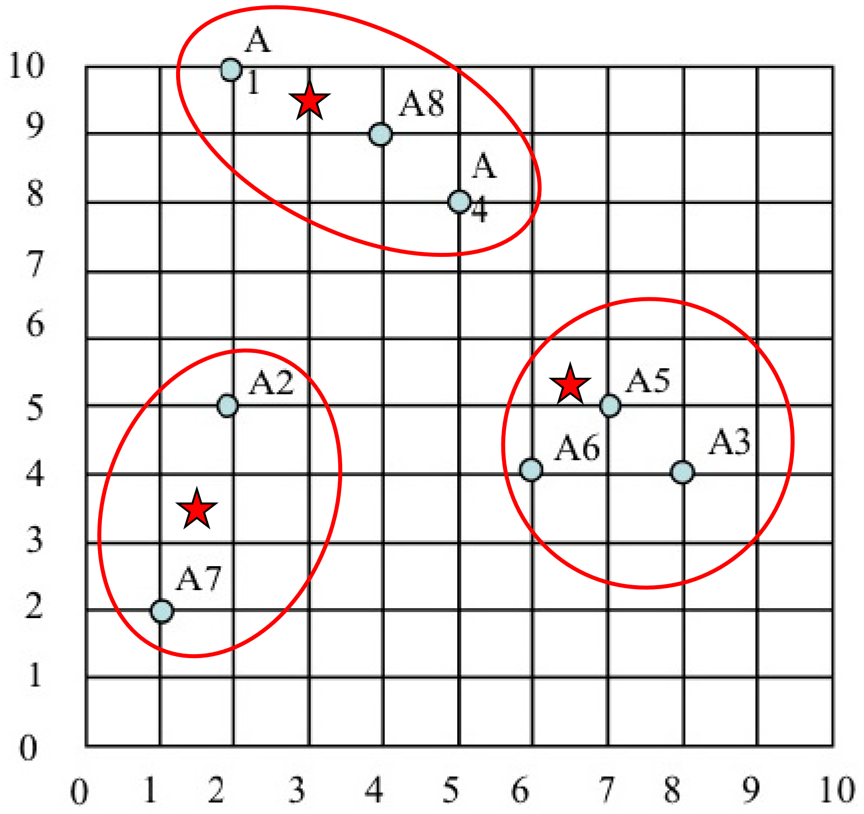
รูปที่ 2.8 ตัวแทนกลุ่มข้อมูลของแบบ A1 – A8 ประมวลผลรอบที่ 2

เนื่องจากมีแบบบางตัวถูกเปลี่ยนกลุ่มข้อมูล ได้แก่ แบบ A8 ในรอบที่ 1 อยู่กลุ่มข้อมูลที่ 2 แต่ในรอบที่ 2 เปลี่ยนมาอยู่ในกลุ่มข้อมูลที่ 1 จึงยังไม่ใช่เงื่อนไขหยุดการทำงาน ดังนั้นจะมีการประมวลผลในรอบต่อไป

รอบที่ 3

คำนวณจุดศูนย์กลางของกลุ่มข้อมูลใหม่ โดยใช้ค่าเฉลี่ยของแบบทั้งหมดในกลุ่มข้อมูล ดังต่อไปนี้
จุดศูนย์กลางกลุ่มข้อมูลที่ 1 (C1) คือ ((2 + 4) / 2, (10 + 9) / 2) = (3, 9.5)
จุดศูนย์กลางกลุ่มข้อมูลที่ 2 (C2) คือ ((8 + 5 + 7 + 6) / 4, (4 + 8 + 5 + 4) / 4) = (6.5, 5.25)
จุดศูนย์กลางกลุ่มข้อมูลที่ 3 (C3) คือ ((2 + 1) / 2, (5 + 2) / 2) = (1.5, 3.5)
ค่าเฉลี่ย C1, C2, C3 แสดงด้วยสัญลักษณ์รูปดาวในรูปที่ 2.9

กลุ่มข้อมูลใหม่ที่ได้ คือ {A1, A4, A8}; {A3, A5, A6} และ {A2, A7} แสดงตามรูปด้านล่างนี้ เนื่องจาก
A1, A4, A8 อยู่ใกล้ C1 มากกว่า C2, C3 ดังนั้น A1, A4, A8 จึงอยู่ในกลุ่มข้อมูลเดียวกัน
A3, A5, A6 อยู่ใกล้ C2 มากกว่า C1, C2 ดังนั้น A3, A5, A6 จึงอยู่ในกลุ่มข้อมูลเดียวกัน
A2, A7 อยู่ใกล้ C3 มากกว่า C1, C2 ดังนั้น A2, A7 จึงอยู่ในกลุ่มข้อมูลเดียวกัน



รูปที่ 2.9 ตัวแทนกลุ่มข้อมูลของแบบ A1 – A8 ประมวลผลรอบที่ 3

เนื่องจากมีแบบบางตัวถูกเปลี่ยนกลุ่มข้อมูล ได้แก่ แบบ A4 จึงยังไม่ใช่เงื่อนไขหยุดการทำงาน ดังนั้นจะมีการประมวลผลในรอบต่อไป

รอบที่ 4

คำนวณจุดศูนย์กลางของกลุ่มข้อมูลใหม่ โดยใช้ค่าเฉลี่ยของแบบทั้งหมดในกลุ่มข้อมูล ดังต่อไปนี้

จุดศูนย์กลางกลุ่มข้อมูลที่ 1 (C1) คือ ((2 + 5 + 4) / 3, (10 + 8 + 9) / 3) = (3.66, 9)

จุดศูนย์กลางกลุ่มข้อมูลที่ 2 (C2) คือ ((8 + 7 + 6) / 3, (4 + 5 + 4) / 3) = (7, 4.33)

จุดศูนย์กลางกลุ่มข้อมูลที่ 3 (C3) คือ ((2 + 1) / 2, (5 + 2) / 2) = (1.5, 3.5)

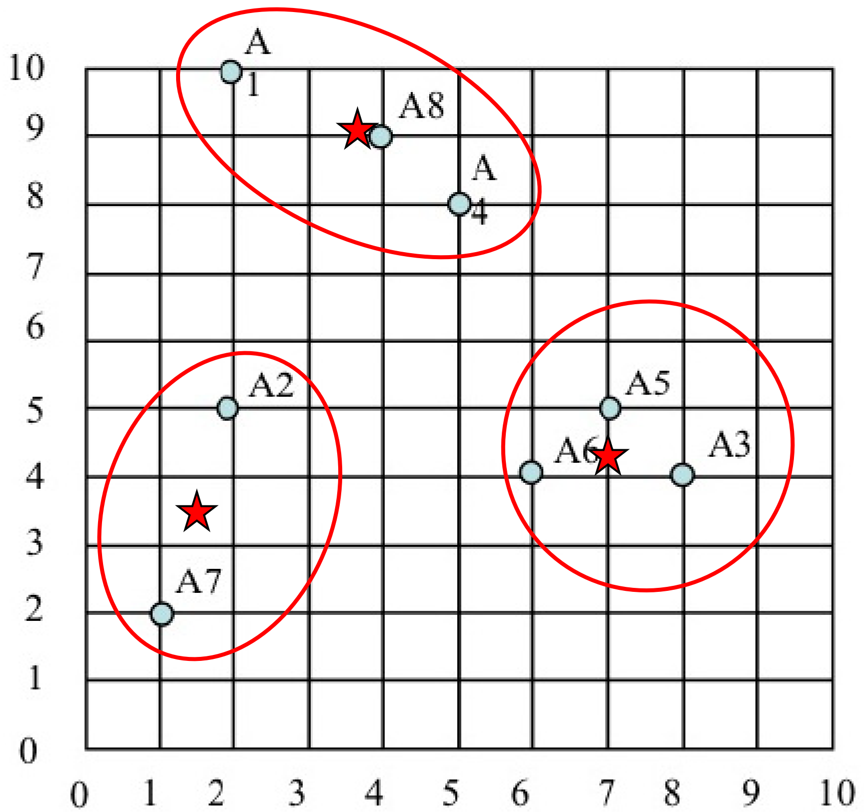
ค่าเฉลี่ย C1, C2, C3 แสดงด้วยสัญลักษณ์รูปดาวในรูปที่ 2.10

กลุ่มข้อมูลใหม่ที่ได้ คือ {A1, A4, A8}; {A3, A5, A6} และ {A2, A7} แสดงตามรูปด้านล่างนี้ เนื่องจาก

A1, A4, A8 อยู่ใกล้ C1 มากกว่า C2, C3 ดังนั้น A1, A4, A8 จึงอยู่ในกลุ่มข้อมูลเดียวกัน

A3, A5, A6 อยู่ใกล้ C2 มากกว่า C1, C2 ดังนั้น A3, A5, A6 จึงอยู่ในกลุ่มข้อมูลเดียวกัน

A2, A7 อยู่ใกล้ C3 มากกว่า C1, C2 ดังนั้น A2, A7 จึงอยู่ในกลุ่มข้อมูลเดียวกัน



รูปที่ 2.10 ตัวแทนกลุ่มข้อมูลของแบบ A1 – A8 ประมวลผลรอบที่ 4

เนื่องจากแบบทุกตัวไม่มีการเปลี่ยนกลุ่มข้อมูลแล้ว จึงเป็นเงื่อนไขหยุดการทำงานและจะไม่มีการประมวลผลในรอบต่อไปอีก

โดยทั่วไปแล้ว การจัดกลุ่มข้อมูลเป็นกระบวนการที่ถูกใช้ในช่วงการเตรียมก่อนการประมวลผล เนื่องจากในช่วงเริ่มต้นจากการเก็บรวมรวมข้อมูลดิบเพื่อสร้างเป็นเซตข้อมูลนั้น แบบทั้งหมดอาจยังไม่มีการกำหนดชั้นข้อมูลมาก่อน จึงจำเป็นต้องใช้การจัดกลุ่มข้อมูลเพื่อแบ่งแบบที่มีความคล้ายกันเข้าไว้ในกลุ่มข้อมูลเดียวกัน แล้วทำการระบุชั้นข้อมูลให้แบบในแต่ละกลุ่มข้อมูล กล่าวคือแบบในกลุ่มข้อมูลเดียวกันจะมีชั้นข้อมูลเหมือนกัน หลังจากนี้ก็จะเข้าสู่ขั้นตอนการสร้างตัวแบบในการทำนายชั้นข้อมูลดังที่จะกล่าวถึงในบทต่อไปหลังจากนี้

5. การกำหนดสาระสำคัญของเซตข้อมูล

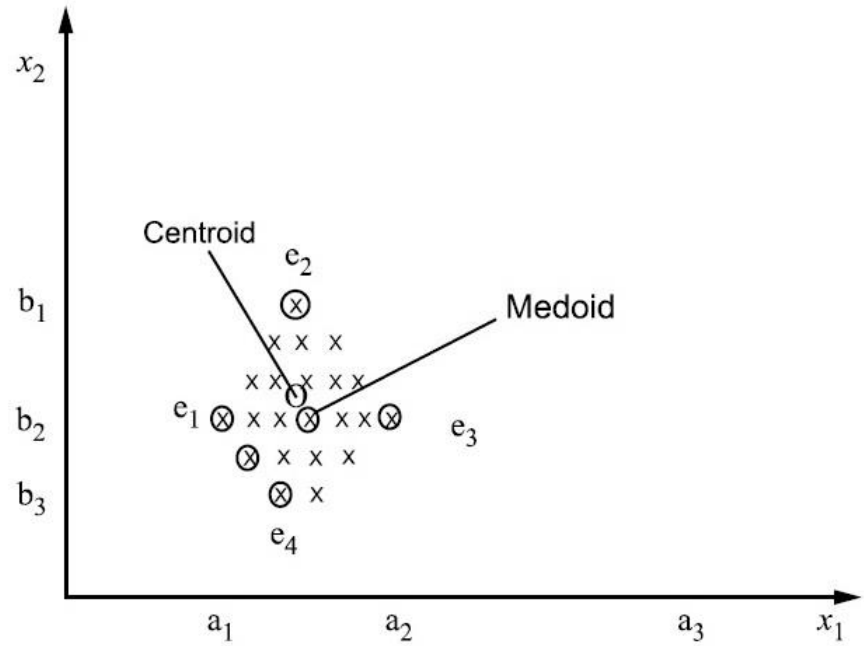
ในกรณีที่เซตข้อมูลมีแบบจำนวนมหาศาล การนำแบบทั้งหมดมาใช้ในการประมวลผลอาจไม่เหมาะสมนักเนื่องจากต้องใช้เวลาประมวลผลยาวนาน ดังนั้นเพื่อต้องการลดเวลาประมวลผลดังกล่าว จึงอาจเลือกเพียงบางแบบหรือกลุ่มของแบบจากแต่ละกลุ่มข้อมูลมาเป็นตัวแทนในการประมวลผลแทน วิธีการนี้เรียกว่า การกำหนดสาระสำคัญของเซตข้อมูล ซึ่งแบ่งออกได้เป็น 2 ประเภทได้แก่ ตัวแทนเดี่ยว (Single Representative) และ ตัวแทนหมู่ (Multiple Representative) ในแต่ละกลุ่มข้อมูล โดยมีรายละเอียด ดังต่อไปนี้

- **ตัวแทนเดี่ยว** คือการใช้แบบเพียงแบบเดียวเท่านั้นต่อกลุ่มข้อมูลเป็น จุดศูนย์กลางกลุ่มข้อมูล (Cluster Centre) โดยทั่วไปอาจใช้ เซนทรอยด์ (Centroid) หรือ มีดอยส์ (Medoid) มีรายละเอียดดังต่อไปนี้
  - **เซนทรอยด์** คือ ค่าเฉลี่ยตัวอย่าง (Sample Mean) วิธีการหาเซนทรอยด์เป็นการนำเอาค่าในแต่ละฟีเจอร์ของแบบทั้งหมดมาคำนวณเพื่อหาค่าเฉลี่ยออกมา เนื่องจากเซนทรอยด์คำนวณจากค่าเฉลี่ยดังนั้นจึงอาจเป็นแบบที่ไม่มีอยู่จริงในเซตข้อมูลก็ได้
  - **มีดอยส์** คือ แบบที่อยู่ตำแหน่งกึ่งกลางมากที่สุดในกลุ่มข้อมูล วิธีการหามีดอยส์เริ่มต้นจากการหาค่า มัธยฐาน (Median) ของแต่ละฟีเจอร์ออกมา จากนั้นเปรียบเทียบว่าแบบใดในกลุ่มข้อมูลอยู่ใกล้กลุ่มของค่ากลางนี้มากที่สุดจะถูกนำมาใช้เป็นมีดอยส์ เนื่องจากมีดอยส์คือการเลือกแบบใดแบบหนึ่งในกลุ่มข้อมูลเพื่อนำมาเป็นตัวแทนเดี่ยว ดังนั้นจึงเป็นแบบที่มีอยู่จริงในเซตข้อมูลแน่นอน
- **ตัวแทนหมู่** คือการใช้กลุ่มของแบบเป็น ตัวแทนกลุ่มข้อมูล (Cluster Representative) เนื่องจากในกลุ่มข้อมูลมีขนาดใหญ่มาก มีจำนวนแบบมากมาย การใช้แบบเพียงแบบเดียวอาจไม่เหมาะสมที่จะเป็นตัวแทนของแบบทุกแบบในกลุ่มข้อมูล

ตัวอย่างที่ 10 รูปที่ 2.11 แสดงแบบทั้งหมดในกลุ่มข้อมูลหนึ่งในเซตข้อมูล ประกอบไปด้วยค่าทั้งหมด 2 ฟีเจอร์ แสดงตามแกน x_1 และ x_2 นอกจากนี้ ค่า a_1, a_2, a_3 คือค่าฟีเจอร์บนแกน x_1 ในขณะที่ b_1, b_2, b_3 คือค่าฟีเจอร์บนแกน x_2

สำหรับกลุ่มข้อมูลนี้ ถ้าใช้ตัวแทนเดี่ยวอาจใช้ เซนทรอยด์ หรือ มีดอยส์ เป็นตัวแทนก็ได้ จากรูปจะเห็นได้ว่า เซนทรอยด์เป็นแบบที่ไม่มีอยู่จริง แต่มีดอยส์เป็นแบบที่มีอยู่จริง การใช้ตัวแทนเดี่ยวในตัวอย่างนี้เพื่อเจาะจงไปที่จุดศูนย์กลางของกลุ่มข้อมูล

ในทางกลับกัน กลุ่มข้อมูลนี้อาจใช้ตัวแทนหมู่ก็ได้ จากรูปมีการใช้แบบ 4 จำนวนแบบที่อยู่บริเวณขอบของกลุ่มข้อมูลเป็นตัวแทน ได้แก่แบบ e_1, e_2, e_3 และ e_4 การใช้ตัวแทนหมู่ในตัวอย่างนี้เพื่อให้ครอบคลุมทุกอาณาเขตในกลุ่มข้อมูล



รูปที่ 2.11 ตัวแทนกลุ่มข้อมูล

6. การประเมินค่าตัวจำแนกแบบ

โดยทั่วไปเมื่อรวบรวมเซตข้อมูลมาได้แล้ว จะทำการแบ่งแบบในเซตข้อมูลออกเป็น 2 ส่วน ได้แก่ เซตฝึกฝน (Training Set) และ เซตทดสอบ (Test Set) โดยทั่วไป อัตราส่วนระหว่างเซตฝึกฝนและเซตทดสอบคือ 3 : 2 กล่าวคือ เซตข้อมูลจะถูกแบ่งออกเป็น 5 ส่วนเท่า ๆ กัน แล้วนำ 3 ส่วนมาใช้เป็นเซตฝึกฝน ในขณะที่เซตทดสอบใช้อีก 2 ส่วนที่เหลือ หรืออีกวิธีหนึ่งที่มีการทำการทดลองกัน

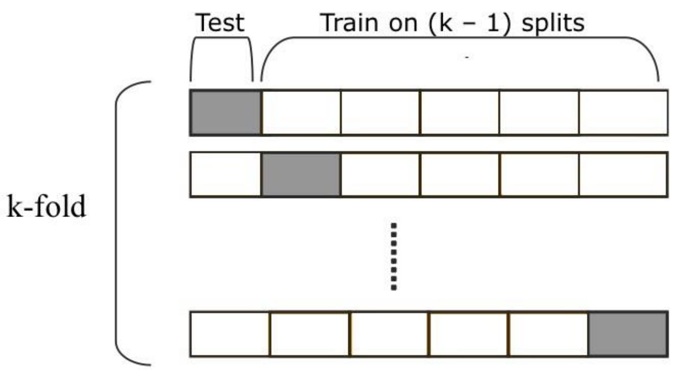
อาจจะแบ่งเซตฝึกฝนและเซตทดสอบออกเป็น 2 ส่วนเท่า ๆ กันในอัตราส่วน 1 : 1 ก็ได้ เซตทั้งสองชนิดมีลักษณะ ดังต่อไปนี้

- **เซตฝึกฝน** คือกลุ่มของแบบที่ใช้สำหรับให้ตัวจำแนกแบบเรียนรู้ฟีเจอร์ของแต่ละชั้นข้อมูล แบบในเซตฝึกฝนจะต้องมีชั้นข้อมูลระบุอยู่อย่างชัดเจน เพื่อให้ตัวจำแนกแบบสามารถแยกความแตกต่างของแบบจากชั้นข้อมูลต่าง ๆ ได้ แบบในเซตประเภทนี้เรียกว่าแบบฝึกฝน
- **เซตทดสอบ** คือกลุ่มของแบบที่ใช้สำหรับวัดประสิทธิภาพของตัวจำแนกแบบ โดยทั่วไปจะซ่อนชั้นข้อมูลซึ่งเปรียบเสมือนคำตอบไว้ไม่ให้ตัวจำแนกแบบรู้ เพื่อให้ตัวแบบทดลองทำนาย จำนวนแบบที่ทายถูกจะถูกนำมาคำนวณหาค่าความแม่นยำของตัวจำแนกแบบ แบบในเซตประเภทนี้เรียกว่าแบบทดสอบ

การวัดค่าความแม่นของตัวจำแนกแบบ สามารถทำการทดลองได้หลายวิธี เช่น การแทนที่ซ้ำ (Re-substitute Estimate) วิธีแยกข้อมูล (Holdout Method) และ การพับเคครั้ง (K-fold Cross-validation) เป็นต้น แต่ละวิธีมีรายละเอียด ดังต่อไปนี้

- **การแทนที่ซ้ำ** คือ การวัดค่าความแม่นจากเซตฝึกฝนแต่เพียงอย่างเดียว โดยไม่จำเป็นต้องใช้เซตทดสอบเลย อธิบายอย่างง่ายคือเซตฝึกฝนและเซตทดสอบคือเซตเดียวกัน การที่จะทดลองในลักษณะนี้ได้นั้น ต้องมีสมมติฐานว่าแบบฝึกฝนเป็นตัวแทนที่ดีของแบบทั้งหมด อย่างไรก็ตาม วิธีนี้มีข้อเสียคือตัวจำแนกแบบทำการเรียนรู้เซตฝึกฝน จึงเห็นชั้นข้อมูลของแบบแต่ละแบบ เมื่อนำไปทดลองทำนายในเซตฝึกฝนซึ่งเป็นเซตเดิมที่เคยรู้จำมาก่อนหน้านี้แล้ว ความแม่นยำที่ได้จึงอาจสูงเกินจริง เปรียบเทียบได้กับการนำการบ้านไปเป็นข้อสอบสำหรับนักเรียน เด็กจึงอาจจำคำตอบมาตอบโดยไม่รู้วิธีทำก็เป็นได้ เมื่อเปรียบเทียบเรื่องข้อสอบกับกระบวนการทางด้านการรู้จำแบบแล้ว ตัวจำแนกแบบอาจรู้จำค่าฟีเจอร์ในลักษณะเจาะจงเพียงไม่กี่ค่าเพื่อทำนายชั้นข้อมูล แต่สำหรับ โปรแกรมประยุกต์ในโลกความเป็นจริง (Real-world Application) ค่าฟีเจอร์ของแบบฝึกฝนและแบบทดสอบมีความแตกต่างกัน ไม่ได้มีค่าเหมือนกันทุกประการ ดังนั้นตัวจำแนกแบบจึงไม่ควรจำค่าฟีเจอร์แบบกระชับเกินไป (รู้จำค่าเดี่ยว) แต่ควรจำแนกแบบหลวม (รู้จำช่วงของค่า) ปัญหานี้เรียกว่า ปัญหาการแน่นเกินไป (Over-fitting Problem) ซึ่งมักจะเกิดขึ้นเมื่อเซตทดสอบและเซตฝึกฝนเป็นเซตเดียวกัน

- **วิธีแยกข้อมูล** คือการแบ่งเซตข้อมูลออกเป็น 3 ส่วน เซตฝึกฝนประกอบด้วย 2 ส่วน และ เซตทดสอบอีก 1 ส่วน หรืออาจแบ่งเป็นสัดส่วนอื่น เช่น แบบทดสอบและแบบฝึกฝนอย่างละครึ่ง จุดประสงค์ของการแยกเซตทั้งสองให้เป็นอิสระต่อกัน เพื่อแก้ปัญหาการแน่นเกินไป เปรียบเทียบได้กับการออกข้อสอบให้มีความแตกต่างจากแบบฝึกหัด ยกตัวอย่างเช่นในวิชาคณิตศาสตร์ ตัวเลขในแบบฝึกหัดและข้อสอบอาจไม่เหมือนกันซักทีเดียว แต่ใช้วิธีทำเดียวกันในการแก้ปัญหา นักเรียนที่ทำแบบฝึกหัดมากพอจะทำข้อสอบได้ โดยไม่จำเป็นต้องจำเฉลยมาตอบเหมือนการแทนที่ซ้ำ เมื่อเปรียบเทียบเรื่องข้อสอบกับกระบวนการทางด้านการรู้จำแบบแล้ว ตัวจำแนกแบบจำเป็นต้องรู้จำค่าเป็นช่วงในแต่ละชั้นข้อมูล ทดแทนการรู้จำเพียงค่าเดี่ยว เนื่องจากเซตฝึกฝนและเซตทดสอบมีความแตกต่างกัน
- **การพับเคครั้ง** คือการแบ่งเซตข้อมูลออกเป็น k ส่วนที่เท่ากัน (k-fold) นำข้อมูล 1 ส่วนเป็นเซตทดสอบ นำส่วนที่เหลือ (k – 1) รวมกันเป็นเซตฝึกฝน ทำการทดลองทั้งสิ้นจำนวน k ครั้ง ข้อมูลในแต่ส่วนจะถูกใช้เป็นเซตทดสอบหนึ่งครั้ง วิธีการทดลองชนิดนี้มีลักษณะคล้ายวิธีแยกข้อมูล ความแตกต่างจากวิธีดังกล่าวคือ วิธีแยกข้อมูลมีเซตทดสอบและเซตฝึกฝนอย่างละหนึ่งเซต แต่การพับเคครั้งมีเซตทดสอบและเซตฝึกฝนอย่างละเคเซต รูปที่ 2.12 แสดงวิธีการแบ่งเซตต้นฉบับออกเป็นเซตฝึกฝนและเซตทดสอบสำหรับการทดลองแบบการพับเคครั้ง



รูปที่ 2.12 การพับเคครั้ง (**ที่มา:** https://www.appnationconference.com/article/generals)

ตัวแปรที่ใช้ประเมินค่าประสิทธิภาพของตัวจำแนกแบบ ได้แก่ ความแม่น (Accuracy) เวลาในการออกแบบ (Design Time) เวลาในการจำแนกแบบ (Classification Time) พื้นที่ (Space) ความสามารถในการอธิบายได้ (Explanation Ability) และ ความทนต่อสัญญาณรบกวน (Noise Tolerance) ปัจจัยแต่ละตัวมีความสำคัญ ดังต่อไปนี้

- **ความแม่น** คือตัวแปรหลักที่ใช้ประเมินค่าประสิทธิภาพในการทำนายของตัวจำแนกแบบ ซึ่งสามารถคำนวณได้จาก อัตราส่วนระหว่างจำนวนครั้งที่ทายแบบทดสอบได้อย่างถูกต้องและจำนวนครั้งในการทำนายแบบทดสอบทั้งหมด ปัจจัยนี้แสดงให้เห็นว่าเมื่อถึงเวลานำตัวจำแนกแบบไปใช้งานจริงแล้ว จะสามารถทำนายแบบได้ถูกต้องประมาณร้อยละเท่าใด
- **เวลาออกแบบ** คือเวลาที่ใช้ในการสร้างตัวจำแนกแบบจากการเรียนรู้เซตฝึกฝน จัดเป็นต้นทุนที่เกิดขึ้นครั้งเดียว (One-Time Cost) เพราะว่าหลังจากที่ตัวจำแนกแบบถูกสร้างขึ้นแล้ว สามารถนำตัวแบบไปทำนายแบบได้หลายครั้ง โดยไม่จำเป็นต้องสร้างตัวแบบทุกครั้งก่อนการทำนายแบบแต่ละตัว ตัวจำแนกแบบอาจถูกสร้างขึ้นใหม่ได้ถ้าต้องการปรับปรุงประสิทธิภาพการทำนายให้ดีขึ้นกว่าเดิม
- **เวลาในการจำแนกแบบ** คือเวลาที่ตัวจำแนกแบบใช้ในการทำนายชั้นข้อมูลของแบบที่ไม่รู้จัก จัดเป็น ต้นทุนที่เกิดขึ้นซ้ำ (Recurring Cost) เพราะว่าโดยส่วนใหญ่แล้วแบบที่ต้องการทำนายมีจำนวนมากมาย ถ้าต้นทุนนี้มีค่าสูงเกินไปอาจไม่สามารถนำไปใช้งานจริงได้ ดังนั้นการแก้ปัญหาอาจมีความจำเป็นต้องเปลี่ยนไปใช้ตัวจำแนกแบบชนิดอื่นที่ทำนายได้เร็วขึ้น แม้ว่าความแม่นจะลดลงก็ตาม
- **พื้นที่** คือขนาดของหน่วยความจำที่จำเป็นต้องใช้สำหรับบรรจุตัวจำแนกแบบ โดยทั่วไปปัจจัยนี้อาจไม่ต้องคำนึงถึงมากนักถ้านำไปใช้งานในอุปกรณ์ที่มีพื้นที่จัดเก็บข้อมูลสูงเช่น เครื่องคอมพิวเตอร์ส่วนบุคคล อย่างไรก็ตามในบางอุปกรณ์ที่มีพื้นที่เก็บข้อมูลอย่างจำกัดเช่น โทรศัพท์มือถือ พื้นที่การจัดเก็บข้อมูลเป็นปัจจัยที่ต้องถูกคำนึงถึง ดังนั้นการแก้ปัญหาอาจมีการกำหนดสาระสำคัญให้กับเซตฝึกฝน ซึ่งจะทำให้ลดพื้นที่ในการจัดเก็บข้อมูล ได้มากกว่าการเก็บทั้งเซตข้อมูล
- **ความสามารถในการอธิบายได้** คือการอธิบายผู้ใช้ถึงเหตุผลในการเลือกชั้นข้อมูลของแบบ โดยส่วนใหญ่ต้องอธิบายโดยใช้ภาษมนุษย์ได้ เพราะง่ายต่อการเข้าใจเช่น การใช้เงื่อนไข “ถ้า แล้ว” ตัวจำแนกบางประเภทมีคุณสมบัตินี้เช่น ต้นไม้การตัดสินใจ แต่ตัวจำแนกบางประเภทก็ขาดคุณสมบัตินี้เช่น เครื่องกลเวกเตอร์สนับสนุน ประโยชน์ของคุณสมบัตินี้พบมากในทางการแพทย์ ตัวอย่างเช่น ซอฟต์แวร์สามารถใช้เงื่อนไข “ถ้าแล้ว” ในการอธิบายคนไข้ได้เช่นถ้าคนไข้ทานอาหารที่มีน้ำตาลสูง แล้วเขามีโอกาสเป็นโรคเบาหวาน เป็นต้น นอกจากนี้ แพทย์สามารถตรวจสอบได้ว่า การทำนายของตัวจำแนกแบบมีความสมเหตุสมผลหรือไม่ผ่านเงื่อนไข “ถ้าแล้ว”

- **ความทนต่อสัญญาณรบกวน** คือความสามารถของตัวจำแนกแบบในการจัดการกับสัญญาณรบกวนได้ดี โดยที่สัญญาณรบกวนคือแบบที่มีความผิดปกติและแตกต่างอย่างเด่นชัดจากแบบส่วนใหญ่ ในเซตข้อมูลที่มีสัญญาณรบกวนมาก ตัวจำแนกแบบอาจทำนายชั้นข้อมูลที่มีสัญญาณรบกวนผิดพลาด เนื่องจากแบบมีฟีเจอร์ที่แตกต่างจากแบบโดยทั่วไป ตัวจำแนกแบบบางประเภทสามารถจัดการกับสัญญาณรบกวนได้ดีเช่น ตัวจำแนกเบส์ แต่ตัวจำแนกแบบบางประเภทไม่สามารถจัดการได้เช่น ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุด อย่างไรก็ตามสัญญาณรบกวนอาจถูกกำจัดออกจากเซตข้อมูลได้ก่อนการสร้างตัวแบบ เพื่อรับประกันได้ว่าจะไม่ทำให้ตัวจำแนกแบบมีประสิทธิภาพที่แย่ลง

7. การสกัดฟีเจอร์

การสกัดฟีเจอร์ (Feature Extraction) เป็นขั้นตอน ก่อนการประมวลผล (Pre-processing) ที่ถูกใช้ในช่วงเตรียมข้อมูล สำหรับ ลดขนาดมิติ (Dimensionality Reduction) ของข้อมูล โดยการคัดแยกฟีเจอร์ที่สำคัญออกมา หรือรวมข้อมูลต่าง ๆ เข้าเป็นฟีเจอร์เดียวกัน เซตข้อมูลที่ผ่านการสกัดฟีเจอร์จะมีขนาดเล็กลง ส่งผลให้การประมวลผลเร็วขึ้น

7.1 การจำแนกเชิงเส้นของฟิชเชอร์

การจำแนกเชิงเส้นของฟิชเชอร์ (Fisher’s Linear Discriminant) คือ การฉายเงา (Projection) ของแบบในปริภูมิหลายมิติลงบนเส้นตรง หลังจากนั้น กระบวนการจำแนกแบบจะเกิดขึ้นในปริภูมิ 2 มิติแทน ในกรณีที่เซตข้อมูลมี 2 ชั้นข้อมูล การฉายเงา จะทำการหาระยะห่างที่มากที่สุดระหว่าง ค่าเฉลี่ย (Mean) และ ระยะห่างที่น้อยที่สุดระหว่าง ความแปรปรวน (Variance) ของสองชั้นข้อมูล โดยการหาค่ามากที่สุดจาก เกณฑ์ของฟิชเชอร์ (Fisher’s Criterion) ซึ่งมีนิยาม ดังนี้

$$J(V) = \frac{|mean_1 - mean_2|^2}{s_1^2 + s_2^2}$$

- J(V) คือ เกณฑ์ของฟิชเชอร์
- $mean_1$, $mean_2$ คือ ค่าเฉลี่ยของชั้นข้อมูลที่ 1 และ 2 ตามลำดับ
- s_1, s_2 คือ ความแปรปรวนของชั้นข้อมูลที่ 1 และ 2 ตามลำดับ

การหาค่ามากที่สุดจากเกณฑ์ของฟิชเชอร์สามารถหาได้จาก เมทริกซ์ความแปรปรวนร่วมผกผัน (Inverse Covariance Matrix) ซึ่งการคำนวณต่าง ๆ กำลังจะกล่าวถึงตามขั้นตอน ดังต่อไปนี้

กำหนดให้ x_i คือ สมาชิกของเวกเตอร์ N หลัก ใน 1 มิติ ดังนั้น ค่าเฉลี่ยของเซตข้อมูล คือ

$$mean = \frac{1}{N}\sum_{i=1}^{N} x_i$$

ในกรณีของข้อมูลใน D มิติ ค่าเฉลี่ยจะอยู่ในรูปเวกเตอร์ขนาด D หน่วย เมื่อ D คือจำนวนมิติ

ถ้ามี K ชั้นข้อมูล (C_1, C_2, …, C_k) ค่าเฉลี่ยในชั้นข้อมูล C_k ที่บรรจุสมาชิกจำนวน N_k ตัว คือ

$$mean = \frac{1}{N_k}\sum_{x_i \in c_k} x_i$$

เมทริกซ์การกระจายระหว่างชั้นข้อมูล (Between-class Scatter Matrix) คือ

$$\sigma_B = \sum_{k=1}^{K} N_k(mean_k - mean)(mean_k - mean)^T$$

เมทริกซ์การกระจายภายในชั้นข้อมูลข้อมูล (Within-class Scatter Matrix) คือ

$$\sigma_W = \sum_{k=1}^{K}\sum_{x_i \in c_k} N_k(x_i - mean_k)(x_i - mean_k)^T$$

เมทริกซ์การแปลง (Transformation Matrix) ถูกใช้ในการย้ายตำแหน่งแบบในชั้นข้อมูลให้มีการแบ่งแยกที่ชัดเจนยิ่งขึ้น มีรูปแบบตามสมการ ด้านล่างนี้

$$J(V) = \frac{V^T\sigma_B V}{V^T\sigma_w V}$$

J(V) คือฟังก์ชันเกณฑ์ โดยที่การจำแนกเชิงเส้นของฟิชเชอร์ มีจุดมุ่งหมายในการหาค่าสูงสุดของฟังก์ชันนี้ผ่านเวกเตอร์ V ตามสมการด้านล่างนี้

$$\sigma_B V = \lambda\sigma_w V$$

σ_B และ σ_W คือ เวกเตอร์ลักษณะเฉพาะ (Eigenvector) $\{v_1, v_2, \ldots, v_D\}$ และ สเกลาร์ λ คือ ค่าลักษณะเฉพาะ (Eigenvalue)

สมการข้างต้นนี้ ทำการฉายเงาของแบบในปริภูมิ D มิติ ลงสู่ปริภูมิย่อย d มิติ ($d < D$) โดยที่ $V_d = \{v_1, v_2, \ldots, v_d\}$ เมื่อ d มีค่าสูงสุด ในส่วนเพิ่มเติม การฉายเงาของเวกเตอร์ x_i ลงบน ปริภูมิย่อย d มิติ อธิบายได้ตามสมการด้านล่างนี้

$$y = V_d^T x$$

ในกรณีของปัญหาสองชั้นข้อมูล ค่าต่าง ๆ ที่เกี่ยวข้อง คำนวณได้ ดังต่อไปนี้

$$mean_1 = \frac{1}{N_1} \sum_{x_i \in c_1} x_i$$

$$mean_2 = \frac{1}{N_2} \sum_{x_i \in c_2} x_i$$

$$\sigma_B = N_1(mean_1 - mean)(mean_1 - mean)^T + N_2(mean_2 - mean)(mean_2 - mean)^T$$

$$\sigma_W = \sum_{k=1}^{K} \sum_{x_i \in c_k} N_k(x_i - mean_k)(x_i - mean_k)^T$$

$$\sigma_B V = \lambda \sigma_w V$$

หมายความว่า

$$\sigma_w^{-1} \sigma_B V = \lambda V$$

เนื่องจาก $\sigma_B V$ อยู่ในทิศทางของ $mean_1 - mean_2$ ดังนั้น

$$V = \sigma_w^{-1}(mean_1 - mean_2)$$

ตัวอย่างที่ 11 ถ้ามีแบบจำนวน 6 แบบ ประกอบไปด้วย (2, 2) ; (4, 3) และ (5, 1) อยู่ในชั้นข้อมูลที่ 1 ในขณะที่ แบบ (1, 3) ; (5, 5) และ (3, 6) อยู่ในชั้นข้อมูลที่ 2

ค่าเฉลี่ยของชั้นข้อมูลที่ 1 คือ

$$mean_1 = \begin{bmatrix} \frac{2+4+5}{3} \\ \frac{1+5+3}{3} \end{bmatrix} = \begin{bmatrix} 3.66 \\ 2 \end{bmatrix}$$

ค่าเฉลี่ยของชั้นข้อมูลที่ 2 คือ

$$mean_2 = \begin{bmatrix} \frac{1+5+3}{3} \\ \frac{3+5+6}{3} \end{bmatrix} = \begin{bmatrix} 3 \\ 4.66 \end{bmatrix}$$

เมทริกซ์การกระจายภายในชั้นข้อมูลข้อมูล คือ

$$\sigma_W = \begin{bmatrix} 2-3.66 \\ 2-2 \end{bmatrix} \times [2-3.66 \quad 2-2] +$$

$$\begin{bmatrix} 4-3.66 \\ 3-2 \end{bmatrix} \times [4-3.66 \quad 3-2] +$$

$$\begin{bmatrix} 5-3.66 \\ 1-2 \end{bmatrix} \times [5-3.66 \quad 1-2] +$$

$$\begin{bmatrix} 1-3 \\ 3-4.66 \end{bmatrix} \times [1-3 \quad 3-4.66] +$$

$$\begin{bmatrix} 5-3 \\ 5-4.66 \end{bmatrix} \times [5-3 \quad 5-4.66] +$$

$$\begin{bmatrix} 3-3 \\ 6-4.66 \end{bmatrix} \times [3-3 \quad 6-4.66] +$$

$$\sigma_W = \begin{bmatrix} -1.66 \\ 0 \end{bmatrix} \times [-1.66 \quad 0] +$$

$$\begin{bmatrix} 0.33 \\ 1 \end{bmatrix} \times [-1.66 \quad 0] +$$

$$\begin{bmatrix} 1.33 \\ -1 \end{bmatrix} \times [1.33 \quad -1] +$$

$$\begin{bmatrix} -2 \\ -1.66 \end{bmatrix} \times [-2 \quad -1.66] +$$

$$\begin{bmatrix} 2 \\ 0.33 \end{bmatrix} \times [2 \quad 0.33] +$$

$$\begin{bmatrix} 0 \\ 1.33 \end{bmatrix} \times [0 \quad 1.33] +$$

$$\sigma_W = \begin{bmatrix} 12.63 & 2.98 \\ 2.98 & 6.63 \end{bmatrix}$$

$$\sigma_w^{-1} = \frac{1}{74.88} \begin{bmatrix} 6.63 & -2.98 \\ -2.98 & 12.63 \end{bmatrix}$$

ทิศทางถูกกำหนดโดย

$$V = \sigma_w^{-1}(mean_1 - mean_2)$$

$$= \frac{1}{74.88} \begin{bmatrix} 6.63 & -2.98 \\ -2.98 & 12.63 \end{bmatrix} \times \begin{bmatrix} 0.66 \\ -2.66 \end{bmatrix}$$

$$= \frac{1}{74.88} \begin{bmatrix} 12.30 \\ -34.2 \end{bmatrix}$$

$$= \begin{bmatrix} 0.164 \\ -0.457 \end{bmatrix}$$

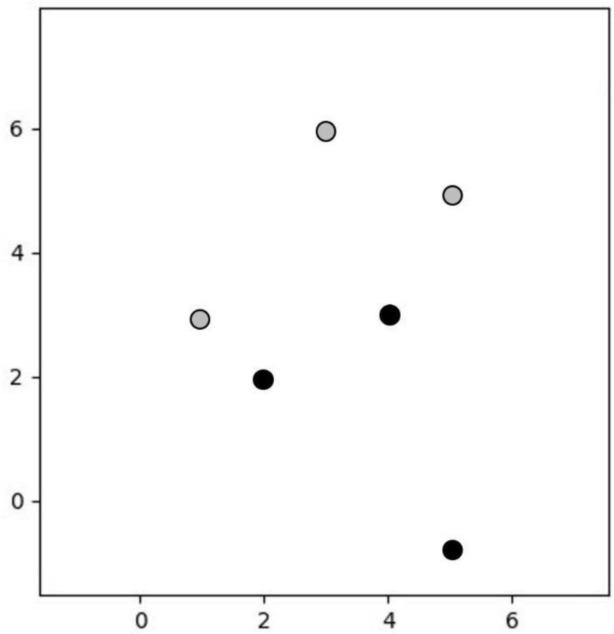
ทดลองแทนค่า v_x^t ตามตารางที่ 2.6 ค้นพบว่า

- แบบอยู่ในชั้นข้อมูลที่ 1 ถ้า $v_x^t \geq -0.715$
- แบบอยู่ในชั้นข้อมูลที่ 2 ถ้า $v_x^t \leq -1.207$

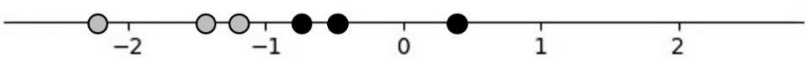
รูปท่ 2.13 และ 2.14 แสดงการฉายเงาของแบบทั้ง 6 ในปริภูมิ 2 มิติ ลงสู่ปริภูมิย่อย d มิติ

ตารางที่ 2.6 การแบ่งแยกเชิงเส้นในปริภูมิย่อย 1 มิติ

Feature 1	Feature 2	Class	v_x^t
2	2	1	–0.586
4	3	1	–0.715
5	1	1	0.363
1	3	2	–1.207
5	5	2	–1.465
3	6	2	–2.250



รูปที่ 2.13 แบบก่อนการฉายเงาในปริภูมิ 2 มิติ



รูปที่ 2.14 แบบหลังการฉายเงาในปริภูมิย่อย 1 มิติ

7.2 การวิเคราะห์องค์ประกอบหลัก

การวิเคราะห์องค์ประกอบหลัก (PCA: Pricipal Component Analysis) คือ การแปลงกลุ่มของฟีเจอร์ในเซตข้อมูลที่อาจมีบางฟีเจอร์มีความสัมพันธ์กัน เป็นกลุ่มของฟีเจอร์ใหม่ที่ไม่มีความสัมพันธ์กัน ซึ่งเรียกว่าองค์ประกอบหลัก โดยองค์ประกอบหลักแรกที่ PCA เลือก ได้แก่ องค์ประกอบหลักที่มีค่าความแปรปรวน (Variance) สูงสุด ตีความได้ว่าองค์ประกอบหลักแรกสามารถอธิบาย ข่าวสาร (Information) ในเซตข้อมูลได้มากที่สุด ในขณะที่ องค์ประกอบหลักในลำดับที่เหลือจะมีความแปรปรวนลดหลั่นกันไป

ถ้า x คือ เซตของเวกเตอร์ N หลัก ใน D มิติ ค่าเฉลี่ยของเซตข้อมูล คือ

$$m_x = E\{x\}$$

เมทริกซ์ความแปรปรวนร่วม (Covariance Matrix) คือ

$$C_x = E\{(x - m_x)(x - m_x)^T\}$$

องค์ประกอบของเมทริกซ์ C_x ได้แก่ c_{ij} เป็นตัวแทนความแปรปรวนขององค์ประกอบตัวแปรสุ่ม x_i และ x_j แต่เมทริกซ์ความแปรปรวนร่วมจากสมการข้างต้น เป็นการหาความแปรปรวนขององค์ประกอบเดียว ดังนั้น $x_i = x_j$ จึงถูกแทนในตัวแปร x ของสมการนี้ ในส่วนเพิ่มเติม เมทริกซ์นี้มีคุณสมบัติเป็น เมทริกซ์สมมาตร (Symmetric Matrix) โดยที่ ฐานหลักเชิงตั้งฉาก (Orthogonal Basis) สามารถคำนวณหาได้จากเวกเตอร์ลักษณะเฉพาะ e_i และค่าลักษณะเฉพาะ λ_i ที่เกี่ยวข้อง จากการแก้สมการด้านล่างนี้

$$C_x e_i = \lambda_i e_i \text{ เมื่อ } i = 1, \dots, n$$

จากผลลัพธ์ของสมการข้างต้นได้แก่เวกเตอร์ลักษณะเฉพาะจำนวนหนึ่ง ทำการเรียงลำดับเวกเตอร์เหล่านี้จากมากไปน้อยนำไปสู่ ฐานหลักเชิงตั้งฉากที่เรียงลำดับแล้ว (Ordered Orthogonal Basis) เวกเตอร์ลักษณะเฉพาะแรกมีนัยสำคัญสูงสุดจะถูกเลือกมาใช้ในขั้นตอนถัดไป ซึ่งเป็นตัวแทนของความแปรปรวนสูงสุดของข้อมูล

กำหนดให้ A คือ เมทริกซ์แถว (Row Matrix) ที่มาจากเวกเตอร์ลักษณะเฉพาะที่ถูกเลือกจากขั้นตอนก่อนหน้า ดังนั้น การแปลงเวกเตอร์ x เป็นไปตามสมการ ด้านล่างนี้

$$y = A(x - m_x)$$

เมื่อได้เวกเตอร์ใหม่ y แล้ว สามารถแปลงกลับไปเป็นเวกเตอร์ดั้งเดิม x ได้ตามสมการด้านล่างนี้ อย่างไรก็ตาม ค่า x ที่ได้จากการแปลงกลับจะมีค่าไม่เท่าเดิม เนื่องจากกระบวนการแปลงได้สูญเสียข้อมูลบางส่วนของแบบไป ดังที่จะได้เห็นในตัวอย่างถัดไป

$$x = A^T y + m_x$$

สมการข้างต้นนำเวกเตอร์คุณลักษณะที่มีขนาดใหญ่ที่สุดเพียงตัวเดียวมาคำนวณ อย่างไรก็ตาม เวกเตอร์คุณลักษณะที่มีขนาดใหญ่ที่สุด K เวกเตอร์แรก อาจนำมาใช้ในการฉายเงาลงบนแกนพิกัดในปริภูมิ K มิติ ดังนี้

$$y = A_k(x - m_x)$$

และ

$$x = A_K^T y + m_x$$

ตัวอย่างที่ 12 เซตข้อมูลประกอบไปด้วย 4 แบบ ใน 2 มิติ ซึ่งแบบในชั้นข้อมูลที่ 1 ได้แก่ (1, 1) และ (1, 2) ในขณะที่แบบในชั้นข้อมูลที่ 2 ได้แก่ (4, 4) และ (5, 4)

ค่าเฉลี่ยของแบบทั้งหมด คือ

$$\text{mean} = \begin{bmatrix} \frac{1+1+4+5}{4} \\ \frac{1+2+4+4}{2} \end{bmatrix}$$

$$= \begin{bmatrix} 2.75 \\ 2.75 \end{bmatrix}$$

เมทริกซ์ความแปรปรวนร่วม คือ

$$C_x = \frac{1}{4-3}\begin{bmatrix} 1-2.75 \\ 1-2.75 \end{bmatrix} \times [1-2.75 \quad 1-2.75] +$$

$$\frac{1}{4-3}\begin{bmatrix} 1-2.75 \\ 2-2.75 \end{bmatrix} \times [1-2.75 \quad 2-2.75] +$$

$$\frac{1}{4-3}\begin{bmatrix} 4-2.75 \\ 4-2.75 \end{bmatrix} \times [4-2.75 \quad 4-2.75] +$$

$$\frac{1}{4-3}\begin{bmatrix} 5-2.75 \\ 4-2.75 \end{bmatrix} \times [5-2.75 \quad 4-2.75] +$$

$$= \frac{1}{3}\begin{bmatrix} -1.75 \\ -1.75 \end{bmatrix} \times [-1.75 \quad -1.75] +$$

$$\frac{1}{3}\begin{bmatrix} -1.75 \\ -0.75 \end{bmatrix} \times [-1.75 \quad -0.75] +$$

$$\frac{1}{3}\begin{bmatrix}1.25\\1.25\end{bmatrix}\times\begin{bmatrix}1.25 & 1.25\end{bmatrix} +$$

$$\frac{1}{3}\begin{bmatrix}2.25\\1.25\end{bmatrix}\times\begin{bmatrix}2.25 & 1.25\end{bmatrix} +$$

$$= \frac{1}{3}\begin{bmatrix}3.0625 & 3.0625\\3.0625 & 3.0625\end{bmatrix} +$$

$$\frac{1}{3}\begin{bmatrix}3.0625 & 1.3125\\1.3125 & 0.5625\end{bmatrix} +$$

$$\frac{1}{3}\begin{bmatrix}1.5625 & 1.5625\\1.5625 & 1.5625\end{bmatrix} +$$

$$\frac{1}{3}\begin{bmatrix}5.0625 & 2.8125\\2.8125 & 1.5625\end{bmatrix}$$

$$= \frac{1}{3}\begin{bmatrix}12.75 & 8.75\\8.75 & 6.75\end{bmatrix}$$

$$= \begin{bmatrix}4.25 & 2.92\\2.92 & 2.25\end{bmatrix}$$

จากเงื่อนไขของเวกเตอร์ลักษณะเฉพาะและค่าลักษณะเฉพาะ ทำการแก้สมการ ดังต่อไปนี้

$$\begin{vmatrix}4.25-\lambda & 2.92\\2.92 & 2.25-\lambda\end{vmatrix} = (4.25-\lambda)(2.25-\lambda)-(2.92)^2$$

$$= \lambda^2 - 6.5\lambda + 1.0361$$

ดังนั้นประมาณค่าได้ดังนี้

$$\lambda_1 = 6.336$$

$$\lambda_2 = 0.1635$$

กำหนดให้

$$e_1 = \begin{bmatrix}e_{11}\\e_{12}\end{bmatrix}$$

$$e_2 = \begin{bmatrix}e_{21}\\e_{22}\end{bmatrix}$$

ดังนั้น

$$C_xe_1 = \lambda_1e_1$$

$$C_xe_2 = \lambda_2e_2$$

แทนค่าตัวแปรลงใน 2 สมการข้างต้น ได้ดังต่อไปนี้

$$\begin{bmatrix} 4.25 & 2.92 \\ 2.92 & 2.25 \end{bmatrix} \begin{bmatrix} e_{11} \\ e_{12} \end{bmatrix} = 6.3360 \begin{bmatrix} e_{11} \\ e_{12} \end{bmatrix}$$

$$\begin{bmatrix} 4.25 & 2.92 \\ 2.92 & 2.25 \end{bmatrix} \begin{bmatrix} e_{21} \\ e_{22} \end{bmatrix} = 0.1635 \begin{bmatrix} e_{21} \\ e_{22} \end{bmatrix}$$

แก้สมการ เพื่อหาค่าเวกเตอร์ e_1 และ e_2 (อ้างถึงแบบฝึกหัด 9) ค้นพบว่า e_2 มีขนาดเล็กมากเมื่อเปรียบเทียบกับ e_1 ดังนั้น e_2 จะไม่ถูกนำมาใช้ และ เวกเตอร์ลักษณะเฉพาะที่มีนัยสำคัญสูงสุดได้แก่ e_1

$$e_1 = \begin{bmatrix} 0.814 \\ 0.581 \end{bmatrix}$$

กำหนดให้ x_1, x_2, x_3, x_4 คือแบบก่อนการแปลง และ y_1, y_2, y_3, y_4 คือแบบหลังการแปลง ทำการแปลงแบบแรก (1, 1) ไปยังเวกเตอร์ลักษณะเฉพาะ ดังนี้

$$y_1 = A_k (x_1 - m_x)$$

$$= [0.814 \quad 0.581] \times \begin{bmatrix} -1.75 \\ -1.75 \end{bmatrix}$$

$$= -2.44$$

ในทำนองเดียวกัน (อ้างถึงแบบฝึกหัด 10)

$$y_2 = -1.86$$

$$y_3 = 1.74$$

$$y_4 = 2.56$$

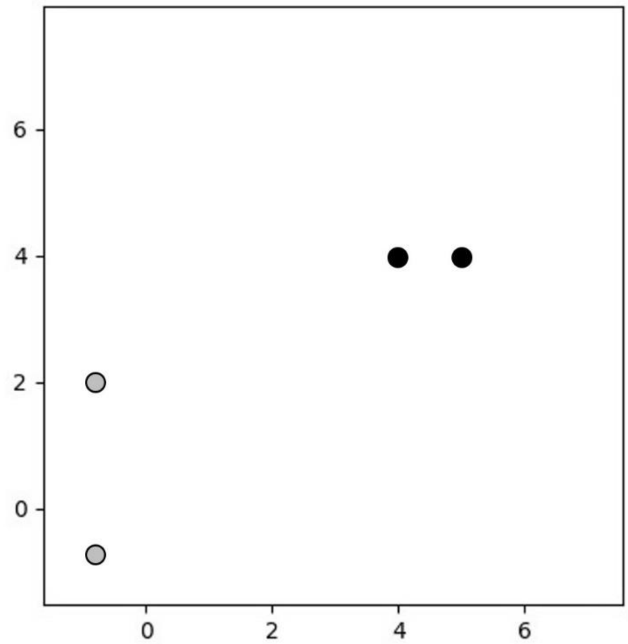
เมื่อต้องการแปลงค่าแบบใหม่กลับไปเป็นแบบดั้งเดิม ค่าของแบบอาจต่างไปจากตอนแรก เนื่องจากในกระบวนการแปลงของ PCA แบบจะสูญเสียข่าวสารบางอย่างไป สำหรับแบบแรกทดลองแปลงกลับดังนี้ (แปลง y_1 กลับเป็น x_1) โดยใช้ฟังก์ชันการแปลง

$$x_1 = A_K^T y_1 + m_x$$

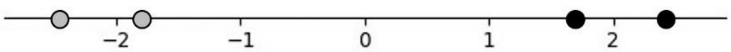
$$= [0.814 \quad 0.581]^T \times (-2.44) + \begin{bmatrix} 2.75 \\ 2.75 \end{bmatrix}$$

$$= \begin{bmatrix} 0.76 \\ 1.322 \end{bmatrix}$$

จะสังเกตได้ว่าค่าที่ได้จะไม่เท่ากับ (1, 1) แต่เป็นค่าประมาณ ในส่วนเพิ่มเติม แบบก่อนและหลังการแปลง แสดงตามรูปที่ 2.15 และ 2.16 ต่อไปนี้



รูปที่ 2.15 แบบก่อนการลดมิติโดย PCA



รูปที่ 2.16 แบบหลังการลดมิติโดย PCA

8. การคัดเลือกฟีเจอร์

สำหรับกระบวนการรู้จำแบบ ฟีเจอร์ทั้งหมดในเซตข้อมูลอาจไม่ได้มีความสำคัญทั้งหมด การกำจัดบางฟีเจอร์ที่ไร้ประโยชน์อาจส่งผลให้การจำแนกแบบมีความแม่นยำสูงขึ้น ดังนั้น ฟีเจอร์ที่ดีย่อมส่งผลต่อการเพิ่มประสิทธิภาพของตัวแบบ การคัดเลือกฟีเจอร์ทำให้มั่นใจได้ ดังนี้

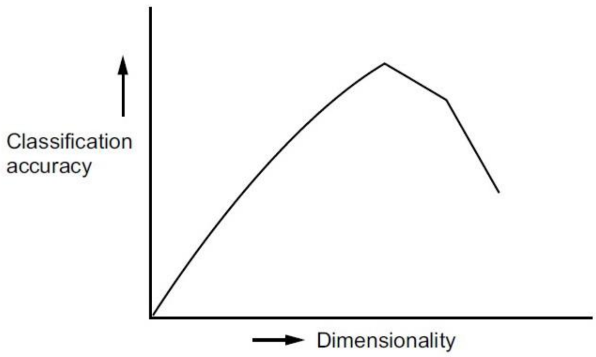
1. เป็นการลดค่าใช้จ่ายของการจำแนกแบบและการออกแบบตัวจำแนก เนื่องจากการลดขนาดของมิติโดยใช้ฟีเจอร์จำนวนน้อยลง ส่งผลดีต่อการะบวนการรู้จำแบบ ดังต่อไปนี้
    a. ช่วยให้การนำเสนอข้อมูลเรียบง่ายขึ้น
    b. ช่วยให้ใช้เวลาในการสร้างตัวแบบแลการะจำแนกแบบเร็วขึ้น
    c. ช่วยให้ใช้หน่วยความจำน้อยลง

2. เป็นการปรับปรุงประสิทธิภาพการทำนายให้แม่นยำขึ้น ด้วยเหตุผลดังต่อไปนี้

    a. ประสิทธิภาพของตัวจำแนกแบบขึ้นอยู่กับ ความสัมพันธ์ของแบบในชั้นข้อมูลเดียวกัน จำนวนของฟีเจอร์ และ ความซับซ้อนในการสร้างตัวจำแนกแบบ โดยทั่วไปแล้ว การเพิ่มจำนวนของฟีเจอร์และจำนวนแบบส่งผลให้ตัวแบบเรียนรู้และจำแนกแบบได้ดีขึ้น อย่างไรก็ตาม การเพิ่มขนาดของเซตข้อมูลจนถึงจุดหนึ่งอาจส่งผลเสียต่อกระบวนการจำแนกแบบได้ กล่าวคือ ทำนายแบบได้ผิดพลาดมากยิ่งขึ้น

    b. ถ้าเซตข้อมูลมีจำนวนแบบทดสอบน้อย การเพิ่มจำนวนฟีเจอร์อาจนำไปสู่ปรากฏการณ์จุดยอด แสดงตามกราฟในรูปที่ 2.17 จากกราฟเห็นได้ว่า การเพิ่มจำนวนมิติเมื่อฟีเจอร์ยังคงมีจำนวนน้อย สามารถเพิ่มความแม่นยำของตัวแบบได้ ในทางกลับกัน ผลร้ายจะเกิดขึ้นถ้าจำนวนของฟีเจอร์มีปริมาณเกินค่า ๆ หนึ่ง

    c. เมื่อจำนวนมิติเพิ่มขึ้น การคำนวณระยะทางจากแต่ละแบบไปบังแบบที่ใกล้ที่สุดและแบบที่ไกลที่สุด มีค่าไม่ต่างกันมาก ผลลกระทบนี้มีต่อ ปัญหาเพื่อนบ้านใกล้ที่สุด ดังที่จะกล่าวถึงในบทต่อไป ซึ่งลดมิติให้ผลลัพธ์ที่ดีกว่า



รูปที่ 2.17 ปรากฏการณ์จุดยอดและผลร้ายของการปรับมิติ

โดยทั่วไปแล้ว ขั้นตอนการคัดเลือกฟีเจอร์เป็นการค้นหาเซตย่อยที่แตกต่างกันทั้งหมดของฟีเจอร์ เพื่อเลือกเซตย่อยที่ดีที่สุด ถ้าเซตข้อมูลมีฟีเจอร์ปริมาณมาก กระบวนการค้นหานี้จะใช้เวลายาวนาน ดังนั้นจึงอาจไม่สามารถใช้การค้นหาที่ดีที่สุดได้ แต่จำเป็นต้องใช้การค้นหาเซตย่อยที่ดีรองลองไป เพื่อให้การประมวลผลเร็วขึ้น

ฟังก์ชันเกณฑ์ (Criterion Function) J(X) ถูกใช้ในการคำนวณหา ความผิดพลาดการจำแนก (Classification Error) ของเซตย่อยฟีเจอร์ (แทนด้วยสัญลักษณ์ X) กำหนดให้ P_e คือ ความน่าจะเป็นของความผิดพลาดการจำแนก ฟังก์ชันดังกล่าวคำนวณได้ ตามสมการด้านล่างนี้

$$J(X) = (1 - P_e)$$

ถ้าค่าของฟังก์ชัน J มีค่าเพิ่มขึ้น หมายความว่าเซตย่อยของฟีเจอร์มีคุณภาพมากขึ้น (ตัวแบบจำแนกแบบได้แม่นยำมากขึ้น) ในทางกลับกัน ค่า J ที่ลดลง หมายความว่าเซตย่อยของฟีเจอร์มีคุณภาพน้อยลง (ตัวแบบจำแนกแบบได้แม่นยำลดลง)

8.1 การค้นหาแบบทั้งหมด

การค้นหาที่ตรงไปตรงมาที่สุดได้แก่ การค้นหาแบบทั้งหมด (Exhaustive Search) เป็นการค้นหาเซตย่อยฟีเจอร์ทุก ๆ รูปแบบที่เป็นไปได้ เพื่อตัดสินเซตย่อยฟีเจอร์ที่ดีที่สุด ถ้าจำนวนฟีเจอร์ทั้งหมดในเซตข้อมูลมีค่าเท่ากับ d ฟีเจอร์ และ เซตย่อยมีขนาด m ฟีเจอร์ ดังนั้น การค้นหาซับเซตย่อยที่มีความผิดพลาดการจำแนกต่ำสุด ต้องค้นหาทั้งหมด ตารางที่ 2.7 แสดงเซตย่อยต่าง ๆ ของเซตข้อมูลที่มี 5 ฟีเจอร์ กำหนดให้ทุกเซตย่อยประกอบด้วย 3 ฟีเจอร์ เมื่อกำหนดให้ค่า 0 คือฟีเจอร์ที่ถูกกำจัดออก ในขณะที่ 1 คือฟีเจอร์ที่ถูกบรรจุลงในเซตย่อย

ตารางที่ 2.7 การค้นหาแบบละเอียด

Sl. No.	f1	f2	f3	f4	f5
1.	0	0	1	1	1
2.	0	1	0	1	1
3.	0	1	1	0	1
4.	0	1	1	1	0
5.	1	0	0	1	1
6.	1	0	1	0	1
7.	1	0	1	1	0
8.	1	1	0	0	1
9.	1	1	0	1	0
10.	1	1	1	0	0

เทคนิคนี้ไม่สามารถนำไปใช้ได้จริงในทางปฏิบัติ สำหรับเซตข้อมูลที่ d และ m มีค่าสูงมาก ยกตัวอย่างเช่น ถ้า d = 24 และ m = 12 แล้ว จำนวนเซตย่อยทั้งหมดจะมีประมาณ 2.7 ล้านเซต

8.2 การค้นหาแบบแตกกิ่งและจำกัดเขต

การค้นหาแบบแตกกิ่งและจำกัดเขต (Branch and Bound Search) หลีกเลี่ยงการค้นหาเซตย่อยทุกแบบที่เป็นไปได้เพื่อค้นหาเซตย่อยที่ดีที่สุดเหมือนการค้นหาแบบละเอียด แต่ใช้การค้นหาเซตย่อยที่ดี (ไม่ถึงกับดีที่สุด) ร่วมกับการตัดสินค่า เกณฑ์ (Criterion) ร่วมกับ ขอบเขต (Bound) ดังที่จะกล่าวต่อไป การค้นหาชนิดนี้มีสมมติฐานเป็น ฟังก์ชันที่มีค่าไม่เพิ่มขึ้นหรือไม่ลดลงเพียงอย่างเดียว (Monotone)

กำนดให้ (Z_1, Z_2, …, Z_l) คือฟีเจอร์จำนวน l = d – m ฟีเจอร์ ที่ถูกกำจัดทิ้งจากทั้งหมด d ฟีเจอร์ เพื่อให้ได้มาซึ่งเซตย่อย m ฟีเจอร์ แต่ละตัวแปร Z_i มีค่าลำดับตั้งแต่ 1 ถึง d อย่างไรก็ตาม ลำดับของฟีเจอร์นี้ไม่สำคัญและมีค่าไม่ซ้ำกัน เมื่อพิจารณาตามแค่ลำดับของ Z_i ดังนั้น $Z_1 < Z_2 < \ldots < Z_l$ เกณฑ์ของการคัดเลือกฟีเจอร์ คือ $J_l(Z_1, \ldots, Z_l)$ ปัญหาการคัดเลือกฟีเจอร์ของเซตย่อยคือการค้นหาเซตย่อยที่ดีที่สุด $Z_1^*, \ldots, Z_l^*$ ดังนั้น

$$J_l(Z_1^*, \ldots, Z_l^*) = max\, J_l(Z_1, \ldots, Z_l)$$

ถ้าเกณฑ์ J เป็นฟังก์ชันที่มีค่าไม่เพิ่มขึ้นหรือไม่ลดลงเพียงอย่างเดียว แล้ว

$$J_1(Z_1) \geq J_2(Z_1, Z_2) \geq \ldots \geq J_l(Z_1, \ldots, Z_l)$$

หมายความว่า เซตย่อยฟีเจอร์ไม่ดีไปกว่าเซตที่มีขนาดใหญ่กว่าที่บรรจุเซตย่อยนี้

กำหนดให้ B คือ ขอบเขตล่าง ของค่าเกณฑ์ที่ดีที่สุด $J_l(Z_1^*, \ldots, Z_l^*)$ นั่นคือ

$$B \leq J_l(Z_1^*, \ldots, Z_l^*)$$

ถ้า $J_k(Z_1, \ldots, Z_k) < B$ เมื่อ $k \leq l$ แล้ว

$$J_l(Z_1, \ldots, Z_k, Z_{k+1}, \ldots, Z_l) \leq B$$

สำหรับค่าทั้งหมดที่เป็นไปได้ของ $Z_{k+1}, \ldots, Z_l$

ขั้นตอนวิธีการค้นหาแบบแตกกิ่งและจำกัดเขต เริ่มต้นจากการสร้างโหนดรากซึ่งแทนเซตของทุกฟีเจอร์ จากนั้นทำการสร้างต้นไม้ย่อยต่อจากโหนดรากนี้ และคำนวณค่าเกณฑ์อย่างต่อเนื่องทุกต้นไม้ย่อย เมื่อไหร่ก็ตาม เกณฑ์ของโหนดใดมีค่าน้อยกว่าขอบเขต B แล้ว โหนดทั้งหมดที่ถูกสร้างต่อจากโหนดนี้จะมีเกณฑ์น้อยกว่า B ด้วย ดังนั้น โหนดเหล่านี้จะไม่ใช่คำตอบที่ดีที่สุด และ ต้นไม้ย่อยภายใต้โหนดนี้จะไม่ถูกสร้างต่อไป สุดท้ายแล้ว คำตอบที่ดีที่สุดมาจากโหนดที่มีค่าเกณฑ์สูงสุด

ตัวอย่างที่ 13 พิจารณาเซตข้อมูลที่ประกอบด้วย 4 ฟีเจอร์ ได้แก่ f1, f2, f3 และ f4 รูปที่ 2.18 แสดงต้นไม้ของการค้นหาแบบแตกกิ่งและจำกัดเขตเมื่อฟีเจอร์ถูกลบออกทีละหนึ่งฟีเจอร์ ตัวเลขประจำโหนดแสดงลำดับขอโหนดที่ถูกสร้างในต้นไม้ เมื่อโหนดใบถูกสร้างแล้วค่าเกณฑ์ของโหนดใบจะถูกประเมิน ยกตัวอย่าง การสร้างต้นไม้ในรูปที่ 2.18 ถูกอธิบาย ดังต่อไปนี้

เมื่อโหนด 3 ถูกสร้าง เนื่องจากค่าเกณฑ์คือ 25 ดังนั้น ขอบเขตถูกตั้งค่าเริ่มต้นให้เป็น 25

โหนดต่อไป (โหนด 4) เกณฑ์คือ 45 ซึ่งมากกว่า 25 ดังนั้น ขอบเขตถูกปรับปรุงค่าใหม่เป็น 45

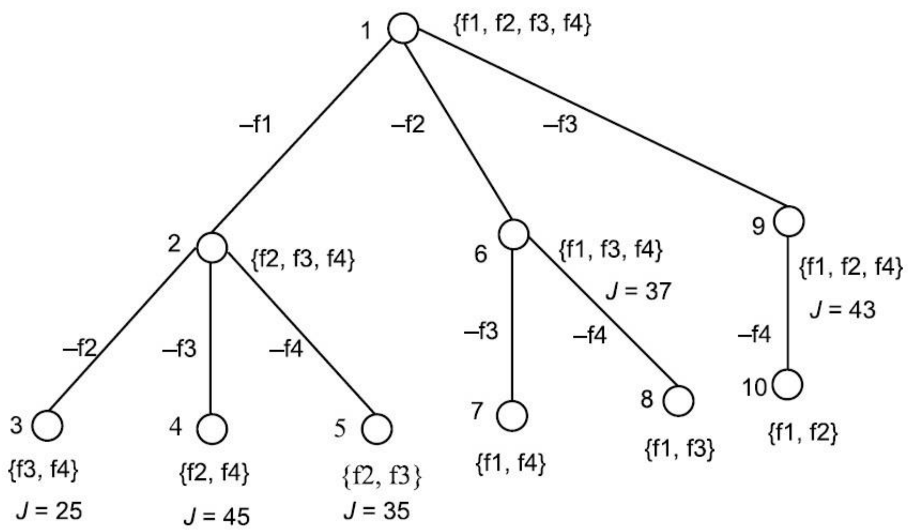
โหนด 5 มีเกณฑ์น้อยกว่าขอบเขตล่าสุด ดังนั้น ขอบเขตยังคงเป็นค่าเดิมคือ 45

เมื่อโหนด 6 ถูกประเมิน ค้นพบว่าเกณฑ์คือ 37 ซึ่งน้อยกว่าขอบเขตล่าสุด ดังนั้นจะไม่มีการสร้างโหนดลูกต่อจากโหนดนี้

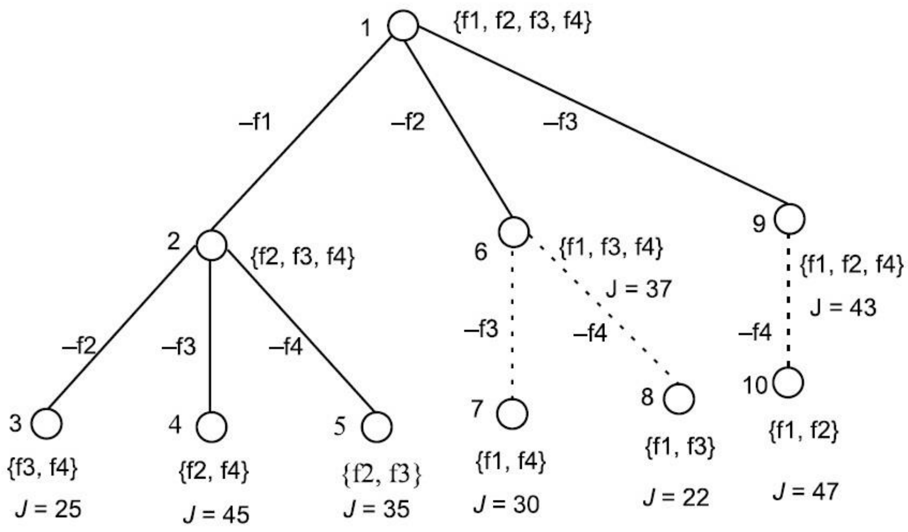
เนื่องจากการค้นหาชนิดนี้มีสมมติฐานโมโนโทน ดังนั้น โหนด 7 และ 8 จะมีค่าเกณฑ์ต่ำกว่า 37

ในทำนองเดียวกัน โหนด 9 มีเกณฑ์ 43 ดังนั้นจึงไม่มีการสร้างต้นไม้ย่อยต่อจากโหนดนี้ เนื่องจากเกณฑ์น้อยกว่าขอบเขต

ท้ายที่สุดแล้ว ฟีเจอร์ที่ถูกกำจัดคือ f1 และ f3 โดยที่ ผลลัพธ์สุดท้ายแสดงผลตาม รูปที่ 2.19



รูปที่ 2.18 ต้นไม้ เมื่อ d = 4 และ m = 2



รูปที่ 2.19 การค้นหาแบบแตกกิ่งและจำกัดเขต

8.3 การค้นหาฟีเจอร์เดี่ยวที่ดีที่สุด

การค้นหาฟีเจอร์เดี่ยวที่ดีที่สุด (Best Individual Feature Selection) มีแนวคิดที่เข้าใจได้ง่าย กล่าวคือ ฟีเจอร์ที่ดีที่สุดจำนวนหนึ่งเท่านั้นที่จะถูกเลือกใช้ นอกจากนี้ แต่ละฟีเจอร์ในเซตข้อมูลจะถูกประเมินอย่างเป็นอิสระต่อกัน ทำให้การคำนวณไม่ยุ่งยากซับซ้อน อย่างไรก็ตาม การค้นหาประเภทนี้อาจล้มเหลวต่อการนำไปใช้งาน เนื่องจาก ความไม่อิสระระหว่างฟีเจอร์สมควรต้องถูกนำมาพิจารณาประกอบกัน

8.4 การค้นหาตามลำดับ

การค้นหาตามลำดับ (Sequential Search) เป็นวิธีการสำหรับเพิ่มฟีเจอร์ที่จำเป็น หรือลบฟีเจอร์ที่ไม่สำคัญ การค้นหาประเภทนี้แบ่งออกได้เป็น 2 ประเภท ได้แก่ การค้นหาเดินหน้าตามลำดับ (SFS: Sequential Forword Selection) และ การค้นหาแบบถอยหลังตามลำดับ (SBS: Sequential Backword Selection) ดังที่จะอธิบายรายละเอียดต่อไป เนื่องจากการค้นหาเหล่านี้ไม่ได้สำรวจเซตย่อยทุกเซตที่เป็นไปได้ ดังนั้นขั้นตอนวิธีนี้ไม่ได้รับประกันผลลัพธ์ที่ดีที่สุด ซึ่งต่างจากการค้นหาแบบละเอียดและการค้นหาแบบแตกกิ่งและจำกัดเขต นอกจากนี้ ข้อเสียของการค้นหาเหล่านี้คือ ผลกระทบแบบเป็นลำดับขั้น (Nesting Effect) ดังที่จะกล่าวถึงในหัวข้อถัดไป

8.4.1 การค้นหาแบบเดินหน้าตามลำดับ

การค้นหาประเภทนี้พิจารณาเพิ่มฟีเจอร์ที่สำคัญทีละหนึ่งฟีเจอร์ในลักษณะ ล่างขึ้นบน (Bottom-up) กล่าวคือ เริ่มต้นจากเซตว่างแล้วเพิ่มขนาดเซตจากเล็กไปใหญ่ อย่างไรก็ตาม วิธีการนี้ได้รับผลกระทบแบบเป็นลำดับขั้น เนื่องจากฟีเจอร์ที่ถูกพิจารณาให้เพิ่มเข้าไปในเซตแล้ว จะไม่สามารถถูกลบทิ้งได้ในภายหลัง

8.4.2 การค้นหาแบบถอยหลังตามลำดับ

การค้นหาประเภทนี้พิจารณาลบฟีเจอร์ที่ไม่มีประโยชน์ทีละหนึ่งฟีเจอร์ในลักษณะ บนลงล่าง (Top-down) กล่าวคือ เริ่มต้นจากเซตที่ประกอบไปด้วยฟีเจอร์ทุกฟีเจอร์แล้วลดขนาดเซตจากใหญ่ไปเล็ก อย่างไรก็ตาม วิธีการนี้ประสบปัญหาจากผลกระทบแบบเป็นลำดับขั้น เนื่องจากฟีเจอร์ที่ถูกลบทิ้งออกจากเซตไปแล้ว จะไม่สามารถถูกเพิ่มได้ในภายหลัง

8.5 การค้นหาแบบลอยตามลำดับ

จากผลกระทบการซ้อนในที่กล่าวถึงไว้ก่อนหน้านี้ ยังไม่มีทฤษฎีใดที่บ่งบอกได้ถึง จำนวนของฟีเจอร์ที่ควรเพิ่ม (l) โดย SFS และ จำนวนของฟีเจอร์ที่ควรลบ (r) โดย SBS ดังนั้น SFS และ SBS จึงถูกพัฒนาต่อมาเป็น การค้นหาแบบลอยตามลำดับ (Sequential Floating Search) เพื่อให้มีประสิทธิภาพมากยิ่งขึ้น กล่าวคือ การค้นหาแบบใหม่นี้ ไม่มีความจำเป็นต้องระบุค่า l และ r อีกต่อไป เนื่องจากจำนวนของการเพิ่มและลบฟีเจอร์เป็นแบบพลวัติ มีการเปลี่ยนแปลงตลอดเวลา ไม่ถูกระบุค่าตายตัว โดยการพิจารณาค่าของฟังก์ชันเกณฑ์ที่มีค่าสูงสุด ในแต่ละขั้นตอน มีเพียงฟีเจอร์เดียวเท่านั้นที่ถูกเพิ่มหรือลบ สำหรับวิธีใหม่นี้ มิติของฟีเจอร์จะไม่มีการเปลี่ยนแปลงแบบโมโนโทน (เพิ่มขึ้นหรือลดลงเพียงอย่างเดียว) เหมือน SFS หรือ SBS แต่อาจจะ เพิ่มขึ้น (Floating Up) หรือ ลดลง (Floating Down) ได้ตลอดเวลาอย่างยืดหยุ่น การค้นหาแบบลอยตามลำดับแบ่งออกได้เป็น 2 ประเภท ได้แก่

- การค้นหาเดินหน้าแบบลอยตามลำดับ (SFFS: Sequential Floating Forward Search)
  SFFS ทำการเพิ่มฟีเจอร์ก่อนแล้วจึงลบฟีเจอร์ สลับกันไปมา

- การค้นหาถอยหลังแบบลอยตามลำดับ (SFBS: Sequential Floating Backward Search)
  SFBS มีการทำงานคล้ายกับ SFFS แต่ทำการลบฟีเจอร์ก่อนแล้วจึงเพิ่มฟีเจอร์ทีหลัง สลับกันไปมา

การค้นหาเดินหน้าแบบลอยตามลำดับ (SFFS)

1. เริ่มต้นให้เซตย่อยฟีเจอร์เป็นเซตว่าง กำหนดให้ค่า k แทน จำนวนฟีเจอร์ เริ่มต้น k = 0
2. เพิ่มฟีเจอร์ที่มีความสำคัญมากที่สุด ไปยังเซตย่อยขนาด k แล้วปรับค่า k = k + 1
3. ลบฟีเจอร์ที่มีความสำคัญน้อยที่สุดจากเซตย่อยนี้ โดยพิจารณาเงื่อนไขในของขั้นตอนที่ 4
4. **IF** เซตย่อยปัจจุบัน คือ เซตย่อยที่ดีที่สุดขนาด k – 1 **THEN**

    ให้ค่า k = k – 1 และ ไปยังขั้นตอนที่ 3

    ELSE

    เพิ่มฟีเจอร์ที่ถูกลบบเข้าไปในเซตย่อยปัจจุบัน และ ไปยังขั้นตอนที่ 2

ตัวอย่างที่ 14 พิจารณาเซตข้อมูลที่มี 4 ฟีเจอร์ ได้แก่ f1, f2, f3, f4 ขั้นตอนวิธี SFFS แสดงดังต่อไปนี้

1. กำหนดเซตย่อยฟีเจอร์เป็นเซตว่าง (F = ∅)
2. ฟีเจอร์ที่มีความสำคัญมากที่สุดถูกค้นพบ ได้แก่ f3 ดังนั้น F = {f3}
3. ฟีเจอร์ที่มีความสำคัญน้อยที่สุดถูกค้นพบ ได้แก่ f3 แสดงว่า f3 สามารถถูกลบทิ้งได้
4. เนื่องจากการลบ f3 ไม่ได้ทำให้ค่าเกณฑ์ดีขึ้น ดังนั้น เก็บ f3 ไว้
5. ฟีเจอร์ที่มีความสำคัญมากที่สุดที่ถูกค้นพบ ได้แก่ f2 ดังนั้น F = {f3, f2}
6. ฟีเจอร์ที่มีความสำคัญน้อยที่สุดที่ถูกค้นพบ ได้แก่ f2 ดังนั้น ไม่เป็นไปตามเงื่อนไข
7. ฟีเจอร์ที่มีความสำคัญมากที่สุดที่ถูกค้นพบ ได้แก่ f1 ดังนั้น F = {f3, f2, f1}
8. ฟีเจอร์ที่มีความสำคัญน้อยที่สุดที่ถูกค้นพบ ได้แก่ f3 ถ้าให้เซตย่อยที่ดีกว่า ดังนั้น F = {f1, f2}

คำตอบที่ดีที่สุด คือ F = {f1, f2} ถ้า m = 2 ถ้าขั้นตอนที่ 8 ฟีเจอร์ที่มีความสำคัญน้อยที่สุดที่ถูกค้นพบ ได้แก่ f1 แล้ว F = {f2, f3} เหมือนขั้นตอนที่ 5 ดังนั้น การทำงานของขั้นตอนวิธีนี้จะนำไปสู่ การวนซ้ำ

การค้นหาถอยหลังแบบลอยตามลำดับ (SFBS)

1. เริ่มต้น k = N เมื่อ N คือจำนวนของฟีเจอร์
2. ลบฟีเจอร์ที่มีความสำคัญน้อยที่สุด ไปยังเซตย่อยขนาด k แล้วปรับค่า k = k – 1
3. เพิ่มฟีเจอร์ที่มีความสำคัญมากที่สุดนอกเซตย่อยนี้ โดยพิจารณาเงื่อนไขของขั้นตอนที่ 4
4. **IF** เซตย่อยปัจจุบัน คือ เซตย่อยที่ดีที่สุดขนาด k + 1 **THEN**

    ให้ค่า k = k + 1 และ ไปยังขั้นตอนที่ 3

    ELSE

    ลบฟีเจอร์ที่ถูกเพิ่มออกจากเซตย่อยปัจจุบัน และ ไปยังขั้นตอนที่ 2

เนื่องจาก SFBS เริ่มต้นจากเซตที่บรรจุทุกฟีเจอร์ จากนั้นทำการลบออกทีละฟีเจอร์ วิธีการนี้ไม่ค่อยจะให้คำตอบที่ดีนัก นอกจากนี้ ถ้ามิติมีขนาดใหญ่ กล่าวคือ d มีขนาดใหญ่มาก แต่ะ m มีขนาดเล็กมาก จะส่งผลให้ใช้เวลาประมวลผลยาวนาน ดังนั้น ควรใช้ SFFS เพื่อหลีกเลี่ยงผลกระทบดังกล่าว

8.6 วิธีเข้าสู่ค่ามากน้อยที่สุด

วิธีเข้าสู่ค่ามากน้อยที่สุด (Max-Min Approach) ประหยัดเวลาประมวลได้เหนือกว่าวิธีการก่อนหน้านี้ทั้งหมดที่ประมวลผลบนปริภูมหลายมิติ เนื่องจากวิธีการใหม่นี้ประมวลผลบนปริภูมิเพียง 2 มิติ อย่างไรก็ตาม ผลลัพธ์ที่ได้ไม่น่าพอใจนัก

กำหนดให้

f_i = ฟีเจอร์จากเซตของฟีเจอร์ที่ถูกพิจารณา $F_k = \{f_1, \ldots, f_k\}$ มาจากการเลือกฟีเจอร์ลำดับที่ i

g_i = ฟีเจอร์ลำดับที่ j จากเซตของฟีเจอร์ที่ไม่ถูกพิจารณา

$\Delta J(g_j, f_i) = | J(g_j, f_i) – J(f_i) |$

วิธีการเข้าสู่ค่ามากน้อยที่สุด ฟีเจอร์ g_j จะถูกเลือกใช้งาน ถ้าเป็นไปตามเงื่อนไขด้านล่างนี้

$\max_{\forall g_j} \min_{\forall f_i} \Delta J(g_j, f_i)$

ข้อเสียของวิธีเข้าสู่ค่ามากน้อยที่สุดคือการสูญเสียข่าวสารของฟีเจอร์ที่ขาดการพิจารณา เนื่องจากวิธีการใหม่นี้พิจารณาเพียงแค่ 2 ฟีเจอร์เท่านั้น

ตัวอย่างที่ 14 พิจารณาเซตข้อมูลที่มี 4 ฟีเจอร์ f1, f2, f3 และ f4 ค่าของฟังก์ชันเกณฑ์ระหว่างสองฟีเจอร์แสดงตาม ตารางที่ 2.8

ตารางที่ 2.8 ฟังก์ชันเกณฑ์ของเซตย่อยฟีเจอร์

Feature	f1	f2	f3	f4
f1	10	30	35	55
f2	30	20	40	53
f3	35	40	30	42
f4	55	53	42	40

ถ้า J (f1) คือ ฟังก์ชันเกณฑ์ โดยใช้เพียงแค่ฟีเจอร์ f1 ดังนั้น ฟังก์ชันเกณฑ์ทั้งหมด ได้แก่

J (f1) = 10; J (f2) = 20; J (f3) = 30; J (f4) = 40

ฟังก์ชันเกณฑ์เหล่านี้ คือค่าตาม เส้นทแยงมุม (Diagonal) ของตารางที่ 2.8

เริ่มต้นพิจารณา f1 ถ้า J (f1, f2) = 30, J (f1, f3) = 35 และ J (f1, f4) = 55 แล้ว

Δ J (f1, f2) = 10; Δ J (f1, f3) = 5; Δ J (f1, f4) = 15

ค่าน้อยที่สุด คือ Δ J (f1, f3) = 5

ต่อไป พิจารณา f2 ถ้า J (f2, f1) = 30, J (f2, f3) = 40 และ J (f2,f4) = 53 แล้ว

Δ J (f2, f1) = 20; Δ J (f2, f3) = 10; Δ J (f2, f4) = 13

ค่าน้อยที่สุด คือ Δ J (f2, f3) = 10

ต่อไป พิจารณา f3 ถ้า J (f3, f1) = 35, J (f3, f2) = 40 และ J (f3, f4) = 42 แล้ว

Δ J (f3, f1) = 25; Δ J (f3, f2) = 20; Δ J (f3, f4) = 2

ค่าน้อยที่สุด คือ Δ J (f3, f4) = 2

ต่อไป พิจารณา f4 ถ้า J (f4, f1) = 55, J (f4, f2) = 53 และ J (f4, f3) = 42 แล้ว

Δ J (f4, f1) = 45; Δ J (f4, f2) = 33; Δ J (f4, f3) = 12

ค่าน้อยที่สุด คือ Δ J (f4, f3) = 12

สุดท้าย หาค่ามากที่สุดท่ามกลางค่าน้อยที่สุดทั้งหมด ดังนั้น f4 ถูกเลือกเป็นฟีเจอร์สำหรับกระบวนการรู้จำแบบต่อไป

จากการคำนวณจากตัวอย่างข้างต้นจะเห็นได้ว่า มีเพียงแค่สองฟีเจอร์เท่านั้นที่ถูกพิจารณาในแต่ละครั้ง ถ้าจำนวนของฟีเจอร์ที่พิจารณาในแต่ละครั้งมีมากกว่าสองฟีเจอร์ ผลลัพธ์ที่ได้โดยวิธีการนี้อาจแตกต่างกันออกไป

อภิปราย

เวลาในการออกแบบคือเวลาในการสร้างตัวจำแนกแบบ จัดเป็นต้นทุนที่ทำเพียงครั้งเดียว หลังจากตัวจำแนกแบบถูกสร้างเรียบร้อยแล้ว ตัวแบบสามารถนำไปใช้ในการทำนายแบบได้ไม่จำกัดจำนวนครั้งจนกว่าจะมีการปรับปรุงหรือสร้างตัวแบบใหม่ ดังนั้นเวลาในการสร้างตัวแบบจัดเป็นเวลาที่แม้ใช้เวลายาวนานก็อาจยอมรอหรือยอมรับต้นทุนตรงนี้ได้ ในทางกลับกัน เวลาในการทำนายคือเวลาในการจำแนกแบบจัดเป็นต้นทุนสะสม เพราะการทำนายแต่ละครั้งต้นทุนตรงนี้จะถูกนับรวมไปด้วย ถ้าเวลาในการทำนายใช้เวลายาวนานและจำเป็นต้องทำนายแบบปริมาณมากอย่างต่อเนื่อง ต้นทุนลักษณะนี้อาจไม่เหมาะสมกับโปรแกรมประยุกต์บางประเภท เช่น ปัญญาประดิษฐ์ในเกมส์คอมพิวเตอร์ เพราะเกมส์ต้องไหลลื่นอยู่ตลอดเวลาเพื่อให้ผู้เล่นรู้สึกสนุกสนานในการเล่น ไม่ควรให้ผู้เล่นต้องรอการตัดสินใจของคอมพิวเตอร์นานเกินไปได้ หรือการทำนายแพ็กเกจการบุกรุกในเครือข่ายคอมพิวเตอร์ จำเป็นที่ต้องรู้ผลในทันทีทันใด เพื่อสกัดกั้นไม่ให้แพ็กเกจดังกล่าวเข้ามาทำอันตรายระบบเครือข่ายได้ ในทางตรงกันข้าม เวลาในการวินิจฉัยของตัวแบบทำนายมะเร็งสามารถรอได้ เพราะคนไข้ไม่จำเป็นต้องรู้ผลทันทีทันใด คนไข้อาจได้รับการนัดหมายให้มาฟังผลในภายหลังได้ หรือการตรวจจับพื้นที่ทางภูมิศาสตร์ว่าพื้นที่ใดอาจมีทรัพยากรใต้ดินเช่นน้ำมันหรือแร่ธาตุ ก็ไม่จำเป็นต้องรีบใช้เวลาในการทำนายเป็นต้น ดังนั้นเวลาในการทำนายจะช้าหรือเร็ว ควรรอได้หรือรอไม่ได้ ขึ้นอยู่กับโปรแกรมประยุกต์ที่ต้องการใช้งานเป็นสำคัญ

สรุป

ลักษณะของเซตข้อมูลที่จัดว่ามีคุณภาพดีสำหรับปัญหาการจำแนกแบบได้แก่ เซตข้อมูลที่แบบในแต่ละชั้นข้อมูลต้องมีค่าฟีเจอร์ที่มีความแตกต่างกันอย่างเห็นได้ชัด เนื่องจากถ้าแบบจากชั้นข้อมูลทั้งหมดมีค่าฟีเจอร์คล้ายกัน ตัวจำแนกแบบจะไม่สามารถระบุได้ว่าแบบที่ต้องการทำนายควรอยู่ในชั้นข้อมูลไหน

โครงสร้างข้อมูลที่ถูกใช้งานมากที่สุดสำหรับแทนแบบในเครื่องคอมพิวเตอร์ได้แก่ เวกเตอร์ เนื่องจากเวกเตอร์มีการเก็บค่าเป็นจำนวนจริง จึงทำให้ง่ายต่อการใช้ฟังก์ชันทางคณิตศาสตร์เช่นหน่วยวัดความใกล้เคียง ซึ่งเป็นเครื่องมือที่ถูกใช้ในการเปรียบเทียบว่าแบบสองแบบมีความคล้ายกันมากน้อยเพียงใด โดยที่หน่วยวัดความใกล้เคียงที่นิยมใช้กันมากที่สุดได้แก่ระยะทางยุคลิดที่ถูกใช้ในการคำนวณก่อนการทำนายแบบ

ขั้นตอนวิธีที่ใช้ในการจัดกลุ่มข้อมูลที่มีแนวคิดพื้นฐานที่สุดคือเคมีนส์ โดยมีพารามิเตอร์เคซึ่งแทนจำนวนกลุ่มข้อมูลที่ผู้ใช้สามารถกำหนดค่าตามต้องการได้ ผลลัพธ์ที่ได้คือกลุ่มข้อมูลจำนวนเคกลุ่มข้อมูล โดยมีคุณสมบัติคือแบบในกลุ่มข้อมูลเดียวกันมีความคล้ายกันมากกว่าแบบจากต่างกลุ่มข้อมูลกัน เคมีนส์ถูกใช้ในการแบ่งแบบออกเป็นกลุ่มต่าง ๆ เพื่อกำหนดชื่อชั้นข้อมูลก่อนนำเข้าสู่กระบวนการการรู้จำแบบ

การกำหนดสาระสำคัญของเซตข้อมูลคือการเลือกตัวแทนของแบบจากแต่ละกลุ่มข้อมูล โดยที่ตัวแทนที่เลือกมาอาจจะเลือกเพียงแบบเดียวต่อกลุ่มข้อมูลหรือหลายแบบต่อกลุ่มข้อมูลก็ได้ โดยทั่วไปการกำหนดสาระสำคัญถูกใช้หลังจากการจัดกลุ่มข้อมูลเรียบร้อยแล้ว กล่าวคือเมื่อเซตข้อมูลผ่านการกำหนดสาระสำคัญแล้ว ย่อมส่งผลให้ใช้เวลาคำนวณในการจำแนกแบบเร็วขึ้น เพราะการคำนวณใด ๆ บนตัวแทนเพียงไม่กี่ตัวย่อมใช้เวลาน้อยกว่าการคำนวณแบบทั้งหมดในเซตข้อมูล ซึ่งมีปริมาณที่แตกต่างกันสูงมาก

การประเมินค่าตัวจำแนกแบบว่ามีประสิทธิภาพมากน้อยเพียงใด สามารถประเมินได้จากปัจจัยต่าง ๆ ดังต่อไปนี้คือ ความแม่นยำในการทำนายแบบ เวลาในการสร้างตัวจำแนกแบบ เวลาในการทำนายแบบ พื้นที่ในการจัดเก็บตัวจำแนกแบบ ความสามารถในการอธิบายขั้นตอนการทำนายเป็นภาษามนุษย์ และ การรับมือกับสัญญาณรบกวนได้ดี

การสกัดฟีเจอร์ถูกใช้สำหรับลดขนาดมิติของเซตข้อมูลในช่วงก่อนการประมวลผล ตัวอย่างของการสกัดฟีเจอร์ได้แก่ การจำแนกเชิงเส้นของฟิชเชอร์ ที่ฉายเงาของแบบในปริภูมิหลายมิติลงบนเส้นตรง และ PCA ที่รวมฟีเจอร์ต่าง ๆ เข้าเป็นฟีเจอร์เดียวกัน โดยที่ ฟีเจอร์ใหม่ที่ได้จาก PCA จะไม่มีความสัมพันธ์กัน ซึ่งต่างจากฟีเจอร์ดั้งเดิมที่อาจมีความสัมพันธ์กัน อย่างไรก็ตาม การแปลงฟีเจอร์ใหม่กลับไปเป็นฟีเจอร์ดั้งเดิมโดย PCA อาจทำให้ค่าที่ได้คลาดเคลื่อนออกไปบ้าง

การคัดเลือกฟีเจอร์ถูกใช้สำหรับค้นหาฟีเจอร์ที่มีความสำคัญเพื่อเก็บไว้ในเซตข้อมูล พร้อมทั้งกำจัดฟีเจอร์ที่ไร้ประโยชน์ออกจากเซตข้อมูล มีจุดประสงค์เพื่อให้การจำแนกแบบมีประสิทธิภาพมากยิ่งขึ้น อันได้แก่ ตัวแบบที่ได้จะมีความแม่นยำในการทำนายสูงขึ้น นอกจากนี้ ด้วยขนาดของเซตข้อมูลที่ลดลง ส่งผลให้เวลาที่ใช้ในการประมวลผลเร็วขึ้นด้วย และยังใช้หน่วยความจำน้อยลง รวมถึงการนำเสนอข้อมูลทำได้อย่างเรียบง่ายกว่าเดิม

เนื้อหาในบทต่อไปจะกล่าวถึงตัวจำแนกแบบประเภทแรก ซึ่งเป็นตัวจำแนกแบบพื้นฐานที่มีความเรียบง่ายมากที่สุด สามารถทำความเข้าใจแนวคิดได้ไม่ยาก ตัวจำแนกแบบชนิดนี้คือขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดซึ่งทำนายแบบ โดยใช้หน่วยวัดความใกล้เคียงในการเปรียบเทียบแบบที่ต้องการทำนายกับแบบที่มีอยู่แล้วในเซตข้อมูล กล่าวคือแบบที่ต้องการทำนายอยู่ใกล้แบบใดมากที่สุดจะทำนายให้อยู่ในชั้นข้อมูลเดียวกันกับแบบนั้น

แบบฝึกหัด

1. จงเสนอแนวคิดในการเติมค่าที่ขาดหายไปของเซตข้อมูลที่ไม่สมบูรณ์

2. จงเสนอแนวคิดในการตรวจจับข้อมูลรบกวนในเซตข้อมูล

3. จงแสดงให้เห็นว่าระยะทางยุคลิดยกกำลังสองเป็น ฟังก์ชันเมตริก หรือ ฟังก์ชันนอนเมตริก

4. จงคำนวณระยะทางเพื่อนบ้านร่วมกันระหว่าง D และ E โดยกำหนดเซตของแบบ ดังต่อไปนี้

   A = (1, 1); B = (2, 2);
   C = (2, 1); D = (3, 1)
   E = (4, 4); F = (5, 4);
   G = (6, 4); H = (6, 5)

5. จงคำนวณระยะทางมัธยฐานเคในข้อ 3 กำหนดให้ค่า k = 1 ระยะทางที่ได้มีความแตกต่างกันหรือไม่

6. จากเซตของแบบด้านล่างนี้ จงแสดงการจัดกลุ่มข้อมูลด้วยวิธีการเคมีนส์โดยใช้ระยะทางแมนฮัตตัน กำหนดให้ k = 4 และ จุดศูนย์กลางกลุ่มข้อมูลเริ่มต้น ได้แก่ แบบที่ขีดเส้นใต้ไว้

   (1, 3); <u>(3, 4)</u>;
   (3, 6); <u>(3, 8)</u>;
   (4, 5); (4, 7);
   <u>(5, 1)</u>; (5, 5);
   (7, 3); <u>(7, 5)</u>;
   (8, 5); (9, 4)

7. จากตัวอย่างที่ 8 ถ้าเปลี่ยนจุดศูนย์กลางกลุ่มข้อมูลเริ่มต้น เป็น A, B, C กลุ่มข้อมูลที่ได้ยังคงเหมือนเดิมหรือไม่ ถ้าไม่เหมือนเดิม กลุ่มข้อมูลชุดไหนมีคุณภาพที่ดีกว่ากัน

8. จงหา เซนทรอยด์ และ มีดอยส์ สำหรับเซตของแบบด้านล่างนี้ แล้ววิเคราะห์ว่าควรใช้จุดใดเป็นตัวแทนของเซตดังกล่าว

   (1, 1); (1, 3);
   (1, 4); (2, 2);
   (2, 3); (3, 1);
   (3, 4); (4, 2)

9. จงแสดงวิธีการแก้สมการเพื่อคำนวณหาค่า e_1 และ e_2 ในตัวอย่างที่ 12

10. จงแสดงวิธีการคำนวณเพื่อหาค่า y_2, y_3, y_4 ในตัวอย่างที่ 12

11. จงแสดงวิธีการหาค่า x_2, x_3, x_4 จากการแปลงข้อมูลจาก y_2, y_3, y_4 ในตัวอย่างที่ 2 ให้สังเกตค่าที่ได้ว่ามีค่าเท่ากันหรือไม่

บทที่ 3
ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุด

จากเนื้อหาในบทที่แล้วที่ได้อธิบายถึงหน่วยวัดความใกล้เคียง ซึ่งเป็นเครื่องมือที่ใช้สำหรับวัดค่าระยะทางระหว่างแบบสองแบบว่าอยู่ห่างกันเท่าใด ในบทนี้จะมีการประยุกต์ใช้หน่วยวัดดังกล่าวสำหรับการจำแนกแบบว่าแบบทดสอบควรอยู่ในชั้นข้อมูลใด กล่าวคือแบบที่อยู่ในชั้นข้อมูลเดียวกันควรอยู่ใกล้เคียงกัน ในขณะที่แบบที่อยู่ต่างชั้นข้อมูลกันควรอยู่ห่างไกลกัน ดังนั้นจากแนวคิดดังกล่าว แบบที่ต้องการทำนายควรมีชั้นข้อมูลเดียวกันกับแบบเดี่ยวหรือกลุ่มของแบบที่อยู่ใกล้ที่สุด เนื้อหาในบทนี้กล่าวถึง ตัวจำแนกประเภทต่าง ๆ ที่ประยุกต์ใช้กฎเพื่อนบ้านใกล้ที่สุด ได้แก่ ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุด ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเคตัว ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเคตัวที่ถูกดัดแปร เพื่อนบ้านใกล้รัศมีอาร์ และ ตัวจำแนกระยะทางสั้นที่สุด

วัตถุประสงค์การเรียนรู้

- ทราบแนวคิดกฎเพื่อนบ้านใกล้ที่สุดได้
- ประยุกต์ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดกับการจำแนกแบบได้
- แยกความแตกต่างระหว่างเพื่อนบ้านใกล้ที่สุดแต่ละขั้นตอนวิธีได้
- เข้าใจตัวจำแนกระยะทางสั้นที่สุดร่วมกับการกำหนดสาระสำคัญของเซตข้อมูลได้

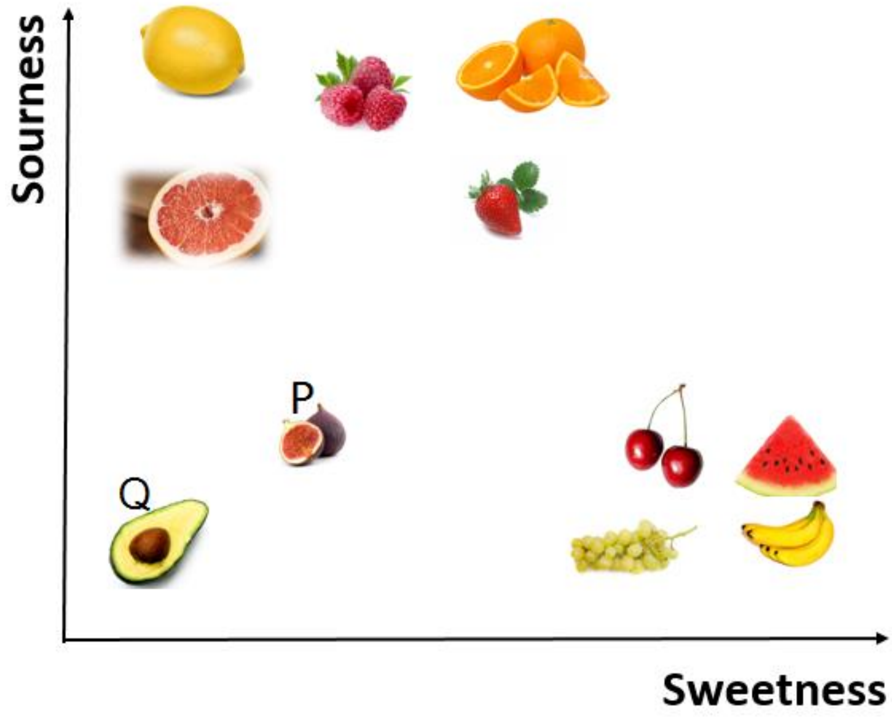
1. ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุด

กฎเพื่อนบ้านใกล้ที่สุด (Nearest Neighbour Rule) เป็นวิธีการตัดสินใจพื้นฐานสำหรับนำไปใช้ในการจำแนกประเภทแบบจากการพิจารณาแบบที่อยู่ใกล้ที่สุด คำอธิบายของกฎดังกล่าวแสดงตามข้อความด้านล่างนี้

“เพื่อนบ้านใกล้ที่สุดของแบบทดสอบที่กำลังพิจารณา คือแบบฝึกฝนในเซตข้อมูลเดียวกัน ที่เมื่อวัดจากหน่วยวัดความใกล้ระหว่างแบบทั้งสองแล้ว ให้ค่าระยะทางที่สั้นที่สุด”

ตัวอย่างของการประยุกต์ใช้กฎเพื่อนบ้านใกล้ที่สุดแสดงตามรูปที่ 3.1 กราฟดังกล่าวแสดงปริมาณ ความหวาน (Sweetness) และ ความเปรี้ยว (Sourness) ของผลไม้ชนิดต่าง ๆ บนแกน X และ Y ตามลำดับ จากรูป ผลไม้ที่มีความหวานมากได้แก่ เชอรี่ แตงโม องุ่น กล้วย สังเกตได้ว่า ผลไม้กลุ่มนี้กระจายตัวอยู่ในบริเวณเดียวกัน ในขณะที่ผลไม้ที่มีความเปรี้ยวมากได้แก่ เลมอน ราสเบอรี่ ส้ม เกรฟฟรุ๊ต สตรอเบอรี่

ซึ่งกระจายอยู่ตัวใกล้กัน ผลไม้ที่มีรสจืดได้แก่อโวคาโด (Q) เพียงชนิดเดียว โดยทั่วไปแล้วแบบที่มีความคล้ายกันย่อมอยู่ใกล้เคียงกัน ยกตัวอย่างเช่น ระยะทางระว่างเชอรี่และแตงโมสั้นกว่าระยะทางระว่างเชอรี่และเลมอน เพราะว่าเชอรี่และแตงโมมีความหวานจัดเหมือนกัน สมมติว่ามีผลไม้ชนิดหนึ่งซึ่งไม่ทราบว่าหวานจัด เปรี้ยวจัด หรือจืด ได้แก่ ผลไม้ P จึงทำการวัดค่าระยะทางโดยใช้หน่วยวัดความใกล้เคียง พบว่า P อยู่ใกล้อโวคาโด (Q) มากที่สุด ดังนั้นจึงทำการจำแนกแบบได้ว่า ผลไม้ P ควรมีรสจืดเหมือน Q เป็นต้น



รูปที่ 3.1 ตัวอย่างกฎเพื่อนบ้านใกล้ที่สุด

สำหรับโปรแกรมประยุกต์ที่บรรจุเซตข้อมูลขนาดใหญ่มาก ซึ่งประกอบไปด้วยแบบจำนวนมหาศาล การประยุกต์ใช้กฎเพื่อนบ้านใกล้ที่สุดส่งผลให้ใช้เวลาในการจำแนกแบบยาวนานกว่าตัวจำแนกแบบประเภทอื่น ๆ เนื่องจากจำเป็นต้องมีการคำนวณระยะทางทั้งหมดจากแบบที่ต้องการทำนายไปยังแบบทดสอบแต่ละแบบในเซตข้อมูลนั่นเอง

ตัวจำแนกที่ประยุกต์ใช้กฎเพื่อนบ้านใกล้ที่สุดมีแนวคิดพื้นฐานที่สุดได้แก่ ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุด (NN: Nearest Neighbour Algorithm) ซึ่งกำหนดให้แบบใหม่จะมีป้ายชื่อชั้นข้อมูลเดียวกับแบบฝึกฝนที่อยู่ใกล้ที่สุดเพียงแบบเดียวเท่านั้น ตัวอย่างการคำนวณของตัวแบบชนิดนี้แสดงตามตัวอย่างที่ 1 ด้านล่างนี้

ตัวอย่างที่ 1 กำหนดเซตฝึกหัดของแบบจำนวน 18 แบบ ด้านล่างนี้

X_1 = (0.8, 0.8, 1)	X_2 = (1.0, 1.0, 1)
X_3 = (1.2, 0.8, 1)	X_4 = (0.8, 1.2, 1)
X_5 = (1.2, 1.2, 1)	X_6 = (4.0, 3.0, 2)
X_7 = (3.8, 2.8, 2)	X_8 = (4.2, 2.8, 2)
X_9 = (3.8, 3.2, 2)	X_{10} = (4.2, 3.2, 2)
X_{11} = (4.4, 2.8, 2)	X_{12} = (4.4, 3.2, 2)
X_{13} = (3.2, 0.4, 3)	X_{14} = (3.2, 0.7, 3)
X_{15} = (3.8, 0.5, 3)	X_{16} = (3.5, 1.0, 3)
X_{17} = (4.0, 1.0, 3)	X_{18} = (4.0, 0.7, 3)

แบบแต่ละแบบแสดงได้ด้วยเวกเตอร์ X_i = (a, b, c) ตัวแปรแต่ละตัวสามารถอธิบายได้ ดังต่อไปนี้

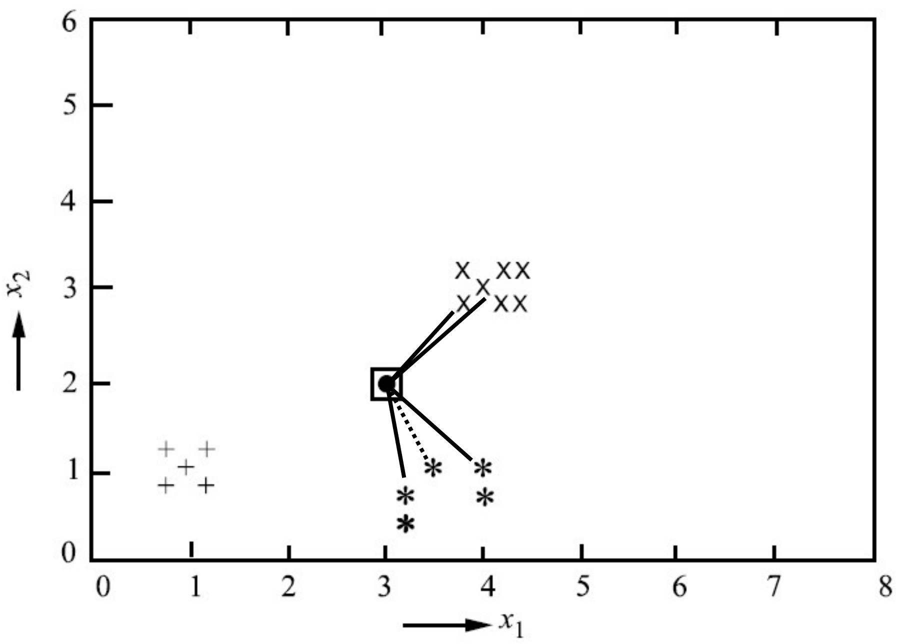
- X_i คือ แบบลำดับที่ i
- a คือ ตำแหน่งในระนาบ x_1
- b คือ ตำแหน่งในระนาบ x_2
- c คือ ป้ายชื่อชั้นข้อมูล หรือ ชั้นข้อมูล

จากรูปที่ 3.1 ชั้นข้อมูล 1 แสดงด้วยเครื่องหมาย “+” ชั้นข้อมูล 2 แสดงด้วยเครื่องหมาย “X” และ ชั้นข้อมูล 3 แสดงด้วยเครื่องหมาย “*” ถ้ากำหนดแบบทดสอบ แสดงด้วยสัญลักษณ์สี่เหลี่ยมจัตุรัสภายในมีวงกลมทึบสีดำ P = (3.0, 2.0) ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุด มีการทำงานตามลำดับ ดังต่อไปนี้

1. คำนวณระยะทางทั้งหมดจากแบบ P ไปยังแบบ X_1, X_2, X_3, …, X_{18} จำนวนระยะทางทั้งหมดจะเท่ากับจำนวนแบบฝึกฝนทั้งหมดในเซตข้อมูล
2. จากระยะทางทั้งหมดจำนวน 18 ค่า ทำการเปรียบเทียบว่า ระยะทางใดมีค่าต่ำที่สุด ปรากฏว่า ระยะทางระหว่าง P และ X_{16} มีค่าน้อยที่สุด (แสดงด้วยเส้นประ) สัญลักษณ์ d^2 จากสมการด้านล่างนี้แทนฟังก์ชันระยะทางยุคลิด

$$d^2(P, X_{16}) = \sqrt{(3.5-3.0)^2 + (1.0-2.0)^2} = 1.12$$

3. กำหนดให้แบบ P มีชั้นข้อมูลเดียวกับแบบที่อยู่ใกล้ที่สุด (X_{16}) ได้แก่ชั้นข้อมูลที่ 3 (*)



รูปที่ 3.2 เซตข้อมูลตัวอย่าง และ P = (3.0, 2.0)

2. ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเคตัว

ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเคตัว (kNN: k-Nearest Neighbour Algorithm) มีลักษณะคล้าย NN แต่ kNN พิจารณาแบบใกล้เคียงเป็นกลุ่มของแบบจำนวนทั้งหมด k แบบ แทนการหาแบบที่อยู่ใกล้ที่สุดเพียงแบบเดียวเหมือน NN กระบวนการจำแนกแบบใช้ป้ายชื่อชั้นข้อมูลของแบบส่วนใหญ่ที่อยู่ใกล้ที่สุดเป็นตัวตัดสิน ข้อดีของ kNN คือมีความแม่นยำโดยรวมมากกว่า NN และสามารถรับมือกับแบบที่เป็นสัญญาณรบกวนได้ดีถ้าสัญญาณรบกวนนั้นอยู่ใกล้แบบที่ต้องการทำนายมากที่สุด กรณีนี้ถ้าใช้ NN เป็นตัวทำนาย ผลการทำนายแบบจะเป็นชั้นข้อมูลเดียวกับสัญญาณรบกวนซึ่งไม่ถูกต้อง

ในเซตข้อมูลที่มีขนาดใหญ่ ความแม่นยำของ kNN ขึ้นอยู่กับการเลือกค่า k ที่เหมาะสม โดยสามารถกำหนดค่า k ให้มีค่าสูงขึ้นได้ เพื่อลดความผิดพลาด แต่จะใช้เวลาในการประมวลผลนานขึ้น ค่า k ที่เหมาะสม สามารถหาได้จากการทดลองในการวัดความแม่นยำในเซตทดสอบหลายครั้งด้วยค่า k ที่แตกต่างกัน แล้วเลือกค่า k ที่ให้ค่าความแม่นยำสูงสุดมาใช้ โดยมาก k ถูกกำหนดค่าให้เป็นจำนวนคี่ เพื่อหลีกเลี่ยงการเท่ากันของแบบที่อยู่ใกล้ที่สุดที่มาจากต่างชั้นข้อมูลกัน

ตัวอย่างการคำนวณของตัวแบบชนิดนี้แสดงตามตัวอย่างที่ 2 ถึง 4 ด้านล่างนี้ โดยทั่วไปผลการจำแนกแบบระหว่าง NN และ kNN อาจเหมือนหรือต่างกันก็ได้ ซึ่งจะได้เห็นการเปรียบเทียบระหว่างตัวจำแนกแบบทั้งสองจากตัวอย่างดังกล่าว

ตัวอย่างที่ 2 จากรูปที่ 3.1 กำหนดให้ k = 5 จะได้ว่า เพื่อนบ้านใกล้ที่สุดจำนวน 5 แบบ (แสดงด้วยเส้นทั้งหมด) X_{16}, X_7, X_{14}, X_6 และ X_{17} โดยที่แต่ละแบบอยู่ในชั้นข้อมูลต่าง ๆ ดังต่อไปนี้

X_6 อยู่ในชั้นข้อมูลที่ 2

X_7 อยู่ในชั้นข้อมูลที่ 2

X_{14} อยู่ในชั้นข้อมูลที่ 3

X_{16} อยู่ในชั้นข้อมูลที่ 3 และเป็นแบบที่อยู่ใกล้ P มากที่สุด

X_{17} อยู่ในชั้นข้อมูลที่ 3

จากการตัดสินโดยใช้เสียงข้างมาก kNN ทำนายแบบ P อยู่ในชั้นข้อมูลที่ 3 (*) เนื่องจากแบบเพื่อนบ้านใกล้ที่สุดจำนวน 5 แบบ มาจากชั้นข้อมูลที่ 3 (*) จำนวน 3 แบบ และ ชั้นข้อมูลที่ 2 (x) จำนวน 2 แบบ ในตัวอย่างนี้ NN และ kNN ทำนายแบบโดยให้ผลลัพธ์เหมือนกัน

ตัวอย่างที่ 3 จากเซตข้อมูลในตัวอย่างที่ 1 ถ้ากำหนดให้แบบ P คือ (4.2, 1.8) แสดงตามรูปที่ 3.2 และกำหนดให้ค่า k เท่ากับ 5 แล้ว จะได้แบบใกล้เคียงที่สุด ได้แก่ X_{17}, X_{16}, X_8, X_7, X_{11} โดยที่แต่ละแบบอยู่ในชั้นข้อมูลต่าง ๆ ดังต่อไปนี้

X_7 อยู่ในชั้นข้อมูลที่ 2

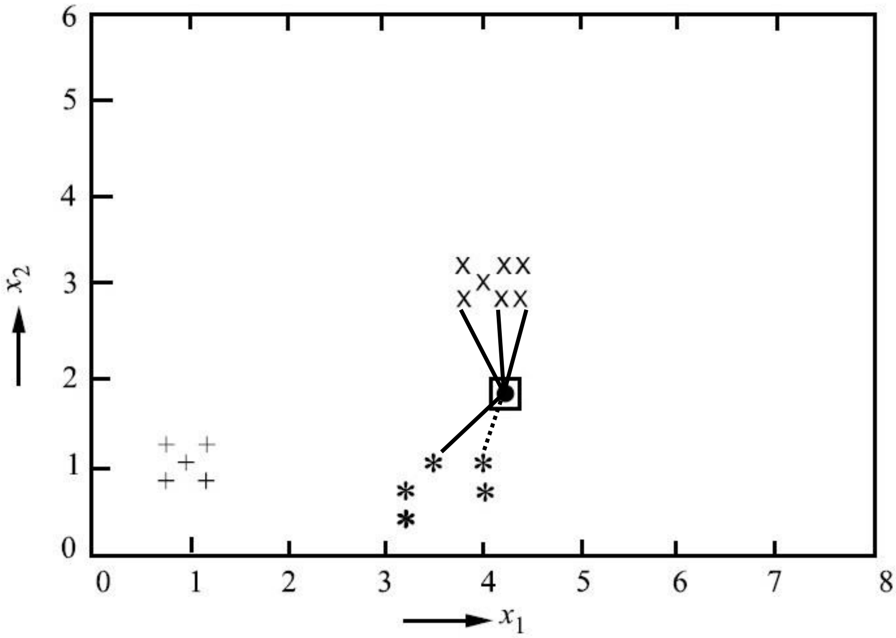
X_8 อยู่ในชั้นข้อมูลที่ 2

X_{11} อยู่ในชั้นข้อมูลที่ 2

X_{16} อยู่ในชั้นข้อมูลที่ 3

X_{17} อยู่ในชั้นข้อมูลที่ 3 และเป็นแบบที่อยู่ใกล้ P มากที่สุด

การทำนาย P โดย NN และ kNN ให้ผลลัพธ์ต่างกัน กล่าวคือ NN ทำนาย P อยู่ในชั้นข้อมูลที่ 3 (*) เนื่องจาก แบบที่ใกล้ที่สุด (แสดงด้วยเส้นประ) คือ X_{17} ซึ่งอยู่ในชั้นข้อมูลที่ 3 (*) ในขณะที่ kNN ทำนาย P อยู่ในชั้นข้อมูลที่ 2 (x) เนื่องจาก แบบใกล้เคียง 5 แบบ (แสดงด้วยเส้นประและเส้นทึบ) ได้แก่ X_{17}, X_{16} อยู่ในชั้นข้อมูลที่ 3 (*) และ X_8, X_7, X_{11} อยู่ในชั้นข้อมูลที่ 2 (x)



รูปที่ 3.3 เซตข้อมูลตัวอย่าง และ P = (4.2, 1.8)

ตัวอย่างที่ 4 จากรูปที่ 3.3 ถ้ากำหนดค่า k มีค่าเท่ากับ 5 แล้ว แบบที่ใกล้ที่สุดเคตัวได้แก่แบบที่ 5, 6, 7, 8, 9 และ 10 โดยที่แต่ละแบบอยู่ในชั้นข้อมูลต่าง ๆ ดังต่อไปนี้

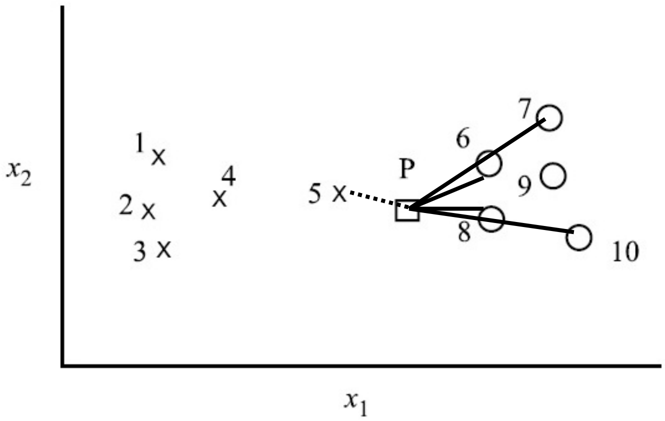
แบบที่ 5 อยู่ในชั้นข้อมูลที่ 1 (แทนด้วยสัญลักษณ์กากบาท)

แบบที่ 5 เป็นแบบที่อยู่ใกล้ P มากที่สุด

แบบที่ 6, 7, 8, 9 และ 10 อยู่ในชั้นข้อมูลที่ 2 (แทนด้วยสัญลักษณ์วงกลม)

สำหรับตัวอย่างนี้ แบบที่ 5 แสดงพฤติกรรมเป็นสัญญาณรบกวน เนื่องจากอยู่ไกลมากจากแบบส่วนใหญ่ (แบบที่ 1, 2, 3 และ 4) จากชั้นข้อมูลเดียวกัน (x) สำหรับกรณีนี้ NN จะทำนาย P ให้มีป้ายชื่อเดียวกับสัญญาณรบกวน คือชั้นข้อมูลที่ 1 (x)

แบบส่วนใหญ่ของ kNN ได้แก่แบบที่ 6, 7, 8, 9 และ 10 อยู่ในชั้นข้อมูลที่ 2 (o) จึงทำนายให้แบบ P มีป้ายชื่อข้อมูล คือชั้นข้อมูลที่ 2 (o) ดังนั้น สัญญาณรบกวนจึงไม่มีผลกับ kNN ตัวอย่างนี้แสดงให้เห็นว่าขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเคตัว สามารถรับมือกับเซตข้อมูลที่มีสัญญาณรบกวนได้กว่าขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุด



รูปที่ 3.4 แบบ P ที่ถูกจำแนกได้อย่างถูกต้องโดยใช้ kNN

3. ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเคตัวที่ถูกดัดแปร

ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเคตัวที่ถูกดัดแปร (MkNN: Modified k-Nearest Neighbour Algorithm) มีลักษณะคล้าย kNN โดยเริ่มต้นจากการหาแบบใกล้เคียงที่สุด k แบบก่อน แต่สิ่งที่แตกต่างกันคือ MkNN จะทำการคำนวณ น้ำหนัก (Weight) มาใช้ในการทำนายชั้นข้อมูลของแบบทดสอบ และน้ำหนักของเพื่อนบ้านแต่ละแบบ สามารถคำนวณได้ ดังนี้

$$w_j = \begin{cases} (d_k - d_j) / (d_k - d_1) & \text{if } d_k \neq d_1 \\ 1 & \text{if } d_k = d_1 \end{cases}$$

การหาค่าน้ำหนักเริ่มต้นจาก คำนวณระยะทางระหว่างแบบที่ต้องการทำนาย ไปยังแบบแต่ละแบบในกลุ่มเพื่อนบ้านใกล้ที่สุด จากขั้นตอนนี้จะได้ค่าระยะทางทั้งหมดจำนวน k ค่า จากนั้นทำการเรียงลำดับค่าระยะทางทั้งหมดจากน้อยสุดไปมากสุด ตัวแปรแต่ละตัวจากสูตร มีคำอธิบาย ดังต่อไปนี้

- d_j คือ ระยะทางระหว่างแบบที่ต้องการทำนายไปยังเพื่อนบ้านใกล้ที่สุดลำดับที่ j
- wj คือ ค่าน้ำหนักระหว่างแบบที่ต้องการทำนายไปยังเพื่อนบ้านใกล้ที่สุดลำดับที่ j
- j = 1, 2, 3, …, k
- d_1 คือ ระยะทางน้อยที่สุด ได้แก่ระยะทางระหว่างแบบที่ต้องการทำนายไปยังแบบที่ใกล้ที่สุด
- d_k คือ ระยะทางมากที่สุด ได้แก่ระยะทางระหว่างแบบที่ต้องการทำนายไปยังแบบที่ไกลที่สุด

จากการคำนวณค่าระยะทาง ค่าของ w_j มีค่าอยู่ระหว่าง 0 ถึง 1 โดยเริ่มจากค่ามากที่สุด 1 (สำหรับแบบที่ใกล้ที่สุดด้วยระยะทาง d_1) ลดลงไปจนถึง ค่าน้อยที่สุด 0 (สำหรับแบบที่ไกลที่สุดด้วยระยะทาง d_k) การจำแนกแบบโดยใช้ MkNN มีขั้นตอนวิธี ดังต่อไปนี้

1. คำนวณหาแบบเพื่อนบ้านใกล้ที่สุดจำนวน k แบบ
2. คำนวณค่าระยะทาง d_j และน้ำหนัก w_j ระหว่างแบบที่ต้องการทำนายและเพื่อนบ้านใกล้ที่สุดลำดับที่ j เมื่อ j = 1, 2, 3, …, k
3. หาค่าผลรวมของน้ำหนักจากแต่ละชั้นข้อมูลของเพื่อนบ้านทั้งหมดที่อยู่ในชั้นข้อมูลเดียวกัน
4. ผลรวมของน้ำหนักจากชั้นข้อมูลใดมีค่ามากที่สุด จำแนกแบบทดสอบเป็นชั้นข้อมูลนั้น

การทำนายชั้นข้อมูลที่ผลรวมน้ำหนักของแบบใกล้เคียงมีค่ามากที่สุด คือ ผลลัพธ์การทำนายของแบบทดสอบ ด้วยการใช้น้ำหนักเป็นตัวตัดสิน ส่งผลให้แบบที่เป็นสัญญาณรบกวนมีผลกระทบน้อยมากกับ MkNN ตัวอย่างการคำนวณของตัวแบบชนิดนี้แสดงตามตัวอย่างที่ 5 ด้านล่างนี้

ตัวอย่างที่ 5 จากรูปที่ 3.1 กำหนดให้ k = 5 จะได้ว่า เพื่อนบ้านใกล้ที่สุดเคตัวได้แก่ X_6, X_7, X_{14}, X_{16} และ X_{17} จากนั้นทำการคำนวณระยะทางระหว่างแบบ P ไปยังเพื่อนบ้านใกล้ที่สุด โดยเริ่มต้นจากแบบที่อยู่ใกล้ที่สุดไปจนถึงแบบที่อยู่ไกลที่สุด ได้ระยะทางทั้งหมดที่เรียงลำดับจากน้อยไปมาก ดังนี้

$d(P, X_{16}) = d_1 = 1.12$

$d(P, X_7) = d_2 = 1.13$

$d(P, X_{14}) = d_3 = 1.32$

$d(P, X_6) = d_4 = 1.41$

$d(P, X_{17}) = d_5 = 1.41$

ค่าน้ำหนักของแต่ละแบบ คำนวณได้ ดังนี้

$w_{16} = (1.41 – 1.12) / (1.41 – 1.12) = 1.00$

$w_7 = (1.41 – 1.13) / (1.41 – 1.12) = 0.97$

$w_{14} = (1.41 – 1.32) / (1.41 – 1.12) = 0.31$

$w_6 = (1.41 – 1.41) / (1.41 – 1.12) = 0.00$

$w_{17} = (1.41 – 1.41) / (1.41 – 1.12) = 0.00$

ผลรวมของน้ำหนักแต่ละชั้นข้อมูล คำนวณได้ ดังนี้

ชั้นข้อมูลที่ 1 มีผลรวมเป็น 0.00 เนื่องจากไม่มีเพื่อนบ้านแบบใดที่อยู่ในชั้นข้อมูลนี้
ชั้นข้อมูลที่ 2 มีผลรวมเป็น 0.97 ($w_7 + w_6$) จากเพื่อนบ้าน X_7 และ X_6
ชั้นข้อมูลที่ 3 มีผลรวมเป็น 1.31 ($w_{16} + w_{14} + w_{17}$) จากเพื่อนบ้าน X_{16}, X_{14} และ X_{17}

ดังนั้น MkNN ทำนายว่า P อยู่ในชั้นข้อมูลที่ 3 เนื่องจากมีผลรวมของค่าน้ำหนักสูงสุด สำหรับตัวอย่างนี้ kNN และ MkNN ให้ผลลัพธ์การทำนายที่เหมือนกัน ในส่วนเพิ่มเติม kNN และ MkNN อาจไม่จำเป็นต้องให้ผลการทำนายที่เหมือนกันก็ได้ขึ้นอยู่กับเซตข้อมูล

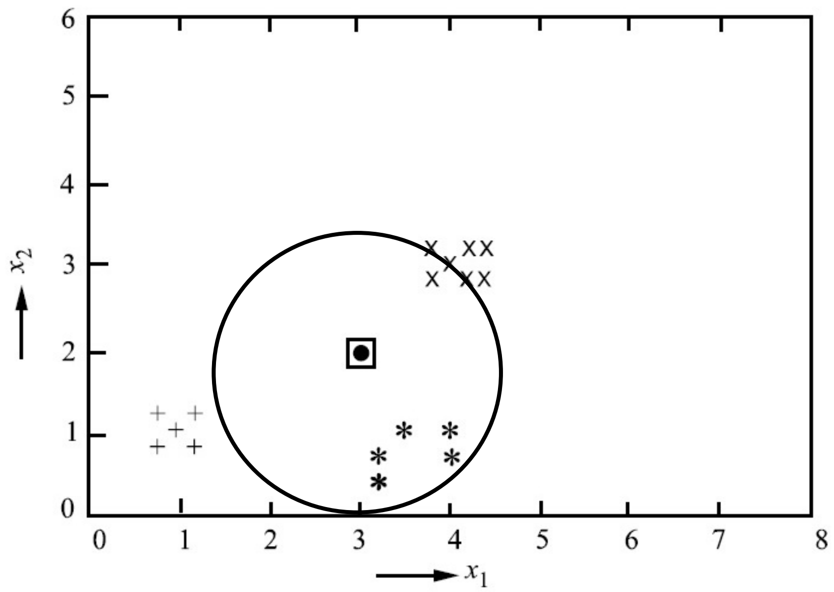
4. เพื่อนบ้านใกล้รัศมีอาร์

เพื่อนบ้านใกล้รัศมีอาร์ (rNN: r Near Neighbours) ใช้ป้ายชื่อชั้นข้อมูลของแบบส่วนใหญ่เป็นตัวตัดสินเช่นเดียวกับ kNN สิ่งที่แตกต่างกันคือ rNN พิจารณาแบบทั้งหมดในรัศมีอาร์โดยวัดจากแบบทดสอบ สำหรับกรณีที่รัศมีมีขนาดเล็กเกินไปจนไม่มีแบบใดอยู่ในอาณาเขต การตัดสินจะทำนายให้แบบทดสอบมีป้ายชื่อชั้นข้อมูลเดียวกับแบบฝึกฝนส่วนใหญ่ในเซตข้อมูล การจำแนกแบบโดยใช้ rNN มีขั้นตอนวิธี ดังต่อไปนี้

1. กำหนดค่า r ซึ่งแทนรัศมี
2. กำหนดพื้นที่ทรงกลม รัศมี r ที่มีแบบที่ต้องการทำนาย (P) เป็นเซนทรอยด์
3. ตรวจสอบดูว่ามีแบบใดบ้างอยู่ภายในพื้นที่ดังกล่าว
4. จากพื้นที่ดังกล่าว นับจำนวนแบบในแต่ละชั้นข้อมูลมีจำนวนเท่าไหร่
5. ถ้าชั้นข้อมูลใดมีแบบจำนวนมากที่สุด จำแนกแบบ P ให้มีชั้นข้อมูลดังกล่าว

ข้อเสียของตัวจำแนกชนิดนี้คือ ความยากในการเลือกค่า r ได้อย่างเหมาะสม ถ้าค่า r มีค่าต่ำเกินไปจนไม่มีแบบใด ๆ อยู่ภายในทรงกลม จะทำให้ตัว rNN ไม่สามารถตัดสินใจได้ ในทางกลับกัน ถ้าค่า r มีค่าสูงเกินไป แบบทั้งหมดจะถูกบรรจุอยู่ในทรงกลม ส่งผลให้ rNN ทำนายแบบจากชั้นมูลที่มีจำนวนแบบมากที่สุด ตัวอย่างการจำแนกแบบโดยใช้ rNN แสดงตาม ตัวอย่างที่ 6 ด้านล่างนี้

ตัวอย่างที่ 6 จากรูปที่ 3.1 กำหนดให้ r = 1.45 จะได้ว่า แบบทั้งหมดในรัศมีอาร์ ประกอบไปด้วย X_6, X_7, X_8, X_9 ซึ่งอยู่ในชั้นข้อมูลที่ 2 และ X_{14}, X_{16}, X_{17} ซึ่งอยู่ในชั้นข้อมูลที่ 3 ดังนั้น rNN ทำนายว่าแบบ P อยู่ในชั้นข้อมูลที่ 2 เพราะในชั้นข้อมูลนี้มีแบบอยู่ภายในพื้นที่ดังกล่าวมากที่สุด



รูปที่ 3.5 รัศมีอาร์

5. ตัวจำแนกระยะทางสั้นที่สุด

ตัวจำแนกระยะทางสั้นที่สุด (MDC: Minimal Distance Classifier) ตัดสินจากแบบที่ใกล้ที่สุด เช่นเดียวกับ NN สิ่งที่แตกต่างกันคือ MDC พิจารณาตัวแทนชั้นข้อมูลใกล้ที่สุด ได้แก่ เซนทรอยด์ (Centroid) ที่หาได้จากค่าเฉลี่ยทั้งหมดของกลุ่มตัวอย่าง สามารถคำนวณได้จากสูตรด้านล่างนี้

$$C_i = \frac{\sum_{j=1}^{N} X_{ij}}{N}$$

ตัวแปรแต่ละตัวจากสูตร มีคำอธิบาย ดังต่อไปนี้

- i คือ ลำดับที่ของชั้นข้อมูล
- $X_{i1}, X_{i2}, X_{i3}, \ldots, X_{iN}$ คือ แบบทดสอบจำนวน N แบบ ในชั้นข้อมูลที่ i
- C_i คือ เซนทรอยด์ในชั้นข้อมูลที่ i

ขั้นตอนวิธีของ ตัวจำแนกแบบ MDC แสดงตามลำดับขั้น ดังต่อไปนี้ และตัวอย่าง

1. คำนวณเซนทรอยด์ในแต่ละชั้นข้อมูล
2. วัดระยะทางจากตัวแบบที่ต้องการทำนาย (P) ไปยังเซนทรอยด์แต่ละตัว
3. เปรียบเทียบว่าเซนทรอยด์จากชั้นข้อมูลใด อยู่ใกล้แบบ P มากที่สุด
4. ถ้า C_k คือ เซนทรอยด์ที่ใกล้ที่สุดจากแบบทดสอบ P แล้ว MDC ทำนายว่า P อยู่ในชั้นข้อมูลที่ k

ตัวอย่างการทำงานแสดงตามตัวอย่างที่ 7 ด้านล่างนี้

ตัวอย่างที่ 7 จากเซตข้อมูลในตัวอย่างที่ 1 และ แบบทดสอบ P = (3.0, 2.0) เซตข้อมูลนี้ประกอบด้วย ชั้นข้อมูลที่ 1 จำนวน 5 แบบ ชั้นข้อมูลที่ 2 จำนวน 7 แบบ และ ชั้นข้อมูลที่ 3 จำนวน 6 แบบ คำนวณเซนทรอยด์แต่ละชั้นข้อมูล ดังนี้

แบบในชั้นข้อมูลที่ 1 ได้แก่ X_1, X_2, X_3, X_4, X_5 ทำการคำนวณเซนทรอยด์ ได้ดังนี้

C_1 = ((0.8 + 1.0 + 1.2 + 0.8 + 1.2) / 5, (0.8 + 1.0 + 0.8 +1.2 +1.2) / 5)

= (1.00, 1.00)

แบบในชั้นข้อมูลที่ 2 ได้แก่ X_6, X_7, X_8, X_9, X_{10}, X_{11}, X_{12} ทำการคำนวณเซนทรอยด์ ได้ดังนี้

C_2 = ((4.0 + 3.8 + 4.2 + 3.8 + 4.2 + 4.4 + 4.4) / 7, (3.0 + 2.8 + 2.8 + 3.2 + 3.2 + 2.8 + 3.2) / 7)

= (4.11, 3.00)

แบบในชั้นข้อมูลที่ 3 ได้แก่ X_{13}, X_{14}, X_{15}, X_{16}, X_{17}, X_{18}

C_3 = ((3.2 + 3.2 + 3.8 + 3.5 + 4.0 + 4.0) / 6, (0.4 + 0.7 + 0.5 + 1.0 + 1.0 + 0.7) /6)

= (3.62, 0.72)

ขณะนี้ มีเซนทรอยด์ 3 ตัว ซึ่งเป็นตัวแทนของแต่ละชั้นข้อมูล แสดงตามรูปที่ 3.5 จากนั้นคำนวณระยะทางยูคลิเดียนระหว่างแบบทดสอบและเซนทรอยด์ของแต่ละชั้นข้อมูล ดังนี้

$d^2(P, C_1) = 2.24$

$d^2(P, C_2) = 1.49$

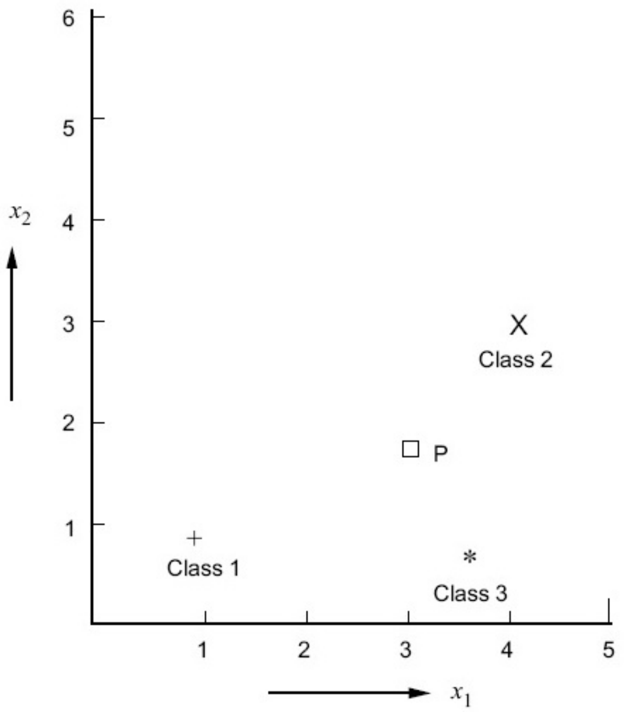
$d^2(P, C_3) = 1.42$

เนื่องจาก C_3 อยู่ใกล้กับ P มากที่สุด ดังนั้น MDC ทำนายว่าแบบ P อยู่ในชั้นข้อมูลที่ 3

สำหรับตัวอย่างนี้ MDC คำนวณระยะทางระหว่างแบบทดสอบและเซนทรอยด์ รวมทั้งหมด 3 ครั้ง ซึ่งเร็วกว่าเมื่อเปรียบเทียบกับ NN, kNN, rNN ที่คำนวณระยะทางระหว่างแบบทดสอบและแบบฝึกฝน รวมทั้งหมด 18 ครั้ง

จากตัวอย่างข้างต้น จุดเด่นของ MDC คือ ความซับซ้อนของเวลา (Time Complexity) กำหนดให้ n คือ จำนวนของแบบฝึกฝน m คือ จำนวนของแบบทดสอบ และ C คือ จำนวนของชั้นข้อมูล MDC ใช้เวลาเป็น O(n + mC) เมื่อ O(n) คือ เวลาที่ใช้ในการคำนวณเซนทรอยด์ และ O(mC) คือ เวลา

ที่ใช้ในการค้นหาเซนทรอยด์ใกล้ที่สุดของแบบทดสอบ ในขณะที่ NN, kNN, rNN ใช้เวลาเป็น O(mn) สำหรับโปรแกรมประยุกต์ขนาดใหญ่แล้ว C มีค่าน้อยกว่า n มาก ๆ ดังนั้น MDC จึงใช้เวลาในการทำนายแบบเร็วกว่า ในส่วนเพิ่มเติม ผลลัพธ์ของ MDC จะให้ผลการจำแนกกลุ่มได้ดีเมื่อเซตข้อมูลมี การแจกแจงแบบปกติ (Normally Distributed)



รูปที่ 3.6 การจำแนกแบบด้วย MDC

6. ตัวอย่างเพิ่มเติม

สำหรับหัวข้อนี้ แสดงโจทย์ตัวอย่างเพิ่มเติม สำหรับการคำนวณโดยใช้ตัวจำแนกชนิดต่าง ๆ ตามหัวข้อก่อนหน้า ตัวอย่างแต่ละตัวอย่างมีรายละเอียดการคำนวณ ดังต่อไปนี้

ตัวอย่างที่ 8 พิจารณาจุดใน 2 มิติ (แบบที่มี 2 สดมภ์) ด้านล่างนี้ โดยที่สดมภ์สุดท้ายเป็นตัวแทนของชั้นข้อมูล และกำหนดแบบทดสอบ P = (3.8, 3.1) จากนั้นทำการคำนวณค่าระยะทางยูคลิเดียนทั้งหมดระหว่างแบบฝึกฝนแต่ละแบบไปยังแบบ P แสดงได้ตามตารางที่ 3.1

(1, 1, 1); (1, 2, 1); (1, 3, 1); (2, 1, 1); (2, 2, 1); (2, 3, 1); (2, 3.5, 1);

(2.5, 2, 1); (3.5, 1, 1); (3.5, 2, 1); (3.5, 3, 2); (3.5, 4, 2); (4.5, 1, 2);

(4.5, 2, 2); (4.5, 3, 2); (5, 4, 2); (5, 5, 2); (6, 3, 2); (6, 4, 2); (6, 5, 2);

ตารางที่ 3.1 เซตข้อมูล 2 มิติ

ลำดับ	แบบ	ชั้นข้อมูล	ระยะทางไปยัง P
1	(1, 1)	1	3.50
2	(1, 2)	1	3.01
3	(1, 3)	1	2.80
4	(2, 1)	1	2.77
5	(2, 2)	1	2.11
6	(2, 3)	1	1.80
7	(2, 3.5)	1	1.84
8	(2.5, 2)	1	1.70
9	(3.5, 1)	1	2.12
10	(3.5, 2)	1	1.14
11	(3.5, 3)	2	0.32
12	(3.5, 4)	2	0.95
13	(4.5, 1)	2	2.21
14	(4.5, 2)	2	1.30
15	(4.5, 3)	2	0.71
16	(5, 4)	2	1.50
17	(5, 5)	2	2.25
18	(6, 3)	2	2.20
19	(6, 4)	2	2.38
20	(6, 5)	2	2.91

การคำนวณหาค่าระยะทางยูคลิเดียนตามตารางข้างต้น ไม่มีความจำเป็นต้องทำให้อยู่ในรูปปกติ เนื่องจากช่วงของค่าที่เป็นไปได้ทั้งหมดในแต่ละสดมภ์มีค่าไม่แตกต่างกันอย่างเห็นได้ชัด ในส่วนเพิ่มเติม การคำนวณในตารางนี้ใช้ทศนิยม 2 ตำแหน่งปัดเศษ เมื่อทศนิยมตำแหน่งถัดไปมีค่าตั้งแต่ 5 ขึ้นไป

ตัวอย่างที่ 8.1 ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเคตัว เมื่อกำหนดให้ k มีค่าเท่ากับ 3 ทำการค้นหาแบบที่อยู่ใกล้แบบ P ที่สุดจำนวน 3 แบบ ได้แก่

แบบที่ 11 = (3.5, 3, 2)

แบบที่ 12 = (3.5, 4, 2)

แบบที่ 15 = (4.5, 3, 2)

เนื่องจากเพื่อนบ้านใกล้ที่สุดทั้งหมดอยู่ในชั้นข้อมูลที่ 2 ดังนั้น NN จำแนก P อยู่ในชั้นข้อมูลที่ 2

ตัวอย่างที่ 8.2 ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเคตัวที่ถูกดัดแปร กำหนดให้ k มีค่าเท่ากับ 5 ทำการค้นหาแบบที่อยู่ใกล้แบบ P ที่สุดจำนวน 5 แบบ ได้แก่ แบบที่ 10, 11, 12, 14, 15 จากนั้นทำการคำนวณระยะทางจากแบบ P ไปยังเพื่อนบ้านใกล้ที่สุด 5 แบบ ได้ดังนี้

แบบที่ 10 มี d(P, (3.5, 2, 1)) = 1.14

แบบที่ 11 มี d(P, (3.5, 3, 2)) = <u>0.32</u> ซึ่งเป็นระยะทางที่สั้นที่สุด

แบบที่ 12 มี d(P, (3.5, 4, 2)) = 0.95

แบบที่ 14 มี d(P, (4.5, 2, 2)) = <u>1.30</u> ซึ่งเป็นระยะทางที่ยาวที่สุด

แบบที่ 15 มี d(P, (4.5, 3, 2)) = 0.71

ทำการคำนวณค่าน้ำหนักของแต่ละแบบจากเพื่อนบ้านใกล้ที่สุด 5 แบบ ได้ดังนี้

แบบที่ 10 มี $w_{(3.5,\ 2,\ 1)}$ = (1.30 – 1.14) / (1.30 – 0.32) = 0.13 / 0.98 $\approx$ 0.13

แบบที่ 11 มี $w_{(3.5,\ 3,\ 2)}$ = 1.00 เนื่องจากระยะทางระหว่างแบบนี้ไปยัง P สั้นที่สุด

แบบที่ 12 มี $w_{(3.5,\ 4,\ 2)}$ = (1.30 – 0.95) / (1.30 – 0.32) = 0.35 / 0.98 $\approx$ 0.36

แบบที่ 14 มี $w_{(4.5,\ 2,\ 2)}$ = 0.00 เนื่องจากระยะทางระหว่างแบบนี้ไปยัง P ยาวที่สุด

แบบที่ 15 มี $w_{(4.5,\ 3,\ 2)}$ = (1.30 – 0.71) / (1.30 – 0.32) = 0.59 / 0.98 $\approx$ 0.60

ทำการหาผลรวมค่าน้ำหนักจากแต่ละชั้นข้อมูล ได้ดังนี้

ชั้นข้อมูลที่ 1 มีผลรวมน้ำหนักเท่ากับ 0.13

ชั้นข้อมูลที่ 2 มีผลรวมน้ำหนักเท่ากับ 1.00 + 0.36 + 0.00 + 0.60 = 1.96

ชั้นข้อมูลที่ 3 มีผลรวมน้ำหนักเท่ากับ 0,00

เนื่องจากค่าน้ำหนักของชั้นข้อมูลที่ 2 มีค่าสูงสุด ดังนั้น MkNN จำแนก P อยู่ในชั้นข้อมูลที่ 2

ตัวอย่างที่ 8.3 เพื่อนบ้านใกล้จำนวนอาร์ตัว กำหนดให้ r มีค่าเท่ากับ ครึ่งหนึ่งของค่าที่มากที่สุดในสดมภ์ *"ระยะทางไปยัง P"* ดังนั้น r = 3.5 / 2 = 1.75 จากนั้นทำการตรวจสอบดูว่า มีแบบใดบ้างที่ระยะทางจากแบบนั้นไปยัง P มีค่าไม่เกิน 1.75 จะได้ว่าแบบทั้งหมดที่เป็นไปตามเงื่อนไขนี้ ได้แก่

แบบที่ 8 เนื่องจาก d(P, (2.5, 2, 1)) = 1.70

แบบที่ 10 เนื่องจาก d(P, (3.5, 2, 1)) = 1.14

แบบที่ 11 เนื่องจาก d(P, (3.5, 3, 2)) = 0.32

แบบที่ 12 เนื่องจาก d(P, (3.5, 4, 2)) = 0.95

แบบที่ 14 เนื่องจาก d(P, (4.5, 2, 2)) = 1.30

แบบที่ 15 เนื่องจาก d(P, (4.5, 3, 2)) = 0.71

แบบที่ 16 เนื่องจาก d(P, (5, 4, 2)) = 1.50

เนื่องจากแบบทั้งหมดที่อยู่ในทรงกลมรัศมี r เป็นแบบจากชั้นข้อมูลที่ 1 จำนวน 2 แบบ และ แบบจากชั้นข้อมูลที่ 2 จำนวน 5 แบบ ดังนั้น rNN จำแนก P อยู่ในชั้นข้อมูลที่ 2

ตัวอย่างที่ 8.4 ตัวจำแนกระยะทางสั้นที่สุด เริ่มต้นจากคำนวณหาเซนทรอยด์ในแต่ละชั้นข้อมูลจากค่าเฉลี่ยของแต่ละสดมภ์ของแบบทั้งหมด ได้ดังนี้

เซนทรอยด์ในชั้นข้อมูลที่ 1 คือ C_1 = (2.05, 2.05)

เซนทรอยด์ในชั้นข้อมูลที่ 2 คือ C_2 = (4.85, 3.4)

จากนั้นทำการคำนวณระยะทางของเซนทรอยด์ทั้งหมดไปยังแบบ P ได้ดังนี้

$d(P, C_1) = 2.04$

$d(P, C_2) = 1.09$

เนื่องจากระยะทางจากเซนทรอยด์ของชั้นข้อมูลที่ 2 ไปยังแบบ P มีค่าสั้นที่สุด ดังนั้น MDC จำแนก P อยู่ในชั้นข้อมูลที่ 2 สำหรับตัวอย่างนี้ ตัวแบบทุกประเภทจำแนกแบบโดยให้ผลลัพธ์ที่สอดคล้องกัน กล่าวคือทำนาย P อยู่ในชั้นข้อมูลที่ 2 เหมือนกันทั้งหมด

ตัวอย่างที่ 9 เซตข้อมูลไอริส (Iris Dataset) เป็นเซตข้อมูลจากเว็บไซต์ UCI Machine Learning Repository กลุ่มตัวอย่างของแบบจากเซตข้อมูลนี้แสดงตามตารางที่ 3.2 ด้านล่างนี้ รายละเอียดของแต่ละสดมภ์ มีความหมายดังต่อไปนี้

- Sepal Length คือ ความยาวกลีบเลี้ยง มีหน่วยเป็น เซนติเมตร (cm)
- Sepal Width คือ ความกว้างกลีบเลี้ยง มีหน่วยเป็น เซนติเมตร (cm)
- Petal Length คือ ความยาวกลีบดอก มีหน่วยเป็น เซนติเมตร (cm)
- Petal Width คือ ความกว้างกลีบดอก มีหน่วยเป็น เซนติเมตร (cm)
- Class คือ ชั้นข้อมูล ประกอบด้วย 3 สายพันธุ์ได้แก่
  - เซโตซ่า (Setosa)
  - เวอร์สิคัลเลอร์ (Versicolour)
  - เวอร์จินิก้า (Virginica)

ตารางที่ 3.2 เซตข้อมูลไอริส

No.	Sepal Length (cm)	Sepal Width (cm)	Petal Length (cm)	Petal Width (cm)	Class
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
3	4.7	3.2	1.3	0.2	Setosa
4	7.0	3.2	4.7	1.4	Versicolour
5	6.4	3.2	4.5	1.5	Versicolour
6	6.9	3.1	4.9	1.5	Versicolour
7	6.3	3.3	6.0	2.5	Virginica
8	5.8	2.7	5.1	1.9	Virginica
9	7.1	3.0	5.9	2.1	Virginica

สำหรับตัวอย่างนี้ การคำนวณหน่วยวัดความใกล้เคียงใช้ระยะทางแมนแฮตตัน ซึ่งสามารถกลับไปอ่านรายละเอียดการคำนวณได้จากบทก่อนหน้านี้ ระยะทาง (d) จากแบบ P ไปยังแบบแต่ละแบบในเซตข้อมูลไอริส แสดงได้ตามตารางที่ 3.3 ด้านล่างนี้

ตารางที่ 3.3 ระยะทางจากแบบ P ไปยังแบบไอริส

No.	d	w	Class
1	4.2	-	Setosa
2	3.9	0.00	Setosa
3	4.4	-	Setosa
4	3.1	0.53	Versicolour
5	2.4	1.00	Versicolour
6	3.2	0.47	Versicolour
7	4.9	-	Virginica
8	2.7	0.80	Virginica
9	4.9	-	Virginica

ตัวอย่างที่ 9.1 ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเคตัว เมื่อกำหนดให้ k มีค่าเท่ากับ 5 ทำการค้นหาแบบที่อยู่ใกล้แบบ P ที่สุดจำนวน 5 แบบ ได้แก่ แบบที่ 2, 4, 5, 6 และ 8 เนื่องจากแบบทั้งหมด มาจากชั้นข้อมูล Setosa จำนวน 1 แบบ มาจากชั้นข้อมูล Versicolour จำนวน 3 แบบ และมาจากชั้นข้อมูล Verginica จำนวน 1 แบบ ดังนั้น NN จำแนก P อยู่ในชั้นข้อมูล Versicolour

ตัวอย่างที่ 9.2 ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเคตัวที่ถูกดัดแปร กำหนดให้ k มีค่าเท่ากับ 5 ทำการคำนวณค่าน้ำหนักของแบบที่ 2, 4, 5, 6 และ 8 แสดงได้ตามสดมภ์ w ในตารางที่ 3.3 ในสดมภ์ดังกล่าวทำการคำนวณค่าน้ำหนักของแบบเพียง 5 แบบนี้เท่านั้น เนื่องจากแบบอื่นไม่มีผลต่อการตัดสินใจทำนายชั้นข้อมูล จากนั้นคำนวณค่าผลรวมน้ำหนักของแต่ละชั้นข้อมูล ได้ดังต่อไปนี้

น้ำหนักรวมของชั้นข้อมูล Setosa เท่ากับ 0.00

น้ำหนักรวมของชั้นข้อมูล Versicolour เท่ากับ 0.53 + 1.00 + 0.47 = 2.00

น้ำหนักรวมของชั้นข้อมูล Verginica เท่ากับ 0.80

เนื่องจากน้ำหนักรวมของชั้นข้อมูล Versicolour มีค่ามากที่สุด ดังนั้น MkNN จำแนก P อยู่ในชั้นข้อมูล Versicolour

ตัวอย่างที่ 9.3 เพื่อนบ้านใกล้จำนวนอาร์ตัว กำหนดให้ r = 4.5 สำรวจแบบที่ระยะทางจาก P มีค่าไม่เกิน r ได้แก่ แบบที่ 1, 2, 3, 4, 5, 6 และ 8 เนื่องจากแบบทั้งหมด มาจากชั้นข้อมูล Setosa จำนวน 3 แบบ มาจากชั้นข้อมูล Versicolour จำนวน 3 แบบ และมาจากชั้นข้อมูล Verginica จำนวน 1 แบบ จึงเกิดการเสมอกันขึ้น กล่าวคือจำนวนแบบของสองชั้นข้อมูลแรกมีค่าเท่ากัน สำหรับกรณีนี้ rNN สามารถจำแนก P อยู่ในชั้นข้อมูล Setosa หรือ Versicolour ก็ได้ ในทางปฏิบัติการทำนายควรให้คำตอบเพียงคำตอบเดียว จึงอาจทำการสุ่มเลือกชั้นข้อมูลที่เสมอกันก็ได้

ตัวอย่างที่ 9.4 ตัวจำแนกระยะทางสั้นที่สุด เริ่มต้นจากคำนวณหาเซนทรอยด์ในแต่ละชั้นข้อมูลจากค่าเฉลี่ยแต่ละสดมภ์ของแบบทั้งหมด ได้ดังนี้

เซนทรอยด์ในชั้นข้อมูล Setosa คือ C_{Setosa} = (4.9, 3.2, 1.4)

เซนทรอยด์ในชั้นข้อมูล Versicolour 2 คือ $C_{Versicolour}$ = (6.8, 3.2, 4.7)

เซนทรอยด์ในชั้นข้อมูล Verginica คือ $C_{Verginica}$ = (6.4, 3.0, 5.7)

จากนั้นทำการคำนวณระยะทางของเซนทรอยด์ทั้งหมดไปยังแบบ P ได้ดังนี้

$d(P, C_{Setosa}) = 4.2$

$d(P, C_{Versicolour}) = 2.9$

$d(P, C_{Verginica}) = 4.0$

เนื่องจากระยะทางจากเซนทรอยด์ของชั้นข้อมูล Versicolour ไปยังแบบ P มีค่าสั้นที่สุด ดังนั้น MDC จำแนก P อยู่ในชั้นข้อมูล Versicolour

สำหรับตัวอย่างนี้ มีตัวแบบบางประเภทที่จำแนกแบบโดยให้ผลลัพธ์ที่แตกต่างกัน ไม่เหมือนกับตัวอย่างก่อนหน้าที่ให้ผลลัพธ์เหมือนกันหมดทุกตัวแบบ กล่าวคือ ตัวจำแนกแบบทั้งหมดยกเว้น rNN ทำนาย P อยู่ในชั้นข้อมูล Versicolour แต่ rNN ทำนาย P อยู่ในชั้นข้อมูล Setosa หรือ Versicolour ก็ได้ ในโลกแห่งความเป็นจริง ตัวแบบทุกชนิดไม่จำเป็นต้องให้ผลการทำนายเหมือนกัน เหมือนเช่นตัวอย่างนี้เป็นต้น

7. การปรับปรุงประสิทธิภาพการค้นหาเพื่อนบ้านใกล้ที่สุด

สำหรับการจำแนกแบบแล้ว ถ้าเซตข้อมูลมีปริมาณของแบบทดสอบจำนวนมหาศาล ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดจะใช้เวลาประมวลผลยาวนาน ในการแก้ปัญหานี้ วิธีการค้นหาเพื่อนบ้านใกล้ที่สุดจึงถูกปรับปรุงประสิทธิภาพให้ใช้เวลาประมวลผลเร็วขึ้น เทคนิคต่าง ๆ ในหัวข้อนี้เกิดขึ้นในช่วงเวลา การประมวลผลก่อน (Pre-processing) และเป็นต้นทุนทางด้านเวลาที่เพิ่มขึ้น แต่เป็นต้นทุนที่เกิดขึ้นเพียงครั้งเดียว การประมวลผลก่อนนี้มีการประยุกต์ อสมการสามเหลี่ยม (Triangular Inequality) ร่วมกับระยะทางภายในชั้นข้อมูล (Intra-class distance) รายละเอียดต่าง ๆ กำลังจะกล่าวถึง ดังต่อไปนี้

7.1 ขั้นตอนวิธีแบบแตกกิ่งและจำกัดเขต

ขั้นตอนวิธีแบบแตกกิ่งและจำกัดเขต (Branch and Bound Algorithm) ทำการเก็บแบบลงในต้นไม้ (Tree) เพื่อหลีกเลี่ยงการเข้าถึงแบบทุกแบบตามลำดับในการหาเพื่อนบ้านใกล้ที่สุด การเก็บแบบลงในโครงสร้างข้อมูลนี้ทำให้ค้นหาเพื่อนบ้านใกล้ที่สุดได้อย่างมีประสิทธิภาพ โดยการค้นหาภายในกิ่งของต้นไม้เพียงกิ่งเดียว ไม่จำเป็นต้องค้นหาทุกกิ่ง ดังนั้นจึงลดเวลาในการค้นหาลงได้

ขั้นตอนวิธีเริ่มต้นด้วยการ จัดกลุ่มข้อมูล (Clustering) เพื่อแบ่งแบบออกเป็นกลุ่มข้อมูล S_j โดยในแต่ละกลุ่มข้อมูลมีการเก็บค่า เซนทรอยด์ μ_j และ รัศมี r_j (คำนวณจากระยะทางที่ยาวที่สุดระหว่างเซนทรอยด์และแบบในกลุ่มข้อมูล)

กำหนดให้ ระยะทางระหว่างเพื่อนบ้านใกล้ที่สุด X_j^* ในกลุ่มข้อมูล j ของแบบ P ได้แก่

$$d(P, X_j^*) = d$$

จากคุณสมบัติอสมการสามเหลี่ยม สามารถเขียนความสัมพันธ์ของแบบ $X_k \in S_j$ ได้ ดังนี้

$$d(P, \mu_j) \le d(P, X_k) + d(X_k, \mu_j) \le d(P, X_k) + r_j$$

ดังนั้น ขอบเขตล่าง b_j สามารถคำนวณได้ ดังนี้

$$\min_{X_k \in S_j} d(P, X_k) \ge d(P, \mu_j) - r_j = b_j$$

จากคุณสมบัติอสมการสามเหลี่ยม กลุ่มข้อมูลใดที่ $b_j \ge d$ จะถูกละเว้นในการค้นหาเพื่อนบ้านที่ใกล้ที่สุด

ถ้าแต่ละกลุ่มข้อมูลไม่มีแบบใดอยู่ร่วมกันเลย ขั้นตอนวิธีเวียนเกิด (Recursive Algorithm) ได้แก่

ขั้นตอนแรก

- จัดกลุ่มข้อมูลระดับแรกของแบบทดสอบทั้งหมดออกเป็น L เซต S_i เมื่อ i = 1 ถึง L
- คำนวณ เซนทรอยด์ μ_i และ รัศมี r_i ของแต่ละกลุ่มข้อมูล S_i
- จุดกลุ่มข้อมูลระดับที่สองของแบบในแต่ละกลุ่มข้อมูล ออกกลุ่มข้อมูลย่อย $S_{i,j}$ เมื่อ j = 1 ถึง L
- คำนวณ เซนทรอยด์ $\mu_{I,j}$ และ รัศมี $r_{i,j}$ ของแต่ละกลุ่มข้อมูลย่อย $S_{i,j}$
- เวียนเกิดการทำงานของขั้นตอนนี้ต่อไป จนกระทั่งแต่ละกลุ่มเหลือข้อมูลเพียงแบบเดียว

ขั้นตอนหลัง

เพื่อค้นหาเพื่อนบ้านใกล้ที่สุดของจุด P ดำเนินตามขั้นตอนย่อย ดังต่อไปนี้

1. **ขั้นตอนแตกกิ่ง**
   - คำนวณ b_j ของแต่ละกลุ่มข้อมูล เพื่อนบ้านใกล้ที่สุดจะอยู่ในกลุ่มข้อมูลที่มีค่านี้ต่ำสุด
   - ถ้ากลุ่มข้อมูลมีกลุ่มข้อมูลย่อย ให้เวียนเกิดขั้นตอนย่อยนี้ต่อไป
2. **ขั้นตอนจำกัดเขต**
   - ถ้ากลุ่มข้อมูลไม่เป็นไปตามเงื่อนไขของขอบเขต ให้ค้นหาในกลุ่มข้อมูลที่มีค่านี้สูงกว่า
   - นอกเหนือจากกรณีข้างต้น ให้หยุดการทำงาน แล้วส่งคืนเพื่อนบ้านใกล้ที่สุด

ประสิทธิภาพของขั้นตอนวิธีแบบแตกกิ่งและจำกัดเขตถือว่าเหนือกว่าขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดพื้นฐาน เนื่องจากสามารถค้นหาเพื่อนบ้านใกล้ที่สุดได้เร็วกว่านั่นเอง

ตัวอย่างที่ 10 พิจารณาแบบจาก ตัวอย่างที่ 1 ดังนี้

X_1 = (0.8, 0.8, 1)	X_2 = (1.0, 1.0, 1)
X_3 = (1.2, 0.8, 1)	X_4 = (0.8, 1.2, 1)
X_5 = (1.2, 1.2, 1)	X_6 = (4.0, 3.0, 2)
X_7 = (3.8, 2.8, 2)	X_8 = (4.2, 2.8, 2)
X_9 = (3.8, 3.2, 2)	X_{10} = (4.2, 3.2, 2)
X_{11} = (4.4, 2.8, 2)	X_{12} = (4.4, 3.2, 2)
X_{13} = (3.2, 0.4, 3)	X_{14} = (3.2, 0.7, 3)
X_{15} = (3.8, 0.5, 3)	X_{16} = (3.5, 1.0, 3)
X_{17} = (4.0, 1.0, 3)	X_{18} = (4.0, 0.7, 3)

การทำงานของขั้นตอนวิธีแบบแตกกิ่งและจำกัดเขตแสดงตามรูปที่ 3.7 เริ่มต้นจากจัดกลุ่มข้อมูลของแบบทั้งหมด ดังนี้

- ❖ **กลุ่มข้อมูลที่ 1** ประกอบไปด้วยแบบจากชั้นข้อมูลที่ 1 จากนั้นให้แบ่งกลุ่มข้อมูลอีกครั้ง จะได้กลุ่มข้อมูลย่อย 1a และ 1b
- ❖ **กลุ่มข้อมูลที่ 2** ประกอบไปด้วยแบบจากชั้นข้อมูลที่ 2 จากนั้นให้แบ่งกลุ่มข้อมูลอีกครั้ง จะได้กลุ่มข้อมูลย่อย 2a และ 2b
- ❖ **กลุ่มข้อมูลที่ 3** ประกอบไปด้วยแบบจากชั้นข้อมูลที่ 3 จากนั้นให้แบ่งกลุ่มข้อมูลอีกครั้ง จะได้กลุ่มข้อมูลย่อย 3a และ 3b

แบบแต่ละแบบจะถูกบรรจุให้อยู่ในกลุ่มข้อมูลย่อยจากการเวียนเกิดขั้นตอนแรก แสดงตามรูปที่ 3.8 จากนั้นคำนวณค่าเซนทรอยด์และรัศมีของแต่ละกลุ่มข้อมูล ได้ดังต่อไปนี้

- ❖ **กลุ่มข้อมูลที่ 1** เซนทรอยด์คือ (1.0, 1.0) และ รัศมีเท่ากับ 0.283
- ❖ **กลุ่มข้อมูลที่ 2** เซนทรอยด์คือ (4.11, 3.0) และ รัศมีเท่ากับ 0.369
- ❖ **กลุ่มข้อมูลที่ 3** เซนทรอยด์คือ (3.62, 0.72) และ รัศมีเท่ากับ 0.528

พิจารณาแบบ P = (3.0, 2.0) ในระดับแรก พิจารณาค่า b_j ของแต่ละกลุ่มข้อมูล ดังนี้

$$b_1 = d(P, \mu_1) - r_1 = 1.953$$

$$b_2 = d(P, \mu_2) - r_2 = 1.125$$

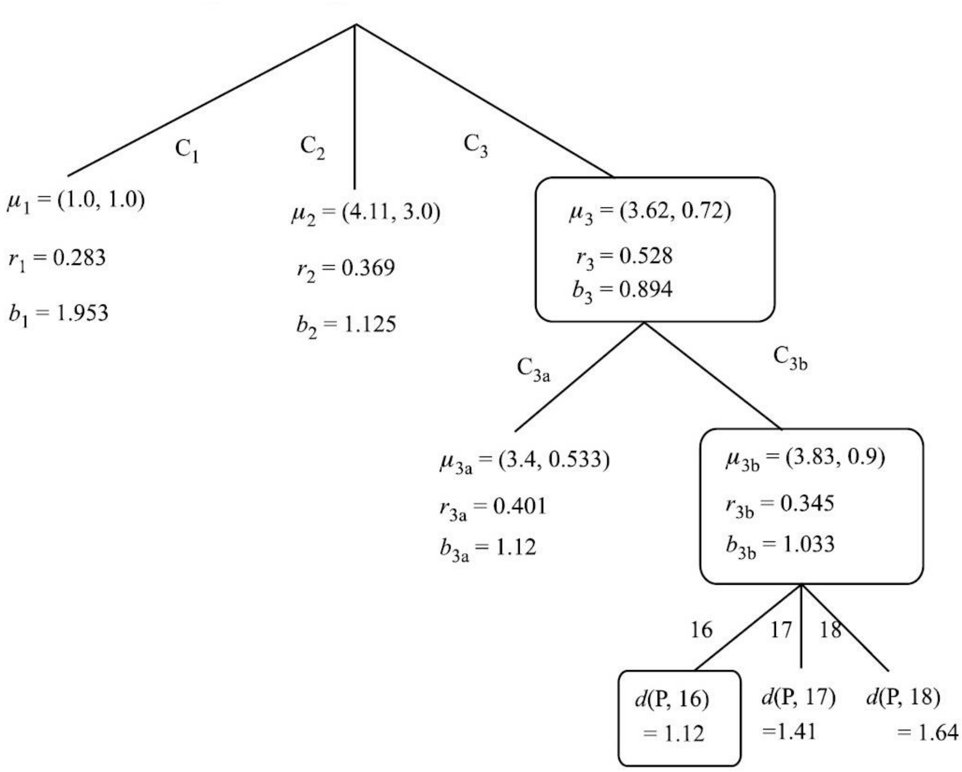
$$b_3 = d(P, \mu_3) - r_3 = 0.894$$

เนื่องจาก b_3 มีค่าน้อยที่สุด ดังนั้น กลุ่มข้อมูลที่ 3 ถูกพิจารณาให้ค้นหาต่อไป

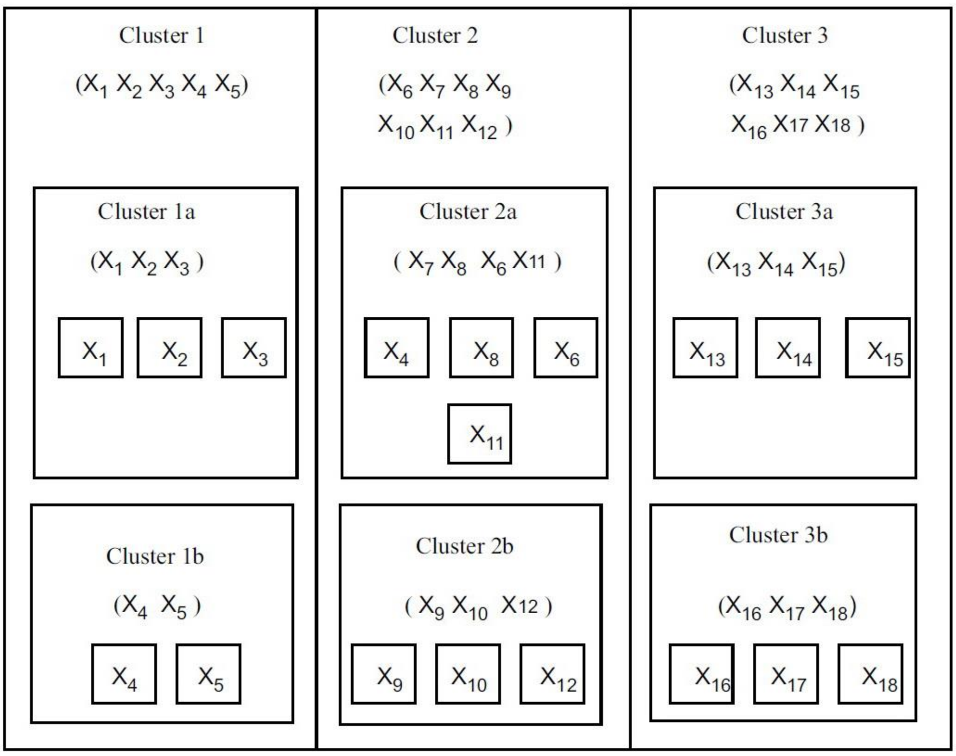
คำนวณค่าเซนทรอยด์และรัศมีของกลุ่มข้อมูลที่ 3 ได้ดังต่อไปนี้

- ❖ **กลุ่มข้อมูลที่ 3a** เซนทรอยด์คือ (3.4, 0.533) และ รัศมีเท่ากับ 0.401 ดังนั้น $b_{3a} = 1.12$
- ❖ **กลุ่มข้อมูลที่ 3b** เซนทรอยด์คือ (3.83, 0.9) และ รัศมีเท่ากับ 0.345 ดังนั้น $b_{3b} = 1.033$

กลุ่มข้อมูลที่ 3b ถูกค้นหาต่อไป และค้นพบว่า X_{16} อยู่ใกล้ P มากที่สุด



รูปที่ 3.7 ขั้นตอนวิธีแบบแตกกิ่งและจำกัดเขต



รูปที่ 3.8 การจัดกลุ่มข้อมูลและกลุ่มข้อมูลย่อย

ตัวอย่างที่ 10 (ต่อ)

ขอบเขต d ระหว่าง P และ X_{16} มีค่าเท่ากับ 1.12

เนื่องจาก b_1 และ b_2 มีค่ามากกว่า d ดังนั้น กลุ่มข้อมูลที่ 1 และ 2 ไม่จำเป็นต้องค้นหาต่อไป

จึงสรุปได้ว่า X_{16} คือเพื่อนบ้านใกล้ที่สุดของ P

7.2 ขั้นตอนวิธีลูกบาศก์

ขั้นตอนวิธีลูกบาศก์ (Cube Algorithm) ถูกใช้ระหว่างช่วงประมวลผลก่อนเพื่อ ฉายเงา (Projection) ของแบบลงบนแต่ละแกน ผลลัพธ์ที่ได้คือ ลูกบาสก์หลายมิติ (Hypercube) ขนาดของด้าน ℓ รูปที่ 3.9 แสดงสี่เหลี่ยมจัตุรัสในปริภูมิ 2 มิติ โดยที่ ℓ คือความยาวของแต่ละด้าน โดยมีเซนทรอยด์อยู่ที่ P = (p_1, p_2) และถูกฉายเงาลงบนแกนทั้ง 2 ด้าน จากการเพิ่มหรือลดค่า ℓ อย่างหมาะสมเป็นลำดับขั้น ส่งผลให้ลูกบาสก์หลายมิติครอบคลุมเพื่อนบ้านใกล้ที่สุดที่ต้องการค้นหา k แบบ ขั้นตอนวิธีบาศก์แสดง มีวิธีการดังต่อไปนี้

ขั้นตอนที่ 1

กำหนดค่าตัวแปรต่าง ๆ ดังต่อไปนี้

B_i คือ ตัวแปรชุดของจำนวนจริง เก็บค่าที่เรียงลำดับแล้ว n ค่า ของฟีเจอร์ i

K_i คือ ตัวแปรชุดของจำนวนเต็ม เก็บ ดัชนี (Index) ของแบบที่อยู่ใน $B_i(j)$

Z คือ ตัวแปรชุดของจำนวนเต็ม n ค่า แต่ละค่ากำหนดค่าเริ่มต้นเป็น 1

d คือ จำนวนมิติ (จำนวนฟีเจอร์)

n คือ จำนวนแบบทั้งหมด

i = 1, …, d

j = 1, …, n

ขั้นตอนที่ 2

กำหนดให้ $P = (p_1, \ldots, p_d)$ คือ เวกเตอร์ทดสอบ เมื่อ p_i แทนค่าของฟีเจอร์ลำดับที่ i

ขั้นตอนที่ 3

สำหรับแต่ละแกน i ให้ตั้งค่าตัวชี้ ptr_{i1} ไปยังตำแหน่งใน A_i ของสมาชิกตัวแรกสุด (a_F) เมื่อ $a_F \geq p_i - ½$

สำหรับแต่ละแกน i ให้ตั้งค่าตัวชี้ ptr_{i2} ไปยังตำแหน่งใน A_i ของสมาชิกตัวสุดท้าย (a_L) เมื่อ $a_L \leq p_i + ½$

ขั้นตอนที่ 4

สำหรับแต่ละแกน i เข้าถึง (Scan) ตัวแปรชุด K_i ตามลำดับ ตั้งแต่ $K_i(ptr_{i1})$ ถึง $K_i(ptr_{i2})$

สำหรับแต่ละตำแหน่งใน K_i ให้เลื่อนบิตไปทางซ้าย (Shift Left) 1 บิต ดังนั้น ค่าในช่องของ Z ชี้ไปยังตำแหน่งนี้

ในทางกลับกัน เมื่อนำค่าในแต่ละตำแหน่งของ K_i คูณด้วย 2 ทำให้ค่าในช่องของ Z ชี้ไปยังตำแหน่งนี้ได้เช่นกัน

ขั้นตอนที่ 5

นับตำแหน่งใน Z ที่มีค่า 2^d และให้ L เท่ากับค่าที่นับได้ ส่งผลให้ตัวนับนี้แทนจำนวนแบบที่ตกอยู่ในลูกบาสก์หลายมิติ

ขั้นตอนที่ 6

กำหนดให้ k แทน จำนวนเพื่อนบ้านใกล้ที่สุดที่ต้องการ

ถ้า L = k แล้วหยุดการทำงาน

ถ้า L < k แล้วเพิ่มค่า l อย่างเหมาะสม คืนค่า (Reset) ตัวชี้ ให้เข้าถึงและเลื่อนบิตค่าที่เพิ่งเพิ่มเข้าไปใหม่ในช่องของ K_i และไปยังขั้นตอนที่ 4

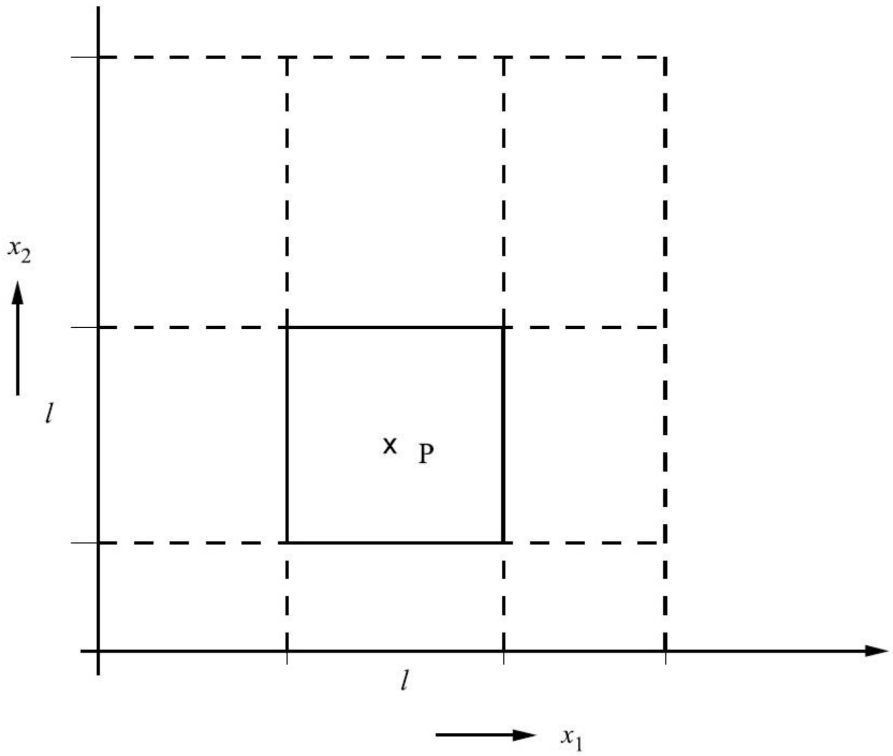
ถ้า L > k ลดค่า l อย่างเหมาะสม คืนค่าตัวชี้ ให้เข้าถึงและเลื่อนบิตค่าที่เพิ่งลบออกจากช่องของ K_i ไปทางขวา 1 บิต และไปยังขั้นตอนที่ 4

ตัวอย่างที่ 11 รูปที่ 3.10 แสดงเซตของแบบ 2 มิติ ได้แก่ แบบ 1, 2, 3, 4 , 5, 6 จากชั้นข้อมูล × และแบบ 7, 8, 9, 10, 11, 12 จากชั้นข้อมูลที่ O

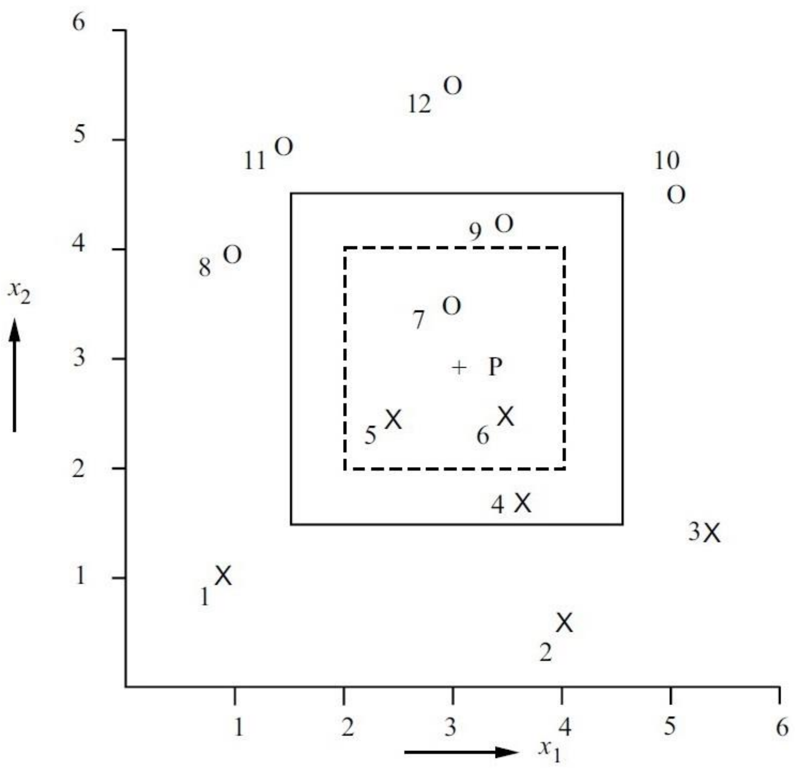
ถ้าต้องการหาเพื่อนบ้านใกล้ที่สุด k ตัว (k = 5) ของ P เมื่อ l ถูกให้ค่าเป็น 2 ซึ่งเป็นไปตามเงื่อนไข $2 \leq p_1 \leq 4$ และ $2 \leq p_2 \leq 4$ แล้วค้นพบว่ามีเพียง 3 แบบ ที่ตกอยู่ภายในสี่เหลี่ยมจัตุรัส

ถ้า l ถูกขยับค่าเพิ่มขึ้นอีก 0.5 ค้นพบว่ามี 5 แบบที่ตกอยู่ภายในสี่เหลี่ยมจัตุรัส จึงได้เพื่อนบ้านใกล้ที่สุดครบ 5 แบบ เนื่องจากป้ายชื่อชั้นข้อมูลส่วนใหญ่มาจากชั้นข้อมูล × จำนวน 3 แบบ โดยที่เหลือ 2 แบบ มาจากชั้นข้อมูล O ดังนั้น P ถูกจำแนกให้อยู่ในชั้นข้อมูล O

ขั้นตอนวิธีบาศก์ใช้พื้นที่เก็บข้อมูลสูงลิ่ว เนื่องจากในแต่ละแกนต้องการพื้นที่สำหรับเก็บค่าฟีเจอร์ที่เรียงลำดับแล้ว N ค่า และดัชนีของค่าเหล่านี้ รวมทั้งหมด 2dN เวิร์ด นอกจากนี้ เวลาในการประมวลผลของการเรียงลำดับตัวแปรชุดที่มีขนาด dN เวิร์ด จำเป็นต้องใช้ออเดอร์ dN log N สำหรับการเปรียบเทียบ



รูปที่ 3.9 การฉายเงาของบาสก์หลายมิติ



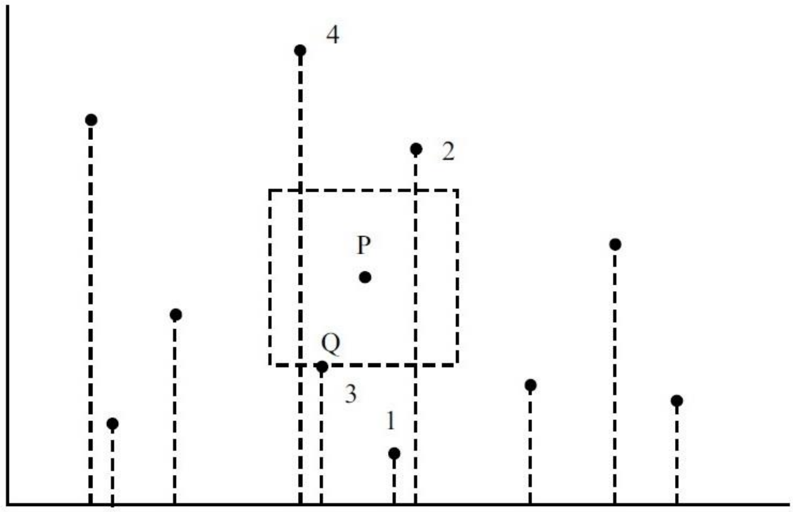
รูปที่ 3.10 การจัดกลุ่มโดยใช้ขั้นตอนวิธีบาสก์

7.3 ขั้นตอนวิธีการทอดเงา

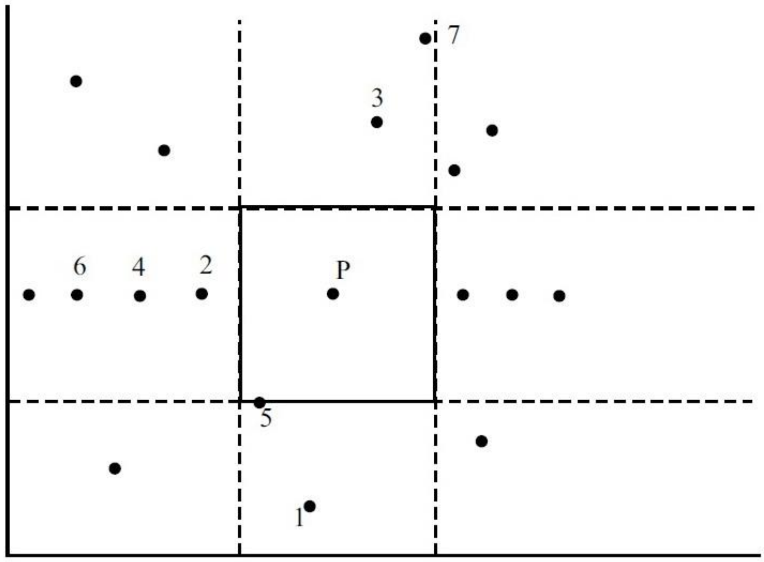
ขั้นตอนวิธีทอดเงา (Projection) หลากหลายวิธีถูกนำมาใช้ในการค้นหาเพื่อนบ้านที่ใกล้ที่สุด วิธีที่เรียบง่ายที่สุดคือการทอดเงาลงบนปริภูมิ 1 มิติ เนื่องจากถ้าแบบถูกเรียงลำดับเรียบร้อยแล้ว การหาเพื่อนบ้านใกล้ที่สุดของ P สามารถทำได้โดยค้นหา ค่าก่อน (Predecessor) และ ค่าหลัง (Successor) ของ P แบบที่ใกล้ P มากที่สุดจาก 2 แบบนี้คือเพื่อนบ้านใกล้ที่สุดที่ต้องการหา

ในปริภูมิ 2 มิติ แบบจะถูกเรียงลำดับจากน้อยไปตาม ตามพิกัดแรกของ พิกัดร่วม (Coordinate) จากค่าแรกนี้ สามารถหาเพื่อนบ้านใกล้ที่สุดของ P และเก็บเพื่อนบ้านนี้ (Q) ลงหน่วยความจำ ถ้าระยะทางตามแกน x จาก P ไปยังแบบอื่น มีค่ามากกว่าระยะทางจริงระหว่าง P และ Q แล้ว การค้นหาจะสิ้นสุดลง และ Q คือเพื่อนบ้านใกล้ที่สุดของ P กระบวนการที่กล่าวมานี้แสดงตามรูปที่ 3.11

การประมวลผลนี้ถูกปรับปรุงให้สามารถทอดเงาบนทั้งแกน x และ y แทนที่ทอดเงาบนแกน x เพียงแกนเดียว โดยดำเนินการบนแกนทั้งสองแบบคู่ขนานกัน กล่าวคือ ขั้นตอนแรกดำเนินการบนแกน x ขั้นตอนถัดไปดำเนินการบนแกน y และสลับแกนไปมา การค้นหาจะจบลง ถ้ามีแกนใดแกนหนึ่งที่ทำให้การค้นหาสิ้นสุดลงได้ รูปที่ 3.12 แสดงการค้นหาดังกล่าว



รูปที่ 3.11 การค้นหาเพื่อนบ้านใกล้ที่สุดจากการทอดเงาลงบนแกน x



รูปที่ 3.12 การค้นหาเพื่อนบ้านใกล้ที่สุดโดยทอดเงาลงบน 2 แกน

ตัวอย่างที่ 12 จากตัวอย่างในรูปที่ 3.2 ถ้ากำหนดให้แบบทดสอบคือ P = (3, 2)

จัดเรียงแบบทั้งหมดจากค่าตามแกน x จากน้อยไปมาก ได้แก่

X_{13}, X_{14}, X_{16}, X_7, X_9, X_{15}, X_6, X_{17}, X_{18}, X_8 ฯลฯ

ระยะทางตามแกน x ของแบบเหล่านี้ ไปยัง P ได้แก่

0.2, 0.2, 0.5, 0.8, 0.8, 1.0, 1.0, 1.0, 1.2. 1.2

ระยะทางจริงจาก X_{13}, X_{14}, X_{16}, X_7, X_9, X_{15}, X_6, X_{17}, X_{18} ไปยัง P ได้แก่

1.162, 1.315, 1.118, 1,131, 1.44, 1.7, 1.414, 1.414, 1.64

ในแต่ละแบบเหล่านี้ ให้เก็บแบบที่ใกล้ที่สุดไปยัง P ลงหน่วยความจำ

เริ่มต้นจาก X_{13} เนื่องจากเป็นแบบแรก จึงมีระยะทางน้อยที่สุดไปยัง P และถูกบันทึกลงหน่วยความจำ

เมื่อ X_{16} ถูกเข้าถึง ระยะทางสั้นที่สุดเปลี่ยนเป็น 1.118

เมื่อ X_{18} ถูกเข้าถึง แบบที่ใกล้ P มากที่สุดยังคงเป็น X_{16}

แบบต่อไป X_8 มีระยะทางตามแกน x คือ 1.2 ซึ่งมากกว่าระยะทางจริงของแบบที่ใกล้ที่สุด X_{16}

จึงไม่จำเป็นต้องตรวจสอบแบบต่อไปอีก และ X_{16} คือ แบบที่ใกล้ที่สุดไปยัง P

7.4 การตัดแบ่งแบบเรียงลำดับ

การตัดแบ่งแบบเรียงลำดับ (Ordered Partitions) ถูกใช้สำหรับลดต้นทุนการคำนวณเพื่อนบ้านใกล้ที่สุด k ตัว จากแบบฝึกฝน n แบบ ในปริภูมิ d มิติ แบบฝึกฝนถูกตัดแบ่งตามค่าพิกัดในแต่ละแกนออกเป็น บล็อก (Block) ความกว้างของแต่ละบล็อคสามารถปรับแต่งให้บรรจุแบบจำนวนเท่ากันได้ (หรือใกล้เคียงกันได้)

การค้นหาเพื่อนบ้านใกล้ที่สุดเริ่มต้นจากระดับ 0 ของต้นไม้ โดยที่แต่ละกิ่งของต้นไม้สามารถคำนวณ ระยะทางยกกำลังสอง (Square Distance) r_l^2 ซึ่งถูกนิยามบนความสัมพันธ์เวียนเกิด ดังสมการด้านล่างนี้

$$r_l^2 = \begin{cases} r_{l-1}^2 + min((x_l - a_l)^2, (x_l - b_l)^2), for\ l \neq d \\ r_{d-1}^2 + (x_d - a_d)^2, for\ l = d \end{cases}$$

ในแต่ละระดับ l ระยะทางยกกำลังสอง r_l^2 ถูกคำนวณจากแบบทดสอบ โหนดใดที่มีค่าระยะทางยกกำลังสองสั้นที่สุดจะถูกตัดสินให้ค้นหาเพื่อนบ้านใกล้ที่สุดของแบบทดสอบตามโหนดนี้ เมื่อใดก็ตามที่ โหนดปลายทาง (Terminal Node) ถูกเข้าถึง ระยะทางยกกำลังสองจะถูกบันทึกและระยะทางสั้นที่สุด d จะถูก

เก็บลงหน่วยความจำ อย่างไรก็ตาม โหนดในระดับใด ๆ ที่มีค่าระยะทางมากกว่า d จะถูกละเว้นในการค้นหา จากที่กล่าวมาทั้งหมด การตัดแบ่งแบบเรียงลำดับใช้เวลาประมวลผลเป็นค่าคงที่ (Constant) โดยไม่เกี่ยวข้องกับจำนวนของแบบทดสอบ

ตัวอย่างที่ 13 พิจารณาแบบ 3 มิติ ดังต่อไปนี้

X_1 = (1.0, 1.0, 1.0)	X_2 = (1.2, 2.0, 4.0)
X_3 = (2.0, 1.5, 3.0)	X_4 = (2.5, 3.0, 5.0)
X_5 = (3.0, 7.0, 6.0)	X_6 = (3.5, 2.5, 3.5)
X_7 = (4.0, 6.0, 2.5)	X_8 = (4.5, 5.5, 4.5)
X_9 = (5.0, 1.5, 2.0)	X_{10} = (5.5, 6.5, 1.5)
X_{11} = (6.0, 8.0, 7.5)	X_{12} = (7.0, 9.0, 8.0)

รูปที่ 3.13 แสดงการตัดแบ่งแบบตามมิติแรก ตามด้วยมิติที่สอง และมิติที่สาม ตามลำดับ

ถ้าแบบ P = (7.0, 7.0, 7.0) การค้นหาด้วยการตัดแบ่ง เริ่มต้นจากกิ่งล่างสุด ซึ่งแทนค่าระหว่าง 4.5 ถึง 7.0 ของมิติแรก

ในขั้นตอนถัดไป กิ่งที่สองตามมิติที่สองที่มีค่าอยู่ระหว่าง 6.5 ถึง ∞ ถูกพิจารณาเพื่อทำการค้นหาเพื่อนบ้านใกล้ที่สุดต่อไป

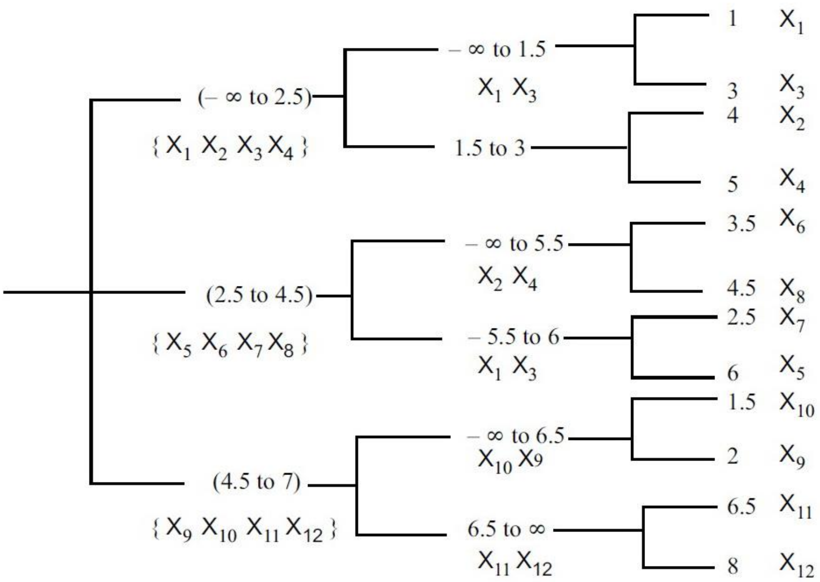
สุดท้าย X_{11} ถูกเลือก เนื่องจากเป็นแบบที่ใกล้ที่สุดของ P ซึ่งให้ระยะทางยกกำลังสองมีค่าเท่ากับ
$(7 - 7)^2 + (7 - 6.5)^2 + (7 - 6.5)^2 = 0 + 0.25 + 0.25 = 0.5$

สำหรับกิ่งแรกตามมิติที่สองที่มีค่าอยู่ระหว่าง $-\infty$ ถึง 6.5 ก็ถูกค้นหาด้วย เนื่องจากมีค่าระยะทางยกกำลังสองเท่ากันกับกิ่งที่สองในมิติเดียวกัน

สองกิ่งที่เหลือในมิติแรกได้แก่สองกิ่งแรกไม่ถูกค้นหา เนื่องจากระยะทางยกกำลังสองมีค่าเกิน 0.5 กล่าวคือ กิ่งแรกมีค่าระยะทางยกกำลังสองเท่ากับ $(7 - 2.5)^2 = 20.25$

และ กิ่งสองมีค่าระยะทางยกกำลังสองเท่ากับ $(7 - 4.5)^2 = 6.25$

ดังนั้นจากการค้นหาทั้งหมด สรุปได้ว่า X_{11} เป็นเพื่อนบ้านใกล้ที่สุดของ P



รูปที่ 3.13 การตัดแบ่งของแบบสำหับการค้นหาโดยใช้การตัดแบ่งแบบเรียงลำดับ

7.5 ต้นไม้เคดี

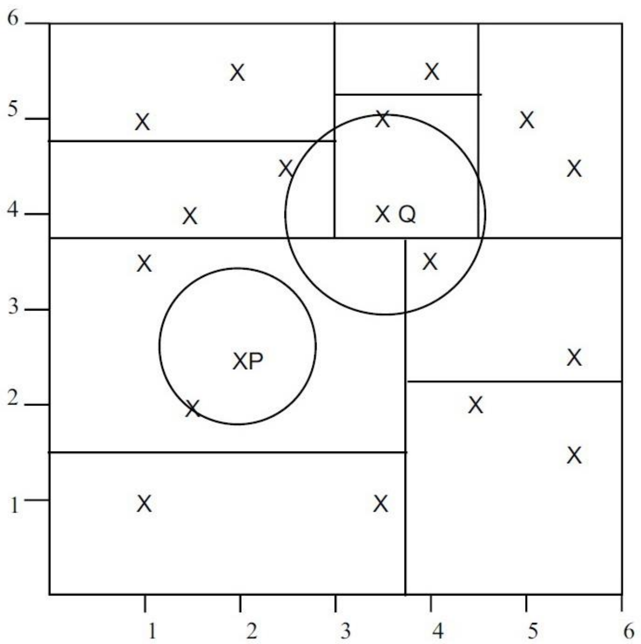
ต้นไม้เคดี (k-d Tree) คือต้นไม้ทวิภาคซึ่งเป็นโครงสร้างข้อมูลใน ขั้นตอนวิธีการค้นหาแบบเพิ่ม (Incremental Search Algorithm) สำหรับค้นหาเพื่อนบ้านใกล้ที่สุด ขั้นตอนวิธีนี้จัดได้ว่ามีประสิทธิภาพโดยสามารถลดการคำนวณที่ซ้ำซ้อนลงได้ รากของต้นไม้เคดีเป็นตัวแทนของปริภูมิ k มิติ ที่เก็บแบบทดสอบทั้งหมด ในขณะที่แต่ละโหนดเป็นตัวแทนของปริภูมิย่อยที่เก็บเซตย่อยของแบบทดสอบ ปริภูมิย่อยของ โหนดไม่ปลายทาง (Non-terminal Node) ถูกตัดแบ่งเป็นสองส่วนโดย ระนาบหลายมิติ (Hyperplane) ที่ตั้งฉากกับแกนใดแกนหนึ่งจากแกนพิกัดร่วม k แกน ส่งผลให้ได้โหนดลูก 2 โหนด ระนาบหลายมิตินี้ ต้องอยู่ในตำแหน่งที่แบ่งแบบฝึกฝนประมาณครึ่งหนึ่งจากโหนดพ่อแม่ไปยังโหนดลูกแต่ละโหนดได้ อย่างไรก็ตาม โหนดลูกที่บรรจุแบบฝึกฝนน้อยกว่า ขีดแบ่ง (Threshold) จะกลายเป็นโหนดใบ (Leaf Node) ของต้นไม้เคดี ในส่วนเพิ่มเติม โหนดลูกแต่ละโหนดในต้นไม้เคดีเป็นตัวแทนของปริมาตรหลายมิติ (Hyperplane) รูปที่ 3.14 แสดงการสร้างต้นไม้เคดีสำหรับแบบสองมิติ โดยที่แต่ละโหนดใบเป็นตัวแทนของแต่ละบริเวณสี่เหลี่ยมในรูป

การค้นหาเพื่อนบ้านใกล้ที่สุดโดยใช้ต้นไม้เคดีที่สร้างจากแบบฝึกฝน เริ่มต้นจากระบุตำแหน่งของโหนดใบจากตำแหน่งของแบบทดสอบ จากนั้นคำนวณระยะทางจากแบบฝึกฝนทั้งหมดในโหนดใบไปยังแบบทดสอบ แบบฝึกฝนที่มีระยะทางสั้นที่สุดไปยังแบบทดสอบเป็นตัวเลือกที่มีโอกาสเป็นเพื่อนบ้านใกล้ที่สุดของแบบทดสอบนี้ ระยะทางสั้นที่สุดนี้ (ใช้สัญลักษณ์เป็น r) ถูกกำหนดเป็น ขอบเขตบน

(Upper Bound) สำหรับระยะทางไปยังเพื่อนบ้านใกล้ที่สุดจริง ถ้าสร้าง ทรงกลมหลายมิติ (Hyperspere) S_r ที่มีจุดศูนย์กลางอยู่ที่แบบทดสอบ รัศมี r กรณีต่าง ๆ ที่เกิดขึ้น มีดังต่อไปนี้

- ถ้า S_r อยู่ภายในปริมาตรหลายมิติของโหนดใบอย่างมิดชิด กรณีนี้การคำนวณระยะทางจะเกิดขึ้นกับแบบของโหนดใบโหนดนี้เท่านั้น เนื่องจากเพื่อนบ้านใกล้ที่สุดคือแบบจากโหนดนี้แน่นอน
- ถ้า S_r ไม่อยู่ในปริมาตรหลายมิติของโหนดใบอย่างมิดชิด ให้ไต่ต้นไม้เคดีขึ้นไปหนึ่งระดับไปยังโหนดพ่อแม่ และไต่ต้นไม้ลงหนึ่งระดับไปยังโหนดลูกอีกโหนด เพื่อพิจารณาโหนดใบที่ซ้อนทับกับ S_r แบบของโหนดใบโหนดนี้จะถูกคำนวณระยะทางด้วย เพื่อค้นหาระยะทางสั้นที่สุดของแบบจากโหนดลูกทั้งสอง ระยะทางสั้นที่สุดค่าใหม่ที่ได้ต้องนำไปปรับปรุงค่า r กระบวนการไต่ขึ้นต้นไม้เคดีนี้จะดำเนินการจนกระทั่ง S_r ถูกล้อมรอบโดยปริมาตรหลายมิติของโหนดไม่ปลายทางอย่างมิดชิด

ตัวอย่างที่ 14 รูปที่ 3.14 แสดงต้นไม้เคดีสำหรับเซตของแบบ พิจารณาแบบ P เนื่องจากพื้นที่สี่เหลี่ยมของโหนดใบปกคลุมวงกลมที่มีจุดศูนย์กลางอยู่ที่แบบนี้อย่างมิดชิด ดังนั้น เพื่อนบ้านใกล้ที่สุดของ P จะอยู่ภายในพื้นที่สี่เหลี่ยมนี้แน่นอน พิจารณาแบบ Q เนื่องจาก วงกลมที่มีจุดศูนย์กลางอยู่ที่แบบนี้ซ้อนทับกับพื้นที่สี่เหลี่ยมของโหนดใบหลายโหนด ดังนั้น จึงมีความจำเป็นต้องค้นหาเพื่อนบ้านใกล้ที่สุดจากพื้นที่สี่เหลี่ยมที่ซ้อนทับนี้ทั้งหมด



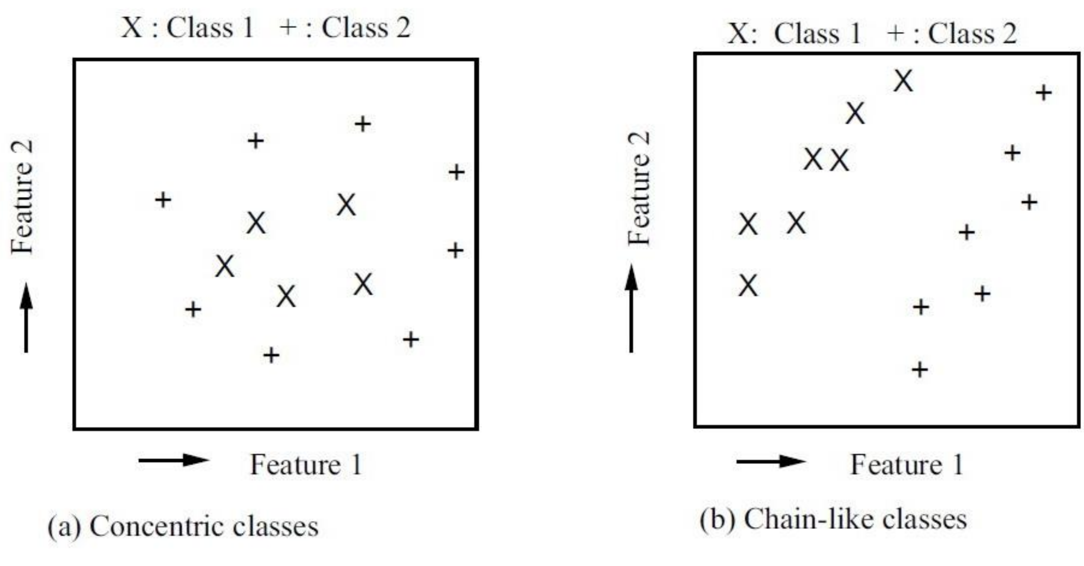
รูปที่ 3.14 การสร้างต้นไม้เคดีและการค้นหาโหนดใกล้ที่สุด

8. ขั้นตอนวิธีการย่อ

โปรโตไทป์ (Prototype) คือแบบตัวแทนจากแบบทั้งหมด โดยมากใช้กับเซตของแบบที่มีขนาดใหญ่ ตัวอย่างของตัวจำแนกแบบที่ใช้โปรโตไทป์ได้แก่ MDC จากหัวข้อที่ 5 ซึ่งใช้เซนทรอยด์เป็นโปรโตไทป์ ผลที่ได้จากการใช้โปรโตโทป์คือเวลาการจำแนกแบบที่เร็วขึ้น เนื่องจากเซตฝึกฝนมีขนาดเล็กลง วิธีการใหม่ในหัวข้อนี้ประยุกต์ใช้โปรโตไทป์เช่นเดียวกับ MDC ซึ่งจัดอยู่ในประเภท การเลือกโปรโตไทป์ (Prototype Selection)

ถึงแม้ว่า MDC เป็นตัวจำแนกแบบที่มีข้อดีทางด้าน ประสิทธิภาพต่อเวลา (Time-effidient) แต่ล้มเหลวในการจำแนกแบบ ถ้าเซนทรอยด์ไม่ได้เป็นตัวแทนที่ดีของชั้นข้อมูล เหตุการณ์นี้เกิดขึ้นบ่อยครั้งในเซตข้อมูล ที่ชั้นข้อมูลมีลักษณะ เหมือนลูกโซ่ (Chain-like) ทอดยาวไปในทิศทางเดียว หรือเกิดขึ้นในกรณีที่ชั้นข้อมูลตั้งแต่สองชั้นข้อมูลขึ้นไป มีจุดศูนย์กลางร่วมกัน (Concentric) ซึ่งหมายความว่าแบบในชั้นข้อมูลที่แตกต่างกันมีการผสมผสานปนเปกัน ดังแสดงในรูปที่ 3.5 ดังนั้น การเลือกใช้ตัวแทน (Representative) มากกว่าหนึ่งตัวแทนจากแต่ละชั้นข้อมูลเป็นทางเลือกที่ดีกว่า

ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดในหัวข้อนี้ ได้แก่ เพื่อนบ้านใกล้ที่สุดแบบย่อ (CNN: Condensed Nearest Neighbour) และ เพื่อนบ้านใกล้ที่สุดแบบย่อที่ถูกปรับปรุง (MCNN: Modified Condensed Nearest Neighbour) วิธีการเหล่านี้มีจุดมุ่งหมายเพื่อรักษาความถูกต้องของกระบวนการจำแนกแบบโดยการสร้าง เซตย่อ (Condensed Set) ซึ่งมีความสอดคล้องกับแบบจากเซตดั้งเดิม วิธีการนี้ทำให้มั่นใจได้ว่าแบบดั้งเดิมทั้งหมดถูกจำแนกได้อย่างถูกต้องภายใต้กฎเพื่อนบ้านใกล้ที่สุด



รูปที่ 3.15 ตัวอย่างของเซตข้อมูลที่มีเซนทรอยด์ร่วมกันและรูปทรงลูกโซ่

ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดจำนวนมากให้ความสอดคล้องกันของแบบตัวแทนและแบบดั้งเดิม แต่ไม่สามารถลดขนาดเซตย่อให้เล็กที่สุดได้ สำหรับจุดอ่อนของ CNN คือมีความเป็นไปได้ที่เซตย่อจะบรรจุแบบจากเซตดั้งเดิมทั้งหมด ทำให้ไม่สามารถลดเวลาประมวลผลลงได้ นอกจากนี้ CNN ยังมีความอ่อนไหวต่อลำดับการประมวลผลของแบบจากเซตฝึกฝน กล่าวคือถ้าลำดับของแบบฝึกฝนไม่เหมือนเดิมอาจส่งผลให้เซตย่อเปลี่ยนแปลงไปจากเดิมได้

ขั้นตอนวิธีการย่อ

CNN เป็นขั้นตอนวิธีแรกที่นำเสนอในหัวข้อนี้ โดยเป็นเทคนิคในกลุ่มของการเลือกโปรโตไทป์ที่ได้รับความนิยมในการใช้งานสูงที่สุด สำหรับขั้นตอนวิธีนี้ แต่ละแบบฝึกฝนถูกพิจารณาทีละแบบเพื่อหาเพื่อนบ้านใกล้ที่สุดจากเซตย่อ ถ้าป้ายชื่อชั้นข้อมูลของแบบที่ถูกพิจารณาและเพื่อนบ้านใกล้ที่สุดเหมือนกัน แบบที่ถูกพิจารณาจะถูกกำจัดออกไป นอกเหนือจากนี้ แบบที่ถูกพิจารณาจะถูกเพิ่มเข้าไปในเซตย่อ หลังจากแบบฝึกฝนถูกพิจารณาจนครบรอบแล้ว รอบต่อไปจะถูกพิจารณาใหม่แบบวนซ้ำ การวนรอบจะดำเนินไปอย่างต่อเนื่อง จนกระทั่งถึงรอบที่ไม่มีแบบใดถูกเพิ่มเข้าไปในเซตย่อเลย สถานะนี้ของเซตย่อเรียกว่า ไม่มีการเปลี่ยนแปลง (Consistent) และ ขั้นตอนวิธีจะหยุดการทำงานลง

กำหนดให้ *Train* คือเซตของคู่จำนวน N สมาชิก แต่ละคู่ประกอบไปด้วยแบบฝึกฝนและป้ายชื่อชั้นข้อมูล

$$Train = \{ (X_1, \theta_1), (X_2, \theta_2), \ldots, (X_N, \theta_N) \}$$

กำหนดให้ *Condensed* คือเซต กำหนดให้มีค่าเริ่มต้นเป็นเซตว่าง

ขั้นตอนวิธีการย่อ มีการทำงานตามลำดับ ดังต่อไปนี้

ขั้นตอนที่ 1

เลือกคู่แรกจาก *Train* แล้วเพิ่มเข้าไปใน *Condense*

Reduced = Train – Condense

ขั้นตอนที่ 2

เลือกแบบแรกจาก *Reduce* และ ค้นหาเพื่อนบ้านใกล้ที่สุดที่อยู่ใน *Condense*

ขั้นตอนที่ 2 (ต่อ)

ถ้าแบบจาก *Reduce* และ เพื่อนบ้านใกล้ที่สุดจาก *Condense* มีป้ายชื่อชั้นข้อมูลแตกต่างกัน ให้เพิ่มแบบจาก *Reduce* เข้าไปใน *Condense*

ลบแบบที่ถูกเลือกจาก *Reduce*

ขั้นตอนที่ 3

ทำซ้ำขั้นตอนที่ 2 จนกระทั่ง *Reduce* เป็นเซตว่าง

ขั้นตอนที่ 4

Reduce = Train – Condense

ทำซ้ำขั้นตอนที่ 2 และ 3

ขั้นตอนที่ 5

ถ้า *Condense* (หรือ *Reduce*) ไม่มีการเปลี่ยนแปลง ในขั้นตอนที่ 4 จำนวน 2 รอบ ติดต่อกัน ให้หยุดการทำงาน

นอกเหนือจากนี้ ทำซ้ำขั้นตอนที่ 2, 3, 4

ตัวอย่างที่ 15 พิจารณาแบบด้านล่างจาก 3 ชั้นข้อมูล ชั้นข้อมูลที่ 1 แสดงด้วย กากบาท ชั้นข้อมูลที่ 2 แสดงด้วย วงกลม ชั้นข้อมูลที่ 3 แสดงด้วย + แบบทั้งหมดแสดงตามรูปที่ 3.16

X_1 = (1.0, 1.0, 1), X_2 = (1.0, 2.0, 1), X_3 = (1.5, 1.5, 1), X_4 = (2.0, 2.0, 1),

X_5= (3.0, 2.0, 2), X_6 = (4.0, 2.0, 2), X_7 = (4.0, 3.0, 2), X_8 = (5.0, 2.5, 2),

X_9 = (2.0, 3.0, 3), X_{10} = (3.0, 3.5, 3), X_{11} = (2.0, 4.0, 3)

ให้แบบที่ส่งให้ขั้นตอนวิธีได้แก่ X_1 ถึง X_{11} ตามลำดับ

รอบที่ 1

เริ่มต้น X_1 ถูกเพิ่มเข้าไปใน *Condensed* มีเพียง X_1 ที่อยู่ในเซตนี้ เนื่องจาก X_2, X_3, X_4 อยู่ใกล้ X_1 มากที่สุด และแบบเหล่านี้มีป้ายชื่อชั้นข้อมูลเดียวกับ X_1 ดังนั้น จึงไม่ทำอะไร

เนื่องจาก X_5 มีป้ายชื่อชั้นข้อมูลต่างจากของ X_1 ดังนั้น X_5 จึงถูกเพิ่มเข้าไปใน *Condensed*

ตอนนี้ X_6 ถูกเปรียบเทียบกับ X_1 และ X_5 เนื่องจาก X_6 อยู่ใกล้มากที่สุดกับ X_5 ซึ่งมีป้ายชื่อชั้นข้อมูลเดียวกัน ไม่มีอะไรเกิดขึ้น

แบบที่อยู่ใกล้ X_7 และ X_8 มากที่สุด คือ X_5 ซึ่งมีป้ายชื่อชั้นข้อมูลเดียวกัน สองแบบแรกจึงไม่ถูกเพิ่มเข้าไปใน *Condensed*

ระยะทางของ X_9 จาก X_1 คือ 2.236 และระยะทางระหว่าง X_9 และ X_5 คือ 1.414 ด้วยเหตุนี้ X_9 อยู่ใกล้ X_5 มากกว่าแบบอื่นใน *Condensed*

เนื่องจาก X_9 และ X_5 มีป้ายชื่อชั้นข้อมูลแตกต่างกันครับ ตอนนี้ *Condensed* มีแบบ X_1, X_5, X_9

แบบ X_{10} และ X_{11} อยู่ใกล้ X_9 ใน *Condensed* มากที่สุด สองแบบแรกจึงไม่เพิ่มเข้าไปใน *Condensed*

จบรอบแรก

รอบที่ 2

ในรอบที่สองนี้ X_2 และ X_3 อยู่ใกล้ X_1 มากที่สุด ซึ่งมีป้ายชื่อชั้นข้อมูลเดียวกัน จึงไม่ถูกเพิ่มเข้าไปใน Condensed

X_4 มีระยะทางเท่ากันไปสู่ X_5 และ X_9 ถ้า X_4 ถูกพิจารณาให้เข้าใกล้ X_5 มากที่สุด เพื่อละเว้นการเสมอ เนื่องจาก X_4 และ X_5 มีป้ายชื่อชั้นข้อมูลแตกต่างกัน X_4 จึงถูกเพิ่มเข้าไปใน Condensed

X_6, X_7, X_8 อยู่ใกล้ X_5 มากที่สุด และไม่ถูกเพิ่มใน Condensed

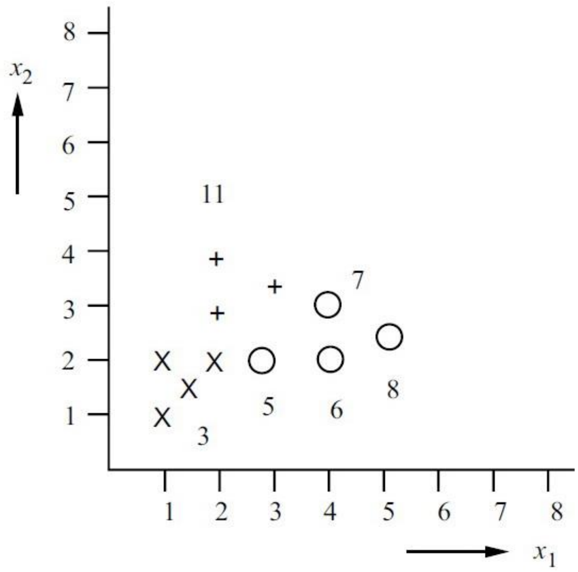
แบบ X_{10} และ X_{11} อยู่ใกล้ X_9 มากที่สุด ซึ่งมีป้ายชื่อชั้นข้อมูลเดียวกัน จึงไม่ถูกเพิ่มเข้าไปในเซตย่อ

ตอนนี้ *Confused* มีแบบ X_1, X_4, X_5, X_9

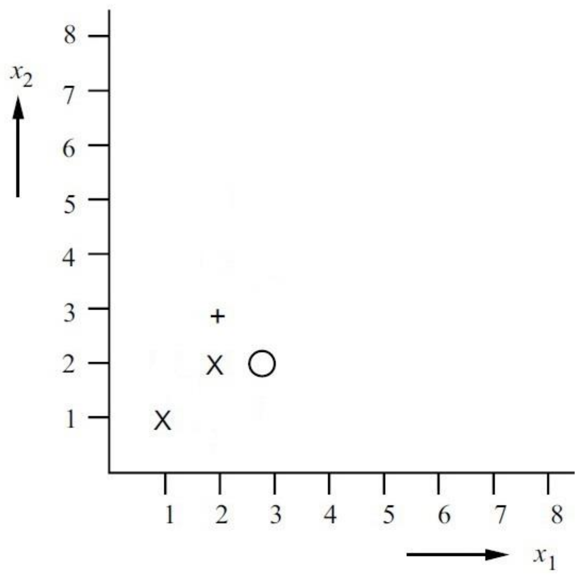
รอบที่ 3

ในรอบต่อไป เนื่องจากไม่มีการเปลี่ยนแปลงในเซตย่อ ดังนั้น

Confused = { X_1, X_4, X_5, X_9 } แสดงตามรูปที่ 3.17



รูปที่ 3.16 เซตข้อมูลตัวอย่างสำหรับขั้นตอนวิธีย่อ



รูปที่ 3.17 เซตย่อ

การได้มาซึ่งเซตย่อจำเป็นต้องใช้เวลาในการประมวลผลเพิ่มเติม แต่เป็นต้นทุนที่เกิดขึ้นครั้งเดียว และเป็นต้นที่เกิดขึ้นขณะ ออฟไลน์ (Off-line) คือในขณะที่ผู้ใช้ไม่ได้ประมวลผลนั่นเอง ซึ่งต่างจากออนไลน์ (On-line) ที่เกิดขึ้นในขณะที่ผู้ใช้กำลังจำแนกแบบ อย่างไรก็ตาม การใช้เซตย่อในการจำแนกแบบส่งผลให้ใช้เวลาที่รวดเร็วกว่าการใช้แบบฝึกฝนทั้งหมด CNN มีประโยชน์ในการลดขนาดของเซตฝึกฝน แต่อาจทำให้ความแม่นยำของตัวแบบลดลง นอกจากนี้ จัดเป็น ข้อดี-ข้อเสีย (Trade-off) ที่ต้องคำนึงถึง CNN ขึ้นอยู่กับลำดับ (Order Dependent) กล่าวคือ เซต Condensed มี ความอ่อนไหว

(Sensitive) กับตำแหน่งของข้อมูลเข้า เซตนี้อาจมีความแตกต่างไปจากเดิม เช่น มีขนาดเพิ่มขึ้น มีขนาดลดลง หรือ อาจมีสมาชิกภายในที่ไม่เหมือนเดิม

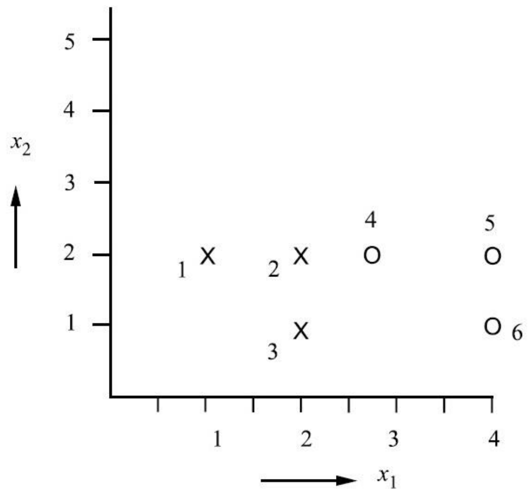
ตัวอย่างที่ 16 จากรูปที่ 3.17 มี 6 แบบ ได้แก่

1: (1, 2), 2: (2. 2) และ 3: (2, 1) ซึ่งอยู่ในชั้นข้อมูล ×

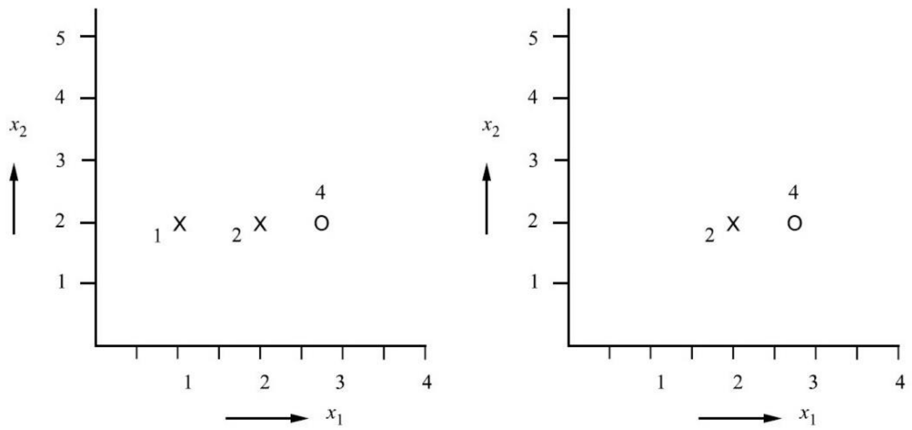
4: (2.7, 2), 5: (4, 2) และ 6: (4, 1) ซึ่งอยู่ในชั้นข้อมูล O

ถ้าลำดับของข้อมูลเข้าคือ 1, 2, 3, 4, 5 และ 6 แล้ว *Condensed* = { 1, 4, 2 }
ถ้าลำดับของข้อมูลเข้าคือ 2, 3, 1, 4, 5 และ 6 แล้ว *Condensed* = { 2, 4 }

เซตย่อทั้งสองมีความแตกต่างกันทั้งสมาชิก ขนาดของเซต และขนาดของชั้นข้อมูล แสดงตามรูปที่ 3.19



รูปที่ 3.18 เซตข้อมูลตัวอย่างสำหรับขั้นตอนวิธีย่อ 2



รูปที่ 3.19 เซตย่อที่แตกต่างกัน

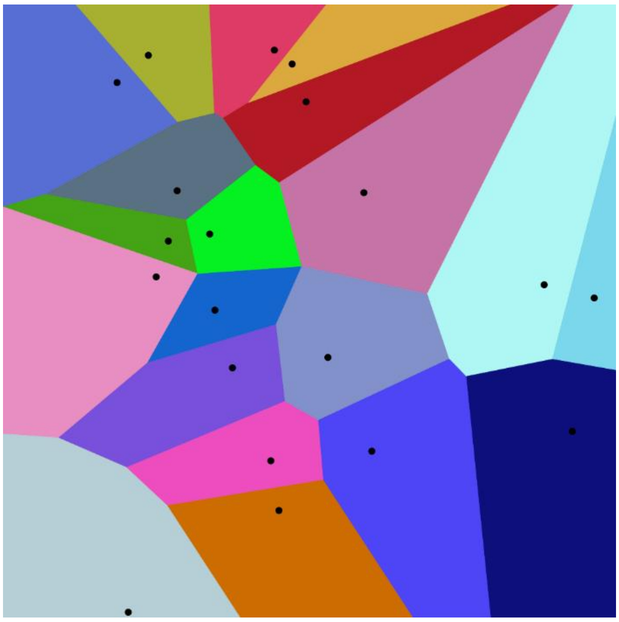
ขั้นตอนวิธีการย่อที่ถูกปรับปรุง

อุปสรรคใหญ่ของปัญหาการจำแนกแบบในเซตข้อมูลที่มีจำนวนมิติสูงมากคือการระบุ ขอบเขต (Boundary) ในแต่ละชั้นข้อมูล อย่างไรก็ตาม MCNN พยายามที่จะแบ่งปริภูมิของชั้นข้อมูลออกเป็นปริภูมิย่อยต่าง ๆ อย่างเรียบง่ายที่ ไม่ซ้อนทับทัน (Non-overlapping) โดยใช้ วิธีการเพิ่มเข้า (Incremental Manner) เพื่อเพิ่มโปรโตไทป์ไปยังเซตโปรโตไทป์ตัวแทน จนกระทั่ง แบบฝึกฝนทั้งหมดถูกจำแนกได้อย่างถูกต้องโดยใช้เซตนี้ ณ จุดนี้ ปริภูมิในแต่ละชั้นข้อมูลถูกแยกแยะกลายเป็น แผนภาพโวโรนอย (Voronoi Diagram) โดยประมาณ

$$R_j = \bigcup_{i=1}^{n} V_{ji}\,, j = 1, \dots, c$$

- n คือจำนวนของปริภูมิในชั้นข้อมูล j
- c คือ จำนวนของชั้นข้อมูล
- R_j คือ ปริภูมิในชั้นข้อมูล j
- V_{ji} คือ ปริภูมิย่อยที่ i ในชั้นข้อมูล j จากแผนภาพโวโรนอย

ในทางคณิตศาสตร์ แผนภาพโวโรนอยคือการแบ่งปริภูมิออกเป็นปริภูมิย่อยที่ติดกัน ในแต่ละปริภูมิย่อยบรรจุ ซีด (Seed) โดยที่ จุดใด ๆ ในปริภูมิย่อยต้องใกล้ซีดของปริภูมินั้นมากกว่าซีดจากปริภูมิอื่น



รูปที่ 3.20 แผนภาพโวโรนอย

(**ที่มา:** https://en.wikipedia.org/wiki/Voronoi_diagram)

MCNN ใช้วิธีการเพิ่มเข้าสร้างเซตของโปรโตไทป์จากแบบตัวแทนของแต่ละชั้นข้อมูลที่สามารถจำแนกแบบฝึกฝนทั้งหมดได้ถูกต้อง ขั้นตอนวิธีเริ่มต้นจากเซตพื้นฐานของโปรไตไทป์ที่ประกอบไปด้วยหนึ่งแบบจากหนึ่งชั้นข้อมูล จากนั้นแบบจากเซตฝึกฝนจะถูกจำแนกโดยโปรไตไทป์เหล่านี้ ในกรณีที่พบแบบที่ถูกจำแนกผิดพลาด โปรโตไทป์จากแต่ละชั้นข้อมูลจะถูกค้นหาใหม่แล้วเพิ่มเข้าไปในเซตของโปรโตไปท์ ณ ขณะนี้ เซตฝึกฝนจะถูกจำแนกอีกครั้งด้วยเซตของโปรไตไทป์ที่มีขนาดใหญ่ขึ้น และอาจมีการค้นหาโปรไตไทป์ใหม่ถ้าพบแบบฝึกฝนที่ถูกจำแนกผิดพลาดอีก กระบวนการนี้ถูกทำซ้ำจนกระทั่งแบบในเซตฝึกฝนถูกจำแนกได้อย่างถูกต้องทั้งหมด

วิธีการที่ใช้ตัวแทนเดี่ยวต่อหนึ่งชั้นข้อมูล เช่น MDC ใช้เซนทรอยด์เป็นแบบตัวแทน มีความเหมาะสมกับเซตข้อมูลที่มีลักษณะเฉพาะตัวอันได้แก่ เซตข้อมูลที่แต่ละชั้นข้อมูลมีการแจกแจงแบบเกาส์ (Gaussian Distribution) ซึ่งต่างจาก MCNN ที่ชั้นข้อมูลไม่จำเป็นต้องมีการแจกแจงดังกล่าว ทำให้มีความยืดหยุ่นต่อการใช้งานมากกว่า

ขั้นตอนวิธีการย่อที่ถูกปรับปรุง มีการทำงานตามลำดับ ดังต่อไปนี้

ขั้นตอนที่ 1

กำหนดให้ *Train* คือ แบบฝึกฝน

$Prototype = \varnothing$

$Typical = \varnothing$

ขั้นตอนที่ 2

ค้นหาแบบทั่ว ๆ ไป สำหรับแต่ละชั้นข้อมูลจาก Train

เพิ่มแบบเหล่านั้นไปยัง Typical

ขั้นตอนที่ 3

$Prototype = Prototype \cup Typical$

ขั้นตอนที่ 4

ด้วย *Prototype* จำแนก *Train* โดยใช้ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุด

ขั้นตอนที่ 5

Sub-group = ∅

เพิ่มแบบที่จำแนกผิดพลาดเข้าไปใน *Sub-group*

ขั้นตอนที่ 6

ถ้า *Sub-group* = ∅ แล้ว หยุดการทำงาน และ *Prototype* คือเซตของโปรโตไทป์สุดท้ายที่ถูกเลือก

ขั้นตอนที่ 7

Typical = ∅

ค้นหาแบบทั่ว ๆ ไป สำหรับแต่ละชั้นข้อมูลใน *Sub-group* และ นำแบบเหล่านั้นใส่ลงใน *Typical*

ขั้นตอนที่ 8

ใช้ *Typical* ในการจำแนก *Sub-group*

ขั้นตอนที่ 9

ค้นหาแบบที่จำนแนกได้อย่างถูกต้อง

Sub-group = ∅

นำแบบที่จำแนกถูกต้องเพิ่มลงใน *Sub-group* และถ้าแบบที่จำแนกผิดพลาดยังมีอยู่ ไปยังขั้นตอนที่ 7

ขั้นตอนที่ 10

ไปยังขั้นตอนที่ 3

แบบทั่วไป (จากเซต *Typical*) ถูกค้นพบในขั้นตอนที่ 5 ถึง 9 แบบเหล่านี้เป็นตัวแทนจากแต่ละชั้นข้อมูล มีเพียงแค่แบบที่จำแนกถูกต้องเท่านั้นที่ถูกเก็บใน *Sub-group* หลังจากนั้น แบบทั่วไปถูกค้นหาอีกครั้ง ขั้นตอนเหล่านี้ดำเนินไปจนกระทั่งเมื่อใช้แบบทั่วไปจำแนกแบบใน Sub-group แล้วไม่มีแบบที่จำแนกผิดพลาดหลงเหลืออยู่ ในส่วนเพิ่มเติม การใช้ขั้นตอนวิธี MCNN จำนวนของแบบที่จำแนกผิดพลาดจากเซตฝึกฝนลดจำนวนลง จนกระทั่งแบบฝึกฝนทั้งหมดถูกจำแนกได้อย่างถูกต้องจากเซตย่อ (เหมือนที่เกิดขึ้นกับกรณีของ CNN)

ตัวอย่างที่ 17 พิจารณาแบบในรูปที่ 3.21 ซึ่งประกอบไปด้วยแบบ ดังต่อไปนี้

X_1 = (1.0, 1.0, 1), X_2 = (1.0, 2.0, 1), X_3 = (1.5, 1.5, 1), X_4 = (2.0, 2.0, 1)

X_5 = (1.0, 3.0, 2), X_6 = (3.0, 2.0, 2), X_7 = (4.0, 2.0, 2), X_8 = (5.0, 2.5, 2)
X_9 = (3.0, 3.5, 2)

X_{10} = (2.0, 3.0, 3), X_{11} = (2.0, 4.0, 3), X_{12} = (3.0, 4.5, 3), X_{13} = (4.0, 3.0, 3)

เซนทรอยด์ C_i สำหรับแต่ละชั้นข้อมูล i ถูกคำนวณได้ ดังต่อไปนี้

C_1 = (1.375, 1.625)

C_2 = (4.0, 3.0)

C_3 = (2.33, 4.0)

C_1 ใกล้ X_3 มากที่สุด C_2 ใกล้ X_7 มากที่สุด และ C_3 ใกล้ X_{11} มากที่สุด

ดังนั้น เซตโปรโตไทป์บรรจุแบบ X_3, X_7, X_{11}

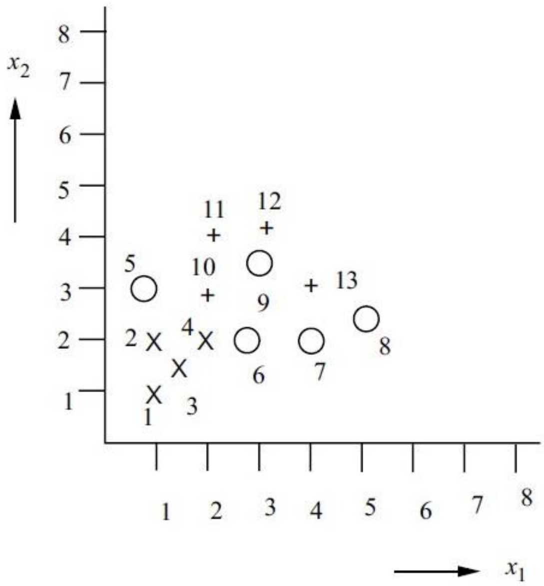
ถ้าเซตฝึกฝนถูกจำแนกโดยใช้เซตโปรโตไทป์ แบบทั้งหมดถูกจำแนกได้อย่างถูกต้องยกเว้น X_{10} ซึ่งระยะทางเท่ากันจาก X_7 และ X_{11}

เนื่องจาก X_7 ปรากฏในเซตฝึกฝนก่อน X_{11} ถ้า X_{10} ถูกจำแนกให้อยู่ในชั้นข้อมูลที่ 2 แล้ว ปรากฏว่าการจำแนกแบบนี้ผิดพลาด เนื่องจาก X_{10} อยู่ในชั้นข้อมูลที่ 3

แบบที่ถูกจำแนกผิดพลาดในแต่ละชั้นข้อมูล ถูกใช้ในการหาเซตต่อไปของแบบทั่วไป (*Typical*) ซึ่งถูกเพิ่มไปยังเซตโปรโตไทป์

ในแต่ละรอบการทำงาน เซตโปรโตไทป์ถูกสร้างจนกระทั่งแบบทั้งหมดถูกจำแนกได้อย่างถูกต้อง

สิ่งที่ MCNN แตกต่างจาก CNN คือ ไม่ขึ้นกับลำดับ (Order Independent) ซึ่งหมายความว่าขั้นตอนวิธีให้เซตโปรโตไทป์เหมือนเดิมไม่ว่าลำดับของแบบในเซตข้อมูลจะเปลี่ยนไปอย่างไร ในขณะที่ความแม่นยำในการจำแนกแบบที่ได้จาก MCNN อยู่ในระดับค่อนข้างดี อย่างไรก็ตาม เวลาที่ใช้ของ MCNN นานกว่า CNN



รูปที่ 3.21 เซตข้อมูลตัวอย่างสำหรับขั้นตอนวิธีย่อที่ถูกปรับปรุง

9. ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดคลุมเครือ

ขั้นตอนวิธีนี้ได้ใช้แนวคิดของ เซตคลุมเครือ (Fuzzy Set) ที่สมาชิกของเซตมีระดับของ การเป็นสมาชิก (Membership) ในทฤษฎีเซตดั้งเดิม สถานะของสมาชิกในเซตมีเพียง 2 สถานะ ได้แก่ เป็นสมาชิก และ ไม่เป็นสมาชิก แต่ในเซตคลุมเครือ สมาชิกในเซตมี ฟังก์ชันการเป็นสมาชิก (Membership Function) ที่ให้ค่าในช่วงของจำนวนเต็ม [0, 1]

ในขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดคลุมเครือ เซตคลุมเครือคือชั้นข้อมูล c ชั้นข้อมูล แต่ละแบบถือว่าอยู่ในทุกชั้นข้อมูล i ด้วยค่าการเป็นสมาชิกซึ่งขึ้นอยู่กับเพื่อนบ้านใกล้ที่สุด k ตัว ในชั้นข้อมูลนั้น ในส่วนเพิ่มเติม ขั้นตอนวิธีนี้ค้นหาเพื่อนบ้านใกล้ที่สุด k ตัวของแบบทดสอบ แล้วให้ค่าการเป็นสมาชิกสำหรับแต่ละจุดไปยังทุกชั้นข้อมูล ถ้า P คือ แบบใหม่ การเป็นสมาชิก $\mu_i(P)$ ของ P ในชั้นข้อมูล i คำนวณได้ ดังสมการด้านล่างนี้

$$\mu_i(P) = \frac{\Sigma_{j=1}^{K} \mu_{ij} \left(\frac{1}{d(P,x_j)^{\frac{2}{m-1}}} \right)}{\Sigma_{j=1}^{K} \left(\frac{1}{d(P,x_j)^{\frac{2}{m-1}}} \right)}$$

μ_{ij} แทน การเป็นสมาชิกในชั้นข้อมูลที่ i จากเวกเตอร์ที่ j ของเซตฝึกฝน วิธีการแรกในการให้ค่าสมาชิกคือ การเป็นสมาชิกเต็มในชั้นข้อมูลที่รู้จักและการไม่เป็นสมาชิกในชั้นข้อมูลที่เหลือ วิธีการอื่นในการให้

ค่าสมาชิกคือ คำนวณระยะทางจากแบบไปยังค่าเฉลี่ยของแบบในทุก ๆ ชั้นข้อมูล กำหนดให้ m คือค่าคงที่ ที่ถูกตั้งค่าโดยผู้ใช้

แต่ละแบบมีค่าการเป็นสมาชิก 1 สำหรับชั้นข้อมูลที่อยู่ในเพื่อนบ้านใกล้ที่สุด k ตัว และ มีค่าการเป็นสมาชิก 0 ไปยังชั้นข้อมูลที่เหลือ แบบทดสอบจะถูกให้ค่าการเป็นสมาชิกไปยังทุกชั้นข้อมูล โดยที่แบบจะถูกจำแนกให้อยู่ในชั้นข้อมูลที่มีค่าการเป็นสมาชิกสูงสุด การคำนวณค่าเหล่านี้ แสดงตามตัวอย่างด้านล่างนี้

ตัวอย่างที่ 18 จากตัวอย่างในรูปที่ 3.2 มีแบบฝึกฝนทั้งหมด ดังต่อไปนี้

X_1 = (0.8, 0.8, 1), X_2 = (1.0, 1.0, 1), X_3 = (1.2, 0.8, 1), X_4 = (0.8, 1.2, 1)
X_5 = (1.2, 1.2, 1)

X_6 = (4.0, 3.0, 2), X_7 = (3.8, 2.8, 2), X_8 = (4.2, 2.8, 2), X_9 = (3.8, 3.2, 2)
X_{10} = (4.2, 3.2, 2), X_{11} = (4.4, 2.8, 2), X_{12} = (4.4, 3.2, 2)

X_{13} = (3.2, 0.4, 3), X_{14} = (3.2, 0.7, 3), X_{15} = (3.8, 0.5, 3), X_{16} = (3.5, 1.0, 3)
X_{17} = (4.0, 1.0, 3), X_{18} = (4.0, 0.7, 3)

ถ้า P = (3.0, 2.0) และ k = 5 แล้ว เพื่อนบ้านใกล้ที่สุด 5 ตัว ได้แก่ 16, 7, 14, 6, 17

เมื่อพิจารณาชั้นข้อมูลที่ 1 เนื่องจากไม่มีแบบจากชั้นข้อมูลที่ 1 ดังนั้น μ_1(P) = 0

สำหรับชั้นข้อมูลที่ 2 μ_{ij} เป็น 1 สำหรับแบบ 6, 7 และเป็น 0 สำหรับ แบบ 16, 14, 17

สำหรับชั้นข้อมูลที่ 3 μ_{ij} เป็น 1 สำหรับแบบ 16, 14, 17 และเป็น 0 สำหรับ แบบ 6, 7

ถ้าผู้ใช้กำหนดค่า m = 2 แล้ว

$$\mu_1(P) = 0$$

$$\mu_2(P) = \frac{\frac{1}{1.414^2} + \frac{1}{1.131^2}}{\frac{1}{1.414^2} + \frac{1}{1.131^2} + \frac{1}{1.118^2} + \frac{1}{1.315^2} + \frac{1}{1.414^2}} = 0.406$$

$$\mu_3(P) = \frac{\frac{1}{1.118^2} + \frac{1}{1.315^2} + \frac{1}{1.414^2}}{\frac{1}{1.118^2} + \frac{1}{1.315^2} + \frac{1}{1.414^2} + \frac{1}{1.414^2} + \frac{1}{1.131^2}} = 0.594$$

ดังนั้น P ถูกกำหนดให้อยู่ในชั้นข้อมูลที่ 3

อภิปราย

ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดมีแนวคิดการประมวลผลที่ไม่ซับซ้อนมากนัก สามารถอธิบายได้อย่างชัดเจนง่ายต่อการทำความเข้าใจ นอกจากนี้ยังพัฒนาโปรแกรมเพื่อสร้างตัวแบบได้อย่างไม่ยาก ตัวจำแนกชนิดนี้ไม่มีเวลาในการสร้างตัวแบบแต่ใช้เวลาในการจำแนกแบบยาวนาน โดยมีความซับซ้อนของเวลาเป็นพหุนามกำลังสองหรือ $O(n^2)$ ดังนั้นจึงไม่เหมาะสมกับโปรแกรมประยุกต์ที่ต้องรีบใช้ผลการทำนายทันที เนื่องจากตัวแบบต้องมีช่วงเวลาในการรอผลการจำแนกแบบ ตัวอย่างของโปรแกรมประยุกต์ที่จำเป็นต้องรีบใช้ผลการทำนายได้แก่ การตรวจจับผู้บุรุกในระบบเครือข่าย เพราะว่าจะได้รีบทำการสกัดกั้นการบุกรุกได้ทัน ในทางกลับกัน โปรแกรมประยุกต์ที่ไม่จำเป็นต้องรีบใช้ผลการทำนายได้แก่ การวินิจฉัยโรค เพราะว่าคนไข้ไม่จำเป็นต้องรับทราบผลการตรวจในทันที สามารถนัดหมายเพื่อมาฟังผลในภายหลังได้

เนื่องจากตัวจำแนกขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุด มีพื้นฐานการทำนายบนการพิจารณาระยะทางระหว่างแบบทดสอบและแบบฝึกฝน ดังนั้นการเลือกหน่วยวัดความใกล้เคียงหรือฟังก์ชันระยะทางที่เหมาะสม ย่อมส่งผลต่อความแม่นยำของการทำนายแบบ นอกจากนี้เซตข้อมูลที่ผ่านการทำให้อยู่ในรูปปกติแล้ว ย่อมให้ผลการทำนายที่สูงกว่าเซตข้อมูลที่ยังไม่ผ่านกระบวนการดังกล่าว เนื่องจากการทำให้อยู่ในรูปปกติเป็นกระบวนการที่ทำให้ค่าในแต่ละฟีเจอร์มีความสำคัญเท่าเทียมกันก่อนการวัดระยะทาง

ปัจจัยที่มีผลต่อประสิทธิภาพการทำนายของตัวจำแนกแบบประเภทนี้คือ การปรับแต่งการทำงานของตัวแบบได้อย่างเหมาะสมผ่านการตั้งค่าพารามิเตอร์ k (จำนวนเพื่อนบ้านใกล้ที่สุดของแบบฝึกฝนที่ต้องการทำนาย) โดยทั่วไปค่าพื้นฐานของ k ที่นิยมใช้กันคือ 5 แต่ไม่ได้รับประกันว่าค่านี้จะให้ผลการทำนายเป็นที่น่าพอใจ ดังนั้นอาจต้องทำการทดลองเพื่อเปรียบเทียบค่า k แต่ละค่าที่ให้ผลดีที่สุด และค่า k ที่ดีที่สุดสำหรับเซตข้อมูลหนึ่ง อาจไม่ใช่ค่า k ที่ดีที่สุดของเซตข้อมูลอื่นก็ได้ นอกจากนี้ไม่นิยมตั้งค่า k เป็นเลขคู่ เนื่องจากในขั้นตอนการจำแนกแบบอาจเกิดกรณีเสมอขึ้นได้เช่น ในเพื่อนบ้านใกล้ที่สุดจำนวน k ตัว อาจมาจากชั้นข้อมูลแรกจำนวน k/2 ตัว และมาจากชั้นข้อมูลที่สองจำนวน k/2 ตัว เนื่องจากจำนวนเพื่อนบ้านใกล้ที่สุดมีจำนวนเท่ากัน ทำให้ไม่สามารถตัดสินได้

ข้อเสียของตัวจำแนกเพื่อนบ้านใกล้ที่สุดคือ การที่ไม่สามารถทำนายได้อย่างมีประสิทธิภาพในเซตข้อมูลที่บรรจุสิ่งรบกวนปริมาณมาก สิ่งรบกวนในที่นี่ได้แก่แบบจากต่างชั้นข้อมูลที่ไม่กระจายตัวในบริเวณชั้นข้อมูลของตนเอง แต่กลับมีตำแหน่งอยู่ในบริเวณของชั้นข้อมูลอื่น ทำให้การจำแนกแบบฝึกฝนของชั้นข้อมูลที่ถูกรบกวนนั้นอาจมีความผิดเพี้ยนเกิดขึ้นได้ นอกจากนี้ในเซตข้อมูลอสมดุลที่สัดส่วนของ

แบบในแต่ละชั้นข้อมูลมีความแตกต่างกันสูงมาก และมีการผสมผสานของแบบจากต่างชั้นข้อมูลกัน การจำแนกแบบในชั้นข้อมูลที่มีปริมาณน้อยอาจไม่ค่อยแม่นยำเท่าที่ควร เนื่องจากเพื่อนบ้านใกล้ที่สุดส่วนใหญ่อาจมาจากชั้นข้อมูลอื่น

สรุป

ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดแบ่งออกเป็นประเภทต่าง ๆ ได้แก่ ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดพิจารณาแบบใกล้ที่สุดเพียงแบบเดียว ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเคตัวพิจารณาแบบใกล้เคียงเป็นกลุ่มและตัดสินจากชั้นข้อมูลส่วนมาก ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุดเคตัวที่ถูกดัดแปรสนใจน้ำหนักแทนป้ายชื่อชั้นข้อมูล เพื่อนบ้านใกล้รัศมีอาร์พิจารณาแบบทั้งหมดในรัศมีอาร์วัดจากแบบทดสอบ ตัวจำแนกระยะทางสั้นที่สุดสนใจตัวแทนของแต่ละชั้นข้อมูลแทนแบบทดสอบทั้งหมด สังเกตได้ว่าตัวจำแนกทุกประเภทข้างต้นใช้หลักการจำแนกแบบในทำนองเดียวกัน กล่าวคือพิจารณาระยะทางระหว่างแบบทดสอบกับแบบฝึกฝนที่อยู่ใกล้กัน อย่างไรก็ตาม ยังมีตัวจำแนกแบบประเภทอื่นที่ไม่ได้ใช้การคำนวณระยะทางในการทำนายแบบ ซึ่งเป็นเนื้อหาในบทต่อไปที่จะกล่าวถึงตัวจำแนกแบบประเภทที่สองได้แก่ ตัวจำแนกเบส์ที่ทำนายแบบใหม่จากการพิจารณาความน่าจะเป็นจากการประยุกต์ใช้ทฤษฎีเบส์

ในกรณีที่เซตข้อมูลบรรจุแบบจำนวนมาก จะทำให้เกิดต้นทุนสูงทางด้านเวลาในการค้นหาเพื่อนบ้านใกล้ที่สุด เพื่อที่จะแก้ปัญหานี้ การประมวลผลก่อนต่าง ๆ จึงถูกคิดค้นขึ้นเพื่อให้ค้นหาเพื่อนบ้านใกล้ที่สุดได้เร็วขึ้น วิธีที่ได้รับความนิยมมากที่สุดได้แก่ ขั้นตอนวิธีการย่อ ที่ใช้เซตย่อซึ่งบรรจุโปรไตไทปซึ่งจุดได้ว่าเป็นตัวแทนจากแบบทั้งหมด เนื่องจากเซตย่อมีขนาดที่เล็กกว่าเซตฝึกฝน จึงทำให้ต้นทุนทางด้านเวลาในการค้นหาเพื่อนบ้านใกล้ที่สุดลดลงไปได้

แบบฝึกหัด

1. จากเซตข้อมูลในรูปที่ 3.2 จงจำแนกแบบ P โดยขั้นตอนวิธีต่าง ๆ ดังต่อไปนี้

    1.1 kNN กำหนดให้ k = 1, 2, 3, 4, 5 ผลลัพธ์ทั้งหมดที่ได้แตกต่างกันหรือไม่

    1.2 MkNN กำหนดให้ k = 5 ผลลัพธ์ที่ได้แตกต่างจาก kNN โดยที่ k = 5 หรือไม่

    1.3 rNN กำหนดให้ r = 1.45 ผลลัพธ์ที่ได้แตกต่างจาก kNN โดยที่ k = 5 หรือไม่

2. จงยกตัวอย่างเซตข้อมูลที่ NN ให้ผลลัพธ์ที่ดีกว่า kNN

3. จงยกตัวอย่างเซตข้อมูลที่ MDC ให้ผลลัพธ์ที่ดี และ เซตข้อมูลที่ MDC ให้ผลลัพธ์ที่แย่

4. จงอภิปรายประสิทธิภาพของ kNN เมื่อกำหนดให้ k = n โดยที่ n คือ จำนวนของแบบฝึกฝนทั้งหมด

5. จงอภิปรายประสิทธิภาพของ rNN ในกรณีที่ แบบฝึกฝนทั้งหมดอยู่ในรัศมี r

6. จากเซตข้อมูลใน ตัวอย่างที่ 9 จงทำการจำแนกแบบ P โดยใช้ตัวแบบทุกประเภทตามตัวอย่างดังกล่าว แต่เปลี่ยนหน่วยวัดความใกล้เคียงเป็นระยะทางยูคลิเดียนแทน แล้วทำการตรวจสอบดูว่าให้ผลการทำนายเหมือนตามตัวอย่างเดิมหรือไม่

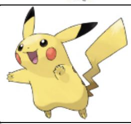
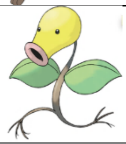
7. จากเซตข้อมูล Pokémon ตามตารางที่ 3.4 ด้านล่างนี้ จงจำแนกแบบ EGG ว่าอยู่ใน Type ใด โดยใช้ตัวจำแนกทุกประเภทในบทนี้ กำหนดให้ใช้หน่วยวัดความใกล้เคียงเป็นระยะทางยูคลิเดียน และตั้งค่าพารามิเตอร์ k = 5 และ r = 100

8. พิจารณาแบบสองมิติที่ประกอบไปด้วยค่าฟีเจอร์ 1, 2 และ ป้ายชื่อชั้นข้อมูล ด้านล่างนี้
   (1, 1, 1), (1, 2, 1), (1, 3, 1), (2, 1, 1), (2, 2, 1), (2, 3, 1), (2, 3.5, 1), (2.5, 2, 1), (3.5, 1, 1), (3.5, 2, 1), (3.5, 3, 2), (3.5, 4, 2), (4.5, 1, 2), (4.5, 2, 2), (4.5, 3, 2), (5, 4, 2), (5, 5, 2), (6, 3, 2), (6, 4, 2), (6, 5, 2)

   a. ค้นหาเซต *Condensed* โดยใช้ CNN

   b. ถ้าแบบของชั้นข้อมูลที่ 2 มาก่อนจากชั้นข้อมูลที่ 1 ตามลำดับด้านล่างนี้ จงค้นหา *Condensed*
      (3.5, 3, 2), (3.5, 4, 2), (4.5, 1, 2), (4.5, 2, 2), (4.5, 3, 2), (5, 4, 2), (5, 5, 2), (6, 3, 2), (6, 4, 2), (6, 5, 2), (1, 1, 1), (1, 2, 1), (1, 3, 1), (2, 1, 1), (2, 2, 1), (2, 3, 1), (2, 3.5,1), (2.5, 2, 1), (3.5, 1, 1), (3.5, 2, 1)

ตารางที่ 3.4 เซตข้อมูลโปเกมอน

Pokémon	Name	Stamina	Attack	Defense	Type
	Caterpie	90	62	66	BUG
	Metapod	100	56	86	BUG
	Butterfree	120	144	144	BUG
	Dratini	82	128	110	DRAGON
	Dragonair	122	170	152	DRAGON
	Dragonite	182	250	212	DRAGON
	Pikachu	70	124	108	ELECTRIC
	Raichu	120	200	154	ELECTRIC
	Bellsprout	100	158	78	GRASS
	Weepinbell	130	190	110	GRASS
	Victreebel	160	222	152	GRASS
	EGG	126	126	122	?

9. จากแบบฝึกหัดข้อ 8 จงค้นหาเซต Condensed โดยใช้ MCNN

10. จากตัวอย่างที่ 17 จงแสดงวิธีการคำนวณที่เหลือ จนกระทั่งขั้นตอนวิธีหยุดการทำงาน

บทที่ 4
ตัวจำแนกเบส์

จากบทก่อนหน้าที่กล่าวถึงตัวจำแนกเพื่อนบ้านใกล้ที่สุด ที่มีแนวคิดโดยการใช้หน่วยวัดความใกล้เคียงสำหรับวัดค่าระยะทางระหว่างแบบทดสอบและแบบฝึกฝน เพื่อตัดสินว่าแบบที่ต้องการทำนายว่าควรอยู่ในชั้นข้อมูลใด อย่างไรก็ตามยังมีตัวจำแนกบางประเภทที่ไม่สนใจความใกล้ไกลของระยะทางระหว่างแบบทดสอบและแบบฝึกฝน แต่สนใจเพียงค่าที่เป็นไปได้จากแต่ละฟีเจอร์ว่าเกิดขึ้นโดยมีความถี่บ่อยครั้งเพียงใดเมื่อเทียบกับแต่ละชั้นข้อมูล เนื้อหาในบทนี้จะกล่าวถึงตัวจำแนกแบบประเภทใหม่ได้แก่ ตัวจำแนกนาอีฟเบส์ที่ประยุกต์ใช้ทฤษฎีเบส์ในการคำนวณความน่าจะเป็นแบบมีเงื่อนไข สำหรับทำนายชั้นข้อมูลของแบบทดสอบ ในส่วนเพิ่มเติม บทนี้ยังกล่าวถึงตัวจำแนกที่อยู่ในประเภทเดียวกันได้แก่ เครือข่ายความเชื่อเบร์เซี่ยน ที่ประยุกต์ใช้โครงสร้างข้อมูลกราฟร่วมกับทฤษฎีเบส์ในการจำแนกแบบทดสอบ

จุดประสงค์การเรียนรู้

- คำนวณความน่าจะเป็นจากทฤษฎีเบส์ได้
- เข้าใจแนวคิดความอิสระจากเงื่อนไขชั้นข้อมูล
- ประมาณค่าพารามิเตอร์ความน่าจะเป็นจากความถี่สำหรับตัวแบบเบส์ได้
- สามารถจำแนกแบบโดยใช้ตัวจำแนกนาอีฟเบส์ได้
- เข้าใจการทำงานของเครือข่ายความเชื่อเบร์เซี่ยน

1. ทฤษฎีเบส์

ตัวจำแนกนาอีฟเบส์ (Naive Bayes Classifier) คือตัวจำแนกความน่าจะเป็นอย่างง่าย เนื่องจากไม่มีความซับซ้อนมากนักในการคำนวณ จึงได้รับความนิยมเป็นอย่างสูงในกระบวนการการรู้จำแบบและการทำเหมืองข้อมูล ตัวแบบประเภทนี้มีลักษณะเด่นคือเป็น ตัวจำแนกเหมาะที่สุด (Optimal Classifier) ซึ่งสามารถจำแนกแบบทดสอบโดยมีค่าเฉลี่ยของความผิดพลาดในการทำนายต่ำที่สุด นอกจากนี้ ตัวจำแนกประเภทนี้ถูกสร้างบนสมมติฐานสองข้อได้แก่ ค่าที่เป็นไปได้ของชั้นข้อมูลต่าง ๆ ในเซตข้อมูลสามารถคำนวณให้อยู่ในรูปของ ความน่าจะเป็นก่อน (Prior Probability) และ การแจกแจงของแบบต่าง ๆ ในแต่ละชั้นข้อมูลต้องสามารถทราบได้ ผลลัพธ์จากการทำนายสำหรับแบบทดสอบจะ

ถูกกำหนดป้ายชื่อจากชั้นข้อมูลที่ ความน่าจะเป็นหลัง (Posterior Probability) มีค่าสูงสุด กล่าวคือตัวแบบต้องคำนวณความน่าจะเป็นหลังของแต่ละชั้นข้อมูลก่อน จากนั้นความน่าจะเป็นหลังค่าใดมีค่ามากที่สุด แบบทดสอบจะถูกทำนายให้อยู่ในชั้นข้อมูลนั้น รายละเอียดของความน่าจะเป็นก่อนและความน่าจะเป็นหลัง จะกล่าวถึงในทฤษฎีด้านล่างนี้

ทฤษฎีเบส์ (Bayes Theorem) ถูกใช้ในการแปลงความน่าจะเป็นก่อนเป็นความน่าจะเป็นหลัง โดยมีพื้นฐานบนแบบที่ต้องการจำแนก จากสูตรด้านล่างนี้

$$P(H_i \mid X) = \frac{P(X \mid H_i)P(H_i)}{P(X)}$$

สมมติว่าทราบค่า $P(H_i)$ ก่อนการสังเกตแบบ X แล้วการจำแนกแบบ X จำเป็นต้องคำนวณค่า $P(H_i \mid X)$ โดยที่ ตัวแปรต่าง ๆ ที่เกี่ยวข้อง มีดังต่อไปนี้

- X คือ แบบที่ไม่ทราบป้ายชื่อชั้นข้อมูล
- H_i คือ สมมติฐาน (Hypothesis) ที่ i โดยแบบ X อยู่ในชั้นข้อมูล C
- $P(H_i)$ คือ ความน่าจะเป็นก่อนของสมมติฐาน H_i
- P(X) คือ ความน่าจะเป็นของแบบ X
- $P(H_i \mid X)$ คือ ความน่าจะเป็นหลังของ H_i ที่เป็นไปตามเงื่อนไขของแบบ X
- $P(X \mid H_i)$ คือ ความเป็นไปได้ (Likelihood)

ตัวอย่างด้านล่างนี้ ได้แก่ตัวอย่างที่ 1 แสดงการตัดสินใจโดยพิจารณาความน่าจะเป็นก่อน และในกรณีที่ถ้าทราบความน่าจะเป็นหลังแล้วการตัดสินใจจะมีคุณภาพดีขึ้น ซึ่งแสดงตามตัวอย่างที่ 2

ตัวอย่างที่ 1 จากการเก็บข้อมูลในร้านกาแฟแห่งหนึ่ง 99% ของลูกค้านิยมดื่มกาแฟ และที่เหลืออีก 1% ชอบดื่มชา ดังนั้น

- P(Coffee Drinker) = 0.99
- P(Tea Drinker) = 0.01

ในกรณีที่ไม่มีข้อมูลมากพอเกี่ยวกับลูกค้าที่กำลังใช้บริการ เช่นนิสัยส่วนตัวเกี่ยวกับรสชาติที่ชอบ หรือสถิติการซื้อเครื่องดื่มในอดีต การทำนายว่าลูกค้าชอบกาแฟจะมีความผิดพลาดเพียง 0.01 โดยเกิดขึ้นในกรณีที่ลูกค้าชอบดื่มชา แต่ทำนายผิดว่าลูกค้าชอบดื่มกาแฟ ดังนั้นเพื่อลดความเสี่ยงในการทำนาย ตัวแบบจะไม่ทำนายว่าลูกค้าจะชอบดื่มชาเลย เพราะมีความผิดพลาดสูงถึง 0.99 นั่นเอง แนวคิดของทฤษฎี

เบส์มีพื้นฐานอยู่บนการทำนายที่ลดความเสี่ยงต่อการผิดพลาดมากที่สุด กล่าวคือทำนายชั้นข้อมูลที่มีโอกาสเกิดขึ้นสูงสุดเสมอ ได้แก่การทำนายว่าลูกค้าจะชอบดื่มกาแฟแต่เพียงอย่างเดียว

ตัวอย่างที่ 2 กำหนดให้ความน่าจะเป็นของวันที่ถนนเปียกมีค่าเท่ากับ 0.3 ดังนั้นความน่าจะเป็นของวันที่ถนนแห้งมีค่าเท่ากับ 0.7 ถ้ามีข้อมูลเพียงเท่านี้ การคาดการณ์ที่ดีที่สุดคือทำนายว่าถนนแห้ง เนื่องจากความผิดพลาดมีแค่ 0.3 แต่ถ้าทำนายว่าถนนเปียกจะมีความผิดพลาดสูงถึง 0.7 ซึ่งมีความเสี่ยงในการทำนายผิดมากกว่า

จากข้อมูลข้างต้น สามารถคำนวณค่าความน่าจะเป็นของวันที่ฝนตกได้ ซึ่งมีค่าเท่ากับ 0.3 และถ้ามีข้อมูลเพิ่มเติมได้แก่ วันที่ถนนเปียกเพราะฝนตกมีโอกาสเกิดขึ้น 90% จากข้อมูลทั้งหมดจะสามารถกำหนดค่าตัวแปรต่าง ๆ ได้ดังต่อไปนี้

- P(H) คือ ความน่าจะเป็นก่อนของวันที่ถนนเปียก หรือ P(H) = 0.3
- P(X) คือ ความน่าจะเป็นของวันที่ฝนตก หรือ P(X) = 0.3
- P(X | H) คือ ความน่าจะเป็นของวันที่ฝนตกจากวันที่ถนนเปียกทั้งหมด หรือ P(X | H) = 0.9
- P(H | X) คือ ความน่าจะเป็นหลังของวันที่ถนนเปียกจากวันที่ฝนตกทั้งหมด

ถ้าต้องการคำนวณความน่าจะเป็นของวันที่ฝนตกแล้วถนนเปียก สามารถใช้ทฤษฎีของเบส์โดยแทนค่าตัวแปรต่าง ๆ แสดงตามสมการด้านล่างนี้

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)} = \frac{0.9 \times 0.3}{0.3} = 0.9$$

ดังนั้น ความผิดพลาดในการคาดการณ์ผิดมีเพียงแค่ 0.1 โดยจะเกิดขึ้นในกรณีที่ทำนายว่าถนนแห้งในวันที่ฝนตกจริง ในตัวอย่างนี้ ถ้าต้องการหาค่าความน่าจะเป็น P(H | X) จำเป็นต้องทราบค่าความน่าจะเป็น P(H), P(X) และ P(X | H) ก่อน ถึงจะคำนวณได้

ตัวอย่างที่ 3 จากการสำรวจพฤติกรรมการบริโภคอาหารและเครื่องดื่มของนักศึกษากลุ่มหนึ่ง ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเชียงใหม่ ว่าเป็นสาวกของเครื่องดื่มโค้กหรือเป๊ปซี่ รวมถึงความชื่นชอบในรสชาติของพิซซ่าฮัทหรือพิซซ่าคอมพานี สำหรับเงื่อนไขของการเก็บข้อมูลคือนักศึกษาจะเลือกเครื่องดื่มได้เพียงชนิดเดียวเท่านั้น ไม่สามารถเลือกทั้งคู่ได้ กล่าวคือถ้านักศึกษาชอบโค้ก แสดงว่า

นักศึกษาไม่ชอบเป๊ปซี่เป็นต้น และในการเลือกยี่ห้อพิซซ่าก็ใช้เงื่อนไขเดียวกันนี้ จากสถิติทั้งหมดที่เก็บรวบรวมได้มีดังนี้

- ความน่าจะเป็นของนักศึกษาที่เป็นสาวกเป๊กซี่เท่ากับ 0.37
- ความน่าจะเป็นของผู้ชื่นชอบรสชาติพิซซ่าฮัทเท่ากับ 0.82
- ความน่าจะเป็นของผู้ชื่นชอบรสชาติพิซซ่าคอมพานีจากสาวกโค้กทั้งหมดเท่ากับ 0.91

จงทำการคำนวณหาความน่าจะเป็นของผู้ชอบดื่มโค้กจากจำนวนผู้ชอบกินพิซซ่าคอมพานีทั้งหมด

จากโจทย์ข้างต้น กำหนดตัวแปรต่าง ๆ ที่เกี่ยวข้อง ดังต่อไปนี้

- P(Pepsi) คือ ความน่าจะเป็นของสาวกเป๊ปซี่ จากโจทย์มีค่าเท่ากับ 0.37
- P(Coke) คือ ความน่าจะเป็นของสาวกโค้ก สามารถคำนวณได้จากความน่าจะเป็นของสาวกเป๊ปซี่จากสมการ

  P(Coke) = 1 – P(Pespsi)

  = 1 – 0.37

  = 0.63

- P(Pizza Company) คือ ความน่าจะเป็นของผู้ชื่นชอบรสชาติพิซซ่าคอมพานี จากโจทย์มีค่าเท่ากับ 0.82
- P(Pizza Company | Coke) คือ ความน่าจะเป็นของผู้ชื่นชอบรสชาติพิซซ่าคอมพานีจากจำนวนสาวกโค้กทั้งหมด จากโจทย์มีค่าเท่ากับ 0.91
- P(Coke | Pizza Company) คือ ความน่าจะเป็นของผู้ชอบดื่มโค้กจากจำนวนผู้ชอบกินพิซซ่าคอมพานีทั้งหมด ตัวแปรนี้เป็นสิ่งที่โจทย์ต้องการหา

แทนค่าตัวแปรทั้งหมดลงในสมการ จะสามารถคำนวณค่า P(Coke | Pizza Company) ได้ดังนี้

$$\mathrm{P(Coke \mid Pizza\ Company)} = \frac{\mathrm{P(Pizza\ Company \mid Coke)P(Coke)}}{\mathrm{P(Company)}}$$

$$= \frac{0.91 \times 0.63}{0.28}$$

$$= 0.699$$

2. การประมาณค่าพารามิเตอร์

ตัวจำแนกนาอีฟเบส์ เป็นตัวแบบที่นำทฤษฎีเบส์มาประยุกต์ใช้ในการทำนายของตัวจำแนกแบบ โดยมีสมมติฐานว่าเซตข้อมูลใด ๆ ที่จะนำมาใช้กับตัวทำนายประเภทนี้ต้องมีคุณสมบัติ ความอิสระจากเงื่อนไขชั้นข้อมูล (Class-conditionally Independent) กล่าวคือค่าของตัวแปรต่าง ๆ ในชั้นข้อมูลเดียวกันไม่มีผลกระทบต่อกันเอง แต่ในความเป็นจริงแล้ว ค่าต่าง ๆ เหล่านี้อาจมีความเกี่ยวข้องกันไม่ว่าทางใดทางหนึ่ง อย่างไรก็ตาม ในตัวจำแนกนี้จะยึดถือสมมติฐานดังกล่าวเพื่อความง่ายในการสร้างตัวแบบ

เหตุการณ์จะถูกพิจารณาว่าเป็นอิสระต่อกันก็ต่อเมื่อ การเกิดขึ้น (หรือไม่เกิดขึ้น) ของเหตุการณ์หนึ่ง ไม่มีผลต่อความน่าจะเป็นของการเกิดขึ้นของอีกเหตุการณ์หนึ่ง ยกตัวอย่างในเซตข้อมูลผู้ป่วยในโรงพยาบาลแห่งหนึ่ง การวินิจฉัยผู้ป่วยว่าเป็น (หรือไม่เป็น) โรคเบาหวาน จะถูกตั้งสมมติฐานว่าไม่มีความเกี่ยวข้องกับอาหารที่ให้น้ำตาลสูง แต่ในความเป็นจริงแล้ว สาเหตุของโรคนี้มาจากการบริโภคอาหารที่มีน้ำตาลสูง จะเห็นได้ว่าสมมติฐานนี้อาจจะขัดกับหลักความเป็นจริง แต่ก็เพื่อให้การสร้างตัวแบบมีการคำนวณที่ไม่ซับซ้อนมากเกินไปจนทำให้เวลาประมวลผลใช้เวลานาน สมมติฐานนี้ทำให้ง่ายต่อการคำนวณเซตข้อมูลขนาดใหญ่ที่มีจำนวนฟีเจอร์มหาศาล เพราะมีความแม่นยำสูงและรวดเร็ว ด้วยวิธีการเช่นนี้จึงเป็นที่มาของการตั้งชื่อตัวจำแนกนี้ว่า ไร้เดียงสา (Naive) เนื่องจากมีการคำนวณที่เรียบง่ายไม่ซับซ้อน จากการศึกษาพบว่าตัวจำแนกนาอีฟเบส์มีประสิทธิภาพใกล้เคียงกับ ต้นไม้การตัดสินใจและโครงข่ายประสาท

ตัวประมาณค่าความน่าจะเป็นจากความถี่ (Frequently-based Probability Estimate) เป็นเครื่องมือที่ถูกใช้ในการคำนวณค่าพารามิเตอร์ที่จำเป็นสำหรับตัวแบบนาอีฟเบย์ ได้แก่ ความน่าจะเป็นของชั้นข้อมูล และความน่าจะเป็นของชั้นข้อมูลที่ระบุค่าฟีเจอร์ที่สนใจ ความน่าจะเป็นเหล่านี้สามารถคำนวณได้จากแบบฝึกฝนในเซตข้อมูล โดยวิธีการคำนวณความน่าจะเป็นดังกล่าวแสดงตาม ตัวอย่างด้านล่างนี้

ตัวอย่างที่ 3 กำหนดให้เซตฝึกฝนมีแบบจำนวน 100 แบบ จากชั้นข้อมูลต่าง ๆ ดังนี้

จำนวนของแบบในชั้นข้อมูลที่ 1 = 40

จำนวนของแบบในชั้นข้อมูลที่ 2 = 30

จำนวนของแบบในชั้นข้อมูลที่ 3 = 30

ความน่าจะเป็นก่อนของชั้นข้อมูลที่ 1 = 40/100 = 0.4

ความน่าจะเป็นก่อนของชั้นข้อมูลที่ 2 = 30/100 = 0.3

ความน่าจะเป็นก่อนของชั้นข้อมูลที่ 3 = 30/100 = 0.3

กำหนดให้ฟีเจอร์ที่ 1 ของเซตข้อมูลมีลักษณะเป็น ทวิภาค (Binary) คือ ค่าที่เป็นไปได้มีแค่ 0 หรือ 1 เท่านั้น ถ้าชั้นข้อมูลที่ 1 มี 30 แบบ ที่ค่าในฟีเจอร์นี้เป็น 0 และ 10 แบบ ที่ค่าในฟีเจอร์นี้เป็น 1 แล้ว

ความน่าจะเป็นก่อนของชั้นข้อมูลที่ 1 ที่ฟีเจอร์ที่ 1 มีค่าเป็น 0 = 30/40 = 0.75

ความน่าจะเป็นก่อนของชั้นข้อมูลที่ 1 ที่ฟีเจอร์ที่ 1 มีค่าเป็น 1 = 10/40 = 0.25

3. ตัวจำแนกนาอีฟเบส์

ตัวจำแนกนาอีฟเบส์ถูกออกแบบโดยการรวมตัวแบบความน่าจะเป็นเข้ากับ กฎการตัดสินใจแมป (MAP: Maximum a Posteriori) โดยกฎนี้จะใช้ฟังก์ชัน “classify” สำหรับการเลือกสมมติฐานที่น่าจะเกิดขึ้นมากที่สุด หรือค่าความน่าจะเป็นหลังสูงสุด ฟังก์ชันดังกล่าวแสดงรายละเอียดการคำนวณดังสมการด้านล่างนี้

$$\text{classify}(f_1,\ldots,f_n) = \arg\max_{c}\, p(C=c)\prod_{i=1}^{n} p(F_i = f_i \mid C=c)$$

ตัวแปรต่าง ๆ ที่เกี่ยวข้อง ได้แก่

n คือ จำนวนฟีเจอร์ในเซตข้อมูล

$F_1, \ldots, F_n$ คือ ฟีเจอร์ ตัวแปรที่ 1 ถึงตัวแปรที่ n ของแบบที่ต้องการจำแนก

$f_1, \ldots, f_n$ คือ ค่าฟีเจอร์ ค่าที่ 1 ถึงค่าที่ n ของแบบที่ต้องการจำแนก

C คือ ตัวแปรชั้นข้อมูล

c คือ ค่าชั้นข้อมูล

argmax คือ ฟังก์ชันค้นหาความน่าจะเป็นสูงสุด จากความน่าจะเป็นของชั้นข้อมูลทั้งหมด

i คือ จำนวนเต็ม ซึ่งมีค่าตั้งแต่ 1 ถึง n

$\prod$ คือ ตัวดำเนินการหาค่า ผลคูณรวม (Product)

p(C = c) คือ ฟังก์ชันความน่าจะเป็น ที่คำนวณจากสัดส่วนของจำนวนแบบที่อยู่ในชั้นข้อมูล c เมื่อเปรียบเทียบกับแบบทั้งหมดในเซตข้อมูล

$p(F_i = f_i \mid C = c)$ คือ ฟังก์ชันความน่าจะเป็น ที่คำนวณจากสัดส่วนของจำนวนแบบในชั้นข้อมูล c ที่ค่าตัวแปร F_i มีค่าเท่ากับ f_i เมื่อเปรียบเทียบกับแบบทั้งหมดในชั้นข้อมูล c

ฟังก์ชัน classify รับข้อมูลเข้าเป็น แบบที่ต้องการจำแนกที่มีค่าฟีเจอร์ $f_1, \ldots, f_n$ และ ได้ผลลัพธ์เป็นชั้นข้อมูล c กล่าวคือแบบฝึกฝน $f_1, \ldots, f_n$ ถูกจำแนกให้อยู่ในชั้นข้อมูล c การจำแนกตัวแบบนาอีฟเบส์ สรุปเป็นขั้นตอน ได้ดังต่อไปนี้

1. คำนวณความน่าจะเป็นของแบบทั้งหมดแต่ละชั้นข้อมูล จากสัดส่วนของแบบในชั้นข้อมูลนั้นเปรียบเทียบกับแบบทั้งหมดในเซตข้อมูล
2. คำนวณความน่าจะเป็นของแบบที่ฟีเจอร์ F_i มีค่า f_i ในแต่ละชั้นข้อมูล เมื่อเปรียบเทียบกับแบบทั้งหมดในชั้นข้อมูลนั้น
3. คำนวณผลคูณรวมของความน่าจะเป็น<u>ที่อยู่ในชั้นข้อมูลเดียวกัน</u>จากข้อ 1 และ ข้อ 2
4. ผลคูณรวมของชั้นข้อมูลใดมีค่าสูงสุด จะจำแนกแบบให้อยู่ชั้นข้อมูลนั้น

ในกรณีที่แบบในชั้นข้อมูลและฟีเจอร์ที่สนใจไม่ปรากฏอยู่ในเซตฝึกฝน ความน่าจะเป็นที่คำนวณได้จะมีค่าเป็นศูนย์ อย่างไรก็ตาม การสร้างตัวแบบนาอีฟเบย์เกิดจากผลคูณรวมของความน่าจะเป็นของแต่ละฟีเจอร์ในชั้นมูลเดียวกัน ดังนั้น ถ้ามีความน่าจะเป็นใดมีค่าเป็นศูนย์แล้ว จะทำให้ค่าผลคูณรวมมีค่าเป็นศูนย์ตาม ซึ่งเท่ากับว่าความน่าจะเป็นทั้งหมดจะไม่ถูกพิจารณา แต่ในความเป็นจริงแล้ว ความน่าจะเป็นที่มากกว่าศูนย์มีความหมายและควรถูกพิจารณา เพื่อแก้ไขปัญหาดังกล่าว ในกระบวนการสร้างตัวแบบจะแทนที่ความน่าจะเป็นศูนย์ด้วยค่าคงที่ขนาดเล็กแทน เพื่อให้ผลคูณรวมไม่เป็นศูนย์ การคำนวณตัวแบบนาอีฟเบส์ แสดงได้ตามตัวอย่างที่ 4 ถึง ตัวอย่างที่ 6

ตัวอย่างที่ 4 กำหนดเซตข้อมูลฝึกฝนที่ประกอบไปด้วย 10 แบบ แสดงตาม ตารางที่ 4.1 และชั้นข้อมูล (Class) ได้แก่ฟีเจอร์ Tasty จงคำนวณว่าแบบใหม่ X ที่มีค่าฟีเจอร์ คือ Cook = Sita, Mood = Bad, Cuisine = Continental ควรถูกจำแนกในชั้นข้อมูลใด

การคำนวณตัวแบบนาอีฟเบส์ ในการจำแนกแบบ X ว่าควรอยู่ในชั้นข้อมูลใด (Tasty เป็น Yes หรือ Tasty เป็น No) มีขั้นตอนดังต่อไปนี้

1) คำนวณความน่าจะเป็นของแบบในแต่ละชั้นข้อมูล

เนื่องจากในเซตข้อมูลมีแบบทั้งหมด 10 แบบ โดยอยู่ในชั้นข้อมูล Yes จำนวน 6 แบบ และอยู่ในชั้นข้อมูล No จำนวน 4 แบบ ดังนั้น

P(Tasty = Yes) = 6/10 = 0.60

P(Tasty = No) = 4/10 = 0.40

2) คำนวณความน่าจะเป็นของฟีเจอร์ Cook = Sita ในแต่ละชั้นข้อมูล

การคำนวณความน่าจะเป็นในหัวข้อย่อยนี้ จะสนใจแต่แบบที่มีค่าฟีเจอร์ Cook = Sita เท่านั้น เนื่องจากเป็นเงื่อนไขเดียวกันกับแบบ X ที่ต้องการจำแนก แต่จะคำนวณแยกเป็น 2 กรณีได้แก่ กรณีที่ Tasty = Yes และ Tasty = No

แบบในชั้นข้อมูล Yes มีจำนวน 6 แบบ โดยมี 2 แบบที่ Cook คือ Sita และ อีก 4 แบบที่ Cook ไม่ใช่ Sita ดังนั้น ความน่าจะเป็นของแบบในชั้นข้อมูล Yes ที่ Cook = Sita คำนวณได้ ดังนี้

P(Cook = Sita | Tasty = Yes) = 2/6 = 0.33

แบบในชั้นข้อมูล No มีจำนวน 4 แบบ โดยไม่มีแบบใดเลยที่ Cook คือ Sita ในขณะที่ แบบทั้งหมด Cook เป็นพ่อครัวคนอื่น ดังนั้น ความน่าจะเป็นของแบบในชั้นข้อมูล No ที่ Cook = Sita คำนวณได้ ดังนี้

P(Cook = Sita | Tasty = No) = 0/4 = 0.00

เนื่องจากความน่าจะเป็นที่คำนวณได้มีค่าเป็นศูนย์ เมื่อนำไปคูณกับความน่าจะเป็นอื่น จะทำให้ผลคูณร่วมกลายเป็นศูนย์ จึงหลีกเลี่ยงโดยการกำหนดค่าคงที่ขนาดเล็กแทน ดังนี้

P(Cook = Sita | Tasty = No) $\longrightarrow$ 0.01

3) คำนวณความน่าจะเป็นของฟีเจอร์ Mood = Bad ในแต่ละชั้นข้อมูล

แบบในชั้นข้อมูล Yes มีจำนวน 6 แบบ โดยมี 2 แบบที่ Mood คือ Bad และ อีก 4 แบบที่ Mood คือ Good ดังนั้น ความน่าจะเป็นของแบบในชั้นข้อมูล Yes ที่ Mood = Bad คำนวณได้ ดังนี้

P(Mood = Bad | Tasty = Yes) = 2/6 = 0.33

แบบในชั้นข้อมูล No มีจำนวน 4 แบบ โดยมี 3 แบบที่ Mood คือ Bad และอีก 1 แบบที่ Mood คือ Good ดังนั้น ความน่าจะเป็นของแบบในชั้นข้อมูล No ที่ Mood = Bad คำนวณได้ ดังนี้

P(Mood = Bad | Tasty = No) = 3/4 = 0.75

4) คำนวณความน่าจะเป็นของฟีเจอร์ Cuisine = Continental ในแต่ละชั้นข้อมูล

แบบในชั้นข้อมูล Yes มีจำนวน 6 แบบ โดยมี 2 แบบที่ Mood คือ Bad และ อีก 4 แบบที่ Mood คือ Good ดังนั้น ความน่าจะเป็นของแบบในชั้นข้อมูล Yes ที่ Mood = Bad คำนวณได้ ดังนี้

แบบในชั้นข้อมูล Yes ที่ Cuisine = Continental มีจำนวน 2 แบบ จากแบบทั้งหมด 6 แบบ และแบบในชั้นข้อมูล No ที่ Cuisine = Continental มีจำนวน 3 แบบทั้งหมด จาก 4 แบบ ดังนั้น ความน่าจะเป็นหลังของ Cuisine คือ Continental และ Tasty มีค่าเป็น Yes และ No คำนวณได้ ดังนี้

P(Cuisine = Continental | Tasty = Yes) = 2/6 = 0.33

P(Cuisine = Continental | Tasty = No) = 3/4 = 0.75

5) คำนวณความน่าจะเป็นของแบบ X ในแต่ละชั้นข้อมูล

ขั้นตอนสุดท้าย ทำการคำนวณความน่าจะเป็นของแบบ X ในกรณีที่อยู่ในชั้นข้อมูล Yes และ No เพื่อทำการเปรียบเทียบกัน โดยที่ความน่าจะเป็นของแบบ X ที่ถูกจำแนกในชั้นข้อมูล Yes ถูกคำนวณจาก ผลคูณรวมของความน่าจะเป็นทั้งหมดจากข้อ 1) – 4) ที่อยู่ในชั้นข้อมูลเดียวกัน (Tasty = Yes) ส่วนการคำนวณความน่าจะเป็นของชั้นข้อมูล No ก็คำนวณในทำนองเดียวกัน ดังนั้น ความน่าจะเป็นของแบบ X ในแต่ละชั้นข้อมูล คำนวณได้ ดังนี้

P(Tasty = Yes | X) = P(Tasty = Yes) x
P(Cook = Sita | Tasty = Yes) x
P(Mood = Bad | Tasty = Yes) x
P(Cuisine = Continental | Tasty = Yes)
= 0.60 x 0.33 x 0.33 x 0.33
= 0.0216

P(Tasty = No | X) = P(Tasty = No) x
P(Cook = Sita | Tasty = No) x
P(Mood = Bad | Tasty = No) x
P(Cuisine = Continental | Tasty = No)
= 0.40 x 0.01 x 0.75 x 0.75
= 0.00225

เนื่องจาก P(Tasty = Yes | X) > P(Tasty = No | X) ดังนั้น X จึงถูกจำแนกให้อยู่ในชั้นข้อมูล Yes หรืออาจกล่าวเป็นคำอธิบายได้ว่า ถ้าวันที่เดินทางไปรับประทานอาหารประเภท Continental ที่กุ๊กในภัตตาคารชื่อ Sita กำลังอารมณ์เสียอยู่ แล้วจะได้อาหารที่มีรสชาติอร่อย

ตารางที่ 4.1 เซตฝึกฝนตัวอย่าง

Cook	Mood	Cuisine	Tasty
Sita	Bad	Indian	Yes
Sita	Good	Continental	Yes
Asha	Bad	Indian	No
Asha	Good	Indian	Yes
Usha	Bad	Indian	Yes
Usha	Bad	Continental	No
Asha	Bad	Continental	No
Asha	Good	Continental	Yes
Usha	Good	Indian	Yes
Usha	Good	Continental	No

ตัวอย่างที่ 5 กำหนดเซตข้อมูลฝึกฝนตามตารางที่ 4.2 จำนวน 6 แบบ ที่เก็บจากแบบสอบถามพฤติกรรมการบริโภคเครื่องดื่มโคล่า แบบสอบถามนี้จะสำรวจว่าผู้บริโภคมีความชื่นชอบ น้ำอัดลมโบราณ (Traditional Cola) หรือไม่ จากการพิจารณาความชื่นชอบเครื่องดื่มโคล่า 3 ยี่ห้อได้แก่ โค้ก (Coke) เป๊ปซี่ (Pepsi) และ เอส (est) ค่าที่จัดเก็บลงเซตข้อมูลมี 2 ประเภท ได้แก่ ดื่ม (DRINK) และ ไม่ดื่ม (NOT_DRINK) จงทำนายพฤติกรรมผู้บริโภครายหนึ่งที่ชอบดื่มเอสแต่ไม่ชอบดื่มโค้กและเป๊ปซี่ว่าจะชอบดื่มน้ำอัดลมโบราณหรือไม่

จากโจทย์สิ่งที่ต้องการจำแนกได้แก่ น้ำอัดลมโบราณ ดังนั้นตัวแปรชั้นข้อมูลคือ Traditional Cola กำหนดให้ให้ผู้บริโภครายหนึ่งคือแบบทดสอบ Y ดังนั้นสามารถเขียน Y ให้อยู่ในรูปของแบบได้ตามสมการด้านล่างนี้

Y = (NOT_DRINK, NOT_DRINK, DRINK)

1) คำนวณความน่าจะเป็นของแบบในแต่ละชั้นข้อมูล

เซตข้อมูลฝึกฝนนี้มีแบบทั้งหมด 6 แบบ โดยอยู่ในชั้นข้อมูล Traditional Cola = NOT_DRINK และ Traditional Cola = DRINK อย่างละครึ่ง ดังนั้น

P(Traditional Cola = NOT_DRINK) = 3/6 = 1/2

P(Traditional Cola = DRINK) = 3/6 = 1/2

2) คำนวณความน่าจะเป็นของฟีเจอร์ Coke = NOT_DRINK ในแต่ละชั้นข้อมูล

ชั้นข้อมูล Traditional Cola = NOT_DRINK มีแบบจำนวน 3 แบบ โดยมีแบบที่ค่าฟีเจอร์ Coke เท่ากับ NOT_DRINK จำนวน 2 แบบ ดังนั้น

P(Coke = NOT_DRINK | Traditional Cola = NOT_DRINK) = 2/3

แบบจากชั้นข้อมูล Traditional Cola = DRINK จำนวน 3 แบบ มีเพียงแบบเดียวที่ค่าฟีเจอร์ Coke เท่ากับ NOT_DRINK ดังนั้น

P(Coke = NOT_DRINK | Traditional Cola = DRINK) = 1/3

3) คำนวณความน่าจะเป็นของฟีเจอร์ Pepsi = NOT_DRINK ในแต่ละชั้นข้อมูล

แบบที่ค่าฟีเจอร์ Pepsi เท่ากับ NOT_DRINK ซึ่งอยู่ในชั้นข้อมูล Traditional Cola = NOT_DRINK มีจำนวน 2 จาก 3 แบบ และอยู่ในชั้นข้อมูล Traditional Cola = DRINK มีจำนวน 1 จาก 3 แบบ ดังนั้น

P(Pepsi = NOT_DRINK | Traditional Cola = NOT_DRINK) = 2/3

P(Pepsi = NOT_DRINK | Traditional Cola = DRINK) = 1/3

4) คำนวณความน่าจะเป็นของฟีเจอร์ est = DRINK ในแต่ละชั้นข้อมูล

มีแบบเพียงแบบเดียวในชั้นข้อมูล Traditional Cola = NOT_DRINK ที่ฟีเจอร์ est มีค่าเท่ากับ DRINK และแบบทั้งหมดในชั้นข้อมูล Traditional Cola = DRINK ฟีเจอร์ est มีค่าเท่ากับ DRINK ดังนั้น

P(est = DRINK | Traditional Cola = NOT_DRINK) = 1/3

P(est = DRINK | Traditional Cola = DRINK) = 3/3

5) คำนวณความน่าจะเป็นของแบบ Y ในแต่ละชั้นข้อมูล

คำนวณความน่าจะเป็นรวมของแบบ Y ในชั้นข้อมูล Traditional Cola = NOT_DRINK และ Traditional Cola = DRINK จากความน่าจะเป็นในชั้นข้อมูลเดียวกันของข้อ 1) – 4) ผลการคำนวณแสดงได้ดังนี้

P(Traditional Cola = NOT_DRINK | Y) = P(Traditional Cola = NOT_DRINK) x
P(Coke = NOT_DRINK | Traditional Cola = NOT_DRINK) x
P(Pepsi = NOT_DRINK | Traditional Cola = NOT_DRINK) x
P(est = DRINK | Traditional Cola = NOT_DRINK)
= 1/2 x 2/3 x 2/3 x 1/3
= 2/27

P(Traditional Cola = DRINK | Y) = P(Traditional Cola = DRINK) x
P(Coke = NOT_DRINK | Traditional Cola = DRINK) x
P(Pepsi = NOT_DRINK | Traditional Cola = DRINK) x
P(est = DRINK | Traditional Cola = DRINK) x
= 1/2 x 1/3 x 1/3 x 3/3
= 1/18

เนื่องจาก P(Traditional Cola = NOT_DRINK | Y) > P(Traditional Cola = DRINK | Y) ดังนั้น Y จึงถูกจำแนกให้อยู่ในชั้นข้อมูล Traditional Cola = NOT_DRINK กล่าวคือ ถ้าผู้บริโภคดื่ม est เพียงยี่ห้อเดียว โดยไม่ดื่ม Coke และ Pepsi แล้วผู้บริโภคจะมีความชื่นชอบในการดื่มน้ำอัดลมโบราณ

ตารางที่ 4.2 เซตฝึกฝนตัวอย่าง 2

Coke	Pepsi	est	Traditional Cola
NOT_DRINK	NOT_DRINK	NOT_DRINK	NOT_DRINK
NOT_DRINK	DRINK	DRINK	NOT_DRINK
DRINK	NOT_DRINK	NOT_DRINK	NOT_DRINK
NOT_DRINK	DRINK	DRINK	DRINK
DRINK	NOT_DRINK	DRINK	DRINK
DRINK	DRINK	DRINK	DRINK

ตัวอย่างที่ 6 จากการสำรวจพฤติกรรมของนักศึกษาภาควิชาวิทยาการคอมพิวเตอร์แห่งหนึ่ง ในการกดหรือไม่กด Like รูปภาพอาหารใน Facebook แสดงตามเซตข้อมูลในตารางที่ 4.3 โดยมีปัจจัยที่เกี่ยวข้องได้แก่

- **FRIEND** รูปภาพเป็นของ Friend นักศึกษาหรือไม่
- **HUNGRY** นักศึกษามีความหิวในขณะรับชมรูปภาพหรือไม่
- **NIGHT** รูปภาพถูกโพสในเวลากลางคืนหรือไม่

กำหนดให้สถานะ 0 คือ ไม่ใช่ และ 1 คือ ใช่ ยกตัวอย่างเช่น แบบในแถวแรกสามารถตีความได้ว่า
“นักศึกษาไม่กดไลค์รูปภาพที่คนโพสไม่ใช่ Friend ขณะนั้นเป็นเวลากลางวัน และนักศึกษากำลังอิ่ม”

ให้ทำนายแบบทดสอบด้านล่างนี้ โดยใช้ตัวจำแนกนาอีฟเบส์
“นักศึกษาจะกดไลค์รูปภาพของ Friend ที่โพสในเวลากลางวัน ในขณะที่นักศึกษากำลังอิ่มอยู่หรือไม่”

สิ่งที่ต้องการทำนายจากตัวอย่างนี้ได้แก่ การกด (หรือไม่กด) ไลค์ ดังนั้น ตัวแปรชั้นข้อมูลคือ LIKE กำหนดให้แบบสดทอบ Z แทน นักศึกษารายหนึ่ง ดังนั้นสมการด้านล่างนี้แทนค่าฟีเจอร์ต่าง ๆ ในแบบทดสอบ Z

$$Z = (1, 0, 0)$$

1) คำนวณความน่าจะเป็นของแบบในแต่ละชั้นข้อมูล

เซตฝึกฝนนี้มี 2 ชั้นข้อมูล โดยที่แต่ละชั้นข้อมูลมีแบบบรรจุอยู่เท่า ๆ กัน ดังนั้น

$$P(LIKE = 0) = 3/6 = 1/2$$

$$P(LIKE = 1) = 3/6 = 1/2$$

2) คำนวณความน่าจะเป็นของฟีเจอร์ FRIEND = 1 ในแต่ละชั้นข้อมูล

ชั้นข้อมูล LIKE = 0 มีเพียงแบบเดียวที่ค่าฟีเจอร์ FRIEND = 1 และ ชั้นข้อมูล LIKE = 1 แบบทุกแบบมีค่าตัวแปรคุณลักษณ FRIEND = 1 ดังนั้น

$$P(FRIEND = 1 \mid LIKE = 0) = 1/3$$

$$P(FRIEND = 1 \mid LIKE = 1) = 3/3$$

3) คำนวณความน่าจะเป็นของฟีเจอร์ HUNGRY = 0 ในแต่ละชั้นข้อมูล

แบบที่ฟีเจอร์ HUNGRY มีค่าเท่ากับ 0 อยู่ในชั้นข้อมูล LIKE = 0 จำนวน 1 แบบ และอยู่ในชั้นข้อมูล LIKE = 1 จำนวน 2 แบบ ดังนั้น

P(HUNGRY = 0 | LIKE = 0) = 2/3

P(HUNGRY = 0 | LIKE = 1) = 1/3

4) คำนวณความน่าจะเป็นของฟีเจอร์ NIGHT = 0 ในแต่ละชั้นข้อมูล

แบบที่ฟีเจอร์ NIGHT มีค่าเท่ากับ 0 อยู่ในชั้นข้อมูล LIKE = 0 จำนวน 1 แบบ และอยู่ในชั้นข้อมูล LIKE = 1 จำนวน 2 แบบ ดังนั้น

P(NIGHT = 0 | LIKE = 0) = 2/3

P(NIGHT = 0 | LIKE = 1) = 1/3

5) คำนวณความน่าจะเป็นของแบบ Z ในแต่ละชั้นข้อมูล

ความน่าจะเป็นรวมของแบบ Z ที่ชั้นข้อมูล LIKE มีค่า 0 และ 1 สามารถคำนวณได้ตามสมการด้านล่างนี้

$$
\begin{aligned}
P(LIKE = 0 \mid Z) &= P(LIKE = 0) \times \\
&\quad P(FRIEND = 1 \mid LIKE = 0) \times \\
&\quad P(HUNGRY = 0 \mid LIKE = 0) \times \\
&\quad P(NIGHT = 0 \mid LIKE = 0) \\
&= 1/2 \times 1/3 \; 2/3 \times 2/3 \\
&= 4/54
\end{aligned}
$$

$$
\begin{aligned}
P(LIKE = 1 \mid Z) &= P(LIKE = 1) \times \\
&\quad P(FRIEND = 1 \mid LIKE = 1) \times \\
&\quad P(HUNGRY = 0 \mid LIKE = 1) \times \\
&\quad P(NIGHT = 0 \mid LIKE = 1) \\
&= 1/2 \times 3/3 \times 1/3 \times 1/3 \\
&= 3/54
\end{aligned}
$$

จากความน่าจะเป็นรวม P(LIKE = 0 | Z) มีโอกาสเกิดมากกว่า P(LIKE = 1 | Z) ดังนั้นจึงทำนาย Z จะอยู่ในชั้นข้อมูล LIKE = 0 สำหรับโจทย์ข้อนี้ นักศึกษาจะไม่กดไลค์รูปภาพของ Friend ที่โพสในเวลากลางวัน ในขณะที่นักศึกษากำลังอิ่มอยู่

ตารางที่ 4.3 เซตฝึกฝนตัวอย่าง 3

FRIEND	HUNGRY	NIGHT	LIKE
0	0	0	0
0	1	0	0
1	0	1	0
1	1	0	1
1	1	1	1
1	0	1	1

4. เครือข่ายความเชื่อเบร์เซี่ยน

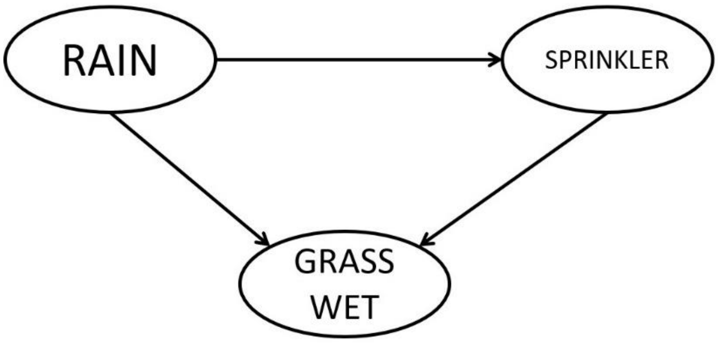
เครือข่ายความเชื่อเบร์เซี่ยน (Bayesian Belief Network) คือตัวแบบที่ประยุกต์ใช้ความน่าจะเป็นจากทฤษฎีเบส์สำหรับการทำนายชั้นข้อมูล ตัวแบบนี้มีลักษณะเป็นกราฟฟิกทำให้มนุษย์สามารถทำความเข้าใจขั้นตอนการจำแนกแบบได้และสามารถออกแบบได้โดยมนุษย์ โครงสร้างข้อมูลที่ถูกประยุกต์ใช้ในตัวจำแนกชนิดนี้ได้แก่ กราฟอวัฏจักรระบุทิศทาง (Directed Acyclic Graph) โดยที่ โหนด (Node) แทนตัวแปรหรือฟีเจอร์ในเซตข้อมูล และ เส้นเชื่อมแบบมีทิศทาง (Directed Edge) แทนความสัมพันธ์แบบมีเงื่อนไข (Conditional Relationships) ถ้ากราฟมีเส้นเชื่อมในทิศทางเริ่มต้นจากโหนด A ไปยังปลายทางที่ โหนด B แล้ว A เป็น พ่อแม่ (Parent) ของ B หรืออาจกล่าวได้ว่า ตัวแปร A มีอิทธิพลต่อตัวแปร B

ตัวอย่างของเครือข่ายความเชื่อเบร์เซี่ยนแสดงตามรูปที่ 4.1 ด้านล่างนี้ กำหนดให้โหนดต่าง ๆ เก็บค่าที่เป็นไปได้เพียง 2 ค่า ได้แก่ จริง (True) และ เท็จ (False) อย่างใดอย่างหนึ่งเท่านั้น โดยแต่ละโหนดมีความหมาย ดังต่อไปนี้

- โหนด RAIN คือ เหตุการณ์ที่ฝนตก ค่าที่เป็นไปได้คือ ฝนตก หรือ ฝนไม่ตก
- โหนด SPRINGER คือ สถานะของปริงเกอร์ ค่าที่เป็นไปได้คือ เปิดใช้งาน หรือ ปิด
- โหนด GRASSWET คือ เหตุการณ์ที่สนามหญ้าเปียก ค่าที่เป็นไปได้คือ เปียก หรือ แห้ง

ตัวอย่างความสัมพันธ์แบบมีเงื่อนไข อธิบายได้ดังนี้

- โหนด RAIN และ โหนด SPRINGER เป็นพ่อแม่ของโหนด GRASSWET หมายความว่า RAIN และ SPRINGER มีอิทธิพลต่อ GRASSWET กล่าวคือสนามหญ้าจะเปียก ถ้าฝนตก หรือมีการเปิดสปริงเกอร์เพื่อรดน้ำ
- โหนด RAIN เป็นพ่อแม่ของโหนด SPRINGER หมายความว่า RAIN มีอิทธิพลต่อ SPRINGER กล่าวคือ สปริงเกอร์จะถูกปิดถ้าฝนตก หรืออาจกล่าวได้ว่า ถ้าสนามหญ้าเปียกด้วยฝน ก็ไม่จำเป็นต้องใช้งานสปริงเกอร์อีกต่อไป ในทางกลับกัน SPRINGER ไม่มีอิทธิพลต่อ RAIN เพราะว่าสถานะของสปริงเกอร์ไม่ส่งผลต่อการตกหรือไม่ตกของฝน



รูปที่ 4.1 ความสัมพันธ์แบบมีเงื่อนไข

กำหนดให้ X คือแบบใหม่ที่ยังไม่ทราบชั้นข้อมูล การทำนายโดยใช้เครือข่ายความเชื่อเบร์เซี่ยนเริ่มต้นจากการคำนวณ ความน่าจะเป็นแบบมีเงื่อนไขของแบบ X ซึ่งแสดงตามสมการด้านล่างนี้

$$P(X_1,...,X_n) = \prod_{i=1}^{n} P(X_i \mid \text{parents}(X_i))$$

ตัวแปรต่าง ๆ ที่เกี่ยวข้อง ได้แก่

n คือ จำนวนฟีเจอร์ในเซตข้อมูล

$X_1, \ldots, X_n$ คือ ฟีเจอร์ ตัวแปรที่ 1 ถึงตัวแปรที่ n ของแบบ X

parents(X_i) คือฟังก์ชันที่ส่งคืน เซตของพ่อแม่ของแบบ X

i คือ จำนวนเต็ม ซึ่งมีค่าตั้งแต่ 1 ถึง n

$\prod$ คือ ตัวดำเนินการหาค่า ผลคูณรวม (Product)

$P(X_i \mid \text{parents}(X_i))$ คือ ความน่าจะเป็นของ X_i ที่มีความสัมพันธ์แบบมีเงื่อนไขกับ พ่อแม่ของ X_i

$P(X_1, \ldots, X_n)$ คือความน่าจะเป็นของแบบที่ต้องการทำนาย

ตัวแบบเครือข่ายความเชื่อเบร์เซี่ยนเป็นตัวจำแนกที่มีข้อดี ดังต่อไปนี้

- **การแสดงภาพ (Visualization)**

  โครงสร้างภายในของตัวแบบสามารถมองเห็นได้อย่างชัดเจน ส่งผลให้การแก้ไขปรับปรุง หรือออกแบบตัวแบบใหม่ ทำได้ง่าย

- **ความสัมพันธ์ (Relationship)**

  ตัวแบบแสดงความสัมพันธ์ (หรือไม่สัมพันธ์) ระหว่างฟีเจอร์ต่าง ๆ ในเซตข้อมูลได้ โดยดูจากโครงสร้างของตัวแบบ

- **การคำนวณ (Computation)**

  ตัวแบบสามารถหาความน่าจะเป็นที่ซับซ้อนได้ โดยคำนวณจากตารางความน่าจะเป็นของตัวแบบ

ขั้นตอนการออกแบบเครือข่ายความเชื่อเบร์เซี่ยน มีสิ่งที่ต้องคำนึงถึง ดังต่อไปนี้

- **ตัวแปร (Variables)**

  เริ่มต้นจากศึกษาปัญหาที่สนใจสร้างตัวจำแนกแบบ ระบุตัวแปรทั้งหมดให้เป็นตัวแทนของฟีเจอร์ในเซตข้อมูล ตัวอย่างของตัวแปรได้แก่ RAIN SPRINKLER และ GLASS WET ตามรูปที่ 4.1

- **ความสัมพันธ์แบบมีเงื่อนไข (Conditional Relationships)**

  วิเคราะห์ความสัมพันธ์แบบมีเงื่อนไขระหว่างตัวแปรต่าง ๆ ว่ามีคู่ของตัวแปรใดบ้างที่มีอิทธิพลต่อกัน ตัวอย่างเช่น ฝนตกมีอิทธิพลต่อสนามหญ้าเปียก ตามรูปที่ 4.1

- **การกระจายความน่าจะเป็น (Probability Distributions)**

  คำนวณความน่าจะเป็นของแต่ละตัวแปรว่ามีโอกาสเกิดขึ้นมากน้อยแค่ไหน เช่น สังเกตการตกของฝนในช่วงเวลาหนึ่งพบว่า ความน่าจะเป็นที่ฝนตกคือ 0.2 ในขณะที่ ความน่าจะเป็นที่ฝนไม่ตกคือ 0.8

สำหรับบางปัญหาจำเป็นต้องใช้ผู้เชี่ยวชาญในสาขานั้น ๆ เช่น ปัญหาการวินิจฉัยโรคมะเร็งเต้านมอาจจะต้องอาศัยแพทย์เฉพาะทางมาช่วยในการออกแบบตัวจำแนกแบบ ตัวอย่างของการคำนวณความน่าจะเป็นในเครือข่ายความเชื่อเบร์เซี่ยน แสดงตามตัวอย่างที่ 7 และ 8 ต่อไปนี้

ตัวอย่างที่ 7 รามเป็นนักเรียน เขารักการดูภาพยนตร์ เขาจะไปโรงภาพยนตร์ถ้าเขามีเงินในกระเป๋าสตางค์ ในทางกลับกัน ถ้าฝนตกเขาจะไม่ไปโรงภาพยนตร์และเลือกที่จะดูโทรทัศน์อยู่ที่บ้านแทน นอกจากนี้ เขายังอุทิศเวลาเพื่อการเรียนอีกด้วย ให้ทุกเหตุการณ์ที่กล่าวมานี้เก็บค่าสถานะเป็นไปได้เพียงแค่ 2 ค่าเท่านั้น คือ จริง (T) หรือ เท็จ (F) จงคำนวณหาความน่าจะเป็นของเหตุการณ์ด้านล่างนี้ โดยอาศัยเครือข่ายความเชื่อเบร์เซี่ยนในรูปที่ 4.2

“รามไม่มีเงินในขณะที่ฝนไม่ตก เขาตัดสินใจไม่ดูโทรทัศน์และออกไปชมภาพยนต์แทน”

กำหนดให้ $\neg$A แทน นิเสธ (Negation) ของ ประพจน์ (Proposition) A กำหนดตัวแปรต่าง ๆ ที่เกี่ยวข้อง ดังต่อไปนี้

- M คือ แรมมีเงินในกระเป๋าสตางค์
- R คือ ฝนตก
- G คือ แรมออกไปดูภาพยนตร์
- S คือ แรมทบทวนบทเรียน
- T คือ แรมดูโทรทัศน์

จากเครือข่ายความเชื่อเบร์เซี่ยน วิเคราะห์ความสัมพันธ์แบบมีเงื่อนไขของตัวแปรที่มีอิทธิพลต่อตัวแปรอื่น ได้ดังต่อไปนี้

- ไม่มีตัวแปรใดมีอิทธิพลต่อตัวแปร M และ R ดังนั้น ความน่าจะเป็นของ M และ R จึงไม่ขึ้นอยู่กับเงื่อนไขของตัวแปรใด
- ตัวแปร M และ R มีผลกระทบต่อตัวแปร G ดังนั้น ความน่าจะเป็นของ G จึงขึ้นอยู่กับเงื่อนไขของ M และ R
- ตัวแปร G มีอิทธิพลต่อตัวแปร S และ T ดังนั้น ความน่าจะเป็นของ S จึงขึ้นอยู่กับเงื่อนไขของ G ในทำนองเดียวกัน ความน่าจะเป็นของ T ขึ้นอยู่กับเงื่อนไขของ G

สัญลักษณ์ต่าง ๆ ในแต่ละองค์ประกอบของเหตุการณ์ในตัวอย่างนี้ มีดังต่อไปนี้

- P($\neg$R) คือ ความน่าจะเป็นขององค์ประกอบ *“ฝนไม่ตก”* เนื่องจาก R เป็นอิสระจากตัวแปรตัวอื่น ความน่าจะเป็นของ R (หรือ $\neg$R) จึงมีความน่าจะเป็นเพียงค่าเดียว
- P($\neg$M) คือ ความน่าจะเป็นขององค์ประกอบ *“รามไม่มีเงิน”* เนื่องจาก M เป็นอิสระจากตัวแปรตัวอื่น ความน่าจะเป็นของ M (หรือ $\neg$M) จึงมีความน่าจะเป็นเพียงค่าเดียว

- $P(G \mid \neg R \text{ and } \neg M)$ คือ ความน่าจะเป็นแบบมีเงื่อนไขขององค์ประกอบ *"รามออกไปชมภาพยนต์ถ้าฝนไม่ตกและเขาไม่มีเงิน"* เนื่องจากตัวแปร R และ M มีอิทธิพลต่อตัวแปร G ดังนั้น ความน่าจะเป็นของ G จึงขึ้นอยู่กับเงื่อนไขของ R และ M แต่จากองค์ประกอบ *"ฝนไม่ตก"* และ *"รามไม่มีเงิน"* ส่งผลให้เงื่อนไขของ G ได้แก่ $\neg R$ และ $\neg M$ ตามลำดับ
- $P(\neg T \mid G)$ คือ ความน่าจะเป็นแบบมีเงื่อนไขขององค์ประกอบ *"รามไม่ดูโทรทัศน์ถ้าเขาไปชมภาพยนต์"* เนื่องจากองค์ประกอบ *"รามไม่ดูโทรทัศน์"* คือ $\neg T$ นอกจากนี้ ตัวแปร G มีอิทธิพลต่อตัวแปร T ดังนั้น ความน่าจะเป็นของ $\neg T$ จึงขึ้นอยู่กับเงื่อนไขของ G แต่จากองค์ประกอบ *"รามไปชมภาพยนต์"* ส่งผลให้เงื่อนไขของ $\neg T$ คือ G

ขั้นตอนต่อไป คำนวณหาค่าต่าง ๆ จากสมการ โดยใช้ตารางความน่าจะเป็นในรูปที่ 4.2

1. คำนวณค่า $P(\neg R)$ เนื่องจาก $P(R) = 0.3$ ดังนั้น

$$\begin{aligned} P(\neg R) &= 1.0 - P(R) \\ &= 1.0 - 0.3 \\ &= 0.7 \end{aligned}$$

2. คำนวณค่า $P(\neg M)$ เนื่องจาก $P(M) = 0.6$ ดังนั้น

$$\begin{aligned} P(\neg M) &= 1.0 - P(R) \\ &= 1.0 - 0.6 \\ &= 0.4 \end{aligned}$$

3. คำนวณค่า $P(G \mid \neg R \text{ and } \neg M)$ เนื่องจากเหตุการณ์ $\neg R$ คือตัวแปร R ที่เก็บค่า F ในทำนองเดียวกัน เหตุการณ์ $\neg M$ คือตัวแป M ที่เก็บค่า F เช่นเดียวกัน ดังนั้น

$$\begin{aligned} P(G \mid \neg R \text{ and } \neg M) &= P(G \mid R = F \text{ and } M = F) \\ &= 0.1 \end{aligned}$$

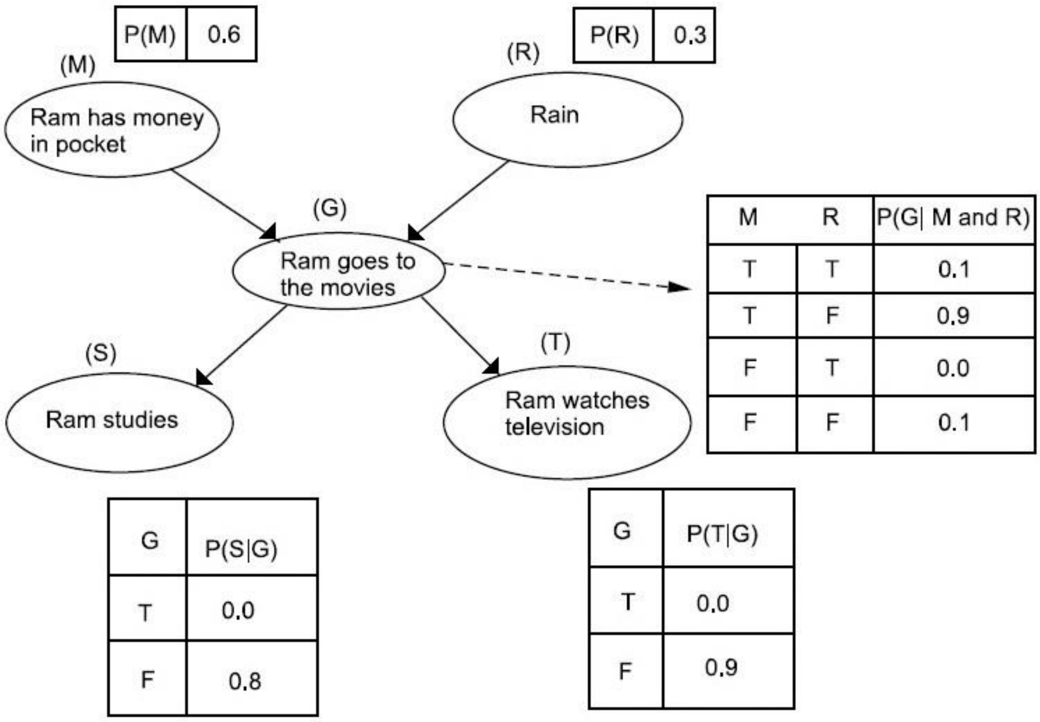
4. คำนวณค่า $P(\neg T \mid G)$ เนื่องจากเหตุการณ์ G คือตัวแปร G ที่เก็บค่า T

$$\begin{aligned} P(\neg T \mid G) &= 1.0 - P(T \mid G) \\ &= 1.0 - P(T \mid G = T) \\ &= 1.0 - 0.0 \\ &= 1.0 \end{aligned}$$

ขั้นตอนสุดท้าย รวมองค์ประกอบทั้งหมดเข้าเป็นเหตุการณ์ แสดงการคำนวณได้ตามสมการด้านล่างนี้

$$
\begin{aligned}
&P(\neg R) \times P(\neg M) \times P(G \mid \neg R \text{ and } \neg M) \times P(\neg T \mid G) \\
&\quad = 0.7 \times 0.4 \times 0.1 \times 1.0 \\
&\quad = 0.028
\end{aligned}
$$

จากโจทย์ในตัวอย่างนี้ ความน่าจะเป็นของเหตุการณ์ *“รามไม่มีเงินในขณะที่ฝนไม่ตก เขาตัดสินใจไม่ดูโทรทัศน์และออกไปชมภาพยนต์แทน”* มีค่าเท่ากับ 0.0028 หรือ 0.28%



รูปที่ 4.2 เครือข่ายความเชื่อของกิจกรรมในช่วงเย็นของราม

ตัวอย่างที่ 8 จากเครือข่ายความเชื่อเบร์เซี่ยนในรูปที่ 4.2 จงแสดงการคำนวณหาค่าความน่าจะเป็นสำหรับเหตุการณ์ด้านล่างนี้

“รามมีเงินในขณะที่ฝนตก เขาตัดสินใจไม่ออกไปชมภาพยนต์และไม่ทบทวนบทเรียน”

กำหนดตัวแปรต่าง ๆ ที่เกี่ยวข้อง ได้แก่ M, R, G, S และ T เหมือนตัวอย่างที่ 7 เนื่องจากเป็นเครือข่ายความเชื่อเบร์เซี่ยนเดียวกัน สัญลักษณ์ต่าง ๆ ในแต่ละองค์ประกอบของเหตุการณ์ในตัวอย่างนี้มีดังต่อไปนี้

- P(R) คือ ความน่าจะเป็นขององค์ประกอบ *"ฝนตก"*
- P(M) คือ ความน่าจะเป็นขององค์ประกอบ *"รามมีเงิน"*
- P(¬G | R and M) คือ ความน่าจะเป็นแบบมีเงื่อนไขขององค์ประกอบ *"รามไม่ออกไปชมภาพยนต์ถ้าฝนตกและเขามีเงิน"* เนื่องจาก *"รามไม่ออกไปชมภาพยนต์"* ดังนั้นองค์ประกอบคือ ¬G นอกจากนี้ ตัวแปร R และ M มีอิทธิพลต่อตัวแปร G ดังนั้น ความน่าจะเป็นของ ¬G จึงขึ้นอยู่กับเงื่อนไขของ R และ M แต่จากองค์ประกอบ *"ฝนตก"* และ *"รามมีเงิน"* ส่งผลให้เงื่อนไขของ ¬G ได้แก่ R และ M ตามลำดับ
- P(¬S| ¬G) คือ ความน่าจะเป็นแบบมีเงื่อนไขขององค์ประกอบ *"รามไม่ทบทวนบทเรียนถ้าเขาไม่ออกไปชมภาพยนต์"* เนื่องจากองค์ประกอบ *"รามไม่ทบทวนบทเรียน"* คือ ¬S นอกจากนี้ ตัวแปร G มีอิทธิพลต่อตัวแปร S ดังนั้น ความน่าจะเป็นของ ¬S จึงขึ้นอยู่กับเงื่อนไขของ ¬G แต่จากองค์ประกอบ *"รามไม่ออกไปชมภาพยนต์"* ส่งผลให้เงื่อนไขของ ¬S ได้แก่ ¬G

ขั้นตอนสุดท้าย รวมองค์ประกอบทั้งหมดเข้าเป็นเหตุการณ์ แสดงการคำนวณได้ตามสมการด้านล่างนี้

$$
\begin{aligned}
&P(\neg R) \times P(\neg M) \times P(G \mid \neg R \text{ and } \neg M) \times P(\neg T \mid G) \\
&\qquad = 0.3 \times 0.6 \times 0.9 \times 0.2 \\
&\qquad = 0.0324
\end{aligned}
$$

จากโจทย์ในตัวอย่างนี้ ความน่าจะเป็นของเหตุการณ์ *"รามไม่มีเงินในขณะที่ฝนไม่ตก เขาตัดสินใจไม่ดูโทรทัศน์และออกไปชมภาพยนต์แทน"* มีค่าเท่ากับ 0.0324 หรือ 3.24%

อภิปราย

ตัวจำแนกเบส์มีพื้นฐานการพิจารณาความน่าจะเป็นของค่าฟีเจอร์ที่อยู่ในชั้นข้อมูลต่าง ๆ โดยไม่สนใจระยะทางระหว่างแบบ จึงมีข้อดีคือใช้เวลาในการจำแนกแบบรวดเร็ว เหมาะสมกับโปรแกรมประยุกต์ที่ต้องการผลลัพธ์ทันทีทันใด อย่างไรก็ตามเนื่องจากการทำนายแบบถูกตัดสินจากความถี่ของค่าที่เกิดขึ้น ดังนั้นในเซตข้อมูลอสมดุลที่สัดส่วนของแบบมีความแตกต่างกันสูงมาก ชั้นข้อมูลที่มีปริมาณแบบน้อยมาก อาจไม่ค่อยถูกทำนาย ส่งผลให้ความแม่นยำของชั้นข้อมูลส่วนน้อยนี้ มีค่าต่ำจนเกินไปจนไม่สามารถนำไปใช้ได้จริง

เนื่องจากเซตข้อมูลที่ใช้ตัวจำแนกนาอีฟเบส์ ต้องถูกตั้งสมมติฐานความอิสระจากเงื่อนไขชั้นข้อมูล ได้แก่ ค่าตัวแปรต่าง ๆ ต้องไม่เกี่ยวข้องกัน แต่ในความเป็นจริง ในบางเซตข้อมูล ค่าของตัวแปรจากสดมภ์ต่าง ๆ อาจมีความเกี่ยวข้องกันก็ได้ ดังนั้นถ้าเกิดกรณีนี้ขึ้น ตัวจำแนกนาอีฟเบส์อาจไม่เหมาะสมที่จะนำมาประยุกต์ใช้ อาจจำเป็นต้องใช้ตัวจำแนกอื่นที่พิจารณาความเกี่ยวข้องกันของแต่ละสดมภ์ ดังที่จะกล่าวถึงในบทต่อไป

ความแม่นยำจากการทำนายของเครือข่ายความเชื่อเบร์เซี่ยน ขึ้นอยู่กับกราฟที่ถูกออกแบบ ดังนั้น ถ้าผู้ออกแบบมีความเชี่ยวชาญย่อมให้ผลลัพธ์ที่เป็นน่าพอใจ ในทางกลับกัน การออกแบบจากจากซอฟต์แวร์อัตโนมัติอาจได้กราฟที่ขาดความสัมพันธ์ที่เหมาะสม จุดอ่อนหลักของการนำเครือข่ายความเชื่อเบร์เซี่ยนไปใช้งาน คือการที่ต้องใช้มนุษย์เข้ามาเกี่ยวข้องกับกระบวนการจำแนกแบบ เพราะมนุษย์แต่ละคนย่อมมีการตัดสินใจที่แตกต่างกัน ในเซตข้อมูลเดียวกันอาจให้ผลการทำนายต่างกันเมื่อเปลี่ยนผู้ออกแบบโครงสร้างข้อมูลนั่นเอง จึงทำให้ผลลัพธ์ที่ได้ขึ้นอยู่กับตัวบุคคลผู้ออกแบบแทน ความแตกต่าง ข้อดี ข้อเสีย ของตัวจำแนกนาอีฟเบส์และเครือข่ายความเชื่อเบร์เซี่ยนแสดงตามตารางที่ 4.4 ด้านล่างนี้

ตารางที่ 4.4 การเปรียบเทียบตัวจำแนกที่ใช้ทฤษฎีเบส์

ตัวจำแนก / เปรียบเทียบ	**ตัวจำแนกนาอีฟเบส์**	**เครือข่ายความเชื่อเบร์เซี่ยน**
ความเหมือน	ประยุกต์ใช้ทฤษฎีเบส์เพื่อคำนวณหาความน่าจะเป็นแบบมีเงื่อนไข	
ความแตกต่าง	ใช้สมการความน่าจะเป็นแบบมีเงื่อนไขสำหรับการจำแนกแบบ	ใช้โครงสร้างข้อมูลชนิดกราฟอวัฏจักรระบุทิศทางสำหรับการจำแนกแบบ
ข้อดี	สมการที่ใช้ในการจำแนกแบบสามารถคำนวณได้ง่าย ไม่ซับซ้อน การใช้ตัวแบบจากผู้ใช้ที่แตกต่างกัน จะให้ผลลัพธ์ที่ตรงกัน	แสดงผลด้วยกราฟฟิก ทำให้ง่ายต่อการทำความเข้าใจตัวแบบ ตัวแปรในเซตข้อมูลมีความเกี่ยวข้องกัน ซึ่งตรงตามหลักความเป็นจริง เช่น การเป็น (หรือไม่เป็น) โรคถุงลมโป่งพอง เกี่ยวข้องกับ พฤติกรรมการสูบบุหรี่
ข้อเสีย	ตัวแปรในเซตข้อมูลถูกตั้งสมมติฐานว่าไม่มีความเกี่ยวข้องกัน ซึ่งผิดกับหลักความเป็นจริง เช่น การเป็น (หรือไม่เป็น) โรคเบาหวาน มีความเกี่ยวข้องกับปริมาณน้ำตาลในอาหาร	จำเป็นต้องใช้ผู้เชี่ยวชาญในการออกแบบตัวแบบ การออกแบบจากผู้ออกแบบต่างกันให้ผลการจำแนกแบบที่แตกต่างกัน

สรุป

ทฤษฎีเบส์ถูกนำใช้ในการคำนวณความน่าจะเป็นหลัง ซึ่งเป็นผลลัพธ์จากความน่าจะเป็นก่อน ทฤษฎีนี้ถูกนำไปประยุกต์ใช้กับตัวจำแนกเบส์เพื่อทำนายชั้นข้อมูลของแบบทดสอบ โดยมีสมมติฐานว่าเซตข้อมูลมีความอิสระจากเงื่อนไขชั้นข้อมูล กล่าวคือการที่ค่าของตัวแปรต่าง ๆ ในชั้นข้อมูลเดียวกันไม่มีผลกระทบซึ่งกันและกัน การใช้งานตัวจำแนกประเภทนี้ ต้องมีการคำนวณค่าพารามิเตอร์โดยใช้ตัวประมาณค่าความน่าจะเป็นจากความถี่ อย่างไรก็ตาม ในบางกรณีความถี่อาจมีค่าเป็นศูนย์ กล่าวคือเหตุการณ์ที่สนใจไม่มีโอกาสเกิดขึ้นเลย ถ้าเกิดกรณีนี้ขึ้น ตัวประมาณค่าความน่าจะเป็นจากความถี่หลีกเลี่ยงความน่าจะเป็นที่เท่ากับศูนย์ โดยทดแทนด้วยค่าคงที่ขนาดเล็กแทน

ตัวจำแนกนาอีฟเบส์เป็นตัวจำแนกความน่าจะเป็นอย่างง่าย ที่ใช้กฎการตัดสินใจแมปในการเลือกชั้นข้อมูลที่มีโอกาสเกิดขึ้นสูงสุดของแบบทดสอบ ในขณะที่เครือข่ายความเชื่อเบร์เซี่ยนคือตัวแบบกราฟฟิกความน่าจะเป็นที่มีลักษณะเป็นกราฟอวัฏจักรระบุทิศทาง ตัวจำแนกทั้งสองชนิดนี้มีพื้นฐานอยู่บนทฤษฎีเบส์เหมือนกัน เนื้อหาในบทต่อไปกล่าวถึงตัวจำแนกประเภทที่สามซึ่งไม่ได้ประยุกต์ใช้หน่วยวัดระยะทางหรือทฤษฎีเบส์ แต่ใช้โครงสร้างข้อมูลชนิดต้นไม้ในการสร้างกฎการตัดสินใจที่อยู่ในรูปแบบ *“ถ้า-แล้ว”* เพื่อทำนายชั้นข้อมูลของแบบทดสอบแทน

แบบฝึกหัด

1. จากการสำรวจค้นพบว่า 0.008% ของประชากรทั้งหมดเป็นโรคมะเร็ง นอกจากนี้ ผลการวินิจฉัยจากห้องปฏิบัติการแห่งหนึ่งให้ผลแบบทวิภาค คือ บวก หรือ ลบ โดยที่ บวก หมายถึง วินิจฉัยว่าคนไข้เป็นโรคมะเร็ง และ ลบ หมายถึงวินิจฉัยว่าคนไข้ไม่เป็นโรคมะเร็ง จากการเก็บประวัติความแม่นยำของผลการวินิจฉัย ได้ข้อสรุปดังนี้ ความถูกต้องของการวินิจฉัยเป็นบวกคือ 98% และ ความถูกต้องของการวินิจฉัยเป็นลบคือ 97% จงคำนวณหาความน่าจะเป็นหลังที่คนไข้จะเป็นโรคมะเร็งหลังจากให้ผลวินิจฉัยเป็นบวก

2. จากเซตข้อมูลด้านล่างนี้ เกี่ยวข้องกับความชอบเครื่องดื่มโคล่าและแบรนด์พิซซ่าของนักศึกษาจำนวน 10 คน จงทำการคำนวณหาค่าความน่าจะเป็นดังต่อไปนี้

ตารางที่ 4.5 เซตฝึกฝนนักศึกษาต่อความชอบเครื่องดื่มโคล่าและยี่ห้อพิซซ่า

Student No.	Cola	Pizza Brand
1	Coke	Pizza Hut
2	Coke	Pizza Company
3	Coke	Pizza Hut
4	Coke	Pizza Hut
5	Coke	Pizza Hut
6	Coke	Continental
7	Pepsi	Pizza Company
8	Pepsi	Pizza Hut
9	Pepsi	Pizza Company
10	Pepsi	Pizza Hut

- ความน่าจะเป็นของผู้ชอบดื่มโค้ก
- ความน่าจะเป็นของผู้ชอบดื่มเป๊ปซี่
- ความน่าจะเป็นของผู้ชอบกินพิซซ่าฮัท
- ความน่าจะเป็นของผู้ชอบกินพิซซ่าคอมพานี
- ความน่าจะเป็นของผู้ชอบดื่มโค้กจากจำนวนผู้ชอบกินพิซซ่าฮัททั้งหมด
- ความน่าจะเป็นของผู้ชอบดื่มโค้กจากจำนวนผู้ชอบกินพิซซ่าคอมพานีทั้งหมด
- ความน่าจะเป็นของผู้ชอบดื่มเป๊ปซี่จากจำนวนผู้ชอบกินพิซซ่าฮัททั้งหมด

- ความน่าจะเป็นของผู้ชอบดื่มเป๊ปซี่จากจำนวนผู้ชอบกินพิซซ่าคอมพานีทั้งหมด
- ความน่าจะเป็นของผู้ชอบกินพิซซ่าฮัทจากจำนวนผู้ชอบดื่มโค้กทั้งหมด
- ความน่าจะเป็นของผู้ชอบกินพิซซ่าฮัทจากจำนวนผู้ชอบดื่มเป๊ปซี่ทั้งหมด
- ความน่าจะเป็นของผู้ชอบกินพิซซ่าคอมพานีจากจำนวนผู้ชอบดื่มโค้กทั้งหมด
- ความน่าจะเป็นของผู้ชอบกินพิซซ่าคอมพานีจากจำนวนผู้ชอบดื่มเป๊ปซี่ทั้งหมด

3. จงจำแนกแบบทดสอบ Y ที่มีค่าฟีเจอร์ คือ Cook = Sita, Mood = Good, Cuisine = Indian จากเซตข้อมูลในตารางที่ 4.1 โดยใช้ตัวจำแนกนาอีฟเบส์

4. จากเซตฝึกฝนในตารางที่ 4.6 กำหนดให้ชั้นข้อมูลคือ Buys_Compute จงจำแนกแบบทดสอบ Y: Age $\leq$ 30, Income = Mediam, Student = Yes, Credit_Rating = Fair โดยใช้ตัวจำแนกนาอีฟเบส์

ตารางที่ 4.6 เซตฝึกฝนตัวอย่าง 2

RID	Age	Income	Student	Credit_Rating	Buys_Computer
1	$\leq$ 30	High	No	Fair	No
2	$\leq$ 30	High	No	Excellent	No
3	31 – 40	High	No	Fair	Yes
4	> 40	Medium	No	Fair	Yes
5	> 40	Low	Yes	Fair	Yes
6	> 40	Low	Yes	Excellent	No
7	31 – 40	Low	Yes	Excellent	Yes
8	$\leq$ 30	Medium	No	Fair	No
9	$\leq$ 30	Low	Yes	Fair	Yes
10	> 40	Medium	Yes	Fair	Yes
11	$\leq$ 30	Medium	Yes	Excellent	Yes
12	31 – 40	Medium	No	Excellent	Yes
13	31 – 40	High	Yes	Fair	Yes
14	> 40	Medium	No	Excellent	No

5. จากเครือข่ายความเชื่อเบร์เซี่ยนในรูปที่ 4.2 ให้พิจารณาว่าสามารถเพิ่มตัวแปร “has exams” เข้าไปในเครือข่ายได้อย่างไร และ ให้เติมค่าความน่าจะเป็นที่สมเหตุสมผลลงในตารางความน่าจะเป็นแบบมีเงื่อนไขสำหรับตัวแปรทุกตัว

6. กำไรของแม่ค้าคนหนึ่งในการขายดอกไม้ขึ้นอยู่กับความสดของดอกไม้ ยิ่งไปกว่านั้น ถ้ามีงานเทศกาล กำไรของเธอจะเพิ่มขึ้น ในทางกลับกัน ช่วงสิ้นเดือนลูกค้าส่วนใหญ่ไม่ซื้อดอกไม้ ถ้าแม่ค้าได้กำไรมากพอ เธอจะกินเลี้ยงอย่างหรูหราสำหรับมื้อค่ำ จากข้อมูลข้างต้น จงวาดเครือข่ายความเชื่อเบร์เซี่ยน พร้อมทั้ง สร้างตารางความน่าจะเป็นแบบมีเงื่อนไขสำหรับตัวแปรทุกตัว (ให้นักศึกษากำหนดค่าความน่าจะเป็นได้เองตามความเหมาะสม) หลังจากนั้น ให้คำนวณความน่าจะเป็นที่แม่ค้าจะกินเลี้ยงอย่างหรูหราสำหรับมื้อค่ำในวันที่ดอกไม้มีความสด

บทที่ 5
ต้นไม้การตัดสินใจ

จากบทก่อนหน้านี้ที่กล่าวถึงตัวจำแนกเบส์และเครือข่ายความเชื่อเบร์เซี่ยน ที่มีแนวคิดของความน่าจะเป็นในการคำนวณค่าโอกาสที่แบบทดสอบจะถูกจำแนกในแต่ละชั้นข้อมูล อย่างไรก็ตาม ยังมีตัวจำแนกประเภทอื่นซึ่งใช้หลักการเดียวกับความน่าจะเป็น ได้แก่ ต้นไม้การตัดสินใจสำหรับการจำแนกแบบที่ประยุกต์ใช้เอนโทรปีร่วมกับเกนข่าวสารเป็นตัววัดสำหรับทำนายแบบทดสอบ ตัวจำแนกชนิดนี้ได้รับความนิยมในการใช้งานอย่างแพร่หลาย เพราะมนุษย์สามารถเข้าใจขั้นตอนการทำงานได้ในลักษณะของสาเหตุและผลลัพธ์ของการทำนายในรูปกฎ ถ้า-แล้ว นั่นเอง รายละเอียดของตัวแบบชนิดนี้และตัววัดดังกล่าวกำลังจะกล่าวถึงในบทนี้ ในส่วนเพิ่มเติม เนื้อหาในบทนี้ยังอธิบายถึงขั้นตอนวิธีในการสร้างขอบเขตการตัดสินใจและระนาบเกินบนฟีเจอร์ เพื่อใช้ในการแบ่งแยกแบบจากต่างชั้นข้อมูลกันออกจากกัน เพื่อให้ตัวแบบสามารถตัดสินใจได้ว่า แบบทดสอบควรถูกจำแนกให้อยู่ในชั้นข้อมูลใดที่มีความเหมาะสมมากที่สุด

จุดประสงค์การเรียนรู้

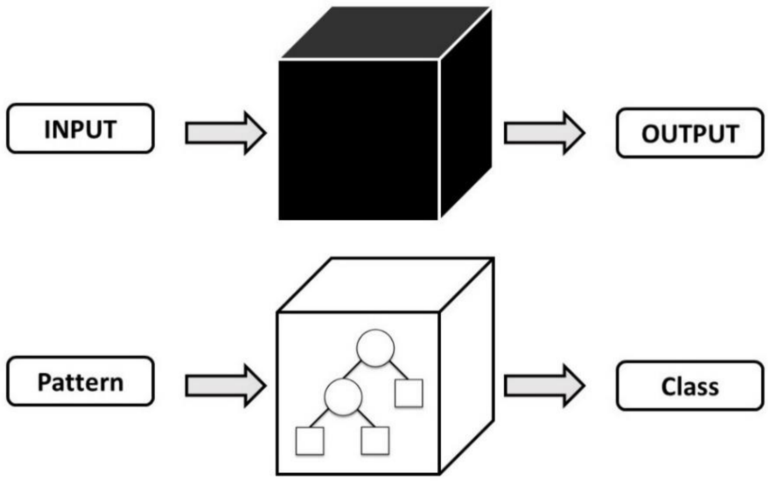
- ประยุกต์ใช้ต้นไม้การตัดสินใจเพื่อการจำแนกแบบได้
- ระบุเส้นทางการจำแนกแบบและกฎการตัดสินใจของต้นไม้การตัดสินใจได้
- เปรียบเทียบจุดแข็งและจุดอ่อนของต้นไม้การตัดสินใจได้
- บ่งบอกปัจจัยที่ใช้แบ่งแยกแบบจากแต่ละชั้นข้อมูลออกจากกันได้
- สามารถวัดค่าการเจือปนของฟีเจอร์ได้
- อุปนัยการสร้างต้นไม้การตัดสินใจได้
- เข้าใจความสำคัญของการเล็มกิ่ง

1. ต้นไม้การตัดสินใจสำหรับการจำแนกแบบ

ต้นไม้การตัดสินใจ (Decision Tree) เป็นตัวแบบที่ได้รับความนิยมสำหรับโปรแกรมประยุกต์ทางด้านการรู้จำแบบ และถูกใช้งานอย่างแพร่หลายในสาขาต่าง ๆ เช่น การแพทย์ การเงินและธนาคาร การศึกษา อุตสาหกรรม วิศวกรรมศาสตร์ และ วิทยาศาสตร์ เป็นต้น ตัวจำแนกแบบชนิดนี้ใช้โครงสร้างข้อมูลชนิดต้นไม้โดยมีคุณสมบัติเด่น ดังต่อไปนี้

- **ความความโปร่งใส (Transparent)**

โดยปกติแล้วตัวจำแนกแบบส่วนใหญ่มีลักษณะเป็น กล่องดำ (Black Box) แสดงตามรูปที่ 5.1 (รูปย่อยบน) ที่ผู้ใช้มองไม่เห็นกระบวนการภายใน ทราบเพียงแค่ข้อมูลเข้าและข้อมูลออกของตัวแบบเท่านั้น ในทางตรงกันข้าม ต้นไม้การตัดสินใจเป็นตัวจำแนกแบบที่มีลักษะเป็น กล่องขาว (White Box) แสดงตามรูปที่ 5.1 (รูปย่อยล่าง) ที่นอกจากผู้ใช้จะทราบข้อมูลเข้าและข้อมูลออกแล้ว ยังมองเห็นโครงสร้างของตัวแบบอย่างชัดเจนว่ามีกระบวนการทำงานภายในเป็นอย่างไร ในส่วนเพิ่มเติม ข้อมูลเข้าของต้นไม้การตัดสินใจได้แก่ แบบที่ต้องการจำแนก และข้อมูลออกได้แก่ ชั้นข้อมูลของแบบดังกล่าว



รูปที่ 5.1 กล่องดำ (บน) และ กล่องขาว (ล่าง)

- **คุณสมบัติเป็นมิตรกับผู้ใช้ (User-friendly Characteristic)**

เนื่องจากตัวแบบเป็นกล่องขาวจึงมีความโปร่งใส ผู้ใช้หน้าใหม่สามารถทำความเข้าใจการทำงานของต้นไม้การตัดสินใจได้ง่ายโดยใช้เวลาเรียนรู้ไม่นาน นอกจากนี้ ผู้ใช้ยังสามารถมองเห็นการจำแนกแบบในแต่ละขั้นตอนของกระบวนการได้อย่างชัดเจน รวมถึงสามารถแปลผลการทำงานเป็นภาษาที่มนุษย์สามารถเข้าใจได้ ต้นไม้การตัดสินใจสามารถใช้งานได้กับเซตข้อมูลที่มีชนิดข้อมูลหลากหลาย ดังต่อไปนี้

  - **ต่อเนื่อง (Continuous)**

    ข้อมูลชนิดต่อเนื่องคือข้อมูลที่สามารถวัดค่าได้ จัดเป็นข้อมูล เชิงปริมาณ (Quantitative) สามารถบันทึกค่าเป็นจำนวนจริง โดยอาจจะอยู่ในรูป

ของจำนวนเต็มหรือทศนิยมก็ได้ ค่าทศนิยมแต่ละตำแหน่งมีความหมายและมีจำนวนตำแหน่งได้ไม่สิ้นสุด ข้อมูลชนิดต่อเนื่องถือว่ามีความละเอียดมากที่สุดและสามารถเปรียบเทียบปริมาณกันได้ถึงความ มากกว่า น้อยกว่า เท่ากับ หรือไม่เท่ากับ ตัวอย่างของข้อมูลชนิดนี้แสดงตาม ตารางที่ 5.1

○ **ประเภท (Categorical)**

ข้อมูลชนิดประเภทคือข้อมูลที่สามารถแบ่งออกเป็นกลุ่มต่าง ๆ ได้ โดยอาจบ่งบอกถึงลักษณะภายนอกหรือคุณสมบัติภายในของวัตถุ ข้อมูลชนิดนี้จัดเป็นข้อมูล เชิงคุณภาพ (Qualitative) และถูกแบ่งออกได้เป็น 2 ประเภทย่อยดังต่อไปนี้

- **นอมินอล (Nominal)**

ข้อมูลชนิดนอมินอลคือข้อมูลที่ไม่สามารถเรียงลำดับในเชิงปริมาณได้ ตัวอย่างของข้อมูลชนิดนี้แสดงตาม ตารางที่ 5.2

- **ออดินอล (Ordinal)**

ข้อมูลชนิดออดินอลคือข้อมูลที่สามารถเรียงลำดับในเชิงปริมาณได้ ตัวอย่างของข้อมูลชนิดนี้แสดงตาม ตารางที่ 5.3

ตารางที่ 5.1 ตัวอย่างชนิดข้อมูลต่อเนื่อง

ข้อมูล	ตัวอย่าง
อายุ	8, 34, 41
อุณหภูมิ	-13, 0, 100
ความเร่ง	1.62, 3.77, 9.81
ชั้นปีของนักศึกษา	1, 2, 3, 4
โคเลสเตอรอลในเลือด	2.49, 2.76, 2.32
เงินเดือน	9k, 20k, 63.106k
SET INDEX	1,048.15 1,044.19, 1,127.24
BMI	18.5, 16,96, 30
GDP	4.0, 4.1, 2.4
พาย	3.1415926536

ตารางที่ 5.2 ตัวอย่างชนิดข้อมูลนอมินอล

ข้อมูล	ตัวอย่าง
เพศ	ชาย หญิง
ชมรม	พุทธศาสนา เทควันโด คณิตศาสตร์
การสูบบุหรี่	สูบ ไม่สูบ
สถานภาพ	โสด สมรส อย่าร้าง
สี	แดง เหลือง น้ำเงิน

ตารางที่ 5.3 ตัวอย่างชนิดข้อมูลออดินอล

ข้อมูล	ตัวอย่าง
ระดับปริญญา	ปริญญาตรี ปริญญาโท ปริญญาเอก
เกรด	F, D, D+, C, C+, B, B+, A
ตำแหน่งวิชาการ	ผศ. รศ. ศ.
การศึกษา	ประถมศึกษา มัธยมศึกษา อุดมศึกษา
การรีวิวอาหาร	❤ ❤❤ ❤❤❤

ต้นไม้การตัดสินใจมีโครงสร้างข้อมูลชนิดทรี องค์ประกอบต่าง ๆ แสดงตาม รูปที่ 5.1 โดยมีรายละเอียดดังต่อไปนี้

- **โหนดภายใน (Internal Node)**

  โหนดภายในเป็นตัวแทนของฟีเจอร์ในเซตข้อมูลแต่ไม่ใช่ป้ายชื่อชั้นข้อมูล โดยทำหน้าที่ทดสอบค่าฟีเจอร์ของแบบที่ต้องการจำแนก เพื่อเลือกเส้นทางตามกิ่งต่าง ๆ ในต้นไม้การตัดสินใจ จากรูปที่ 5.2 โหนดภายใน ได้แก่ I_0 และ I_1

- **ราก (Root)**

  รากคือโหนดภายในที่อยู่บนสุดของต้นไม้ เป็นจุดเริ่มต้นกระบวนการการจำแนกแบบ จากรูปที่ 5.2 ราก ได้แก่ I_0

- **โหนดใบ (Leaf Node)**

  โหนดใบเป็นตัวแทนป้ายชื่อชั้นข้อมูล คือข้อมูลออกหรือผลลัพธ์ของการทำนาย จากรูปที่ 5.2 โหนดใบ ได้แก่ L_1, L_2 และ L_3

- **กิ่ง (Branch)**

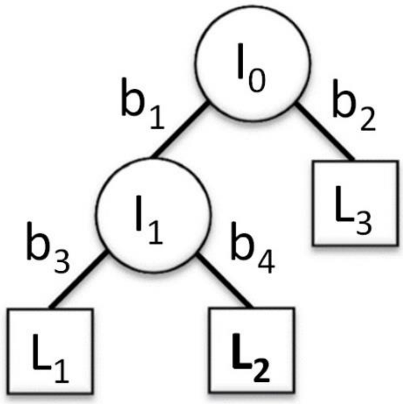
กิ่งเป็นตัวแทนของค่าต่าง ๆ ในฟีเจอร์ แต่ละกิ่งเก็บค่าที่เป็นไปได้จากการตัดสินใจของโหนดภายใน อาจเก็บเป็นค่าเดี่ยวหรือเก็บค่าเป็นช่วงก็ได้ จากรูปที่ 5.2 กิ่ง ได้แก่ b_1, b_2, b_3 และ b_4 ในส่วนเพิ่มเติม คุณสมบัติของกิ่ง มีดังต่อไปนี้

  - **เหตุการณ์ไม่เกิดร่วม (Mutually Distinct)**

    ค่าหรือช่วงของค่าที่เป็นไปได้ในฟีเจอร์ สำหรับแต่ละกิ่งในต้นไม้การตัดสินใจต้องมีค่าไม่ซ้ำกัน หมายความว่า ต้องไม่เกิดหตุการณ์ที่สองกิ่งใด ๆ ที่มีค่าเหมือนกัน

  - **เหตุการณ์ไม่เกิดร่วม (Collectively Exhaustive)**

    ในแต่ละฟีเจอร์ของแบบที่ต้องการจำแนก มีการเปรียบเทียบค่าในฟีเจอร์นั้นว่าตรงกับเงื่อนไขของกิ่งไหน ซึ่งจะมีเพียงเหตุการณ์เดียวเท่านั้นที่เป็นไปได้ หรืออาจกล่าวได้ว่า การตัดสินใจในแต่ละโหนดภายใน จะมีเพียงเส้นทางเดียวจากโหนดพ่อแม่ไปสู่โหนดลูกเท่านั้น



รูปที่ 5.2 โครงสร้างต้นไม้การตัดสินใจ

เหตุการณ์ทั้งสองสามารถอธิบายได้ดังนี้ ถ้าฟีเจอร์ I_0 มีค่าที่เป็นไปได้สองค่าคือ b_1 หรือ b_2 แล้วการตัดสินใจที่โหนดของฟีเจอร์นี้ จะมีเพียงสองเหตุการณ์ (หรือสองเส้นทาง) ที่เป็นไปได้เท่านั้น คือแบบที่ต้องการทำนายมีค่าฟีเจอร์ $I_0 = b_1$ หรือ $I_0 = b_2$ โดยที่ $b_1 \neq b_2$ อย่างไรก็ตาม ในกรณีที่เส้นทางของการตัดสินใจนำไปสู่โหนดใบแล้ว ต้นไม้การตัดสินใจจะให้ข้อมูลออกเป็นผลลัพธ์ของการทำนายแบบนั่นเอง กระบวนการจำแนกแบบตามโครงสร้างต้นไม้การตัดสินใจแสดงตาม ตัวอย่างที่ 1 ต่อไปนี้

ตัวอย่างที่ 1 กำหนดเซตข้อมูล D_1 โดยมีรายละเอียดของฟีเจอร์และชั้นข้อมูลตาม ตารางที่ 5.4 จงทำนายแบบ A ที่มีค่าฟีเจอร์ต่าง ๆ ตามรายละเอียด ด้านล่างนี้

$$A = (b_1, b_4)$$

ตารางที่ 5.4 ฟีเจอร์ของเซตข้อมูล D_1

ฟีเจอร์	ค่า
I_0	b_1, b_2
I_1	b_3, b_4
CLASS	L_1, L_2, L_3

จากแบบ A ที่โจทย์กำหนด มีค่าฟีเจอร์ได้แก่ (b_1, b_4) หมายความว่า ฟีเจอร์แรก (I_0) ของ A มีค่า b_1 และ ฟีเจอร์หลัง (I_1) มีค่า b_2 ในส่วนของชั้นข้อมูล (CLASS) เป็นสิ่งที่ยังไม่ทราบค่าในตอนนี้ เนื่องจากต้องผ่านกระบวนการจำแนกแบบในต้นไม้การตัดสินใจเสียก่อน

ค่าที่เป็นไปได้ของฟีเจอร์ I_0 และ I_1 คือ b_1, b_2 และ b_3, b_4 ตามลำดับ เส้นทางการค้นหาของต้นไม้การตัดสินใจเริ่มต้นจากราก ได้แก่ I_0 ทำการเปรียบค่าฟีเจอร์แรกของแบบที่ต้องการทำนาย (A) ว่ามีค่าเป็นไปตามเงื่อนไขของกิ่งไหน เนื่องจาก $I_0 = b_1$ ดังนั้น เส้นทางการตัดสินใจจะเลือกกิ่งซ้าย (b_1) แล้วลงมาที่โหนดภายใน I_1

ในขั้นตอนถัดไปของการสร้างต้นไม้ตัดสินใจ ทำการเปรียบเทียบค่าฟีเจอร์หลังของ A ซึ่ง $I_1 = b_4$ เส้นทางการตัดสินจะลงไปทางกิ่งขวา (b_4) เข้าสู่โหนดใบ (L_2) ซึ่งเก็บป้ายชื่อชั้นข้อมูล ดังนั้น แบบ A จะถูกต้นไม้การตัดสินใจจำแนกให้อยู่ในชั้นข้อมูล L_2 หรือสามารถเขียนเป็นกฎการตัดสินใจได้ ตามเงื่อนไขด้านล่างนี้

“ถ้า $I_0 = b_1$ และ $I_1 = b_4$ แล้วแบบจะอยู่ในชั้นข้อมูล L_2”

ต้นไม้การตัดสินใจทวิภาค (Binary Decision Tree) คือต้นไม้การตัดสินใจประเภทหนึ่ง การตัดสินใจที่แต่ละโหนดภายในจะมีเพียงแค่สองทางเลือกเท่านั้น ที่แต่ละโหนดภายในมีการต่อเชื่อมเพียงสองกิ่งเท่านั้น โดยที่ กิ่งแรก แทนค่า จริง (True) ในขณะที่ อีกกิ่งแทนค่า เท็จ (False) ตัวแบบชนิดนี้มีโครงสร้างข้อมูลเหมือนกับ ต้นไม้ทวิภาค (Binary Tree) ตัวอย่างของต้นไม้การตัดสินใจประเภทนี้ แสดงตาม รูปที่ 5.3

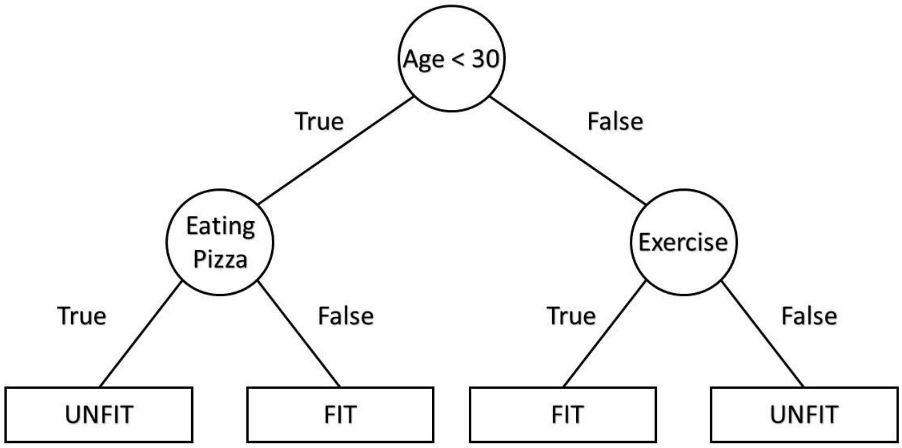
ตัวอย่างที่ 2 จงระบุกฎการตัดสินใจทั้งหมดที่ได้จากต้นไม้การตัดสินใจทวิภาคในรูปที่ 5.3

“ถ้าเขาอายุน้อยกว่า 30 ปี และ กินพิซซ่า แล้วเขาจะสุขภาพไม่ดี”

“ถ้าเขาอายุน้อยกว่า 30 ปี และ ไม่กินพิซซ่า แล้วเขาจะสุขภาพดี”

“ถ้าเขาอายุตั้งแต่ 30 ปี และ ออกกำลังกาย แล้วเขาจะสุขภาพดี”

“ถ้าเขาอายุตั้งแต่ 30 ปี และ ไม่ออกกำลังกาย แล้วเขาจะสุขภาพไม่ดี”



รูปที่ 5.3 ต้นไม้การตัดสินใจสำหรับการจำแนกประเภทสัตว์

ตัวอย่างของต้นไม้การตัดสินใจสำหรับการจำแนกแบบที่มีความสัมพันธ์กับเซตข้อมูล แสดงตามตัวอย่างด้านล่างนี้

ตัวอย่างที่ 3 ต้นไม้การตัดสินใจในรูปที่ 5.4 ถูกออกแบบจากเซตข้อมูลของสัตว์ตามตารางที่ 5.5 ความหมายของแต่ละฟีเจอร์มีรายละเอียด ดังต่อไปนี้

- CLASS แทน ป้ายชื่อชั้นข้อมูลของสัตว์ประเภทต่าง ๆ
- Legs แทน จำนวนขาของสัตว์ ค่าที่เป็นไปได้คือ 2 หรือ 4 ขา
- Horns แทน การมีเขา ค่าที่เป็นไปได้คือ มี หรือ ไม่มี
- Size แทน ขนาด ค่าที่เป็นไปได้คือ เล็ก ปานกลาง ใหญ่
- Colour แทน สี ค่าที่เป็นไปได้คือ ขาว ดำ น้ำตาล เขียว ส้ม
- Beak แทน จะงอยปาก ค่าที่เป็นไปได้คือ มี หรือ ไม่มี
- Sound แทน เสียงร้อง ค่าที่เป็นไปได้มีความแตกต่างกันหมด ตามชนิดของสัตว์

จากการพิจารณาต้นไม้การตัดสินใจ มีข้อสังเกต ดังต่อไปนี้

- ป้ายชื่อชั้นข้อมูลในเซตข้อมูลนี้มีความเกี่ยวเนื่องกับโหนดใบในต้นไม้การตัดสินใจนี้ เนื่องจาก โหนดใบถูกใช้เป็นตัวแทนประเภทของสัตว์
- เส้นทางจากรากจนถึงโหนดใบในต้นไม้การตัดสินใจนำมาซึ่งกฎการตัดสินใจ แสดงตามเงื่อนไขด้านล่างนี้ โดยที่ Legs, Horns และ Size คือ ฟีเจอร์ ในขณะที่ Mouse คือ ป้ายชื่อชั้นข้อมูล ในเซตข้อมูล

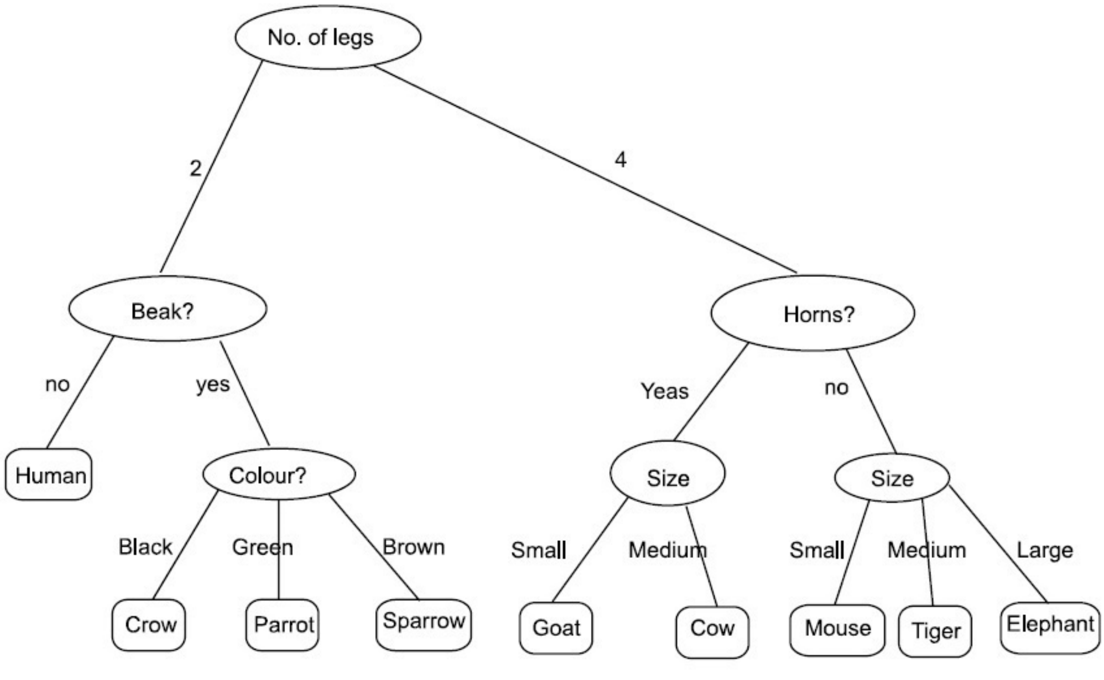
  "ถ้า Legs = 4 และ Horns = False และ Size = Small แล้ว ป้ายชื่อชั้นข้อมูลคือ Mouse"

- กฎสามารถทำความเข้าใจได้ง่าย โดยดูเงื่อนไขของ ผู้มาก่อน (Antecedent) และ ผลลัพธ์เดี่ยว (Single Outcome) ยกตัวอย่างจากกฎด้านบน ผู้มาก่อนได้แก่ Legs, Horn, Size และ ผลลัพธ์เดี่ยว คือ Mouse
- การตัดสินใจเลือกเส้นทาง มาจากการเปรียบค่าในฟีเจอร์ของแบบที่ต้องการทำนายว่าเป็นไปตามเงื่อนไขของกิ่งใด โดยที่ การตัดสินใจจะเกิดขึ้นทุก ๆ โหนดภายในที่อยู่ตามเส้นทางตั้งแต่รากจนถึงใบ และ การตัดสินใจครั้งแรกจะเกิดขึ้นที่ราก
- ฟีเจอร์นอกประเด็น (Irrelevant Feature) คือฟีเจอร์ที่ไม่เกี่ยวข้องกับการจำแนกชั้นข้อมูล ยกตัวอย่างเช่น ภาพยนตร์ที่คนไข้ชื่นชอบย่อมไม่เกี่ยวข้องกับการวินิจฉัยว่าเป็น (หรือไม่เป็น) โรคเบาหวาน เป็นต้น สำหรับเซตข้อมูลนี้ไม่มีฟีเจอร์นอกประเด็นที่ชัดเจน
- ฟีเจอร์ที่มีความเฉพาะเจาะจงกับป้ายชื่อชั้นข้อมูลมากจนเกินไปจะไม่ปรากฏในต้นไม้การตัดสินใจ สำหรับฟีเจอร์ลักษณะนี้ได้แก่ Sound เนื่องจากเสียงร้องของสัตว์เป็นคุณสมบัติเฉพาะตัว การนำต้นไม้การตัดสินใจที่ได้ไปจำแนกสัตว์อื่นที่มีเสียงร้องแตกต่างออกไปย่อมมีความผิดพลาดได้ง่าย
- ต้นไม้การตัดสินใจสามารถสร้างได้จากเซตข้อมูลที่มีฟีเจอร์เป็น ปริมาณ (Numeric) หรือ ประเภท (Category) จากเซตข้อมูลนี้ ฟีเจอร์ปริมาณได้แก่ Legs ในขณะที่ฟีเจอร์ที่เหลือคือฟีเจอร์ประเภท
- ต้นไม้การตัดสินใจนี้ไม่ใช่ต้นไม้การตัดสินใจทวิภาค เนื่องจากฟีเจอร์ Colour มีกิ่งมากกว่า 2 กิ่ง อย่างไรก็ตาม ต้นไม้การตัดสินใจใด ๆ สามารถแปลงให้อยู่ในรูปของต้นไม้การตัดสินใจทวิภาคได้ โดยเปลี่ยนการตัดสินใจที่มีทางเลือกมากกว่า 2 ทางเลือก ให้อยู่ในรูปของการตัดสินใจที่มีเพียง จริง หรือ เท็จ เท่านั้น

- โหนดยิ่งอยู่สูง เซตของป้ายชื่อชั้นข้อมูลยิ่งมีขนาดใหญ่ (มีจำนวนป้ายชื่อจำนวนมาก) ขนาดของเซตจะลดลงในระดับที่ต่ำลงมา เช่น ภายหลังการตัดสินใจที่โหนด Legs กิ่งของค่า 2 ประกอบไปด้วยป้ายชื่อของมนุษย์และนกรวม 4 ป้ายชื่อ ในขณะที่ ภายหลังการตัดสินใจที่โหนด Colour กิ่งของค่า Green ประกอบไปด้วย นกแก้ว เพียงป้ายชื่อเดียว

ตารางที่ 5.5 รายละเอียดของเซตของสัตว์

CLASS	Legs	Horns	Size	Colour	Beak	Sound
Cow	4	Yes	Medium	White	No	Moo
Crow	2	No	Small	Black	Yes	Caw
Elephant	4	No	Large	Black	No	Trumpet
Goat	4	Yes	Small	Brown	No	Bleat
Mouse	4	No	Small	Black	No	Squeak
Parrot	2	No	Small	Green	Yes	Squawk
Sparrow	2	No	Small	Brown	Yes	Chirp
Tiger	4	No	Medium	Orange	No	Roar



รูปที่ 5.4 ต้นไม้การตัดสินใจสำหรับการจำแนกประเภทสัตว์

2. ระนาบเกิน

ในกระบวนการสร้างต้นไม้ตัดสินใจสำหรับการจำแนกแบบ ฟีเจอร์ (หรือกลุ่มของฟีเจอร์) ในเซตข้อมูล จะโดนทดสอบเลือกฟีเจอร์ที่เหมาะสมที่สุด เพื่อนำไปสร้างโหนดภายใน การทดสอบสามารถแบ่งออกได้เป็น 3 ประเภท ได้แก่

1. การทดสอบขนานแกน (Axis-parallel Test)

การทดสอบประเภทนี้พิจารณาเพียงฟีเจอร์เดียวเท่านั้น เป็นการสร้าง ระนาบหลายมิติ หรือ ไฮเปอร์เพลน (Hyperplane) ที่ขนานกับแกนของฟีเจอร์เพื่อแบ่งแยก ปริภูมิ (Space) ออกเป็นสองส่วนออกจากกัน ทำให้แบบในปริภูมิถูกแบ่งแยกออกเป็นสองกลุ่ม การแบ่งแยกลักษณะนี้เรียกว่า การแบ่งแยกแบบขนานแกน (Axis-parallel Split) โดยแสดงตามอสมการด้านล่างนี้ พร้อมทั้งความหมายของตัวแปรต่าง ๆ ที่เกี่ยวข้อง การแบ่งแยกประเภทนี้แสดงตามตัวอย่างที่ 4 และ ระนาบหลายมิติแสดงตาม รูปที่ 5.5

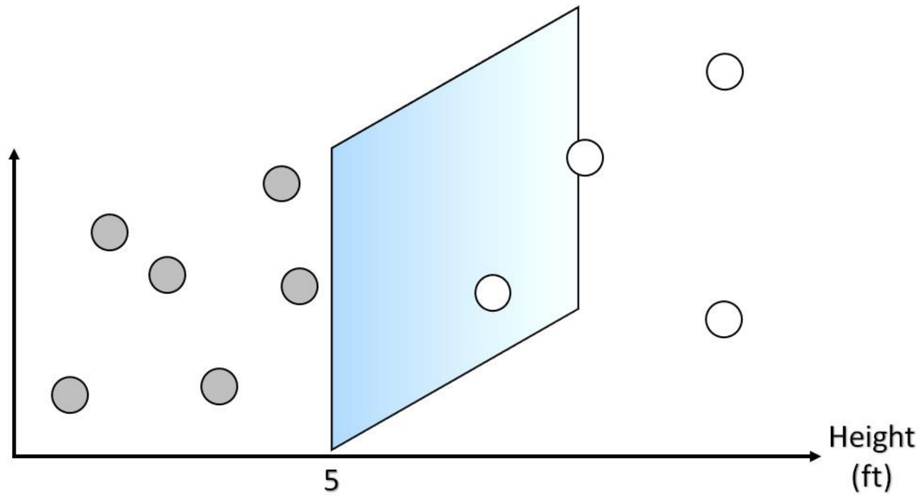
$$x > a_0$$

- x คือ ฟีเจอร์ในเซตข้อมูล
- a_0 คือ ขีดแบ่ง (Threshold)

สาเหตุที่ใช้คำว่าระนาบหลายมิติ เนื่องจากจำนวนฟีเจอร์ในเซตข้อมูลอาจมีมากกว่า 2 ฟีเจอร์และจำนวนมิติคือจำนวนฟีเจอร์นั่นเอง ด้วยเหตุที่ว่าไม่สามารถใช้เส้นตรงแบ่งแยกปริภูมิที่มากกว่าสองมิติออกจากกันได้ จึงจำเป็นต้องใช้ระนาบในการแบ่งแทน เปรียบเทียบได้กับการแบ่งปริมาตรในห้องเรียนที่มีนักศึกษานั่งเรียนอยู่ออกเป็นสองส่วน ไม่สามารถใช้เชือก (เปรียบเทียบได้กับเส้นตรง) ออกจากกันได้ จำเป็นต้องใช้พาร์ติชันติดตั้งจากพื้นถึงเพดาน (เปรียบเทียบได้กับระนาบ) ถึงจะกั้นห้องเรียนออกเป็นสองส่วนได้อย่างเด็ดขาด ส่งผลให้นักศึกษาถูกแยกออกจากกันเป็นสองกลุ่ม

ตัวอย่างที่ 4 การทดสอบแกนขนานตามอสมการด้านบนนี้ ทดสอบความสูงของวัตถุว่ามีค่ามากกว่า 5 ฟุตหรือไม่ กล่าวอีกนัยหนึ่งได้ว่า ระนาบเกิน Height = 5 แบ่งปริภูมิออกเป็นสองส่วน ตามรูปที่ 5.5 ได้แก่ ปริภูมิ Height > 5 และ ปริภูมิ Height ≤ 5 ซึ่งแบ่งแยกแบบออกจากเป็นเป็นสองกลุ่มได้แก่ แบบจากกลุ่มสีขาวและสีเทาตามลำดับ

“Height > 5 ft”



รูปที่ 5.5 ระนาบหลายมิติ Height = 5

2. การทดสอบส่วนผลมเชิงเส้นของฟีเจอร์ (Test on Linear Combination of Features)

การทดสอบประเภทนี้พิจารณาฟีเจอร์เป็นกลุ่ม โดยที่ระนาบเกินไม่จำเป็นต้องขนานแกนของฟีเจอร์ใด แต่มีความเอียงเข้าสู่แกนของฟีเจอร์ใดฟีเจอร์หนึ่ง การแบ่งแยกลักษณะนี้เรียกว่า การแบ่งแยกแบบเอียง (Oblique Split) โดยแสดงตามอสมการด้านล่างนี้ การแบ่งแยกประเภทนี้แสดงตาม ตัวอย่างที่ 5 และ ระนาบหลายมิติแสดงตาม รูปที่ 5.6

$$\sum_{i=1}^{d} a_i x_i > a_0$$

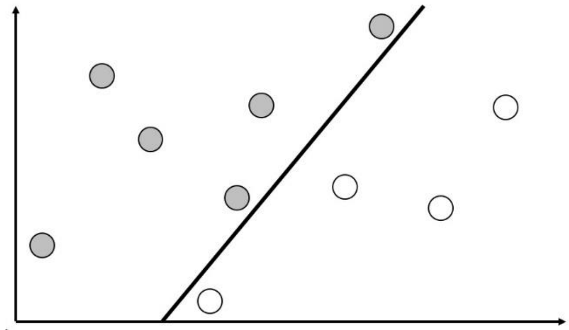
- x_i คือ ฟีเจอร์ลำดับที่ i ในเซตข้อมูล
- a_i คือ น้ำหนักของฟีเจอร์ลำดับที่ i
- a_0 คือ ขีดแบ่ง (Threshold)
- i คือ จำนวนเต็ม มีค่าตั้งแต่ 1 จนถึง d
- d คือ จำนวนฟีเจอร์ทั้งหมดในเซตข้อมูล

ตัวอย่างที่ 5 อสมการด้านล่างนี้เป็นการทดสอบผลรวมเชิงเส้นของฟีเจอร์ Height และ Weight เพื่อแบ่งแยกปริภูมิออกเป็น 2 ส่วน ดังนี้

- ปริภูมิ 0.4 Height + 0.3 Weight > 38
- ปริภูมิ 0.4 Height + 0.3 Weight ≤ 38

โดยตัวอย่างการทดสอบเชิงเส้นบนกลุ่มฟีเจอร์แสดงตาม รูปที่ 5.6

"0.4 Height + 0.3 Weight > 38"



รูปที่ 5.6 การทดสอบเชิงเส้นบนกลุ่มฟีเจอร์

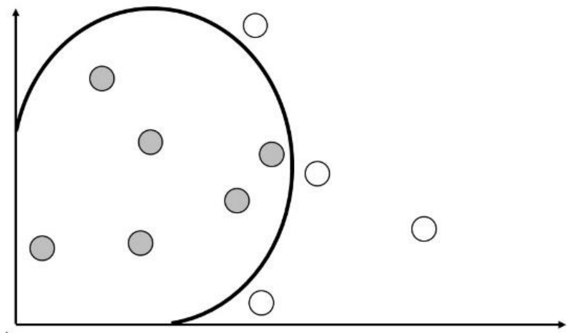
3. การทดสอบส่วนผสมไม่เชิงเส้นของฟีเจอร์

(Test on Non-linear Combination of Features)

การทดสอบประเภทนี้มีส่วนประกอบคือ ฟังก์ชันไม่เชิงเส้นและมีรูปแบบที่กว้างที่สุด การแบ่งแยกประเภทนี้แสดงตามอสมการด้านล่างนี้ และตัวอย่างของฟังก์ชันไม่เชิงเส้นแสดงตามรูปที่ 5.7 ด้านล่างนี้

$f(x) > 0$

- f(x) คือ ฟังก์ชันไม่เชิงเส้น



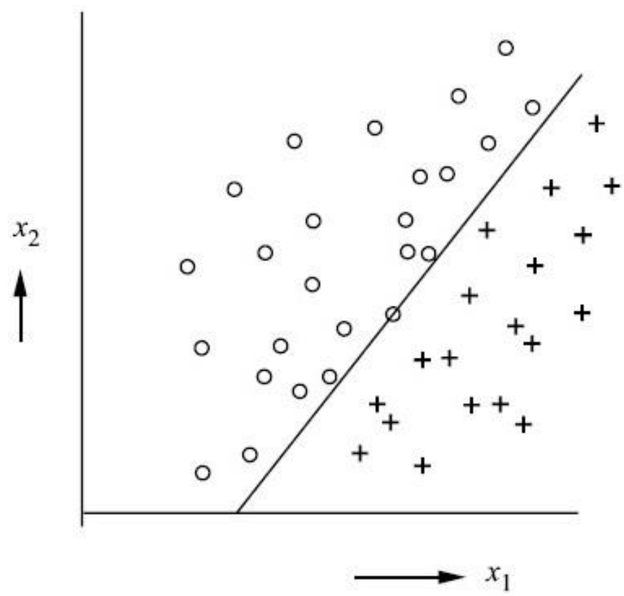
รูปที่ 5.7 การทดสอบไม่เชิงเส้นบนกลุ่มฟีเจอร์

จากการทดสอบทั้ง 3 ประเภท สังเกตได้ว่าการทดสอบขนานแกนเป็นกรณีพิเศษของการทดสอบเชิงเส้นบนกลุ่มฟีเจอร์ในกรณีที่ค่าน้ำหนัก $a_i = 0$ ทุกฟีเจอร์ ยกเว้นฟีเจอร์เดียว ในส่วนเพิ่มเติม การทดสอบขนานแกนและการทดสอบเชิงเส้นบนกลุ่มฟีเจอร์เป็นกรณีพิเศษของการทดสอบไม่เชิงเส้น

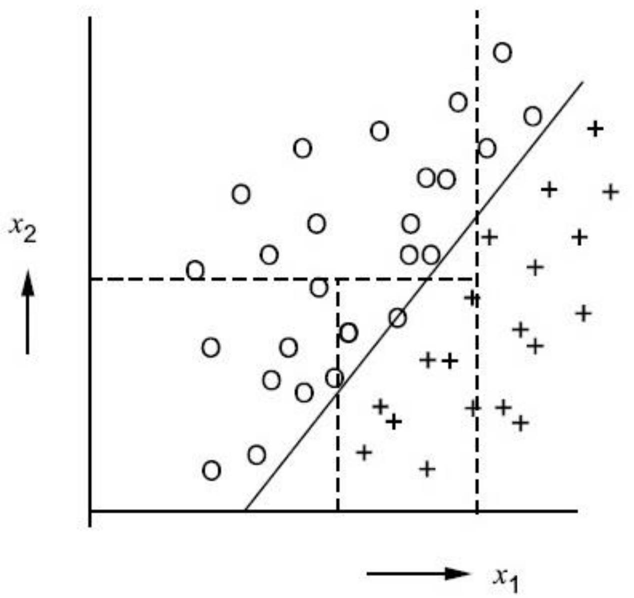
บนกลุ่มฟีเจอร์ จะเห็นได้ว่า การทดสอบไม่เชิงเส้นบนกลุ่มฟีเจอร์มีรูปแบบที่กว้างที่สุด นอกจากนี้ ยังมีการประมวลผลที่ช้าที่สุดอีกด้วย

ต้นไม้การตัดสินใจขนานแกน (Axis-parallel Decision Tree) เป็นต้นไม้การตัดสินใจพื้นฐานที่ใช้การแบ่งแยกแบบขนานแกน เนื่องจากการแบ่งแยกต้องต้องขนานกับแกนของฟีเจอร์ใดฟีเจอร์หนึ่ง (ตั้งฉากกับแกนของฟีเจอร์ที่กำลังพิจารณา) ดังนั้น ขอบเขตการตัดสินใจ (Decision Boundary) จึงมีรูปทรงเป็น ปริภูมิสี่เหลี่ยมมุมฉาก (Rectangular Region) การจำแนกแบบโดยใช้ต้นไม้การตัดสินใจชนิดนี้เรียกว่า การจำแนกประเภทสี่เหลี่ยมมุมฉาก (Rectangular Classification) อย่างไรก็ตาม การแบ่งแยกประเภทนี้ ไม่สอดคล้องกับการกระจายตัวของข้อมูลในหลายเซตข้อมูล เนื่องจาก ขอบเขตตัดสินใจอาจสแบ่งเป็น ปริภูมิสี่เหลี่ยมไม่มุมฉาก (Non-rectangular Region) ดังนั้น ต้นไม้การตัดสินใจขนานแกนจะทำการแบ่งแยกหลายครั้ง โดยปริภูมิสี่เหลี่ยนมมุมฉาก (จำนวนหลายปริภูมิ) ที่ได้จะเป็นค่าโดยประมาณเมื่อเปรียบเทียบกับปริภูมิสี่เหลี่ยมไม่มุมฉาก แสดงตามตัวอย่างที่ 6 ต่อไปนี้

ตัวอย่างที่ 6 พิจารณาเซตข้อมูลในรูปที่ 5.8 และ 5.9 ที่ประกอบด้วย 2 ฟีเจอร์ (x_1 และ x_2) และ 2 ชั้นข้อมูล (o และ +) การกระจายตัวของข้อมูลนี้เหมาะสมสำหรับการแบ่งแยกแบบเอียง (เส้นทึบในรูปที่ 5.8) อย่างไรก็ตาม ถ้าต้องการใช้ต้นไม้การตัดสินใจขนานแกน จำเป็นต้องแบ่งแยกแบบขนานแกนหลายครั้ง (เส้นประในรูปที่ 5.9) และความแม่นยำในการจำแนกแบบที่ได้จะมีความผิดพลาดอยู่บ้าง



รูปที่ 5.8 การแบ่งแยกแบบเอียงและปริภูมิสี่เหลี่ยมไม่มุมฉาก



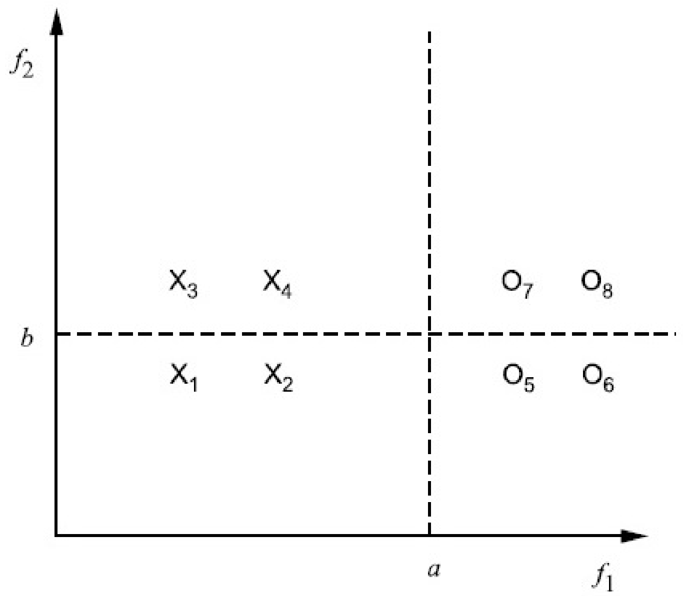
รูปที่ 5.9 การแบ่งแบบขนานแกนและปริภูมิสี่เหลี่ยมมุมฉาก

3. หน่วยวัดการเจือปน

การสร้างต้นไม้การตัดสินใจเริ่มต้นจากการเลือกฟีเจอร์มาสร้างโหนดภายใน ฟีเจอร์ที่มีความสำคัญสูงที่สุดจะถูกเลือกไปสร้างเป็นราก ฟีเจอร์ที่มีความสำคัญรองลงมาจะเป็นโหนดภายในของระดับถัดไป หรืออาจกล่าวได้ว่า โหนดภายในของฟีเจอร์ที่มีความสำคัญมากกว่าจะอยู่ในระดับที่สูงกว่าโหนดภายในของฟีเจอร์ที่มีความสำคัญต่ำกว่า ฟีเจอร์ที่มีความสำคัญต้องมีลักษณะเป็น ฟีเจอร์จำแนก (Discriminative Feature) ซึ่งมีคุณสมบัติคือ เมื่อพิจารณาค่าบนแกนของฟีเจอร์นี้แล้ว แบบในชั้นข้อมูลเดียวกันจะมีค่าคล้ายกันหรือเกาะกลุ่มกัน ในขณะที่แบบจากต่างชั้นข้อมูลกันจะมีค่าแตกต่างกันหรืออยู่ห่างไกลกัน ฟีเจอร์ประเภทนี้มีความเหมาะสมต่อการรู้จำแบบ เนื่องจากส่งผลให้ตัวแบบจำแนกได้อย่างแม่นยำว่าแบบควรอยู่ในชั้นข้อมูลใด ในทางกลับกัน ฟีเจอร์ที่ขาดคุณสมบัติฟีเจอร์จำแนกส่งผลให้แบบจากต่างชั้นมูลกันจะมีค่าคล้ายกันหรือกระจายตัวผสมกันเมื่อพิจารณาค่าบนฟีเจอร์นี้ ทำให้ตัวแบบมีความยากในการจำแนกแบบและอาจทำนายผิดเป็นส่วนใหญ่ ตัวอย่างต่อไปแสดงลักษณะของฟีเจอร์ที่เป็นและไม่เป็นฟีเจอร์จำแนก

ตัวอย่างที่ 7 พิจารณาเซตข้อมูลในรูปที่ 5.10 ที่ประกอบด้วย 2 ฟีเจอร์ (f_1 และ f_2) และ 2 ชั้นข้อมูล (o และ $\times$) ค่า a และ b คือ ค่าบนแกน f_1 และ f_2 ตามลำดับ พิจารณาการแบ่งแยกแบบของค่าบนแกนฟีเจอร์ทั้งสอง ดังต่อไปนี้

- f_1 คือ ฟีเจอร์จำแนก เนื่องจาก เมื่อพิจารณาค่าบนแกนของฟีเจอร์นี้พบว่า การตัดสินใจที่ค่า a สามารถแบ่งแยกแบบจากต่างชั้นข้อมูลกันออกจากกันได้ ดังนี้
    - การตัดสินใจ $f_1 \geq a$ แบบอยู่ในชั้นข้อมูล o ทั้งหมด
    - การตัดสินใจ $f_1 < a$ แบบอยู่ในชั้นข้อมูล × ทั้งหมด
- f_2 ไม่มีคุณสมบัติเป็นฟีเจอร์จำแนก เนื่องจากการพิจารณาค่าบนแกนของฟีเจอร์นี้ ไม่มีค่าใดเลย ที่แบ่งแยกแบบจากชั้นข้อมูล o และ × ออกจากกันได้ การตัดสินใจที่ค่า b สามารถแบ่งแบบออกเป็น 2 กลุ่ม ๆ ละเท่า ๆ กัน ได้ แต่แบบในปริภูมิเดียวกันมาจากชั้นข้อมูลต่างกัน



รูปที่ 5.10 ฟีเจอร์ที่เป็นและไม่เป็นฟีเจอร์จำแนก

จากตัวอย่างที่ผ่านมา f_1 จัดเป็นฟีเจอร์จำแนก เพราะระนาบเกิน $f_1 = a$ สามารถแบ่งเซตข้อมูลออกเป็น 2 ปริภูมิ โดยแต่ละปริภูมิบรรจุแบบจากชั้นข้อมูลเดียวกัน ฟีเจอร์ในลักษณะนี้จัดว่ามี ความบริสุทธิ์ (Purity) การสร้างโหนดภายในจะเลือกจากฟีเจอร์ที่สามารถแบ่งแบบให้มีความบริสุทธิ์สูงที่สุดเท่าที่จะทำได้ แต่แทนที่จะใช้ความบริสุทธิ์ในการเลือกฟีเจอร์จำแนก การสร้างต้นไม้การตัดสินใจจะคำนวณหา การเจือปน (Impurity) แทน สองค่านี้มีความสัมพันธ์กัน กล่าวคือ ยิ่งข้อมูลมีความบริสุทธิ์สูงการเจือปนจะต่ำ แต่ถ้าข้อมูลมีความบริสุทธิ์ต่ำ การเจือปนจะสูง

ภายหลังการแบ่งแยกแบบขนานแกนที่ระนาบเกิน f_1 = a ของฟีเจอร์จำแนก f_1 ซึ่งมีค่าการเจือปนต่ำที่สุด โหนดภายในของฟีเจอร์นี้จะถูกสร้างในต้นไม้การตัดสินใจ และเซตของแบบที่สัมพันธ์กับ f_1 จะถูกแบ่งออกเป็นเซตย่อยของแบบจำนวน 2 เซต ดังนี้

- เซตย่อยของแบบที่ 1 บรรจุแบบ x_1, x_2, x_3, x_4
- เซตย่อยของแบบที่ 2 บรรจุแบบ o_1, o_2, o_3, o_4

หลังจากนั้น ในแต่ละเซตย่อยของแบบจะเริ่มกระบวนการสร้างโหนดภายในจากฟีเจอร์จำแนกในลักษณะของ การเวียนเกิด (Recursive) วนซ้ำต่อไปเรื่อย ๆ จนกระทั่งเซตย่อยของแบบไม่เหลือการเจือปน (เซตย่อยของแบบบรรจุแบบจากชั้นข้อมูลเดียวกันทั้งหมด) โหนดภายในของขั้นตอนนี้คือโหนดลูกของโหนดภายในจากขั้นตอนก่อนหน้า นอกจากนี้ หน่วยวัดการเจือปนมีอยู่หลายชนิด โดยแต่ละชนิดมีรายละเอียดดังต่อไปนี้

1. การเจือปนเอนโทรปี (Entropy Impurity)

การเจือปนชนิดนี้สามารถเรียกสั้น ๆ ได้ว่า เอนโทรปี ซึ่งมีสูตรคำนวณตามสมการด้านล่างนี้

$$i(N) = -\sum_j P(w_j)\, log_2\, P\,(w_j)$$

- i(N) คือ เอนโทรปีของโหนด N
- w_j คือ ชั้นข้อมูล j
- $P(w_j)$ คือ สัดส่วนของแบบในชั้นข้อมูล j ที่โหนด N
- j คือ ค่าคงที่ตั้งแต่ 1 ถึง จำนวนชั้นข้อมูล

สาเหตุที่การคำนวณเอนโทรปีใช้ log ฐาน 2 เพราะสัญญาณดิจิตอลมีเพียง 2 สถานะเท่านั้น ได้แก่ 0 และ 1 ในตำราฉบับนี้จะขอใช้สัญลักษณ์ log แทน $\log_2$ สำหรับตัวอย่างถัดไปแสดงการคำนวณค่าเอนโทรปีจากสมการด้านบนนี้

ตัวอย่างที่ 8 จงคำนวณค่าเอนโทรปีในแต่ละกรณี ดังต่อไปนี้

ตัวอย่างที่ 8.1 เซตของแบบที่โหนดภายใน P บรรจุแบบจากชั้นข้อมูลเดียวกันทั้งหมด

$P(w_1) = 1$

ดังนั้น

$$
\begin{aligned}
i(N) &= -P(w_1) \log P(w_1) \\
&= -1 \log_2 1 \\
&= 0
\end{aligned}
$$

ตัวอย่างที่ 8.2 เซตของแบบที่โหนดภายใน Q บรรจุแบบจาก 2 ชั้นข้อมูล ๆ ละกึ่งหนึ่ง

$$
\begin{aligned}
P(w_1) &= 0.5 \\
P(w_2) &= 0.5
\end{aligned}
$$

ดังนั้น

$$
\begin{aligned}
i(N) &= -P(w_1) \log P(w_1) - P(w_2) \log P(w_2) \\
&= -0.5 \log 0.5 - 0.5 \log_2 0.5 \\
&= 1
\end{aligned}
$$

ตัวอย่างที่ 8.3 เซตของแบบที่โหนดภายใน R บรรจุแบบจาก 3 ชั้นข้อมูล ชั้นข้อมูลแรกมี 4 แบบ ชั้นข้อมูลที่สองมี 5 แบบ และชั้นข้อมูลสุดท้ายมีแบบเดียว

$$
\begin{aligned}
P(w_1) &= 4/10 = 0.4 \\
P(w_2) &= 5/10 = 0.5 \\
P(w_3) &= 1/10 = 0.1
\end{aligned}
$$

ดังนั้น

$$
\begin{aligned}
i(N) &= -P(w_1) \log P(w_1) - P(w_2) \log P(w_2) \\
&= -0.4 \log 0.4 - 0.5 \log 0.5 - 0.1 \log 0.1 \\
&= 1.36
\end{aligned}
$$

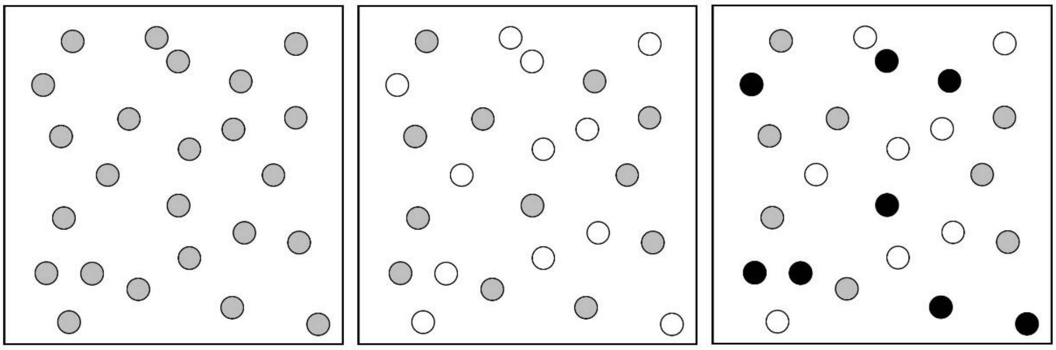
จากทั้งสามตัวอย่างย่อยข้างต้น สรุปได้ดังนี้

- เอนโทรปีมีค่า 0 จากโหนดภายใน P หมายความว่า แบบในเซตย่อยนี้ไม่มีการเจือปนเหลืออยู่เลย (มีความบริสุทธิ์ 100%) เพราะเซตนี้มีแต่แบบจากชั้นข้อมูลเดียวกันเท่านั้น ไม่มีแบบจากชั้นข้อมูลอื่นเลย

- เอนโทรปีของโหนด R มีค่าสูงกว่าเอนโทรปีของโหนด Q หมายความว่า แบบในเซตของโหนด R นี้มีการเจือปนมากกว่า แบบในเซตของโหนด Q กล่าวคือ แบบในโหนด R มาจากการผสมกันของ 3 ชั้นข้อมูล แต่แบบในโหนด Q มาจากการผสมกันของ 2 ชั้นข้อมูลเท่านั้น

รูปที่ 5.11 ด้านล่างนี้แสดงการกระจายตัวของแบบในแต่ละชั้นข้อมูล เมื่อเอนโทรปีมีค่าแตกต่างกัน อธิบายได้ดังนี้

- รูปย่อยซ้าย เอนโทรปีมีค่า 0 แบบไม่มีการเจือปนเลย
- รูปย่อยกลาง เอนโทรปีมีค่าต่ำ แบบมีการเจือปนน้อย
- รูปย่อยขวา เอนโทรปีมีค่าสูง แบบมีการเจือปนมาก



รูปที่ 5.11 แบบในแต่ละชั้นข้อมูลเมื่อเอนโทรปีมีค่าแตกต่างกัน

2. การเจือปนจีนี (Gini Impurity)

การเจือปนชนิดนี้เป็นอีกหนึ่งตัววัดที่เป็นที่นิยมใช้เหมือนเอนโทรปี สูตรคำนวณแสดงตามสมการด้านล่างนี้ โดยสัญลักษณ์ต่าง ๆ มีความหมายเช่นเดียวกันกับสัญลักษณ์ของสมการการเจือปนเอนโทรปี

$$i(N) = \sum_{j} P(w_j) * P(1 - P(w_j))$$

ตัวอย่างด้านล่าง แสดงการคำนวณการเจือปนจีนี

ตัวอย่างที่ 8.4 จากตัวอย่างที่ 8.3 จงคำนวณการเจือปนจีนี

$$\begin{aligned} i(N) &= 0.4 \times (1 - 0.4) + 0.5 \times (1 - 0.5) + 0.1 \times (1 - 0.1) \\ &= 0.58 \end{aligned}$$

3. การเจือปนจำแนกผิดพลาด (Misclassification Impurity)

การเจือปนชนิดนี้จะวัดค่าความน่าจะเป็นที่ต่ำที่สุด ที่แบบทดสอบจะถูกทำนายผิดพลาด โดยสูตรคำนวณแสดงตามสมการด้านล่างนี้ โดยสัญลักษณ์ต่าง ๆ มีความหมายเช่นเดียวกันกับสัญลักษณ์ของสมการการเจือปนเอนโทรปี

$$i(N) = 1 - max_j P(w_j)$$

- max_j $P(w_j)$ คือ ค่าสูงสุดในหมู่ $P(w_j)$ ของแต่ละชั้นข้อมูล

ตัวอย่างที่ 8.5 จากตัวอย่างที่ 8.3 จงคำนวณการเจือปนจำแนกผิดพลาด

$$\begin{aligned} i(N) &= 1 - \max(0.4, 0.5 - 0.1) \\ &= 0.5 \end{aligned}$$

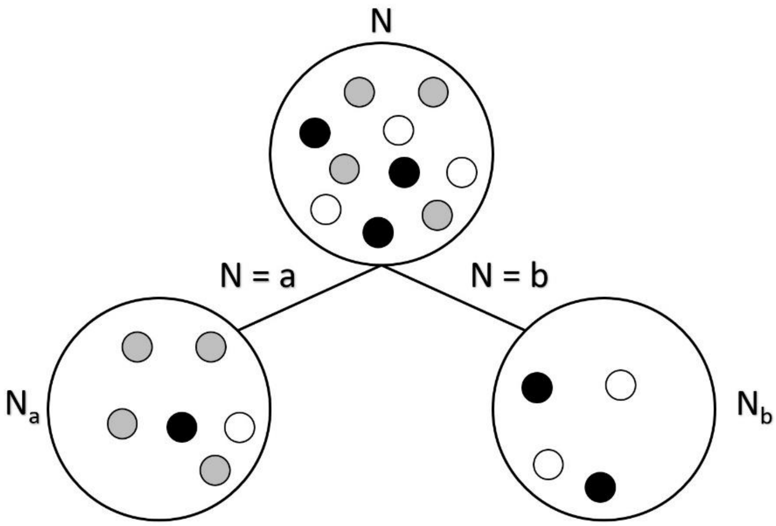
เอนโทรปีและการเจือปนจีนีมีความแตกต่างกัน กล่าวคือเอนโทรปีคือผลรวมของความน่าจะเป็นของแต่ละชั้นข้อมูลในหน่วยลอการิทึม ในขณะที่การคำนวณการเจือปนจีนีสนใจแต่แบบที่ถูกจำแนกผิดพลาดซึ่งคล้ายกับการเจือปนจำแนกผิดพลาด

4. เกนข่าวสาร

เมื่อต้นไม้การตัดสินใจเกิดการแบ่งที่โหนดภายใน (N) ของฟีเจอร์ใด แบบที่สัมพันธ์กับ N จะถูกแบ่งออกเป็นกลุ่มต่าง ๆ ตามค่าที่เป็นไปได้ของฟีเจอร์ N (a หรือ b) ยกตัวอย่างเช่นใน รูปที่ 5.12 โหนดภายใน (ราก) บรรจุแบบจาก 3 ชั้นข้อมูล (แสดงด้วยสีขาว ดำ เทา) เมื่อเกิดการแบ่งขึ้น N จะแตกออกเป็น 2 โหนดภายใน (N_L และ N_R) ตาม 2 เส้นทางคือ กิ่งซ้าย N = a และ กิ่งขวา N = b โดยที่ โหนดภายในใหม่จะเป็นโหนดลูกของโหนดภายในเดิม และแบบจากโหนด N จะโดนแบ่งเป็นกลุ่มตามค่าต่าง ๆ ของ N ดังนี้

- โหนด N บรรจุแบบทั้งหมดจาก 3 ชั้นข้อมูล (ขาว ดำ เทา) มีค่าฟีเจอร์ N = a หรือ N = b
- โหนด N_L ของกิ่ง N = a บรรจุแบบจาก 3 ชั้นข้อมูลบางส่วน (ขาว ดำ เทา) มีค่าฟีเจอร์ N = a เท่านั้น โดยไม่มีแบบใดที่มีค่าฟีเจอร์ N = b
- โหนด N_R ของกิ่ง N = a บรรจุแบบจาก 2 ชั้นข้อมูลบางส่วน (ขาว และ ดำ) มีค่าฟีเจอร์ N = b เท่านั้น โดยไม่มีแบบใดที่มีค่าฟีเจอร์ N = a

เมื่อกระบวนการแบ่งสิ้นสุดลงที่โหนดภายในเดิม โหนดภายในใหม่ก็จะเริ่มต้นการแบ่งในลักษณะของการเวียนซ้ำ (Recursive) ต่อไป จนกระทั่ง สิ้นสุดการสร้างต้นไม้การตัดสินใจ



รูปที่ 5.12 การแบ่งที่โหนดภายใน

ฟีเจอร์ที่ถูกเลือกนำไปสร้างโหนดภายในของต้นไม้การตัดสินใจต้องมีคุณสมบัติคือ การลดค่าการเจือปน (Impurity Drop) ต้องมีค่ามากที่สุดเท่าที่จะเป็นไปได้ โดยที่ค่านี้คำนวณได้จากสมการด้านล่างนี้

การลดค่าการเจือปน = เอนโทรปีของโหนด N – ผมรวมของเอนเทรปีของโหนดลูกของ N

จากรูปที่ 5.12 สูตรการคำนวณสามารถเขียนใหม่ได้ว่า

การลดค่าการเจือปน = เอนโทรปีของโหนด N – เอนโทรปีของโหนด N_a – เอนโทรปีของโหนด N_b

การลดค่าการเจือปนมีชื่อเรียกอีกอย่างหนึ่งว่า เกนข่าวสาร (Information Gain) แสดงได้ตามสมการด้านล่างนี้

$$\Delta i(N) = i(N) - \sum_j P_j * i(N_j)$$

- $\Delta i(N)$ คือ เกนข่าวสาร ของโหนด N
- $i(N)$ คือ การเจือปน ของโหนด N
- $i(N_j)$ คือ การเจือปน ของโหนดลูกของ N ตามกิ่ง N = j
- P_j คือ สัดส่วนของแบบเมื่อค่าฟีเจอร์ N = j
- j คือ ค่าฟีเจอร์ที่เป็นไปได้ทั้งหมดของโหนด N

การคำนวณค่าเกนข่าวสารแสดงตามตัวอย่าง ด้านล่างนี้

ตัวอย่างที่ 9 พิจารณากรณีของแบบจำนวน 100 แบบ โดยมี 40 แบบอยู่ในชั้นข้อมูลที่หนึ่ง (C_1) 30 แบบอยู่ในชั้นข้อมูลที่สอง (C_2) และ 30 แบบอยู่ในชั้นข้อมูลที่สาม (C_3) กำหนดให้ฟีเจอร์ N เกิดการแบ่งแบบไปยัง 2 เส้นทาง โดยที่กิ่งซ้ายถูกแบ่งแบบมา 60 แบบ โดยมาจากชั้นข้อมูลที่ 1, 2, 3 จำนวน 40, 10, 10 แบบ ตามลำดับ และ กิ่งขวาถูกแบ่งแบบมา 40 แบบ โดยมาจากชั้นข้อมูลที่ 2 และ 3 ชั้นข้อมูลละ 20 แบบ ตามลำดับ ข้อมูลทั้งหมดที่กล่าวมานี้แสดงโดยสรุปตาม ตารางที่ 5.6 จงคำนวณค่าเกนข่าวสารเมื่อใช้หน่วยวัด การเจือปนเอนโทรปี การเจือปนจีนี และ การเจือปนจำแนกผิดพลาด

ตารางที่ 5.6 จำนวนแบบในต้นไม้การตัดสินใจหลังจากการแบ่งแยกที่ฟีเจอร์ N

N = a Left Branch N_L	N = b Right Branch N_R	Total	Class
40	0	40	1
10	20	30	2
10	20	30	3

ตัวอย่างที่ 9.1 ใช้หน่วยวัดการเจือปนเอนโทรปี

a. คำนวณค่าสัดส่วนของแบบในแต่ละชั้นข้อมูลที่โหนด N

เนื่องจากที่โหนด N แบบทั้งหมดยังไม่ถูกแบ่งแยกออกจากกัน ดังนั้นสัดส่วนจึงหารด้วยจำนวนแบบทั้งหมด คือ 100 แบบ

$P(C_1) = 40/100 = 0.4$

$P(C_2) = 30/100 = 0.3$

$P(C_3) = 30/100 = 0.3$

การเจือปนเอนโทรปีของโหนด N

$$
\begin{aligned}
i(N) &= -P(C_1) \log P(C_1) - P(C_2) \log P(C_2) - P(C_3) \log P(C_3) \\
&= -0.4 \log 0.4 - 0.3 \log 0.3 - 0.3 \log 0.3 \\
&= 1.38
\end{aligned}
$$

b. คำนวณค่าสัดส่วนของแบบในแต่ละชั้นข้อมูลที่โหนด N_L

เนื่องจากที่โหนด N_L แบบถูกแบ่งแยกออกมาจากโหนด N จำนวน 60 โหนด ดังนั้น

$P(C_1) = 40/60 = 2/3$

$P(C_2) = 10/60 = 1/6$

$P(C_3) = 10/60 = 1/6$

การเจือปนเอนโทรปีของโหนด N_L

$$
\begin{aligned}
i(N_L) &= -P(C_1) \log P(C_1) - P(C_2) \log P(C_2) - P(C_3) \log P(C_3) \\
&= -(2/3) \log (2/3) - (1/6) \log (1/6) - (1/6) \log (1/6) \\
&= 1.25
\end{aligned}
$$

c. คำนวณค่าสัดส่วนของแบบในแต่ละชั้นข้อมูลที่โหนด N_R

เนื่องจากที่โหนด N_R แบบถูกแบ่งแยกออกมาจากโหนด N จำนวน 40 โหนด ดังนั้น

$P(C_1) = 0/40 = 0.0$

$P(C_2) = 20/40 = 0.5$

$P(C_3) = 20/40 = 0.5$

การเจือปนเอนโทรปีของโหนด N_R

เนื่องจาก log 0 ไม่นิยาม ดังนั้นจึงไม่คำนวณค่านี้ และตัดทิ้งไป

$$
\begin{aligned}
i(N_R) &= -P(C_2) \log P(C_2) - P(C_3) \log P(C_3) \\
&= -0.5 \log 0.5 - 0.5 \log 0.5 \\
&= 1.0
\end{aligned}
$$

เนื่องจากแบบที่ถูกแบ่งแยกจากโหนด N มาทางกิ่งซ้ายจำนวน 60 แบบ และ มาทางกิ่งขวาจำนวน 40 แบบดังนั้น

$$
\begin{aligned}
P(C_L) &= 60/100 \\
&= 0.6
\end{aligned}
$$

$$
\begin{aligned}
P(C_R) &= 40/100 \\
&= 0.4
\end{aligned}
$$

จาก a. – c. คำนวณค่าเกนข่าวสารโดยใช้หน่วยวัดการเจือปนเอนโทรปี ได้ดังนี้

$$\begin{aligned}\Delta i(N) &= i(N) - P(C_L) \times i(N_L) - P(C_R) \times i(N_R) \\ &= 1.38 - 0.6 \times 1.25 - 0.4 \times 1.0 \\ &= 0.23\end{aligned}$$

ตัวอย่างที่ 9.2 ใช้หน่วยวัดการเจือปนจีนี

เนื่องจากค่าของสัดส่วนของแบบทุกค่าเหมือนตัวอย่างที่ 9.1 จึงไม่ต้องคำนวณซ้ำ

a. การเจือปนจินีของโหนด N

$$\begin{aligned}i(N) &= P(C_1) \times (1 - P(C_1) + P(C_2) \times (1 - P(C_2) + P(C_3) \times (1 - P(C_3) \\ &= 0.4 \times (1 - 0.4) + 0.3 \times (1 - 0.3) + 0.3 \times (1 - 0.3) \\ &= 0.66\end{aligned}$$

b. การเจือปนจินีของโหนด N_L

$$\begin{aligned}i(N_L) &= P(C_{1L}) \times (1 - P(C_{1L}) + P(C_{2L}) \times (1 - P(C_{2L}) + P(C_{3L}) \times (1 - P(C_{3L}) \\ &= (2/3) \times (1 - 2/3) + (1/6) \times (1 - 1/6) + (1/6) \times (1 - 1/6) \\ &= 0.5\end{aligned}$$

c. การเจือปนจีนีของโหนด N_R

$$\begin{aligned}i(N_R) &= P(C_{2L}) \times (1 - P(C_{2L}) + P(C_{3L}) \times (1 - P(C_{3L}) \\ &= 0.5 \times (1 - 0.5) + 0.5 \times (1 - 0.5) \\ &= 0.5\end{aligned}$$

จาก a. – c. คำนวณค่าเกนข่าวสารโดยใช้หน่วยวัดการเจือปนจีนี ได้ดังนี้

$$\begin{aligned}\Delta i(N) &= i(N) - P(C_L) \times i(N_L) - P(C_R) \times i(N_R) \\ &= 0.66 - 0.6 \times 0.5 - 0.4 \times 0.5 \\ &= 0.16\end{aligned}$$

ตัวอย่างที่ 9.3 ใช้หน่วยวัดการเจือปนจำแนกผิดพลาด

เนื่องจากค่าของสัดส่วนของแบบทุกค่าเหมือนตัวอย่างที่ 9.1 จึงไม่ต้องคำนวณซ้ำ

a. การเจือปนจำแนกผิดพลาดของโหนด N

$$i(N) = 1 - \max(0.4, 0.3, 0.3) = 0.6$$

b. การเจือปนจำแนกผิดพลาดของโหนด N_L

$$i(N_L) = 1 - \max(2/3, 1/6, 1/6) = 0.333$$

c. การเจือปนจำแนกผิดพลาดของโหนด N_R

$$i(N_R) = 1 - \max(0.5, 0.5) = 0.5$$

จาก a. – c. คำนวณค่าเกนข่าวสารโดยใช้หน่วยวัดการเจือปนจำแนกผิดพลาด ได้ดังนี้

$$\begin{aligned} \Delta i(N) &= i(N) - P(C_L) \times i(N_L) - P(C_R) \times i(N_R) \\ &= 0.6 - 0.6 \times 0.333 - 0.4 \times 0.5 \\ &= 0.202 \end{aligned}$$

เนื่องจากในแต่ละโหนดลูกมีจำนวนแบบไม่เท่ากัน ดังนั้นต้องมีการนำเอาค่าการเจือปนในแต่ละกิ่งของต้นไม้การตัดสินใจไปคูณกับค่าน้ำหนักในแต่ละโหนดลูก ค่าน้ำหนักคำนวณได้จากสัดส่วนของแบบในแต่ละชั้นข้อมูลต่อจำนวนแบบทั้งหมด โดยค่าน้ำหนักนี้ยิ่งมีค่าสูง สามารถตีความได้ว่าโหนดลูกโหนดนั้นมีความสำคัญมากนั่นเอง

5. สปริท

สปริท (Split) คือแบบบางส่วนที่ถูกแบ่งแยกออกมาจากแบบทั้งหมดที่สัมพันธ์กับโหนดภายในโดยที่ รากแบ่งแยกแบบทั้งหมดในเซตฝึกฝนออกเป็นสปริทต่าง ๆ ในขณะที่ โหนดภายในที่เป็นโหนดลูกแบ่งแยกแบบจากแต่ละสปริทออกเป็นเซตย่อยต่าง ๆ ในส่วนเพิ่มเติม สปริทคือผลลัพธ์จากกฎการตัดสินใจ ซึ่งอยู่ในรูปแบบด้านล่างนี้

$F_i(x) > a_0$

- $F_i(x)$ คือ ฟังก์ชันของฟีเจอร์ x
- a_0 คือ ค่าคงที่ใด ๆ

หัวข้อที่ 2 ในบทนี้ ได้กล่าวถึงการแบ่งแยกประเภทต่าง ๆ ไว้ ในหัวข้อนี้จะแสดงให้เห็นถึงต้นไม้การตัดสินใจที่ได้จากการแบ่งแยกแต่ละชนิด ดังต่อไปนี้

1. การแบ่งแยกแบบขนานแกน

การแบ่งแยกประเภทนี้พิจารณาค่าของฟีเจอร์เดี่ยวเท่านั้น ตามอสมการด้านล่างนี้ โดยที่การตัดสินใจขึ้นอยู่กับค่า a_0 บนแกนของฟีเจอร์ x_j ถ้าแบบทดสอบใดมีค่าฟีเจอร์นี้มากกว่า a_0 แบบนั้นจะถูกรวมกันเป็นสปริทแรก ในขณะที่แบบที่เหลือทั้งหมดจะรวมกันเป็นสปริทหลัง

$$x_j > a_0$$

- x_j คือ ฟีเจอร์ ลำดับที่ j
- a_0 คือ ค่าคงที่บนแกนของฟีเจอร์ x_j

การสร้างต้นไม้การตัดสินใจจากการแบ่งแยกประเภทนี้ แสดงตามตัวอย่างที่ 10

ตัวอย่างที่ 10 กำหนดเซตข้อมูลที่ประกอบด้วยชั้นข้อมูล X และ O แบบในเซตข้อมูลนี้มีจำนวน 2 ฟีเจอร์ โดยแต่ละแบบมีค่าของฟีเจอร์ ดังต่อไปนี้

$X_1 = (1, 1)$; $X_2 = (2, 1)$
$X_3 = (1, 2)$; $X_4 = (2, 2)$
$O_3 = (6, 1)$; $O_6 = (7, 1)$
$O_7 = (6, 2)$; $O_8 = (7, 2)$
$X_9 = (6, 7)$; $X_{10} = (7, 7)$

รูปที่ 5.13 แสดง พิกัดร่วม (Coordinate) บนแกนของฟีเจอร์ f_1 ละ f_2 ของแต่ละแบบ กำหนดให้ค่าของ a = 4 และ b = 4

การแบ่งแยกแบบขนานแกนพิจารณาทีละแกนของฟีเจอร์เท่านั้น ซึ่งจะแบ่งปริภูมิออกเป็นปริภูมิย่อยที่ขนานกับแกนของฟีเจอร์ ในส่วนเพิ่มเติม การแบ่งแยกประเภทนี้ จะสร้างกฎการตัดสินใจ ดังต่อไปนี้

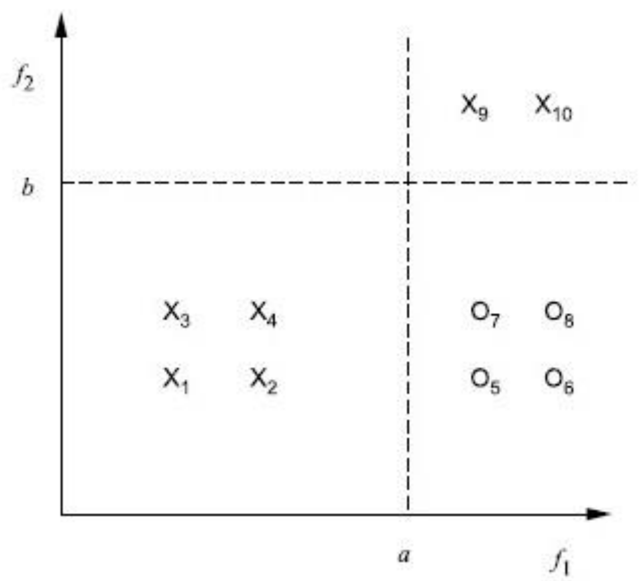
1. ฟีเจอร์ f_1 เป็นฟีเจอร์แรกที่ถูกเลือกมาสร้างเป็นโหนดภายใน เนื่องจากการแบ่งแยกที่ฟีเจอร์นี้สามารถสร้างสปริทที่มีความบริสุทธิ์เกิดขึ้นได้ ดังนั้น รากของต้นไม้การตัดสินใจจึงมีกฎการตัดสินใจคือ $f_1 > a$ แสดงตาม รูปที่ 5.14 สปริททั้งหมดที่ได้จากขั้นตอนนี้ มีดังต่อไปนี้

1.1 สปริทขวา ได้แก่ O_5, O_6, O_7, O_8 และ X_9, X_{10} ซึ่งอยู่ในปริภูมิขวามือของระนาบเกิน f_1 = a และเป็นไปตามกฎการตัดสินใจ f_1 > a สปริทนี้มีความเจือปน เนื่องจากแบบมาจากต่างชั้นข้อมูลกัน ต้นไม้การตัดสินใจจำเป็นต้องเลือกฟีเจอร์เพิ่มเติมเพื่อทำการแบ่งแยกต่อไปได้แก่ฟีเจอร์ f_2 ซึ่งสร้างสปริทที่มีความบริสุทธิ์ได้ ดังนั้น โหนดต่อไปของต้นไม้การตัดสินใจคือโหนดภายใน f_2 > 4 แสดงตาม รูปที่ 5.14 สปริททั้งหมดที่ได้จากฟีเจอร์ f_2 มีดังต่อไปนี้

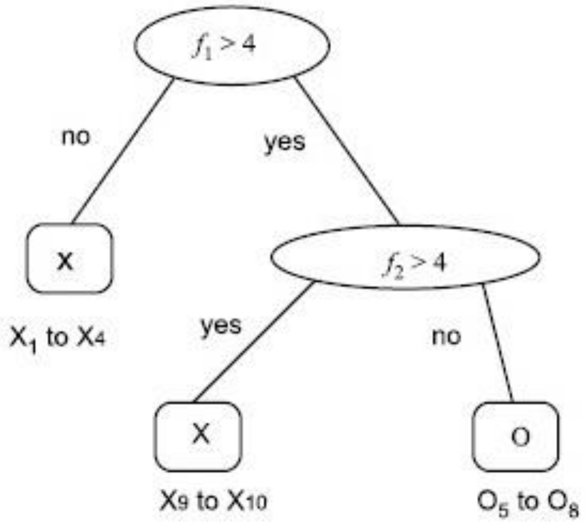
1.1.1 สปริทซ้าย ได้แก่ X_9, X_{10} ต้นไม้การตัดสินใจจึงสร้างโหนดใบของชั้นข้อมูล X ตามกิ่งซ้ายของโหนดภายใน f_2 > 4

1.1.2 สปริทขวา ได้แก่ O_5, O_6, O_7, O_8 ต้นไม้การตัดสินใจจึงสร้างโหนดใบของชั้นข้อมูล O ตามกิ่งขวาของโหนดภายใน f_2 > 4

1.2 สปริทซ้าย ได้แก่ X_1, X_2, X_3, X_4 ซึ่งอยู่ในปริภูมิซ้ายมือของระนาบเกิน f_1 = a และไม่เป็นไปตามกฎการตัดสินใจ f_1 > a สปริทนี้มีความบริสุทธิ์ เนื่องจากแบบทั้งหมดมาจากชั้นข้อมูลเดียวกัน ดังนั้น ต้นไม้การตัดสินใจจึงสร้างโหนดใบของชั้นข้อมูล X ตามกิ่งซ้ายของราก แสดงตามรูปที่ 5.14



รูปที่ 5.13 การแบ่งแยกแบบขนานแกน



รูปที่ 5.14 ต้นไม้การตัดสินใจสำหรับแบบที่ใช้การแบ่งแยกแบบขนานแกน

2. การแบ่งแยกแบบเอียง

การแบ่งแยกประเภทนี้พิจารณากลุ่มของฟีเจอร์ได้ พิจารณาระนาบเกิน $a_1f_1 + a_2f_2 = a_0$ ในรูปที่ 5.15 ซึ่งสามารถแบ่งแยกแบบทั้งหมดออกเป็นสองสปริทที่มีความบริสุทธิ์ได้ รูปแบบทั่วไปของกฎการตัดสินใจแสดงตามอสมการ ด้านล่างนี้

$a \cdot f_1 + b \cdot f_2 + c > 0$

- f_1, f_2 คือ ฟีเจอร์ ลำดับที่ 1 และ 2 ตามลำดับ
- a, b, c คือ ค่าคงที่ใด ๆ

จากรูปแบบอสมการทั่วไปข้างต้น เมื่อแทนค่าฟีเจอร์แรกและฟีเจอร์หลังของแบบ X_2, X_{10}, O_7 ลงในตัวแปร f_1 และ f_2 แล้วจะสามารถสร้าง 3 อสมการได้ตามที่กำลังจะแสดงต่อไปนี้ เนื่องจากแบบจากชั้นข้อมูล O และ X ถูกแบ่งแยกโดยระนาบเกิน $a \cdot f_1 + b \cdot f_2 + c$ ดังนั้น เครื่องหมายอสมการของชั้นข้อมูล O และ X จึงแตกต่างกัน

$$2a + b + c > 0$$

$$7a + 7b + c > 0$$

$$6a + 2b + c < 0$$

เมื่อแก้อสมการทั้งสามแล้ว ตัวแปรต่าง ๆ มีค่า ดังต่อไปนี้

$a = -1$

$b = 1$

$c = 2$

ดังนั้น กฎการตัดสินใจที่ได้จากการแก้อสมการข้างต้น คือ

$-f_1 + f_2 + 2 > 0$

หรือ

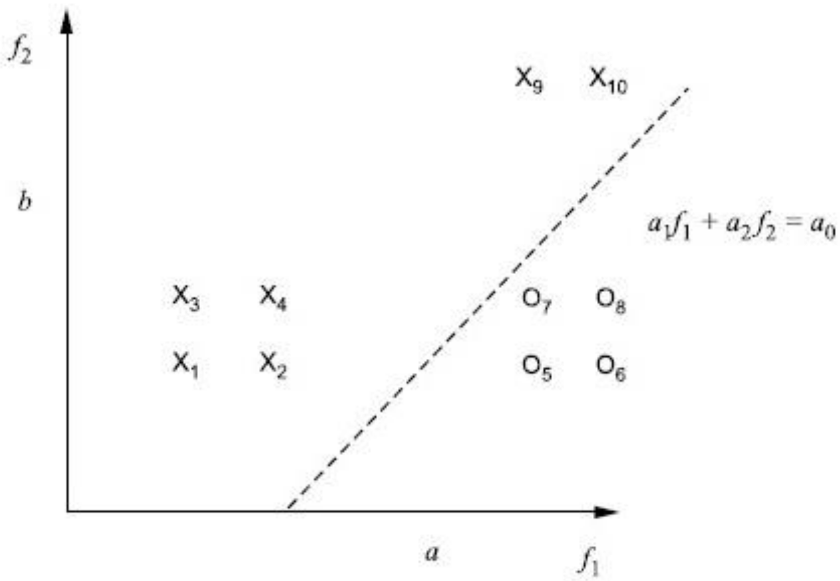
$f_1 - f_2 < 2$

ซึ่งสามารถแบ่งแยกชั้นข้อมูลได้ โดยมีเงื่อนไขในแต่ละชั้นข้อมูล คือ

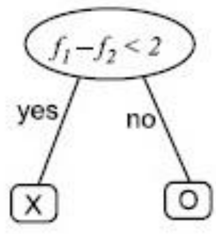
$f_1 - f_2 < 2$ สำหรับชั้นข้อมูล X

$f_1 - f_2 \geq 2$ สำหรับชั้นข้อมูล O

เนื่องจากขอบเขตการตัดสินใจนี้สามารถแบ่งแยกแบบให้มีความบริสุทธิ์ในแต่ละชั้นข้อมูลได้ ดังนั้น โหนดแรกหรือรากของต้นไม้การตัดสินใจ จึงถูกสร้างขึ้นตามรูปที่ 5.16



รูปที่ 5.15 การแบ่งแยกแบบเอียง



รูปที่ 5.16 ต้นไม้การตัดสินใจสำหรับแบบที่ใช้การแบ่งแยกแบบเอียง

6. การเล็มกิ่ง

ปัญหากระชับมากเกินไปเกิดขึ้นกับต้นไม้การตัดสินใจเมื่อมีโหนดภายในจำนวนมาก ส่งผลให้กฎการตัดสินใจมีความเฉพาะจงกับแบบฝึกฝน กล่าวคือแบบฝึกฝนแต่ละตัวจะมีเส้นทางของตัวเองจากรากจนถึงใบในต้นไม้ ตัวจำแนกแบบจะทำนายได้อย่างแม่นยำถ้าแบบในเซตทดสอบและแบบในเซตฝึกฝนมีความคล้ายกัน แต่ในความเป็นจริงแล้ว แบบจากทั้งสองเซตนี้มีความแตกต่างกัน ดังนั้นวิธีการรับมือปัญหานี้ต้องลดความลึกของต้นไม้การตัดสินใจลง โดยใช้เทคนิคที่เรียกว่า การเล็มกิ่ง (Pruning)

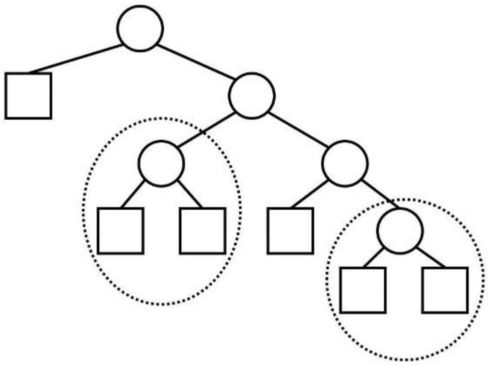
การเล็มกิ่งเริ่มต้นจากการสร้างต้นไม้การตัดสินใจอย่างสมบูรณ์ทั้งต้น จากนั้นเริ่มการเล็มกิ่งโดยค้นหาฟีเจอร์นอกประเด็นได้แก่ฟีเจอร์ที่ไม่เกี่ยวข้องกับการจำแนกชั้นข้อมูล โดยฟีเจอร์ประเภทนี้จะมีค่าเกนข่าวสารต่ำ ฟีเจอร์นอกประเด็นและกิ่งย่อยภายใต้ฟีเจอร์นี้จะถูกค้นหา และลบอย่างต่อเนื่องจนกระทั่งจำนวนโหนดใบลดลงจนถึงค่าที่ผู้ใช้ต้องการ ขั้นตอนวิธีทั้งหมดแสดงตามลำดับ ด้านล่างนี้

ขั้นตอนวิธีสำหรับการเล็มกิ่ง:

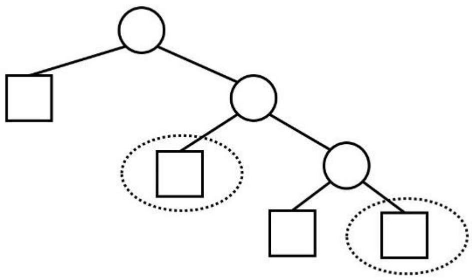
1. เก็บกิ่งทั้งหมดลงโครงสร้างข้อมูล
2. นับจำนวนโหนดใบทั้งหมด
3. ในขณะที่จำนวนของโหนดใบยังเกินจำนวนที่ผู้ใช้กำหนด:
    a. ค้นหากิ่งย่อยที่มีค่าเกนข่าวสารต่ำที่สุด
    b. ลบโหนดลูกทั้งหมดภายใต้กิ่งย่อยนี้
    c. เพิ่มโหนดใบภายใต้กิ่งย่อยนี้
    d. ปรับปรุงค่าจำนวนโหนดทั้งหมด

รูปที่ 5.17 และ 5.18 แสดงต้นไม้การตัดสินใจก่อนและหลังการเล็มกิ่ง คำอธิบายสัญลักษณ์ต่าง ๆ ในรูปทั้งคู่อธิบายได้ ดังต่อไปนี้

- วงกลม เป็นตัวแทนของ โหนดภายใน
- สี่เหลี่ยม เป็นตัวแทนของ โหนดใบ
- วงรีเส้นประ
    - รูปที่ 5.17 เป็นตัวแทนของ โหนดภายในและกิ่งย่อยที่ถูกเล็มกิ่ง
    - รูปที่ 5.18 เป็นตัวแทนของ โหนดใบที่ถูกแทนที่

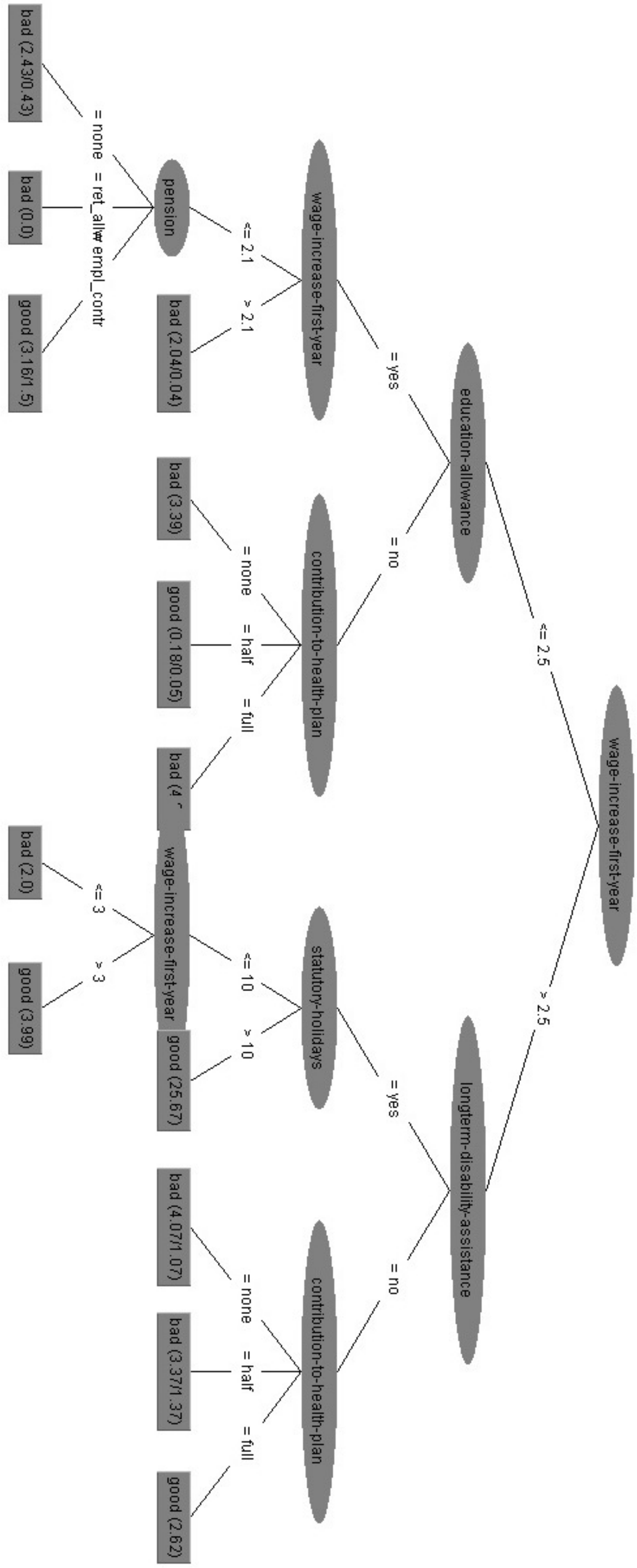


รูปที่ 5.17 ต้นไม้การตัดสินใจที่ยังไม่ผ่านการเล็มกิ่ง

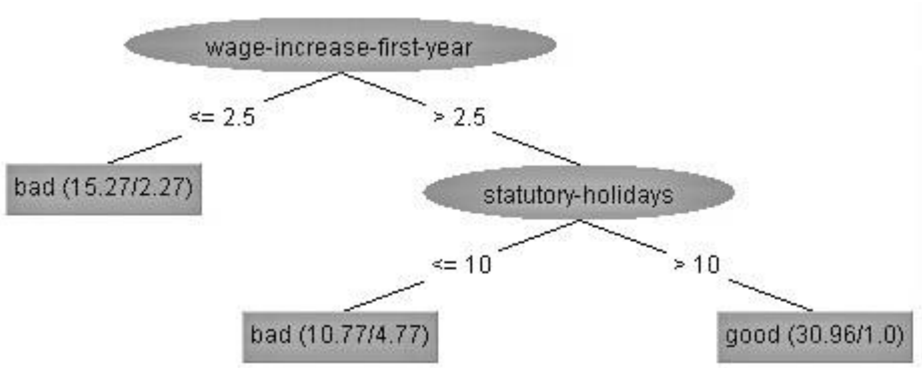


รูปที่ 5.18 ต้นไม้การตัดสินใจที่ผ่านการเล็มกิ่ง

รูปที่ 5.19 และ รูปที่ 5.20 แสดงต้นไม้การตัดสินใจที่สร้างจากเซตข้อมูล labor.arff ในซอฟต์แวร์ Weka โดยเปรียบเทียบให้เห็นถึงรูปร่างของต้นไม้ก่อนและหลังการเล็มกิ่ง ถึงแม้ว่าต้นไม้ทั้งสองต้นนี้ถูกสร้างจากเซตข้อมูลเดียวกันและให้ผลการจำแนกแบบที่คล้ายกันก็ตาม แต่สิ่งที่แตกต่างกันอย่างเห็นได้ชัดคือต้นไม้ที่ผ่านการเล็มกิ่งมีลำต้นที่ตื้นกว่า ส่งผลให้กฎที่ได้มีความซับซ้อนน้อยกว่า สัญลักษณ์ต่าง ๆ ในรูปอธิบายดังนี้ วงรีแทนฟีเจอร์ สี่เหลี่ยมผืนผ้าแทนป้ายชื่อชั้นข้อมูล และ นิพจน์ระหว่างส่วนของเส้นตรงคือกฎ ยกตัวเช่น ถ้าค่าคุณลักษณะ wage-increase-first-year มีค่าไม่เกิน 2.5 แล้วแบบจะถูกจำแนกให้มีป้ายชื่อชั้นข้อมูล bad เป็นต้น



รูปที่ 5.19 labor.arff ก่อนการเล็มกิ่ง



รูปที่ 5.20 labor.arff หลังการเล็มกิ่ง

7. การอุปนัยต้นไม้การตัดสินใจ

การอุปนัย (Induction) หมายถึงการสร้างหน่วยย่อยไปหาหน่วยใหญ่ กระบวนการสร้างต้นไม้การตัดสินใจเริ่มต้นจากการสร้างรากเป็นโหนดแรก ภายหลังการแบ่งแยกที่ราก แต่ละสปริทจะถูกสร้างเป็นต้นไม้ย่อย โดยที่โหนดภายในโหนดแรกของต้นไม้ย่อยคือโหนดลูกของราก จากนั้น การสร้างต้นไม้ย่อยจากแต่ละสปริทที่ถูกแบ่งแยกจะเกิดขึ้นอย่างต่อเนื่องในลักษณะของ การเวียนเกิด (Recursive) จนกระทั่ง ต้นไม้การตัดสินใจถูกสร้างเสร็จสิ้นทั้งต้นตั้งแต่รากจนถึงใบ เครื่องมือที่จำเป็นต้องใช้ในการอุปนัยต้นไม้การตัดสินใจได้แก่หน่วยวัดความเจือปนและเกนข่าวสาร การนำไปประยุกต์ใช้แสดงตามตัวอย่างต่อไปนี้

ตัวอย่างที่ 11 จากเซตฝึกฝนของข้อมูลภัตตาคารในตารางที่ 5.6 จงอุปนัยต้นไม้การตัดสินใจโดยใช้เอนโทรปีเป็นหน่วยวัดความเจือปน สำหรับทำนายว่าอาหารที่สั่งไปจะมีรสชาติอย่างไร กำหนดให้ฟีเจอร์ Tasty แทนชั้นข้อมูล ความหมายของแต่ละฟีเจอร์มีรายละเอียด ดังต่อไปนี้

- Cook แทน กุ๊กที่ปรุงอาหารให้ ซึ่งมี 3 คนให้บริการ ได้แก่ สิธา อาชา และ อุชา
- Mood แทน สภาพอารมณ์ของกุ๊ก ได้แก่ อารมณ์ดี และ อารมณ์เสีย
- Cuisine แทน ประเภทอาหาร ได้แก่ อาหารอินเดีย และ อาหารยุโรป
- Tasty แทน รสชาติอาหาร ได้แก่ อร่อย และ ไร้รสชาติ

ก. การสร้างโหนดราก

เซตข้อมูลในตารางที่ 5.6 ประกอบไปด้วยแบบทั้งหมด จำนวน 12 แบบ โดยอยู่ในชั้นข้อมูล Tasty = Yes จำนวน 8 แบบ และ ชั้นข้อมูล Tasty = No จำนวน 4 แบบ เอนโทรปีคำนวณได้ ดังนี้

$$i(N) = -\frac{4}{12}log(\frac{4}{12}) - \frac{8}{12}log(\frac{8}{12}) = 0.9183$$

ตารางที่ 5.6 เซตข้อมูลภัตตาคาร

Cook	Mood	Cuisine	Tasty
Sita	Bad	Indian	Yes
Sita	Good	Continental	Yes
Asha	Bad	Indian	No
Asha	Good	Indian	Yes
Usha	Bad	Indian	Yes
Usha	Bad	Continental	No
Asha	Bad	Continental	No
Asha	Good	Continental	Yes
Usha	Good	Indian	Yes
Usha	Good	Continental	No
Sita	Good	Indian	Yes
Sita	Bad	Continental	Yes

พิจารณาฟีเจอร์ทั้งสาม เพื่อค้นหาฟีเจอร์ที่เกนข่าวสารมีค่าสูงสุด เพื่อนำมาสร้างเป็นโหนดแรกหรือราก ของต้นไม้การตัดสินใจ ได้ตามขั้นตอนต่อไปนี้

1. **Cook**

(a) Cook = Sita มี 4 แบบ ที่อยู่ในชั้นข้อมูล Yes และ ไม่มีแบบ ที่อยู่ในชั้นข้อมูล No ดังนั้น

$$i(N) = -\frac{4}{4}log(\frac{4}{4}) = 0.0$$

(b) Cook = Asha มี 2 แบบ ที่อยู่ในชั้นข้อมูล Yes และ 2 แบบ ที่อยู่ในชั้นข้อมูล No ดังนั้น

$$i(N) = -\frac{2}{4}log(\frac{2}{4}) - \frac{2}{4}log(\frac{2}{4}) = 1.0$$

(c) Cook = Usha มี 2 แบบ ที่อยู่ในชั้นข้อมูล Yes และ 2 แบบ ที่อยู่ในชั้นข้อมูล No ดังนั้น

$$i(N) = -\frac{2}{4}log(\frac{2}{4}) - \frac{2}{4}log(\frac{2}{4}) = 1.0$$

จาก (a)-(c) คำนวณ เกนข่าวสาร ได้ดังนี้

$$\Delta i(N) = 0.9183 - \frac{4}{12} \times 1.0 - \frac{4}{12} \times 1.0 = 0.2516$$

2. Mood

(a) Mood = Bad มี 3 แบบ ที่อยู่ในชั้นข้อมูล Yes และ 3 แบบ ที่อยู่ในชั้นข้อมูล No ดังนั้น

$$i(N) = -\frac{3}{6}log(\frac{3}{6}) - \frac{3}{6}log(\frac{3}{6}) = 1.0$$

(b) Mood = Good มี 5 แบบ ที่อยู่ในชั้นข้อมูล Yes และ 1 แบบ ที่อยู่ในชั้นข้อมูล No ดังนั้น

$$i(N) = -\frac{5}{6}log(\frac{5}{6}) - \frac{1}{6}log(\frac{1}{6}) = 0.65$$

จาก (a)-(b) คำนวณ เกนข่าวสาร ได้ดังนี้

$$\Delta i(N) = 0.9183 - \frac{6}{12} \times 1.0 - \frac{6}{12} \times 0.65 = 0.0933$$

3. Cuisine

(a) Cuisine = Indian มี 5 แบบ ที่อยู่ในชั้นข้อมูล Yes และ 1 แบบ ที่อยู่ในชั้นข้อมูล No ดังนั้น

$$i(N) = -\frac{5}{6}log(\frac{5}{6}) - \frac{1}{6}log(\frac{1}{6}) = 0.65$$

(b) Cuisine = Continental มีอย่างละ 3 แบบ ที่อยู่ในชั้นข้อมูล Yes และ No ดังนั้น

$$i(N) = -\frac{3}{6}log(\frac{3}{6}) - \frac{3}{6}log(\frac{3}{6}) = 1.0$$

จาก (a)-(b) คำนวณ เกนข่าวสาร ได้ดังนี้

$$\Delta i(N) = 0.9183 - \frac{6}{12} \times 0.65 - \frac{6}{12} \times 1.0 = 0.0933$$

เนื่องจากเกนข่าวสารของฟีเจอร์ Cook มีค่าสูงสุด ดังนั้นฟีเจอร์นี้จึงถูกเลือกเป็นรากของต้นไม้การตัดสินใจ แสดงตามรูปที่ 5.21 หลังจากนั้นแบบฝึกฝนในตารางที่ 5.6 จะถูกแบ่งแยกออกเป็น 3 สปริท ตามกิ่งต่าง ๆ ดังต่อไปนี้

ข. สปริท Cook = Asha ที่บรรจุแบบฝึกฝนตามตารางที่ 5.7 ตามกิ่งซ้ายของราก

ค. สปริท Cook = Sita ที่บรรจุแบบฝึกฝนตามตารางที่ 5.8 ตามกิ่งกลางของราก

ง. สปริท Cook = Usha ที่บรรจุแบบฝึกฝนตามตารางที่ 5.9 ตามกลิ่งขวาของราก

ข. การสร้างโหนดภายในตามกิ่งซ้ายของราก

สปริทในตารางที่ 5.7 ประกอบไปด้วยแบบทั้งหมด จำนวน 4 แบบ โดยอยู่ในชั้นข้อมูล Tasty = Yes และ ชั้นข้อมูล Tasty = No จำนวนเท่า ๆ กัน เอนโทรปีคำนวณได้ ดังนี้

$$i(N) = -\frac{2}{4}log(\frac{2}{4}) - \frac{2}{4}log(\frac{2}{4}) = 1.0$$

สังเกตได้ว่า ค่าเอนโทรปีนี้มีค่าเท่ากับเอนโทรปีของ Asha ในหัวข้อที่ ก.1.(b) เนื่องจากเป็นการพิจารณาแบบที่ค่าฟีเจอร์ Cook = Asha เท่านั้น

ตารางที่ 5.7 สปริทที่ถูกแบ่งแยกไปยังกิ่ง Cook = Asha

Cook	Mood	Cuisine	Tasty
Asha	Bad	Indian	No
Asha	Good	Indian	Yes
Asha	Bad	Continental	No
Asha	Good	Continental	Yes

ต่อไป พิจารณาฟีเจอร์ Mood และ Cuisine เพื่อหาฟีเจอร์ที่เกนข่าวสารมีค่าสูงสุด

1. **Mood**

(a) Mood = Bad มี 2 แบบ ที่อยู่ในชั้นข้อมูล No และ ไม่มีแบบที่อยู่ในชั้นข้อมูล Yes ดังนั้น

$$i(N) = -\frac{2}{2}log(\frac{2}{2}) = 0.0$$

(b) Mood = Good มี 2 แบบ ที่อยู่ในชั้นข้อมูล Yes และ ไม่มีแบบ ที่อยู่ในชั้นข้อมูล No ดังนั้น

$$i(N) = -\frac{2}{2}log(\frac{2}{2}) = 0.0$$

จาก (a)-(b) คำนวณ เกนข่าวสาร ได้ดังนี้

$$\Delta i(N) = 1.0$$

2. **Cuisine**

(a) Cusine = Indian มี 1 แบบ ที่อยู่ในชั้นข้อมูล Yes และ 1 แบบ ที่อยู่ในชั้นข้อมูล No ดังนั้น

$$i(N) = -\frac{1}{2}log(\frac{1}{2}) - \frac{1}{2}log(\frac{1}{2}) = 1.0$$

(b) Cuisine = Continental มีอย่างละ 1 แบบ ที่อยู่ในชั้นข้อมูล Yes และ No ดังนั้น

$$i(N) = -\frac{1}{2}log(\frac{1}{2}) - \frac{1}{2}log(\frac{1}{2}) = 1.0$$

จาก (a)-(b) คำนวณ เกนข่าวสาร ได้ดังนี้

$$\Delta i = 1.0 - \frac{2}{4} \times 1.0 - \frac{2}{4} \times 1.0 = 0.0$$

เนื่องจากเกนข่าวสารของฟีเจอร์ Mood มีค่าสูงสุด ดังนั้น ฟีเจอร์นี้ถูกเลือกเป็นโหนดตามกิ่งซ้ายของต้นไม้การตัดสินใจแสดงตาม รูปที่ 5.21 นอกจากนี้ ภายหลังการแบ่งแยกที่เซตย่อยนี้ สปริทที่ได้มีดังต่อไปนี้

I. สปริท Mood = Bad แบบทั้งหมดมาจากชั้นข้อมูลเดียวกัน ดังนั้น โหนดใบของชั้นข้อมูล No จึงถูกสร้างภายใต้กิ่งซ้าย ตามรูปที่ 5.21

II. สปริท Mood = Good แบบทั้งหมดมาจากชั้นข้อมูลเดียวกัน ดังนั้น โหนดใบของชั้นข้อมูล Yes จึงถูกสร้างภายใต้กิ่งขวา ตามรูปที่ 5.21

ค. การสร้างโหนดภายในตามกิ่งกลางของราก

สปริทในตารางที่ 5.8 แบบทั้งหมดของฟีเจอร์ Cook = Sita อยู่ในชั้นข้อมูล Yes ดังนั้น ชั้นข้อมูลนี้จึงกลายเป็นโหนดใบตามกิ่งกลางของต้นไม้การตัดสินใจแสดงตาม รูปที่ 5.21

ตารางที่ 5.8 สปริทที่ถูกแบ่งแยกไปยังกิ่ง Cook = Sita

Cook	Mood	Cuisine	Tasty
Sita	Bad	Indian	Yes
Sita	Good	Continental	Yes
Sita	Good	Indian	Yes
Sita	Bad	Continental	Yes

ง. การสร้างโหนดภายในตามกิ่งขวาของราก

สปริทในตารางที่ 5.9 ประกอบไปด้วยแบบทั้งหมด จำนวน 4 แบบ โดยอยู่ในชั้นข้อมูล Tasty = Yes และ ชั้นข้อมูล Tasty = No จำนวนเท่า ๆ กัน เอนโทรปีคำนวณได้ ดังนี้

$$i(N) = -\frac{2}{4}log(\frac{2}{4}) - \frac{2}{4}log(\frac{2}{4}) = 1.0$$

สังเกตได้ว่า ค่าเอนโทรปีนี้มีค่าเท่ากับเอนโทรปีของ Usha ในหัวข้อที่ ก.1.(c) เนื่องจากเป็นการพิจารณาแบบที่ค่าฟีเจอร์ Cook = Usha เท่านั้น

ตารางที่ 5.9 สปริทที่ถูกแบ่งแยกไปยังกิ่ง Cook = Usha

Cook	Mood	Cuisine	Tasty
Usha	Bad	Indian	Yes
Usha	Bad	Continental	No
Usha	Good	Indian	Yes
Usha	Good	Continental	No

ต่อไป พิจารณาฟีเจอร์ Mood และ Cuisine เพื่อหาฟีเจอร์ที่เกนข่าวสารมีค่าสูงสุด

1. **Mood**

   (a) Mood = Bad มี 1 แบบ ที่อยู่ในชั้นข้อมูล Yes และ 1 แบบ ที่อยู่ในชั้นข้อมูล No ดังนั้น

   $$i(N) = -\frac{1}{2}log(\frac{1}{2}) - \frac{1}{2}log(\frac{1}{2}) = 1.0$$

   (b) Mood = Good มี 1 แบบ ที่อยู่ในชั้นข้อมูล Yes และ 1 แบบ ที่อยู่ในชั้นข้อมูล No ดังนั้น

   $$i(N) = -\frac{1}{2}log(\frac{1}{2}) - \frac{1}{2}log(\frac{1}{2}) = 1.0$$

   จาก (a)-(b) คำนวณ เกนข่าวสาร ได้ดังนี้

   $$\Delta i = 1.0 - \frac{2}{4} \times 1.0 - \frac{2}{4} \times 1.0 = 0.0$$

2. **Cuisine**

   (a) Cusine = Indian มี 2 แบบ ที่อยู่ในชั้นข้อมูล Yes และ ไม่มีแบบ ที่อยู่ในชั้นข้อมูล No ดังนั้น

   $$i(N) = -\frac{2}{2}log(\frac{2}{2}) = 0.0$$

   (b) Cuisine = Continental มี 2 แบบ ที่อยู่ในชั้นข้อมูล No และ ไม่มีแบบ ที่อยู่ในชั้นข้อมูล No ดังนั้น

   $$i(N) = -\frac{2}{2}log(\frac{2}{2}) = 0.0$$

จาก (a)-(b) คำนวณ เกนข่าวสาร ได้ดังนี้

$$\Delta i = 1.0$$

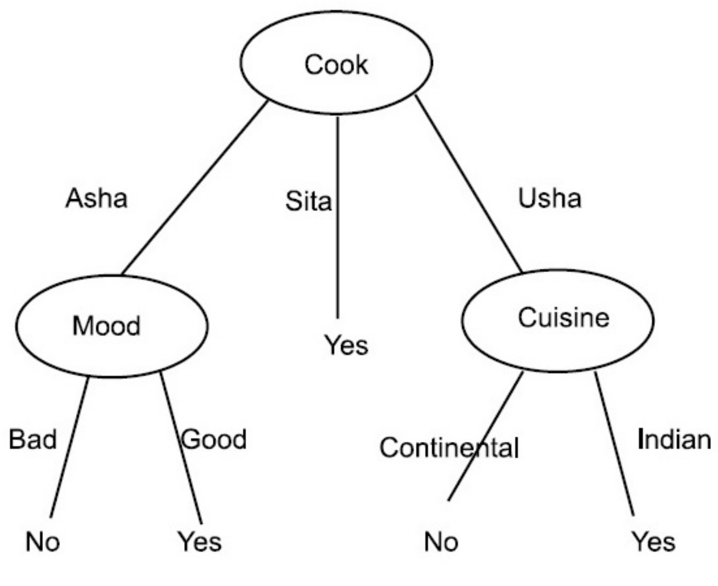
เนื่องจากเกนข่าวสารของฟีเจอร์ Cuisine มีค่าสูงสุด ดังนั้น ฟีเจอร์นี้ถูกเลือกเป็นโหนดตามกิ่งซ้ายของต้นไม้การตัดสินใจแสดงตาม รูปที่ 5.21 นอกจากนี้ ภายหลังการแบ่งแยกที่สปริทนี้ สปริทย่อยที่ได้มีดังต่อไปนี้

I. สปริท Cuisine = Continental แบบทั้งหมดมาจากชั้นข้อมูลเดียวกัน ดังนั้น โหนดใบของชั้นข้อมูล No จึงถูกสร้างภายใต้กิ่งซ้าย ตามรูปที่ 5.21

II. สปริท Cuisine = Indian แบบทั้งหมดมาจากชั้นข้อมูลเดียวกัน ดังนั้น โหนดใบของชั้นข้อมูล Yes จึงถูกสร้างภายใต้กิ่งขวา ตามรูปที่ 5.21

จากรูปที่ 5.21 สามารถวิเคราะห์ต้นไม้การตัดสินใจ ได้หลายประเด็น ดังต่อไปนี้

- พิจารณาตารางที่ 5.7 ถ้าแบบทดสอบมีค่าฟีเจอร์ Cook = Asha แล้ว Cuisine มี ความซ้ำซ้อน (Redundant) ในชั้นข้อมูลทั้ง Yes และ No ดังนั้น จึงไม่สามารถแบ่งแยกแบบโดยฟีเจอร์นี้ได้
- พิจารณาตารางที่ 5.8 ถ้าแบบทดสอบมีค่าฟีเจอร์ Cook = Sita จะสามารถจำแนกแบบให้อยู่ในชั้นข้อมูล Yes ได้ทันที โดยไม่จำเป็นต้องใช้ข้อมูลจาก Mood และ Cuisine
- พิจารณาตารางที่ 5.9 ถ้าแบบทดสอบมีค่าฟีเจอร์ Cook = Usha แล้ว จะไม่สามารถแบ่งแยกแบบได้ ในทำนองเดียวกับฟีเจอร์ Cook = Asha



รูปที่ 5.21 ต้นไม้การตัดสินใจจากเซตข้อมูลภัตตาคาร

ตัวอย่างที่ 12 จงระบุกฎการตัดสินใจทั้งหมด ที่ได้จากต้นไม้การตัดสินใจใน ตัวอย่างที่ 11

กฎการตัดสินใจที่ 1 ถ้า Cook = Asha และ Mood = Bad แล้ว Tasty = No

กฎการตัดสินใจที่ 2 ถ้า Cook = Asha และ Mood = Good แล้ว Tasty = Yes

กฎการตัดสินใจที่ 3 ถ้า Cook = Sita แล้ว Tasty = Yes

กฎการตัดสินใจที่ 4 ถ้า Cook = Usha และ Cuisine = Continental แล้ว Tasty = No

กฎการตัดสินใจที่ 5 ถ้า Cook = Usha และ Cuisine = Indian แล้ว Tasty = Yes

ตัวอย่างที่ 13 จงทำการจำแนกแบบดังต่อไปนี้ โดยใช้กฎการตัดสินใจจาก ตัวอย่างที่ 12

แบบ ก *“กุ๊กคือสิธาที่อารมณ์เสียในขณะทำอาหารอินเดีย”*

กำหนดค่าฟีเจอร์ต่าง ๆ ดังนี้

Cook = Sita, Mood = Bad, Cuisine = Indian

จากกฎการตัดสินใจที่ 3 ในตัวอย่างที่ 12

Tasty = Yes

แบบ ข *“กุ๊กคืออุชาที่อารมณ์ดีในขณะทำอาหารยุโรป”*

กำหนดค่าฟีเจอร์ต่าง ๆ ดังนี้

Cook = Usha, Mood = Good, Cuisine = Continental

จากกฎการตัดสินใจที่ 4 ในตัวอย่างที่ 12

Tasty = No

แบบ ค *“กุ๊กคืออาชาที่กำลังทำอาหารอินเดีย”*

ไม่สามารถจำแนกแบบได้ เนื่องจากกฏที่ค่าฟีเจอร์ Cook = Asha ไม่มีค่าฟีเจอร์ Mood

ตัวอย่างที่ 14 จากเซตฝึกฝนของข้อมูลแพริตีบิตในตารางที่ 5.10 จงอุปนัยต้นไม้การตัดสินใจโดยใช้เอนโทรปีเป็นหน่วยวัดความเจือปน เพื่อทำนายแบบทดสอบ B = (NONE, 0, 0) กำหนดให้ฟีเจอร์ BIT3 แทนชั้นข้อมูล ความหมายของแต่ละฟีเจอร์มีรายละเอียด ดังต่อไปนี้

- Parity แทน ภาวะของแพริตีบิต ประกอบไปด้วย
    - ODD บิตภาวะคี่
    - EVEN บิตภาวะคู่
    - NONE ไม่ทราบภาวะ
- BIT1 แทน บิตที่ 1
- BIT2 แทน บิตที่ 2
- BIT3 แทน บิตที่ 3

ตารางที่ 5.10 เซตข้อมูลแพริตีบิต

Parity	**BIT1**	**BIT2**	**BIT3: Class**
ODD	0	1	0
ODD	0	0	0
EVEN	0	1	1
EVEN	0	0	1
EVEN	1	1	0
EVEN	1	0	0
NONE	1	1	1
NONE	1	0	1

ก. การสร้างโหนดราก

เซตข้อมูลในตารางที่ 5.10 ประกอบไปด้วยแบบทั้งหมด จำนวน 8 แบบ โดยอยู่ในชั้นข้อมูล BIT3 = 0 จำนวน 4 แบบ และ ชั้นข้อมูล BIT3 = 1 จำนวน 4 แบบ เอนโทรปีคำนวณได้ ดังนี้

$$i(N) = -\frac{4}{8}log(\frac{4}{8}) - \frac{4}{8}log(\frac{4}{8}) = 1.0$$

พิจารณาฟีเจอร์ที่เกนข่าวสารมีค่าสูงสุด เพื่อนำมาสร้างเป็นรากของต้นไม้การตัดสินใจ ได้ตามขั้นตอนต่อไปนี้

1. **Parity**

(a) Parity = NONE แบบทั้งหมดอยู่ในชั้นข้อมูล 1 ดังนั้น

$$i(N) = -\frac{2}{2}log(\frac{2}{2}) = 0.0$$

(b) Parity = EVEN มีแบบอย่างละครึ่งอยู่ในชั้นข้อมูลทั้งสอง ดังนั้น

$$i(N) = -\frac{2}{4}log(\frac{2}{4}) - \frac{2}{4}log(\frac{2}{4}) = 1.0$$

(c) Parity = ODD แบบทั้งหมดอยู่ในชั้นข้อมูล 0 ดังนั้น

$$i(N) = -\frac{2}{2}log(\frac{2}{2}) = 0.0$$

จาก (a)-(c) คำนวณ เกนข่าวสาร ได้ดังนี้

$$\Delta i(N) = 1.0 - \frac{4}{8} \times 1.0 = 0.5$$

2. **BIT1**

(a) BIT1 = 1 มีแบบอย่างละครึ่งอยู่ในชั้นข้อมูลทั้งสอง ดังนั้น

$$i(N) = -\frac{2}{4}log(\frac{2}{4}) - \frac{2}{4}log(\frac{2}{4}) = 1.0$$

(b) BIT1 = 0 มีแบบอย่างละครึ่งอยู่ในชั้นข้อมูลทั้งสอง ดังนั้น

$$i(N) = -\frac{2}{4}log(\frac{2}{4}) - \frac{2}{4}log(\frac{2}{4}) = 1.0$$

จาก (a)-(b) คำนวณ เกนข่าวสาร ได้ดังนี้

$$\Delta i(N) = 1.0 - \frac{4}{8} \times 1.0 - \frac{4}{8} \times 1.0 = 0.0$$

3. **BIT2**

(a) BIT2 = 1 มีแบบอย่างละครึ่งอยู่ในชั้นข้อมูลทั้งสอง ดังนั้น

$$i(N) = -\frac{2}{4}log(\frac{2}{4}) - \frac{2}{4}log(\frac{2}{4}) = 1.0$$

(b) BIT2 = 0 มีแบบอย่างละครึ่งอยู่ในชั้นข้อมูลทั้งสอง ดังนั้น

$$i(N) = -\frac{2}{4}log(\frac{2}{4}) - \frac{2}{4}log(\frac{2}{4}) = 1.0$$

จาก (a)-(b) คำนวณ เกนข่าวสาร ได้ดังนี้

$$\Delta i(N) = 1.0 - \frac{4}{8} \times 1.0 - \frac{4}{8} \times 1.0 = 0.0$$

ฟีเจอร์ Parity มีค่าเกนข่าวสารสูงสุด จึงถูกเลือกเป็นรากของต้นไม้การตัดสินใจ แสดงตามรูปที่ 5.22 เนื่องจากค่าที่เป็นไปได้ของฟีเจอร์นี้มี 3 ค่า ดังนั้น แบบฝึกฝนในตารางที่ 5.10 จึงถูกแบ่งออกเป็น 3 สปริท ตามกิ่งต่าง ๆ ดังต่อไปนี้

ข. สปริท Parity = NONE ที่บรรจุแบบฝึกฝนตามตารางที่ 5.11 ตามกิ่งซ้ายของราก

ค. สปริท Parity = EVEN ที่บรรจุแบบฝึกฝนตามตารางที่ 5.12 ตามกิ่งกลางของราก

ง. สปริท Parity = ODD ที่บรรจุแบบฝึกฝนตามตารางที่ 5.13 ตามกลิ่งขวาของราก

ข. การสร้างโหนดภายในตามกิ่งซ้ายของราก

จากสปริทในตารางที่ 5.11 แบบทั้งหมดของฟีเจอร์ Parity อยู่ในชั้นข้อมูล 1 ดังนั้น ชั้นข้อมูลนี้ถูกเลือกให้เป็นโหนดใบของต้นไม้การตัดสินใจ แสดงตามรูปที่ 5.22

ตารางที่ 5.11 สปริทที่ถูกแบ่งแยกไปยังกิ่ง Parity = NONE

Parity	BIT1	BIT2	BIT3: CLASS
NONE	1	1	1
NONE	1	0	1

ค. การสร้างโหนดภายในตามกิ่งกลางของราก

เนื่องจากเป็นการพิจารณาแบบของฟีเจอร์ Parity = EVEN เท่านั้น จึงนำค่าเอนโทรโทรปีของ EVEN ในหัวข้อที่ ก.1.(b) มาใช้ได้เลย

$$i(N) = 1.0$$

จากนั้น ค้นหาฟีเจอร์ที่เกนข่าวสารมีค่าสูงสุด จากฟีเจอร์ที่เหลือ

1. BIT1

(a) BIT1 = 1 แบบทั้งหมดอยู่ในชั้นข้อมูล 0 ดังนั้น

$$i(N) = -\frac{2}{2} log(\frac{2}{2}) = 0.0$$

ตารางที่ 5.12 สปริทที่ถูกแบ่งแยกไปยังกิ่ง Parity = EVEN

Parity	BIT1	BIT2	BIT3: CLASS
EVEN	0	1	1
EVEN	0	0	1
EVEN	1	1	0
EVEN	1	0	0

(b) BIT1 = 0 แบบทั้งหมดอยู่ในชั้นข้อมูล 1 ดังนั้น

$$i(N) = -\frac{2}{2}log(\frac{2}{2}) = 0.0$$

จาก (a)-(b) คำนวณ เกนข่าวสาร ได้ดังนี้

$$\Delta i(N) = 1.0 - \frac{4}{8} \times 1.0 - \frac{4}{8} \times 1.0 = 1.0$$

2. BIT2

(a) BIT2 = 1 มีแบบอย่างละครึ่งอยู่ในชั้นข้อมูลทั้งสอง ดังนั้น

$$i(N) = -\frac{1}{2}log(\frac{1}{2}) - \frac{1}{2}log(\frac{1}{2}) = 1.0$$

(b) BIT2 = 0 มีแบบอย่างละครึ่งอยู่ในชั้นข้อมูลทั้งสอง ดังนั้น

$$i(N) = -\frac{1}{2}log(\frac{1}{2}) - \frac{1}{2}log(\frac{1}{2}) = 1.0$$

จาก (a)-(b) คำนวณ เกนข่าวสาร ได้ดังนี้

$$\Delta i(N) = 1.0 - \frac{2}{4} \times 1.0 - \frac{2}{4} \times 1.0 = 0.0$$

เนื่องจากฟีเจอร์ BIT1 มีค่าเกนข่าวสารสูงสุด จึงถูกเลือกเป็นโหนดตามกิ่งกลางของต้นไม้การตัดสินใจ แสดงตาม รูปที่ 5.22 หลังจากขั้นตอนนี้ สปริทจะโดนแบ่งแยกออกเป็นสปริทย่อย ดังต่อไปนี้

I. สปริท BIT1 = 0 แบบทั้งหมดมาจากชั้นข้อมูล 1 ดังนั้น โหนดใบของชั้นข้อมูล 1 จึงถูกสร้างภายใต้กิ่งซ้าย ตามรูปที่ 5.22

II. สปริท BIT1 = 1 แบบทั้งหมดมาจากชั้นข้อมูล 0 ดังนั้น โหนดใบของชั้นข้อมูล 0 จึงถูกสร้างภายใต้กิ่งขวา ตามรูปที่ 5.22

ง. การสร้างโหนดภายในตามกิ่งขวาของราก

จากสปริทในตารางที่ 5.13 แบบทั้งหมดของฟีเจอร์ Parity อยู่ในชั้นข้อมูล 0 ดังนั้น ชั้นข้อมูลนี้ถูกเลือกให้เป็นโหนดใบของต้นไม้การตัดสินใจ แสดงตามรูปที่ 5.22

ตารางที่ 5.13 สปริทแพริตีบิตที่ถูกแบ่งแยกไปยังกิ่ง Parity = ODD

Parity	BIT1	BIT2	BIT3: CLASS
ODD	0	1	0
ODD	0	0	0

กฎการตัดสินใจทั้งหมด ที่ได้จากต้นไม้การตัดสินใจในตัวอย่างนี้ ได้แก่

กฎการตัดสินใจที่ 1 ถ้า PARITY = NONE แล้ว BIT3 = 1

กฎการตัดสินใจที่ 2 ถ้า PARITY = ODD แล้ว BIT3 = 0

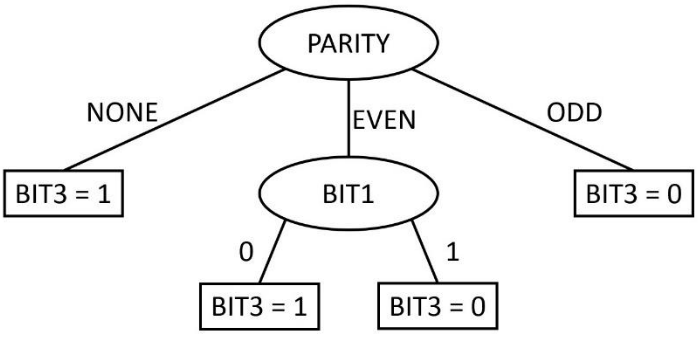
กฎการตัดสินใจที่ 3 ถ้า PARITY = EVEN และ BIT1 = 0 แล้ว BIT3 = 1

กฎการตัดสินใจที่ 4 ถ้า PARITY = EVEN และ BIT1 = 1 แล้ว BIT3 = 0

แบบทดสอบ B = (NONE, 0, 0) มีค่าฟีเจอร์ต่าง ๆ ดังนี้

- PARITY = NONE
- BIT1 = 0
- BIT2 = 0

แบบทดสอบ B เป็นไปตามกฎที่ 1 ดังนั้น B ถูกจำแนกให้อยู่ในชั้นข้อมูล BIT3 = 1



รูปที่ 5.22 ต้นไม้การตัดสินใจจากเซตข้อมูลแพริตีบิต

อภิปราย

ต้นไม้การตัดสินใจเป็นตัวจำแนกแบบ ที่ได้รับความนิยมใช้งานกันอย่างแพร่หลายในปัจจุบัน เนื่องจากใช้งานได้ดีในหลายโปรแกรมประยุกต์ ตัวแบบชนิดนี้มีข้อดีหลายข้อ อย่างไรก็ตามตัวแบบชนิดนี้ยังมีข้อเสียบางข้อที่อาจไม่สามารถนำไปใช้งานได้ในบางกรณี รายการจุดแข็งและจุดอ่อนของต้นไม้การตัดสินใจ มีดังต่อไปนี้

จุดแข็ง

- อธิบายการทำงานของตัวแบบให้ผู้ใช้เข้าใจได้ โดยใช้เวลาไม่นาน
- ทำความเข้าใจและแปลผลกฎการตัดสินใจได้ง่าย
- สามารถสร้างต้นไม้การตัดสินใจได้จากเซตข้อมูลที่มีฟีเจอร์ชนิด ต่อเนื่อง หรือ ประเภท ก็ได้
- บ่งบอกได้ว่าฟีเจอร์ไหนมีความสำคัญสูงสุดสำหรับการจำแนกแบบ
- มองเห็นกระบวนการทำงานภายในเนื่องจากมีลักษณะเป็น กล่องขาว ซึ่งมีลักษณะตรงข้ามกับกล่องดำ ที่ผู้ใช้จะเห็นแค่ข้อมูลเข้าและข้อมูลออกเท่านั้น
- สามารถเพิ่มประสิทธิภาพการทำงานได้ โดยการรวมต้นไม้การตัดสินใจหลาย ๆ ต้นเข้าด้วยกันเป็นตัวจำแนกแบบชนิด ป่าสุ่ม (Random Forrest)

จุดอ่อน

- สำหรับเซตข้อมูลอสมดุลที่สัดส่วนของชั้นข้อมูลมีความแตกต่างกันสูงมาก ต้นไม้การตัดสินใจจะมี อคติ (Bias) กับชั้นข้อมูลที่มีขนาดเล็กมาก
- การสร้างโหนดภายในของต้นไม้การตัดสินใจขนานแกน พิจารณาได้เพียงฟีเจอร์เดียวเท่านั้น ไม่สามารถพิจารณากลุ่มฟีเจอร์ได้
- ขอบเขตการตัดสินใจของต้นไม้การตัดสินใจขนานแกน ต้องมีลักษณะเป็นเส้นตรงที่ตั้งฉากกับแกนของฟีเจอร์เท่านั้น ดังนั้นเซตข้อมูลส่วนใหญ่อาจไม่สามารถแบ่งแยกเชิงเส้นได้ เมื่อพิจารณาเพียงแกนใดแกนหนึ่งเท่านั้น
- มีความ ไม่แน่นอน (Unstable) ถ้าเซตข้อมูลมีการเปลี่ยนแปลงเพียงเล็กน้อย อาจส่งผลให้โครงสร้างของต้นไม้การตัดสินใจมีโอกาสสูงที่จะเปลี่ยนแปลงไปจากเดิม
- ในกระบวนการสร้างต้นไม้การตัดสินใจ มีต้นทุนทางด้านเวลาสูง เนื่องจากการพิจารณาสร้างโหนดแต่ละโหนดต้องมีการแตกตารางในหลายรูปแบบ เพื่อเปรียบเทียบรูปแบบที่เกนข่าวสารมีค่าสูงสุด

- การเล็มกิ่ง มีต้นทุนทางด้านเวลาสูง เนื่องจากต้องมีการสร้างและเปรียบเทียบ ต้นไม้ย่อย (Sub-tree) จำนวนหลายต้น เพื่อเลือกต้นไม้ย่อยที่ดีที่สุด
- สำหรับเซตข้อมูลที่มีชั้นข้อมูลจำนวนมาก ต้นไม้การตัดสินใจมีโอกาสที่จะทำนายผิดพลาดสูง
- การตีความ ป่าสุ่ม ทำได้ยาก ไม่เหมือนการตีความ ต้นไม้การตัดสินใจ เพียงต้นเดียว

สรุป

ต้นไม้การตัดสินใจคือตัวแบบที่ถูกนำไปประยุกต์ใช้อย่างหลากหลายในสาขาวิชาต่าง ๆ เช่น ระบบสนับสนุนการตัดสินใจ ปัญญาประดิษฐ์ การเรียนรู้ของเครื่อง การทำเหมืองข้อมูล และ การรู้จำแบบ ต้นไม้การตัดสินใจเป็นโครงสร้างข้อมูลชนิดทรีซึ่งมีองค์ประกอบ ได้แก่ โหนดราก โหนดภายใน โหนดใบ และ กิ่ง โดยที่ โหนดรากและโหนดภายในเป็นตัวแทนของฟีเจอร์ในเซตข้อมูล โหนดใบเป็นตัวแทนของชั้นข้อมูล กิ่งเป็นตัวแทนของค่าฟีเจอร์ นอกจากนี้ เส้นทางจากโหนดรากจนถึงโหนดใบในต้นไม้การตัดสินใจนำมาซึ่งกลุ่มของกฎการตัดสินใจในรูปแบบ *“ถ้า-แล้ว”* เช่น *“ถ้าฝนไม่ตก และ มีเมฆครึ้ม แล้วเด็กนักเรียนจะออกไปเล่นที่สนามหญ้า”*

ในกระบวนการสร้างต้นไม้ตัดสินใจสำหรับการจำแนกแบบ ฟีเจอร์เดี่ยวหรือกลุ่มของฟีเจอร์ ในเซตข้อมูล จะโดนทดสอบเลือกฟีเจอร์ที่เหมาะสมที่สุด เพื่อนำไปสร้างโหนดภายใน การทดสอบสามารถแบ่งออกได้เป็น 3 ประเภท ได้แก่ การทดสอบขนานแกนที่ใช้การแบ่งแยกแบบขนานแกน การทดสอบเชิงเส้นบนกลุ่มฟีเจอร์ที่ใช้การแบ่งแยกแบบเอียง และ การทดสอบไม่เชิงเส้นบนกลุ่มฟีเจอร์ที่ใช้ฟังก์ชันไม่เชิงเส้นในการแบ่งแยกแบบจากต่างชั้นข้อมูลกันออกจากกัน

ต้นไม้การตัดสินใจขนานแกนคือต้นไม้การตัดสินใจประเภทหนึ่ง ที่แต่ละกฎย่อยจะพิจารณาเพียงฟีเจอร์เดียวเท่านั้น กล่าวคือ กฎ 1 กฎมีความเกี่ยวข้องกับตัวแปรเพียง 1 ตัวแปรเท่านั้น เช่น *“อายุมากกว่า 18 ปี”* ต้นไม้การตัดสินใจชนิดนี้ไม่สามารถพิจารณากลุ่มฟีเจอร์ได้ หมายความว่า ไม่มีกฎย่อยใดที่เกี่ยวข้องกับตัวแปรมากกว่าหนึ่งตัว เช่น ตัวแบบจะไม่สามารถสร้างกฎย่อย *“อายุมากกว่า 18 ปี และ สำเร็จการศึกษาแล้ว และ เงินเดือนต่ำกว่า 2 หมื่นบาท”* ได้ การสร้างขอบเขตการตัดสินใจในลักษณะนี้มีชื่อเรียกว่า การจำแนกประเภทสี่เหลี่ยมมุมฉาก ที่ระนาบเกินต้องตั้งฉากกับแกนของฟีเจอร์เท่านั้น และปริภูมิที่ได้มีลักษณะเป็นรูปทรงสี่เหลี่ยมมุมฉาก

ฟีเจอร์จำแนกมีคุณสมบัติคือการตัดสินใจเลือกระนาบเกินจากค่าบนแกนของฟีเจอร์นี้สามารถแบ่งแบบจากชั้นข้อมูลต่างกันให้ออกจากกันได้ หรืออาจกล่าวได้ว่า ค่าฟีเจอร์ของแบบในชั้นข้อมูลเดียวกันจะมีค่าคล้ายกันหรือเกาะกลุ่มกัน แต่ค่าฟีเจอร์ของแบบจากต่างชั้นข้อมูลกันจะมีค่าแตกต่างกัน

หรือกระจายตัวกัน ฟีเจอร์จำแนกมีความสำคัญในกระบวนการสร้างต้นไม้การตัดสินใจ เนื่องจากเอนโทรปีซึ่งเป็นตัววัดที่ใช้เลือกฟีเจอร์มาสร้างโหนดภายในของต้นไม้การตัดสินใจ จะให้ความสำคัญกับฟีเจอร์จำแนกมากกว่าฟีเจอร์อื่น ๆ

เอนโทรปีคือหน่วยวัดการเจือปน มีค่าแปรผันผกผันกับค่าความบริสุทธิ์ เอนโทรปีมีค่าต่ำถ้าแบบมีการเจือปนน้อยหรือเซตของแบบมาจากชั้นข้อมูลเพียงไม่กี่ชั้น ในทางกลับกัน เอนโทรปีมีค่าสูงถ้าแบบมีการเจือปนมากหรือเซตของแบบมาจากการผสมของหลายชั้นข้อมูล ในขณะที่ เกนข่าวสารคือค่าการลดค่าการเจือปนของโหนดที่เกิดการแบ่งแยก ฟีเจอร์ที่นำไปสร้างโหนดภายในของต้นไม้การตัดสินใจต้องมีค่าเกนข่าวสารสูงสุด

ปัญหากระชับมากเกินไปเกิดขึ้นกับต้นไม้การตัดสินใจที่มีความลึกมากเกินไป ส่งผลเสียให้การจำแนกแบบมีความเฉพาะเจาะจงกับแบบจากเซตฝึกฝน ทำให้การทำนายแบบจากเซตทดสอบขาดความแม่นยำเป็นส่วนใหญ่ การรับมือปัญหาประเภทนี้จำเป็นต้องใช้ขั้นตอนวิธีเล็มกิ่งเพื่อกำจัดฟีเจอร์นอกประเด็นทิ้งไป นอกจากนี้ การเล็มกิ่งส่งผลให้ความลึกของต้นไม้การตัดสินใจมีความตื้นขึ้น

การอุปนัยการต้นไม้การตัดสินใจหมายถึง การสร้างต้นไม้การตัดสินใจจากรากจนถึงใบใน โดยจะเกิดการแบ่งแยกเซตข้อมูลทุกครั้งภายหลังการสร้างโหนดภายใน เซตย่อยแต่ละเซตถูกเรียก สปริท ถ้าการแบ่งแยกได้สปริทที่ประกอบไปด้วยแบบจากชั้นข้อมูลเดียวกัน แล้วการแบ่งแยกจะสิ้นสุดลงและสร้างโหนดใบให้กับตัวแบบ กระบวนการอุปนัยตัวแบบนี้ดำเนินไปในลักษณะของความสัมพันธ์เวียนเกิดและสิ้นสุดลงเมื่อต้นไม้หยุดการแบ่งแยกแบบ

ตัวจำแนกแบบจากเนื้อหาในบทที่แล้วและบทนี้ จำแนกโดยอาศัยหลักความน่าจะเป็นโดยสนใจไปที่จำนวนแบบในแต่ละชั้นข้อมูล ในบทต่อไปจะกล่าวถึงตัวจำแนกแบบที่ทำนายโดยใช้หลักการที่แตกต่างกันออกไป ได้แก่ การเรียนรู้เพอร์เซ็ปตรอน ที่จำแนกแบบโดยใช้ฟังก์ชันการจำแนกเชิงเส้นเพื่อสร้างระนาบที่ใช้แบ่งบริเวณของแบบจากชั้นข้อมูลต่างกันออกจากกัน นอกจากนี้ ในบทต่อไปยังกล่าวถึงโครงข่ายประสาท ซึ่งเป็นตัวจำแนกแบบที่มีการทำงานเลียนแบบสมองของมนุษย์

แบบฝึกหัด

1. จากเซตข้อมูลในตารางที่ 5.14 และ 5.15 สามารถสร้างต้นไม้การตัดสินใจได้ตามรูปที่ 5.23 จงตอบคำถามต่อไปนี้

6.1 จงระบุป้ายชื่อชั้นข้อมูลทั้งหมดในต้นไม้

6.2 จงยกตัวอย่างกฎที่ได้จากต้นไม้มา 2 กฎ

6.3 ฟีเจอร์ใดบ้างเป็นฟีเจอร์นอกประเด็น

6.4 ฟีเจอร์ใดเป็นตัวเลข และ ฟีเจอร์ใดเป็นประเภท

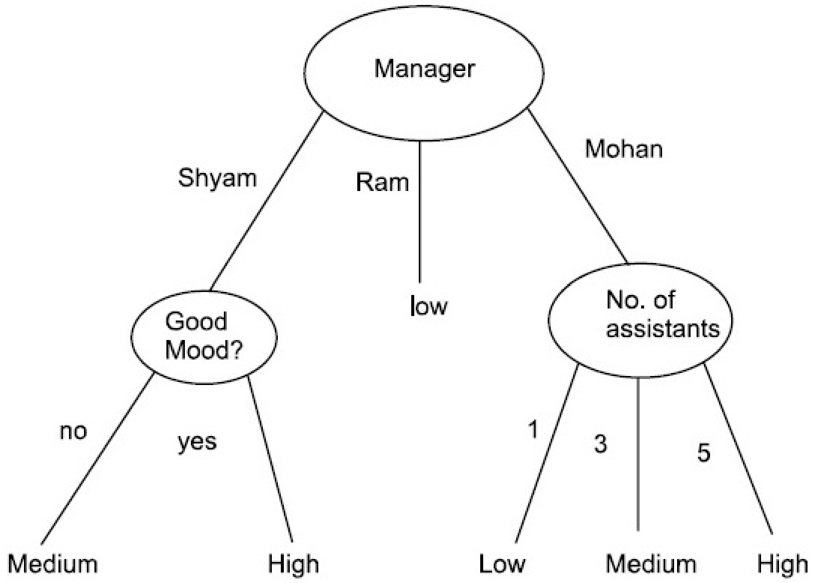
6.5 ภายหลังการตัดสินใจที่ Good Mood? กิ่งของค่า Medium บรรจุไปด้วยแบบใดบ้าง

ตารางที่ 5.14 บันทึกพนักงานในบริษัท

Name	Age	Educational Qualification	Position
Ram	55	B Com	Manager
Shyam	30	BE	Manager
Mohan	40	MSc	Manager

ตารางที่ 5.15 แบบที่ใช้ในการสร้างต้นไม้การตัดสินใจ

Manager	Number of Assistants	Mood	Output
Shyam	3	No	Medium
Shyam	5	No	Medium
Shyam	1	Yes	High
Ram	1	Yes	Low
Ram	5	No	Low
Ram	5	Yes	Low
Mohan	1	No	Low
Mohan	3	Yes	Medium
Mohan	5	No	High



รูปที่ 5.23 ต้นไม้สำหรับการจำแนกแบบ

2. จงสร้างต้นไม้การตัดสินใจสำหรับปัญหาการเลือกภัตตาคารโดยมีฟีเจอร์ ดังต่อไปนี้

   a) Expense: Expensive, Reasonable, Cheap

   b) Location from Home: Far, Near, Very Close

   c) Quality of Food: Very Good, Fair, Bad

   d) Weather: Raining, Sunny, Overcast

3. พิจารณาต้นไม้การตัดสินใจสำหรับจำแนกแบบทั้งหมด 25 แบบ โดยที่ รากแบ่งแบบออกเป็น กิ่งซ้าย 10 แบบ และ กิ่งขวา 15 แบบ นอกจากนี้ ที่กิ่งซ้ายยังมีการแบ่งแบบออกเป็นอีก 3 กิ่งย่อย จำนวน 0, 4 และ 6 แบบ จากกิ่งย่อยซ้ายมือสุดไปขวามือสุดตามลำดับ จงคำนวณ เกนข่าวสาร ของแต่ละกิ่งในต้นไม้ โดยใช้หน่วยวัดการเจือปน ดังต่อไปนี้

   6.1 เอนโทรปี

   6.2 การเจือปนจีนี

   6.3 การเจือปนจำแนกผิดพลาด

4. จงสร้างต้นไม้การตัดสินใจจากเซตฝึกฝน ในตัวอย่างที่ 6 ของบทที่ 4

5. จงสร้างต้นไม้การตัดสินใจจากเซตฝึกฝน ในแบบฝึกหัดข้อ 4 ของบทที่ 4

6. พิจารณาแบบใน 2 มิติ ด้านล่าง พร้อมทั้งตอบคำถาม ดังต่อไปนี้

    6.1 จงวาดรูประนาบเกินของการแบ่งแยกแบบขนานแกนที่ดีที่สุด โดยใช้เอนโทรปี

    6.2 จงวาดรูประนาบเกินของการแบ่งแยกแบบเอียง ในกรณีที่สามารถหาได้

Class 1	**Class 2**
(1, 1)	(6, 1)
(1, 2)	(6, 2)
(2, 1)	(1, 8)
	(2, 7)
	(2, 8)

บทที่ 6
โครงข่ายประสาทและเครื่องกลเวกเตอร์สนับสนุน

ตัวจำแนกแบบที่กล่าวถึงในหลายบทก่อนหน้า ได้แก่ ขั้นตอนวิธีเพื่อนบ้านใกล้ที่สุด ตัวจำแนกเบส์ และ ต้นไม้การตัดสินใจ ตัวแบบทั้งหมดนี้สามารถใช้กับเซตข้อมูลที่มีหลายชั้นข้อมูลได้ ซึ่งแตกต่างกับตัวจำแนกแบบในบทนี้ที่มีลักษณะเฉพาะตัว ได้แก่ เครื่องกลเวกเตอร์สนับสนุน ซึ่งเป็นตัวจำแนกแบบชนิดไบนารี ตัวแบบชนิดนี้ทำการจำแนกแบบในเซตข้อมูลที่บรรจุแบบสองชั้นข้อมูลหรือมากกว่า โดยการสร้างขอบเขตการตัดสินใจในปริภูมิหลายมิติจากแบบในเซตย่อยของเซตฝึกฝน แบบในที่นี้เรียกว่า เวกเตอร์สนับสนุน ในทางเรขาคณิต เวกเตอร์สนับสนุนคือแบบที่อยู่ใกล้ขอบเขตการตัดสินใจมากที่สุด ในการทำความเข้าใจเครื่องกลเวกเตอร์สนับสนุน ความรู้พื้นฐานที่จำเป็นต้องทราบประกอบไปด้วย ฟังก์ชันการจำแนกเชิงเส้น การเรียนรู้เพอร์เซ็ปตรอน และ โครงข่ายประสาท ดังนั้น เนื้อหาในช่วงต้นของบทจะปูพื้นฐานดังกล่าว ก่อนกล่าวถึงเครื่องกลเวกเตอร์สนับสนุนเป็นเรื่องสุดท้าย

จุดประสงค์การเรียนรู้

- เรียนรู้วิธีการจำแนกแบบโดยประยุกต์ฟังก์ชันการจำแนกเชิงเส้น
- เข้าใจการปรับเวกเตอร์น้ำหนักของการเรียนรู้เพอร์เซ็ปตรอน
- ทราบการวิวัฒน์โครงข่ายประสาทจากสมองมนุษย์
- ศึกษาการทำงานของเครื่องกลเวกเตอร์สนับสนุน

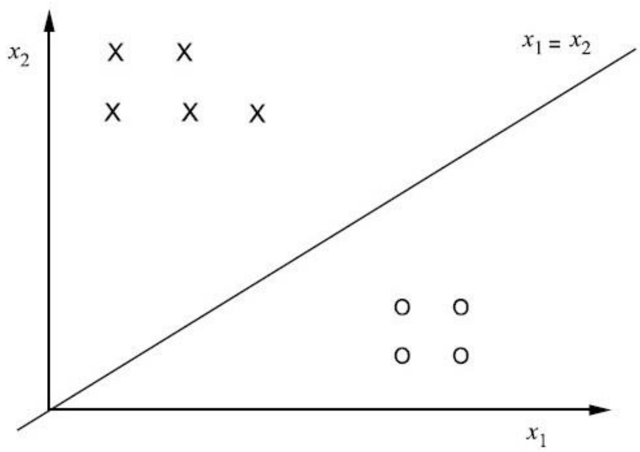
1. ฟังก์ชันการจำแนกเชิงเส้น

ฟังก์ชันการจำแนกเชิงเส้น (Linear Discriminant Function) คือฟังก์ชันที่สามารถใช้แบ่งแยกแบบจากต่างชั้นข้อมูลกันออกจากกัน ตัวอย่างด้านล่างนี้ แสดงการแบ่งแยกแบบในปริภูมิ 2 มิติ ในเซตข้อมูลที่มี 2 ชั้นข้อมูล ออกจากกัน

ตัวอย่างที่ 1 เซตข้อมูลใน ตารางที่ 6.1 ประกอบด้วยแบบจำนวน 9 แบบ โดยอยู่ในชั้นข้อมูล X จำนวน 5 แบบ และ อยู่ในชั้นข้อมูล O จำนวน 4 แบบ ในส่วนเพิ่มเติม x_1 และ x_2 แทนค่าในฟีเจอร์แรกและฟีเจอร์หลังตามลำดับ แบบทั้งหมดในปริภูมิ 2 มิติ แสดงตาม รูปที่ 6.1

ตารางที่ 6.1 รายละเอียดของแบบ

Pattern No.	1	2	Class
1	0.5	3.0	X
2	1.0	3.0	X
3	0.5	2.5	X
4	1.0	2.5	X
5	1.5	2.5	X
6	4.5	1.0	O
7	5.0	1.0	O
8	4.5	0.5	O
9	5.5	0.5	O



รูปที่ 6.1 ฟังก์ชันการจำแนกเชิงเส้น $x_1 = x_2$

พิจารณาสมการเส้นตรง $x_1 = x_2$ ที่แบ่งแยกแบบจากชั้นข้อมูลทั้งสองออกจากกัน โดยแบบจากชั้นข้อมูล X อยู่บริเวณทางฝั่งซ้าย แบบจากชั้นข้อมูล O อยู่บริเวณทางฝั่งขวา หรือกล่าวอีกนัยหนึ่งได้ว่าแบบกลุ่มนี้มีคุณสมบัติ การจำแนกเชิงเส้น (Linearly Separable) เนื่องจากสามารถหาเส้นตรงที่เป็นไปตามเงื่อนไขคือ แบบที่อยู่ในชั้นข้อมูลเดียวกันต้องอยู่ฝั่งเดียวกัน แต่แบบจากชั้นข้อมูลต่างกันต้องอยู่ต่างฝั่งกัน อีกหนึ่งวิธีในการแบ่งแยกแบบออกจากกันคือใช้อสมการ โดยมีรายละเอียด ดังต่อไปนี้

- แบบในชั้นข้อมูล X เป็นไปตามอสมการ $x_1 < x_2$ ที่เทียบเท่ากับ $x_1 - x_2 < 0$ ยกตัวอย่างเช่น แบบที่ 1 มีค่า $x_1 = 0.5$ และ $x_2 = 3.0$ ดังนั้น $0.5 - 3.0 = -2.5 < 0$
- แบบในชั้นข้อมูล O เป็นไปตามอสมการ $x_1 > x_2$ หรือ $x_1 - x_2 > 0$ เช่น แบบที่ 9 เป็นไปตามอสมการ $x_1 - x_2 = 5.5 - 0.5 > 0$

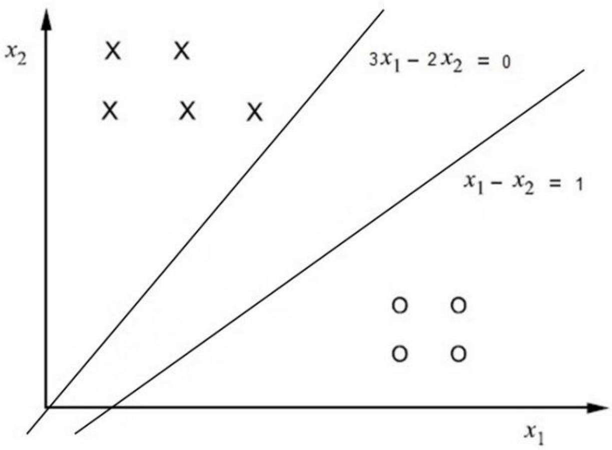
การแบ่งแยกแบบจากชั้นข้อมูล X และ O ด้วยอสมการ $x_1 < x_2$ ข้างต้น อยู่ในลักษณะสมมาตรกัน จากการกระจายตัวของแบบและตำแหน่งของเส้นตรงอาจกล่าวได้ว่า เส้นตรงนี้ได้แบ่งแยกแบบออกจากกันเป็น 2 ฝั่ง ได้แก่

- แบบจากชั้นข้อมูล X อยู่ทาง ฝั่งลบ (Negative Side) ของเส้นตรง เพราะว่า $x_1 - x_2 < 0$
- แบบจากชั้นข้อมูล O อยู่ทาง ฝั่งบวก (Positive Side) ของเส้นตรง เพราะว่า $x_1 - x_2 > 0$

ยังมีเส้นตรงอื่นอีกที่ใช้ในการสร้างขอบเขตการตัดสินใจ เพื่อแบ่งแยกแบบออกจากกันได้ เช่น เส้นตรง $x_1 - x_2 = 1$ โดยที่

- แบบจะอยู่ในชั้นข้อมูล X ถ้า $x_1 - x_2 < 1$
- แบบจะอยู่ในชั้นข้อมูล O ถ้า $x_1 - x_2 > 1$

สังเกตได้ว่า เส้นตรง $x_1 - x_2 = 0$ ขนานกับ $x_1 - x_2 = 1$ ข้อสังเกตที่ได้คือ $x_1 - x_2 = 0$ ผ่านจุดกำเนิด แต่ $x_1 - x_2 = 1$ เลื่อนจากจุดกำเนิดไปทางขวา ส่งผลให้ ชั้นข้อมูล X มีเนื้อที่ของปริภูมิที่มากกว่าของชั้นข้อมูล O นอกจากนี้ ยังมีเส้นตรงอื่นอีกที่แบ่งแยกแบบออกจากกัน ได้แก่ เส้นตรง $3x_1 - 2x_2 = 0$ ที่ไม่ขนานกับเส้นตรง 2 เส้น ข้างต้น



รูปที่ 6.2 ฟังก์ชันการจำแนกเชิงเส้น $x_1 - x_2 = 1$ และ $3x_1 - 2x_2 = 0$

เส้นตรงทุกเส้นที่กล่าวไปข้างต้น สามารถเขียนให้อยู่ในรูปแบบฟังก์ชันด้านล่างนี้ ได้เสมอ

$$f(x) = w_1x_1 + w_2x_2 + b = 0$$

- $f(x)$ คือ ฟังก์ชันการจำแนกเชิงเส้น ในรูปของเส้นตรง
- x_1, x_2 คือ ตัวแปรของฟีเจอร์ 1 และ 2 ตามลำดับ

- w_1, w_2 คือ สัมประสิทธิ์ของตัวแปร x_1 และ x_2 ตามลำดับ
- b คือ ค่าคงที่ใด ๆ

ตัวอย่างการระบุค่าตัวแปรจากฟังก์ชันการจำแนกเชิงเส้นข้างต้น มีดังต่อไปนี้

- เส้นตรง $x_1 - x_2 = 0$ มีค่าตัวแปร $w_1 = 1$, $w_2 = -1$ และ $b = 0$
- เส้นตรง $x_1 - x_2 = 1$ มีค่าตัวแปร $w_1 = 1$, $w_2 = -1$ และ $= -1$

สำหรับปริภูมิที่สูงกว่า 2 มิติ ไม่สามารถใช้เส้นตรงในการแบ่งแยกแบบได้ ดังนั้น ขอบเขตการตัดสินใจจึงมีลักษณะเป็น ระนาบหลายมิติ

$$f(x) = w^t x + b = 0$$

- f(x) คือ ฟังก์ชันการจำแนกเชิงเส้น ในรูปของระนาบหลายมิติ
- x คือ เวกเตอร์ d มิติ ของตัวแปรของแต่ละแต่ละฟีเจอร์
- w^t คือ เวกเตอร์น้ำหนัก d มิติ ของสัมประสิทธิ์ของตัวแปรใน x
- b คือ สเกลาร์ (Scalar)
- d คือ จำนวนมิติของแบบ

ฟังก์ชันการจำแนกเชิงเส้นข้างต้น สามารถแบ่งออกได้เป็น 2 รูปแบบ ดังนี้

- เอกพันธุ์ (Homogeneous) ถ้า $b = 0$
- ไม่เอกพันธุ์ (Non-homogeneous) ถ้า $b \neq 0$

กล่าวได้ว่า ชั้นข้อมูล X และ O สามารถจำแนกเชิงเส้นได้ ถ้าสามารถหาค่า w และ b ที่เป็นไปตามเงื่อนไขด้านล่างนี้

$w^t x + b > 0$ สำหรับ แบบ x ที่อยู่ในชั้นข้อมูลแรก กำหนดให้เป็น O

$w^t x + b < 0$ สำหรับ แบบ x ที่อยู่ในชั้นข้อมูลหลัง กำหนดให้เป็น X

ตัวอย่างของการจำแนกเชิงเส้นแสดงตามตัวอย่างต่อไปนี้

ตัวอย่างที่ 2 พิจารณาแบบทั้ง 9 ในรูปที่ 6.1 และกำหนดค่า $w^t = (1, -1)$ และ $b = -1$ จากนั้นคำนวณค่า $w^t x + b$ ของแต่ละแบบ เก็บค่าลงในตารางที่ 6.2 พบว่า

- แบบจากชั้นข้อมูล X ได้แก่ 5 แบบแรก เป็นไปตามเงื่อนไข $w^tx + b < 0$
- แบบจากชั้นข้อมูล O ได้แก่ 4 แบบหลัง เป็นไปตามเงื่อนไข $w^tx + b > 0$

ดังนั้น ชั้นข้อมูล X และ O จำแนกเชิงเส้นได้ ค่า b ในตัวอย่างนี้ คือ จุดตัดแกน x_2 ซึ่งมีผลต่อตำแหน่งที่ขอบเขตการตัดสินใจพาดผ่าน พิจารณาค่า b ในลักษณะต่าง ๆ ดังนี้

1. พิจารณาค่า $b = 0$ จุดกำเนิดจะอยู่บนขอบเขตการตัดสินใจ เพราะว่า $w^tx + b = 0$ สำหรับ $x = 0$ และ $b = 0$ สำหรับทุกค่าของ w
2. พิจารณาค่า $b > 0$ จุดกำเนิดจะอยู่ทางฝั่งบวก เพราะว่า $w^tx + b > 0$ สำหรับ $x = 0$ และ $b > 0$ สำหรับทุกค่าของ w โดยที่ จุดตัดแกน x_2 อยู่บนจุดกำเนิด
3. พิจารณาค่า $b < 0$ จุดกำเนิดจะอยู่ทางฝั่งลบ เพราะว่า $w^tx + b < 0$ สำหรับ $x = 0$ และ $b < 0$ สำหรับทุกค่าของ w โดยที่ จุดตัดแกน x_2 อยู่ล่างจุดกำเนิด

การสังเกตข้างต้นสามารถใช้กับแบบใดก็ได้ ที่อยู่ในปริภูมิ d มิติ ($d \geq 1$) อย่างไรก็ตาม แบบในตัวอย่างที่ 1 อยู่ในปริภูมิ 2 มิติ ($d = 2$)

ตารางที่ 6.2 ฟังก์ชันการจำแนกเชิงเส้น

Pattern No.	x_1	x_2	Class	$w^tx + b$
1	0.5	3.0	X	–3.5
2	1.0	3.0	X	–3.0
3	0.5	2.5	X	–3.0
4	1.0	2.5	X	–2.5
5	1.5	2.5	X	–2.0
6	4.5	1.0	O	2.5
7	5.0	1.0	O	3.0
8	4.5	0.5	O	3.0
9	5.5	0.5	O	4.0

ตัวอย่างที่ 3 จากแบบใน ตารางที่ 6.1 และ รูปที่ 6.1 พิจารณาเส้นตรง $x_1 = x_2$ ที่สามารถแบ่งแยกแบบจากชั้นข้อมูล X และ O ออกจากกันได้หมด เส้นตรงเส้นนี้คือขอบเขตการตัดสินใจที่สามารถเขียนให้อยู่ในรูปของ $x_1 - x_2 = 0$ ซึ่งมีค่าตัวแปร $w^t = (1, -1)$ และ $b = 0$ นอกจากนี้ จุดกำเนิด (0, 0) ก็อยู่บนเส้นตรงนี้

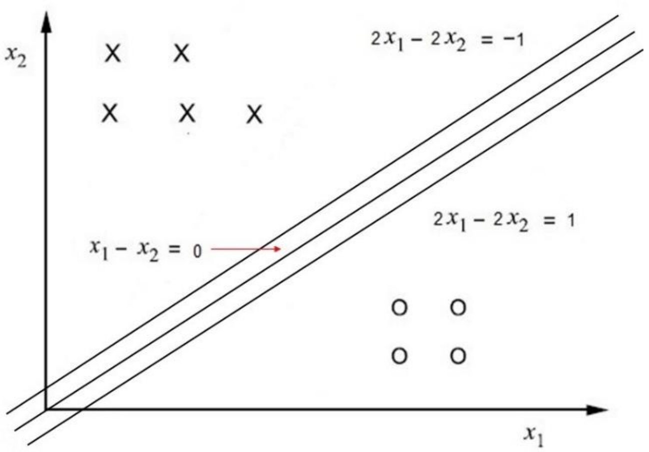
พิจารณาเส้นตรง $2x_1 - 2x_2 = -1$ ที่สามารถเขียนให้อยู่ในรูป $2x_1 - 2x_2 + 1 = 0$ เส้นตรงเส้นนี้มาจาก การเลื่อน (Shift) เส้นตรง $x_1 - x_2 = 0$ ขึ้นทางด้านบนตามแนวแกน x_2 เส้นตรงที่กำลังพิจารณานี้สามารถแบ่งแยกแบบได้ ดังต่อไปนี้

- ชั้นข้อมูล X อยู่ทาง ฝั่งลบ (Negative Side) ของเส้นตรง เพราะว่า $2x_1 - 2x_2 + 1 < 0$
- ชั้นข้อมูล O อยู่ทาง ฝั่งบวก (Positive Side) ของเส้นตรง เพราะว่า $2x_1 - 2x_2 + 1 > 0$ และ จุดกำเนิดอยู่ทางฝั่งนี้

เส้นตรงเส้นต่อไปที่พิจารณาคือ $2x_1 - 2x_2 = 1$ หรือ $2x_1 - 2x_2 - 1 = 0$ ซึ่งมาจากการเลื่อน $x_1 - x_2 = 0$ ลงทางด้านล่างตามแนวแกน x_2 ส่งผลให้จุดกำเนิดอยู่ทางฝั่งลบ ด้วยเหตุที่ว่าชั้นข้อมูลทั้งสองถูกแบ่งแยกออกจากกัน เมื่อนำเส้นตรงที่ถูกพิจารณานี้ไปจำแนกแบบทั้ง 9 แบบ ผลการทำนายเป็นที่ถูกต้องทั้งหมด

ค่าของ b เป็นตัวกำหนดระยะทางการเลื่อนของขอบเขตการตัดสินใจในกรณีที่เป็นเส้นตรงจากตำแหน่งของจุดกำเนิด ในทำนองเดียวกัน ค่า w เป็นตัวกำหนดทิศทางหรือตำแหน่งของขอบเขตการตัดสินใจ ซึ่งจะได้กล่าวถึงในตัวอย่างต่อไป

ถึงแม้ว่าตัวอย่างส่วนใหญ่ในตำราเล่มนี้ พิจารณาขอบเขตการตัดสินใจในรูปของงสันตรงบนปริภูมิ 2 มิติ เพื่ออธิบายแนวคิดพื้นฐาน อย่างไรก็ตาม ข้อสรุปสามารถประยุกต์ได้กับมิติที่สูงขึ้นเพียงแต่ขอบเขตการตัดสินใจจะเป็นระนาบหลายมิติแทนแทน



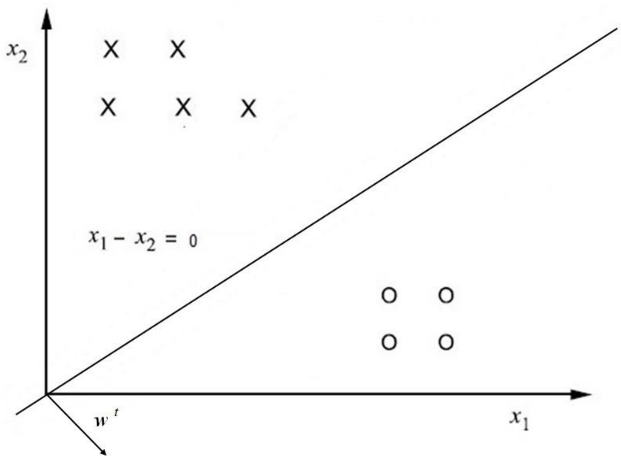
รูปที่ 6.3 ฟังก์ชันการจำแนกเชิงเส้น $2x_1 - 2x_2 = -1$, $2x_1 - 2x_2 = 1$ และ $x_1 - x_2 = 0$

ตัวอย่างที่ 4 จากแบบในรูปที่ 6.1 พิจารณาเส้นตรง $x_1 = x_2$ ซึ่งเหมือนกับเส้นตรง $x_1 - x_2 = 0$ ขอบเขตการตัดสินใจนี้ มีค่าตัวแปร $w^t = (1, -1)$ ซึ่งมีค่าตัวแปรย่อย ได้แก่ $w_1 = 1$ และ $w_2 = -1$ จากรูปที่ 6.3 แสดงให้เห็นว่า เวกเตอร์น้ำหนัก w ตั้งฉากกับ ขอบเขตการตัดสินใจ $x_1 - x_2 = 0$ พิจารณาจุด (α, α) ใด ๆ บนเส้นตรง $x_1 = x_2$

$$
\begin{aligned}
w^t x &= \begin{pmatrix} 1 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} \alpha \\ \alpha \end{pmatrix} \\
&= \alpha - \alpha \\
&= 0
\end{aligned}
$$

ต่อไป พิจารณาการเลื่อนเส้นตรง $x_1 - x_2 = 0$ ขึ้นและลง ได้แก่ $2x_1 - 2x_2 = -1$ และ $2x_1 - 2x_2 = 1$ ตามลำดับ การพิจารณาเส้นตรงที่ขนานกับ $x_1 - x_2 = 0$ เวกเตอร์น้ำหนัก w มีค่าเท่าเดิม คือ (1, –1) มีแต่ตัวแปร b เท่านั้นที่มีการเปลี่ยนค่า รายละเอียดทั้งหมดแสดงตาม ตารางที่ 6.3 โดยหลักที่สองเป็นการปรับค่าขอบเขตการตัดสินใจทั้งหมดให้เป็นเส้นตรงที่มีค่าสัมประสิทธิ์เท่ากัน โดยมีจุดประสงค์เพื่อการเปรียบเทียบค่า w และ b จึงได้ข้อสรุปว่า w ตั้งฉากกับขอบเขตการตัดสินใจทั้งสามที่ขนานกันเหล่านี้

เนื่องจาก w ตั้งฉากกับเส้นตรงทั้งสามเส้นที่ขนานกัน ดังนั้นมุม θ ระหว่างแบบทางฝั่งบวก (แบบจากชั้นข้อมูล O) และ w มีค่าอยู่ในช่วง $-90 < \theta < 90$ โดยส่งผลให้ค่า $\cos \theta > 0$ รายละเอียดเพิ่มเติม จะกล่าวถึงในตัวอย่างถัดไป



รูปที่ 6.3 ฟังก์ชันการจำแนกเชิงเส้น $x_1 - x_2 = 0$ และ เวกเตอร์ w^t

ตารางที่ 6.3 เวกเตอร์น้ำหนักที่ตั้งฉากกับขอบเขตการตัดสินใจ

ขอบเขตการตัดสินใจ	เส้นตรง	w	b
$x_1 - x_2 = 0$	$x_1 - x_2 + 0 = 0$	(1, –1)	0
$2x_1 - 2x_2 + 1 = 0$	$x_1 - x_2 + ½ = 0$	(1, –1)	½
$2x_1 - 2x_2 - 1 = 0$	$x_1 - x_2 - ½ = 0$	(1, –1)	–½

ตัวอย่างที่ 5 จากแบบในรูปที่ 6.1 พิจารณาเส้นตรง $x_1 = x_2$ ในรูปของขอบเขตการตัดสินใจ ที่มีค่าเวกเตอร์ $w^t = (1, -1)$ จากนั้นพิจารณาแบบทั้งหมดในชั้นข้อมูล O ยกตัวอย่าง แบบที่ 7 (5, 1) ในตารางที่ 6.1 ค้นพบว่าค่า โคไซน์ (Cosine) ระหว่าง w และ x มีค่าตามสมการด้านล่าง เมื่อ กำหนดสัญลักษณ์ $\|w^t\|$ และ $\|x\|$ แทนขนาดของเวกเตอร์ w^t และ x ตามลำดับ

$$\cos\theta = \frac{w^t x}{\|w^t\|\|x\|}$$

$$= \frac{\binom{7}{-1}\cdot\binom{5}{1}}{\sqrt{1^2+(-1)^2}\cdot\sqrt{5^2+1^2}}$$

$$= \frac{4}{\sqrt{2}\sqrt{26}}$$

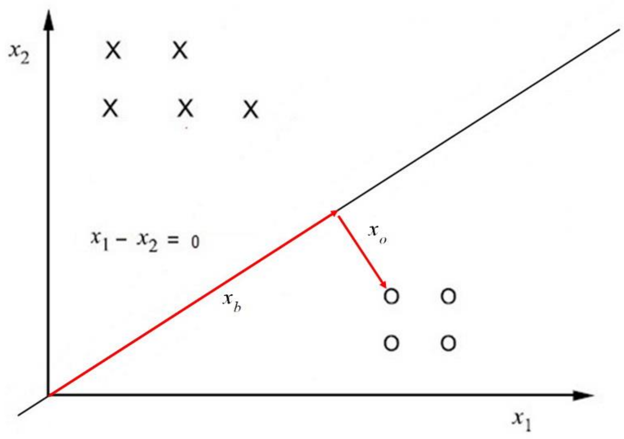
$$> 0$$

เนื่องจาก w^t ตั้งฉากกับขอบเขตการตัดสินใจ และ $\cos\theta$ มีค่าเป็นบวก ดังนั้น แบบทั้งหมดจากชั้นข้อมูล O จึงอยู่ทางฝั่งบวก เปรียบเทียบได้กับค่าโคไซน์ของมุมที่อยู่ใน คอวดรันต์ (Quadrant) ที่ 1 หรือ 4 ซึ่งอยู่ด้านตรงข้ามกับแบบของชั้นข้อมูล X ที่อยู่ในคอวดรันต์ที่ 2 หรือ 3 เมื่อขอบเขตการตัดสินใจเปรียบเสมือนแกน Y เป็นต้น

สำหรับแบบ x ใด ๆ สามารถเขียนให้อยู่ในรูปผลรวมของเวกเตอร์ระหว่าง เวกเตอร์ตามแนวขอบเขตการตัดสินใจ และ เวกเตอร์ที่ตั้งฉากกับเวกเตอร์ก่อนหน้า ตามสมการด้านล่างนี้ ตัวอย่างของเวกเตอร์ทั้งสอง แสดงตามรูปที่ 6.4

$$x = x_b + x_o$$

- x_b คือ เงา (Projection) ตามแนวของเวกเตอร์ x
- x_o คือ ส่วนประกอบเชิงตั้งฉากของเวกเตอร์ x



รูปที่ 6.4 เงาและส่วนประกอบเชิงตั้งฉากของแบบ

เนื่องจาก w^t ตั้งฉากกับขอบเขตการตัดสินใจและมีทิศทางพุ่งไปยังด้านบวก ดังนั้น สามารถเขียน x_o ให้อยู่ในรูปด้านล่างนี้ได้

$$x_o = p\frac{w^t}{\|w^t\|}$$

- p คือ จำนวนจริง ที่มีเงื่อนไขดังนี้
  - $p > 0$ ถ้า x มาจากชั้นข้อมูล O
  - $p < 0$ ถ้า x มาจากชั้นข้อมูล X
- $\frac{w^t}{\|w^t\|}$ คือ เวกเตอร์หนึ่งหน่วย (Unit Vector) ของเวกเตอร์ w

จากข้อสังเกตทั้งหมดข้างต้น สามารถเขียนสมการของ ขอบเขตการตัดสินใจ $f(x)$ ได้ดังนี้

$$f(x) = w^t x + b$$

เนื่องจาก

1. เวกเตอร์ w^t ตั้งฉากกับเวกเตอร์ x ดังนั้น $w^t x_b = 0$
2. เส้นตรง $x_1 = x_2$ ค่า b = 0

ดังนั้น

$$f(x) = w^t x_b + b + w^t x_o$$
$$= 0 + w^t x_o$$

เนื่องจาก $x_o = p\frac{w^t}{\|w^t\|}$ ดังนั้น

$f(x)$ $= pw^t \frac{w^t}{\|w^t\|}$

$= p\|w^t\|$

แก้สมการหาค่า p ได้ดังนี้

$$p = \frac{w^t x + b}{\|w^t\|}$$

- ความหมายของค่า p คือ ระยะทางที่สั้นที่สุด (Shortest Distance) ระหว่าง แบบ x และ ขอบเขตการตัดสินใจ f(x)
- $\|w^t\|$ คือ ขนาดของเวกเตอร์ w^t

ตัวอย่างนี้มีจุดประสงค์เพื่อให้ได้เห็นขั้นตอนการคำนวณระยะทางระหว่างเวกเตอร์และเส้นตรง นอกจากนี้ ระยะทาง p มีชื่อเรียกอีกอย่างหนึ่งว่า ระยะทางปกติ (Normal Distance)

ตัวอย่างที่ 6 พิจารณาแบบในรูปที่ 6.1 และ ขอบเขตการตัดสินใจในรูปของเส้นตรง $x_1 - x_2 - 1 = 0$ จงคำนวณหาระยะทางระหว่างเส้นตรงจากโจทย์ไปยังจุดต่อไปนี้

ตัวอย่างที่ 6.1 จุดกำเนิด

จากเส้นตรง ได้ค่าตัวแปรต่าง ๆ ดังต่อไปนี้

$\|w^t\| = \sqrt{2}$

b = –1

ดังนั้น

$$p = \frac{w^t x + b}{\|w^t\|}$$

$$= \frac{\begin{pmatrix} 1 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \end{pmatrix} - 1}{\sqrt{2}}$$

$$= \frac{-1}{\sqrt{2}}$$

ตัวอย่างที่ 6.2 จุด (1, 1)

ใช้ค่าตัวแปร $\|w^t\|$ และ b จากตัวอย่างที่ 6.1 แสดงการคำนวณได้ดังนี้

$$p = \frac{w^t x + b}{\|w^t\|}$$

$$= \frac{\begin{pmatrix} 1 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} - 1}{\sqrt{2}}$$

$$= \frac{-1}{\sqrt{2}}$$

จาก 2 ตัวอย่างย่อยข้างต้น แสดงให้เห็นว่าระยะทางจากจุด (α, α) เมื่อ α จำนวนจริงใด ๆ ไปยังขอบเขตการตัดสินใจเท่ากับ $\frac{-1}{\sqrt{2}}$ ในส่วนเพิ่มเติม ความหมายของระยะทางที่มีค่าน้อยกว่าศูนย์คือแบบอยู่ทางฝั่งลบของขอบเขตการตัดสินใจ

2. การทำให้อยู่ในรูปปกติ

ขอบเขตการตัดสินใจในปริภูมิ d มิติ เมื่อ $d \geq 1$ สามารถแปลงไปเป็นขอบเขตการตัดสินใจในปริภูมิ d + 1 มิติ โดยการเพิ่มมิติของปริภูมิของเวกเตอร์เดิม ขั้นตอนดังกล่าวแสดงให้เห็นในตัวอย่างต่อไปนี้

ตัวอย่างที่ 7 กำหนดขอบเขตการตัดสินใจ f(x) ในปริภูมิ d มิติ ตามรูปแบบด้านล่างนี้

$$f(x) = w^t x + b = 0$$

ขอบเขตการตัดสินใจในปริภูมิ d มิตินี้ สามารถแปลงเป็นขอบเขตการตัดสินใจที่เพิ่มปริภูมิอีก 1 มิติได้ โดยการเพิ่มค่าไปยังมิติที่ d + 1 ของแบบและเส้นตรงเดิม ดังนี้

$$x_{d+1} = 1$$
$$w^t_{d+1} = b$$

กำหนดฟังก์ชันของขอบเขตการตัดสินใจใหม่ f(x′) ในปริภูมิ d+1 มิติ ตามรูปแบบด้านล่างนี้

$$f(x') = z^t x' = 0$$

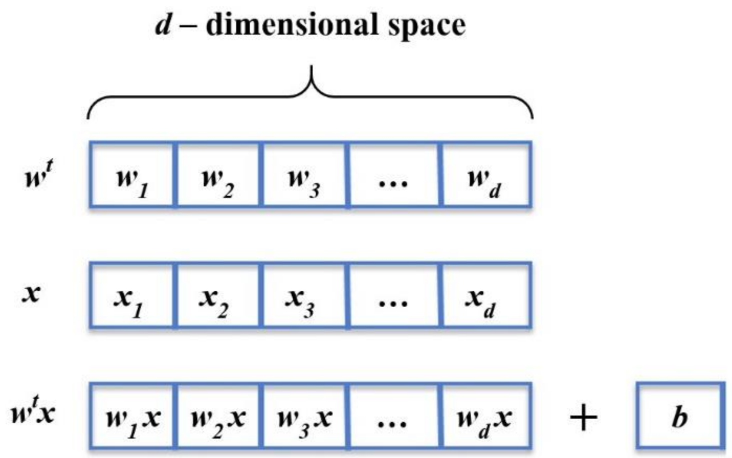
เมื่อ z^t และ x' คือ เวกเตอร์ขนาด d+1 มิติ การแปลงทำได้โดยการเพิ่มค่า b ไปยัง มิติที่ d+1 ของ w^t และ เพิ่มค่า 1 ไปยัง มิติที่ d+1 ของ x' ตามรูปแบบด้านล่างนี้

$$z^t = \begin{pmatrix} w^t \\ b \end{pmatrix}$$
$$x' = \begin{pmatrix} x \\ 1 \end{pmatrix}$$

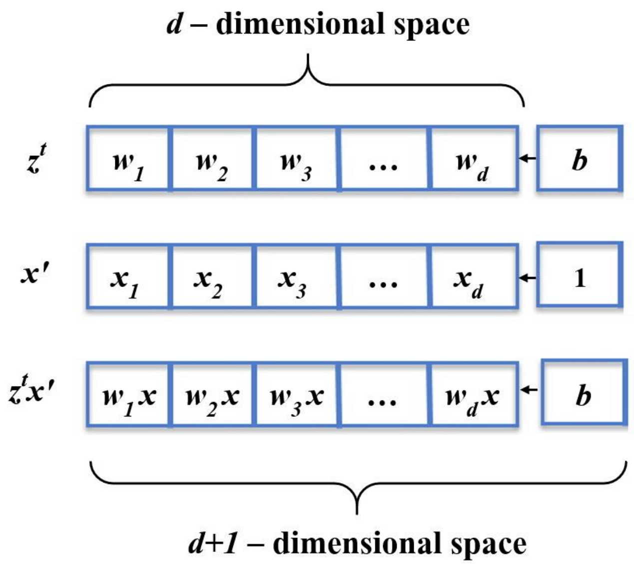
ขั้นตอนทั้งหมดที่กล่าวมานี้ เป็นการแปลงขอบเขตการตัดสินใจไม่เอกพันธ์ f(x) ในปริภูมิ d มิติ เป็นขอบเขตการตัดสินใจเอกพันธ์ f(x′) ในปริภูมิ d + 1 มิติ

เพื่อให้เห็นภาพในการอธิบายกระบวนการแปลงดังกล่าว รูปที่ 6.5 และ 6.6 แสดงขอบเขตการตัดสินใจก่อนและหลังการแปลง โดยที่ ค่า b ในแต่ละรูปมีความแตกต่างกัน ดังนี้

- b ในรูปที่ 6.5 คือ สเกลาร์ ส่งผลให้ $w^t x + b$ เป็นขอบเขตการตัดสินใจไม่เอกพันธ์
- b ในรูปที่ 6.6 คือ สมาชิกเวกเตอร์ในมิติสุดท้าย ส่งผลให้ $z^t x'$ เป็นขอบเขตการตัดสินใจเอกพันธ์



รูปที่ 6.5 $w^t x$ ไม่เอกพันธ์ในปริภูมิ d มิติ



รูปที่ 6.6 $z^t x'$ เอกพันธ์ในปริภูมิ d + 1 มิติ

ตัวอย่างที่ 8 จากเส้นตรง $x_1 - x_2 = 1$ จงเขียนให้อยู่ในรูปแบบ $z^t x'$

เส้นตรง $x_1 - x_2 = 1$ สามารถเขียนให้อยู่ในรูป $x_1 - x_2 - 1 = 0$ กำหนดค่าตัวแปร ดังนี้

$$z^t = \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}$$

$$x' = \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

แทนค่าตัวแปร เพื่อตรวจคำตอบ ได้ดังนี้

$$z^t x' = \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}$$

$$= x_1 - x_2 - 1$$

การแปลง (Transformation) คือ การเพิ่มมิติของเซตข้อมูล มีลำดับการทำงาน ดังต่อไปนี้

1. เพิ่มฟีเจอร์ของทุกแบบอีก 1 มิติ
2. เพิ่มค่า 1 ที่มิติที่เพิ่มเข้าไป

การทำให้อยู่ในรูปปกติ (Normalization) คือ การทำให้แบบทั้งหมดอยู่ฝั่งเดียวกัน มีขั้นตอนดังนี้

1. ใช้วิธีการแปลง
2. ฟีเจอร์ของแบบทางฝั่งบวกให้คงค่าเดิมไว้
3. ฟีเจอร์ของแบบทางฝั่งลบให้เปลี่ยนค่าเป็นเครื่องหมายตรงกันข้าม

ตัวอย่างที่ 9 พิจารณาแบบในตารางที่ 6.1 แบบที่อยู่ทางฝั่งลบได้แก่แบบจากชั้นข้อมูล X (X_1 ถึง X_5) ยกตัวอย่างการทำให้อยู่ในรูปปกติของแบบ X_1 ได้แก่ (0.5, 3.0) ได้ตามขั้นตอน ดังต่อไปนี้

1. เพิ่มฟีเจอร์ของแบบ (0.5, 3.0) ในมิติที่ 3 ด้วยค่า 1 ได้ผลลัพธ์ คือ (0.5, 3.0, 1.0)

2. เปลี่ยนค่าฟีเจอร์ของ (0.5, 3.0, 1.0) เป็นค่าตรงกันข้าม คือ (–0.5, –3.0, –1.0)

ตารางที่ 6.3 แบบ 3 มิติ ที่ทำให้อยู่ในรูปปกติ

Pattern	1	2	3
X_1	–0.5	–3.0	–1
X_2	–1.0	–3.0	–1
X_3	–0.5	–2.5	–1
X_4	–1.0	–2.5	–1
X_5	–1.5	–2.5	–1
X_6	4.5	1.0	1
X_7	5.0	1.0	1
X_8	4.5	0.5	1
X_9	5.5	0.5	1

สำหรับแบบที่เหลือในชั้นข้อมูล X ได้แก่ X_2 ถึง X_5 ก็ทำได้ในทำนองเดียวกัน ผลลัพธ์ทั้งหมดแสดงตามตารางด้านล่างนี้ สังเกตได้ว่าแบบทางฝั่งบวก X_6 ถึง X_9 ซึ่งอยู่ในชั้นข้อมูล O มีการเพิ่มมิติเพียงอย่างเดียวเท่านั้น โดยไม่มีการกลับค่าเป็นตรงกันข้าม

การทำให้แบบอยู่ในรูปปกติเกิดขึ้นก่อนการหาขอบเขตการตัดสินใจ ขั้นตอนนี้มีความจำเป็นเนื่องจากเป็นรูปแบบของข้อมูลเข้าที่ส่งให้การเรียนรู้เพอร์เซ็ปตรอนเรียนรู้ ดังที่จะกล่าวถึงในหัวข้อถัดไป ในตำราฉบับนี้ จากนี้ไปจนจบบท สัญลักษณ์ w^t และ x จะถูกใช้แทนที่สัญลักษณ์ z^t และ x' ตามลำดับ

3. การเรียนรู้เพอร์เซ็ปตรอน

การเรียนรู้เพอร์เซ็ปตรอน (Perceptron Learning) ถูกใช้ในการเรียนรู้เวกเตอร์ w^t สำหรับแบ่งแยกชั้นข้อมูลทั้งสองออกจากกันในกรณีที่ชั้นข้อมูลมีการจำแนกเชิงเส้นได้ เพื่อจุดประสงค์สำหรับการจำแนกแบบ กำหนดข้อมูลเข้าและขั้นตอนวิธีของการเรียนรู้เพอร์เซ็ปตรอน ดังนี้

ข้อมูลเข้า: แบบจำนวน n แบบ $x_1, x_2, x_3, \ldots, x_n$ ใน d+1 มิติ ที่ผ่านการทำให้อยู่ในรูปปกติแล้ว

ขั้นตอนวิธีการเรียนรู้เพอร์เซ็ปตรอน

1. กำหนดค่าเริ่มต้น w เป็น เวกเตอร์ว่าง (Null Vector) โดยการให้ค่า i = 1 และ $w_i = 0$
2. **FOR** j = 1 **TO** n **DO**

   IF $w^t_i x_j \le 0$ **THEN** i = i +1 **AND** $w^t_i = w^t_{i-1} + x_j$
3. ทำซ้ำขั้นตอนที่ 2 จนกระทั่ง i ไม่เกิดการเปลี่ยนค่าจนครบทั้ง n แบบ

วิธีการคำนวณของการเรียนรู้เพอร์เซ็ปตรอน เพื่อปรับน้ำหนักของขอบเขตการตัดสินใจ แสดงตามตัวอย่างต่าง ๆ ดังต่อไปนี้

ตัวอย่างที่ 5 จงแสดงลำดับขั้นตอนของวิธีการเรียนรู้เพอร์เซ็ปตรอน จากแบบในตารางที่ 6.3

กำหนดค่าเริ่มต้น $w^t_1 = \begin{pmatrix}0\\0\\0\end{pmatrix}$

เริ่มต้นคำนวณจากแบบแรก x_1

$$w^t_1 x_1 = \begin{pmatrix}0\\0\\0\end{pmatrix} \cdot \begin{pmatrix}-0.5\\-3.0\\-1.0\end{pmatrix} = (0.0)\,(-0.5) + (0.0)\,(-3.0) + (0.0)\,(-1.0) = 0.0$$

เนื่องจากเงื่อนไขของคำสั่ง IF–THEN เป็นจริง ดังนั้นจึงมีการปรับปรุงค่าน้ำหนัก

$$w^t_2 = w^t_1 + x^t_1 = \begin{pmatrix}0\\0\\0\end{pmatrix} + \begin{pmatrix}-0.5\\-3.0\\-1.0\end{pmatrix} = \begin{pmatrix}0.0-0.5\\0.0-3.0\\0.0-1.0\end{pmatrix} = \begin{pmatrix}-0.5\\-3.0\\-1.0\end{pmatrix}$$

เมื่อค่าน้ำหนักเปลี่ยน ให้นำแบบถัดไปได้แก่ x_2 ต่อจากแบบเดิมที่เคยคำนวณไปแล้ว มาคำนวณต่อ

$$w^t_2x_2 = \begin{pmatrix}-0.5\\-3.0\\-1.0\end{pmatrix}\cdot\begin{pmatrix}-1.0\\-3.0\\-1.0\end{pmatrix} = 10.5$$

เนื่องจาก $w^t_2x_2 > 0$ จึงยังไม่มีการปรับปรุงค่าน้ำหนัก แล้วให้นำแบบถัดไปได้แก่ x_3 มาคำนวณต่อ

$$w^t_2x_3 = \begin{pmatrix}-0.5\\-3.0\\-1.0\end{pmatrix}\cdot\begin{pmatrix}-0.5\\-2.5\\-1.0\end{pmatrix} = 8.75 > 0$$

$$w^t_2x_4 = 9.00 > 0$$

$$w^t_2x_5 = 9.25 > 0$$

$$w^t_2x_6 = -6.25 < 0$$

ปรับปรุงค่าน้ำหนัก ดังนี้

$$w^t_3 = w^t_2 + x^t_6 = \begin{pmatrix}4.0\\-2.0\\0.0\end{pmatrix}$$

$$w^t_3x_7,\ w^t_3x_8,\ w^t_3x_9,\ w^t_3x_1,\ w^t_3x_2,\ w^t_3x_3,\ w^t_3x_4 > 0$$

$$w^t_3x_5 = -1 < 0$$

ปรับปรุงค่าน้ำหนัก ดังนี้

$$w^t_4 = w^t_3 + x^t_5 = \begin{pmatrix}2.5\\-4.5\\-1.0\end{pmatrix}$$

$$w^t_4x_6,\ w^t_4x_7,\ w^t_4x_8,\ w^t_4x_9,\ w^t_4x_1,\ w^t_4x_2,\ w^t_4x_3,\ w^t_4x_4,\ w^t_4x_5 > 0$$

เนื่องจากไม่มีการปรับปรุงค่าน้ำหนักจนครบทุกแบบ จึงได้เวกเตอร์ที่จำแนกแบบทั้งหมดได้อย่างถูกต้อง

$$w^t_4 = 2.5x_1 - 4.5x_2 - 1.0 = 0.0$$

สามารถเขียนให้อยู่ในรูปด้านล่างได้

$$w^t_4 = 5x_1 - 9x_2 - 2 = 0$$

จากการคำนวณข้างต้น สังเกตได้ว่าแบบมีลำดับการคำนวณแบบวนซ้ำ กล่าวคือการคำนวณเริ่มต้นตั้งแต่แบบแรกไปจนถึงแบบสุดท้ายแล้วกลับมาเริ่มต้นที่แบบแรกใหม่ ในกรณีที่น้ำหนักมีค่าเป็น

บวก หมายความว่าแบบถูกจำแนกได้อย่างถูกต้อง ในขณะที่ ถ้าค่าน้ำหนักติดลบแล้วแบบมีความผิดพลาดในการจำแนกแบบ นอกจากนี้ ค่าน้ำหนักที่ปรับปรุงใหม่มีคุณภาพมากกว่าค่าน้ำหนักเดิม หรืออีกนัยหนึ่ง จำนวนของแบบที่ถูกทำนายได้อย่างถูกต้องมีมากกว่า

ตัวอย่างที่ 6 จงใช้วิธีการเรียนรู้เพอร์เซ็ปตรอนจากแบบใน ตารางที่ 6.4 เพื่อจำแนกแบบทดสอบ Y = (1.0, 1.0)

ตารางที่ 6.4 แบบสำหรับการเรียนรู้เพอร์เซ็ปตรอน

Pattern No.	1	2	Class
1	0.5	1.0	X
2	1.0	2.0	X
3	0.5	1.5	X
4	1.5	1.5	X
5	1.5	2.0	X
6	4.5	1.5	O
7	5.0	1.5	O
8	4.5	2.5	O
9	5.5	1.5	O

จากแบบในตารางข้างต้น ให้เพิ่มมิติอีก 1 มิติ จากเดิม 2 มิติ เป็น 3 มิติ แล้วทำให้อยู่ในรูปปกติ โดยการกลับเครื่องหมายของแบบทางฝั่งลบ สุดท้าย ได้ผลลัพธ์ดังตารางด้านล่างนี้

ตารางที่ 6.5 แบบที่ทำให้อยู่ในรูปปกติ

Pattern No.	1	2	3	Class
1	–0.5	–1.0	–1	X
2	–1.0	–2.0	–1	X
3	–0.5	–1.5	–1	X
4	–1.5	–1.5	–1	X
5	–1.5	–2.0	–1	X
6	4.5	1.5	1	O
7	5.0	1.5	1	O
8	4.5	2.5	1	O
9	5.5	1.5	1	O

หาขอบเขตการตัดสินใจ ได้ตามขั้นตอน ต่อไปนี้

กำหนดค่าเริ่มต้น $w^t_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$

$w^t_1 x_1 = 0 \quad \leq 0$

ปรับปรุงค่าน้ำหนักของเวกเตอร์ ดังนี้

$w^t_2 = w^t_1 + x_1 = \begin{pmatrix} -0.5 \\ -1.0 \\ -1.0 \end{pmatrix}$

$w^t_2 x_2 = 3.50 \quad > 0$

$w^t_2 x_3 = 2.75 \quad > 0$

$w^t_2 x_4 = 3.25 \quad > 0$

$w^t_2 x_5 = 3.75 \quad > 0$

$w^t_2 x_6 = -4.75 \quad \leq 0$

ปรับปรุงค่าน้ำหนักของเวกเตอร์ ดังนี้

$w^t_3 = w^t_2 + x_6 = \begin{pmatrix} 4.0 \\ 0.5 \\ 0.0 \end{pmatrix}$

$w^t_3 x_7 = 20.75 \quad > 0$

$w^t_3 x_8 = 19.25 \quad > 0$

$w^t_3 x_9 = 22.75 \quad > 0$

$w^t_3 x_1 = -2.50 \quad \leq 0$

ปรับปรุงค่าน้ำหนักของเวกเตอร์ ดังนี้

$w^t_4 = w^t_3 + x_1 = \begin{pmatrix} 3.5 \\ -0.5 \\ -1.0 \end{pmatrix}$

$w^t_4 x_2 = -1.5 \quad \leq 0$

ปรับปรุงค่าน้ำหนักของเวกเตอร์ ดังนี้

$w^t_5 = w^t_4 + x_2 = \begin{pmatrix} 2.5 \\ -2.5 \\ -2.0 \end{pmatrix}$

$w^t_5x_3 = 4.50 \quad > 0$

$w^t_5x_4 = 2.00 \quad > 0$

$w^t_5x_5 = 3.25 \quad > 0$

$w^t_5x_6 = 5.50 \quad > 0$

$w^t_5x_7 = 6.75 \quad > 0$

$w^t_5x_8 = 3.00 \quad > 0$

$w^t_5x_9 = 8.00 \quad > 0$

$w^t_5x_1 = 3.25 \quad > 0$

$w^t_5x_2 = 4.50 \quad > 0$

เนื่องจากไม่มีการปรับปรุงค่าน้ำหนักจนครบทุกแบบ ดังนั้น ขอบเขตการตัดสินใจ คือ

$$2.5x_1 - 2.5x_2 - 2.0 = 0.0$$

หรือเขียนให้อยู่ในรูปแบบด้านล่างนี้ได้

$$5x_1 - 5x_2 - 4 = 0$$

ทดลองแทนค่าฟีเจอร์ลงในขอบเขตการตัดสินใจ ผลลัพธ์แสดงตามตารางด้านล่าง พบว่าแบบจากชั้นข้อมูล X อยู่ทางฝั่งลบ ในขณะที่ แบบจากชั้นข้อมูล O อยู่ทางฝั่งบวก ดังนั้น สรุปได้ว่าขอบเขตการตัดสินใจนี้จำแนกแบบทั้งหมดได้อย่างถูกต้อง

ตารางที่ 6.6 ชั้นข้อมูลจำแนกเชิงเส้น

Pattern No.	1	2	Class	$5x_1 - 5x_2 - 4$
1	0.5	1.0	X	–6.5
2	1.0	2.0	X	–9.0
3	0.5	1.5	X	–9.0
4	1.5	1.5	X	–4.0
5	1.5	2.0	X	–6.5
6	4.5	1.5	O	11.0
7	5.0	1.5	O	13.5
8	4.5	2.5	O	6.0
9	5.5	1.5	O	16.0

ทำนายแบบทดสอบ Y = (1, 1) โดยการแทนค่าลงไปในขอบเขตการตัดสินใจ ดังนี้

$$(5)(1) - (5)(1) - 4 = -4$$

เนื่องจากแบบทดสอบนี้อยู่ทางฝั่งลบ ดังนั้น Y อยู่ในชั้นข้อมูล X

ตัวอย่างที่ 7 OR Gate คือ ลอจิกเกตพื้นฐาน แสดงตามรูปที่ 6.7 กำหนดให้ชั้นข้อมูลคือผลลัพธ์ของตัวดำเนินการนี้ จงหาขอบเขตการตัดสินใจของแบบใน ตารางที่ 6.7 โดยใช้วิธีการเรียนรู้เพอร์เซ็ปตรอน

รูปที่ 6.7 OR Gate

ตารางที่ 6.7 เซตข้อมูล OR สำหรับการเรียนรู้เพอร์เซ็ปตรอน

Pattern No.	A	B	x
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

ทำแบบในตารางข้างต้นให้อยู่ในรูปปกติ ได้ดังนี้

ตารางที่ 6.8 เซตข้อมูล OR ที่ทำให้อยู่ในรูปปกติ

Pattern No.	A	B	x
1	0	0	–1
2	0	1	1
3	1	0	1
4	1	1	1

กำหนดค่าเริ่มต้น $w^t_1 = \begin{pmatrix}0\\0\\0\end{pmatrix}$

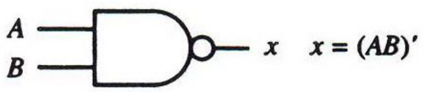
เนื่องจาก $w^t_1 x_1 = 0 \leq 0$ ดังนั้น $w^t_2 = \begin{pmatrix}0\\0\\-1\end{pmatrix}$

เนื่องจาก $w^t_1 x_2 = -1 \leq 0$ ดังนั้น $w^t_3 = \begin{pmatrix}0\\1\\0\end{pmatrix}$

(การคำนวณในส่วนที่ข้ามไป สามารถทำเพิ่มเติมได้จากแบบฝึกหัด ข้อ 8)

เมื่อคำนวณมาจนถึง $w^t_{10} = \begin{pmatrix}2\\2\\1\end{pmatrix}$ เวกเตอร์น้ำหนักนี้ จำแนกแบบทั้งหมดได้อย่างถูกต้อง

ตัวอย่างที่ 8 NAND Gate คือ ลอจิกเกตพื้นฐาน แสดงตามรูปที่ 6.8 กำหนดให้ชั้นข้อมูลคือผลลัพธ์ของตัวดำเนินการนี้ จงหาขอบเขตการตัดสินใจของแบบในตารางที่ 6.9 โดยใช้วิธีการเรียนรู้เพอร์เซ็ปตรอน



รูปที่ 6.8 NAND Gate

ตารางที่ 6.9 เซตข้อมูล NAND สำหรับการเรียนรู้เพอร์เซ็ปตรอน

Pattern No.	A	B	x
1	0	0	1
2	0	1	1
3	1	0	1
4	1	1	0

ทำแบบในตารางข้างต้นให้อยู่ในรูปปกติ ได้ดังนี้

ตารางที่ 6.10 เซตข้อมูล NAND ที่ทำให้อยู่ในรูปปกติ

Pattern No.	A	B	x
1	0	0	–1
2	0	–1	–1
3	–1	0	–1
4	1	1	0

กำหนดค่าเริ่มต้น $w^t_1 = \begin{pmatrix}0\\0\\0\end{pmatrix}$

$w^t_1x_1 = 0 \qquad \leq 0$

ปรับปรุงค่าน้ำหนักของเวกเตอร์ ดังนี้

$w^t_2 = w^t_1 + x_1 = \begin{pmatrix}0\\0\\-1\end{pmatrix}$

$w^t_2x_2$, $w^t_2x_3 = 1 > 0$ แต่ $w^t_2x_4 = -1 \leq 0$ ดังนั้น $w^t_3 = \begin{pmatrix}1\\1\\0\end{pmatrix}$

$w^t_3x_1 \leq 0$ ดังนั้น ปรับปรุงค่าน้ำหนัก $w^t_4 = \begin{pmatrix}1\\1\\-1\end{pmatrix}$

$w^t_4x_2 \leq 0$ ดังนั้น ปรับปรุงค่าน้ำหนัก $w^t_5 = \begin{pmatrix}1\\0\\-2\end{pmatrix}$

$w^t_5x_3 > 0$ แต่ $w^t_5x_4 \leq 0$ ดังนั้น ปรับปรุงค่าน้ำหนัก $w^t_6 = \begin{pmatrix}2\\1\\-1\end{pmatrix}$

$w^t_6x_1 > 0$ แต่ $w^t_6x_2 \leq 0$ ดังนั้น ปรับปรุงค่าน้ำหนัก $w^t_7 = \begin{pmatrix}2\\0\\-2\end{pmatrix}$

$w^t_7x_3 \leq 0$ ดังนั้น ปรับปรุงค่าน้ำหนัก $w^t_8 = \begin{pmatrix}-1\\0\\-3\end{pmatrix}$

$w^t_8x_4 \leq 0$ ดังนั้น ปรับปรุงค่าน้ำหนัก $w^t_9 = \begin{pmatrix}0\\1\\-2\end{pmatrix}$

$w^t_9x_1$, $w^t_9x_2$, $w^t_9x_3 > 0$ แต่ $w^t_9x_4 \leq 0$ ดังนั้น ปรับปรุงค่าน้ำหนัก $w^t_{10} = \begin{pmatrix}0\\2\\-1\end{pmatrix}$

$w^t_{10}x_1 > 0$ แต่ $w^t_9x_2 \leq 0$ ดังนั้น ปรับปรุงค่าน้ำหนัก $w^t_{11} = \begin{pmatrix}0\\-1\\-2\end{pmatrix}$

$w^t_{11}x_3 > 0$ แต่ $w^t_{11}x_4 \leq 0$ $w^t_{12} = \begin{pmatrix}1\\0\\-1\end{pmatrix}$

$w^t_{12}x_1$, $w^t_{12}x_2 > 0$ แต่ $w^t_{12}x_3 \leq 0$ ดังนั้น $w^t_{13} = \begin{pmatrix}0\\0\\-2\end{pmatrix}$

$w^t_{13}x_4 \leq 0$ ดังนั้น ปรับปรุงค่าน้ำหนัก $w^t_{14} = \begin{pmatrix}0\\0\\-1\end{pmatrix}$

$w^t_{14}x_1$, $w^t_{14}x_2$, $w^t_{14}x_3 > 0$ แต่ $w^t_{14}x_4 \leq 0$ ดังนั้น ปรับปรุงค่าน้ำหนัก $w^t_{15} = \begin{pmatrix}0\\0\\0\end{pmatrix}$

สังเกตได้ว่าค่าน้ำหนักปัจจุบันมีค่ากลับไปเท่ากับค่าน้ำหนักเริ่มต้น ($w^t_{15} = w^t_1$) ดังนั้น การปรับปรุงค่าน้ำหนักต่อไปจะเกิดการวนซ้ำค่าน้ำหนัก ส่งผลให้จำนวนรอบของขั้นตอนวิธีการเรียนรู้เพอร์เซ็ปตรอนเป็นอนันต์และโปรแกรมจะไม่หยุดการทำงาน การเกิดเหตุการณ์เช่นนี้ หมายความว่าชั้นข้อมูลไม่สามารถจำแนกเชิงเส้นได้

รูปแบบทั่วไปของการจำแนกเชิงเส้น

ฟังก์ชันการจำแนกเชิงเส้นสามารถขยายแนวคิด เพื่อแก้ปัญหาชั้นข้อมูลที่ไม่สามารถจำแนกเชิงเส้นได้ โดยใช้รูปแบบเอกพันธ์ ยกตัวอย่าง ดังนี้

กำหนดค่า

$$f(x) = 1 + x + x^2$$

สามารถเขียนให้อยู่ในรูปของ

$$f(x') = z^t x' = 0$$

เมื่อ

$$z^t = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

และ

$$x' = \begin{pmatrix} x^2 \\ x \\ 1 \end{pmatrix}$$

การนำฟังก์ชันนี้ไปใช้ แสดงให้เห็นในตัวอย่างถัดไป

ตัวอย่างที่ 9 พิจารณาตัวจำแนกทวิภาค ซึ่งจำแนกชั้นข้อมูล ดังนี้

- จำแนกชั้นข้อมูลเป็น O ถ้า $f(x) > 0$ เมื่อแบบอยู่ทางด้านบวก
- จำแนกชั้นข้อมูลเป็น X ถ้า $f(x) < 0$ เมื่อแบบอยู่ทางด้านลบ

กำหนดค่า

$$f(x) = a + bx + cx^2$$

จากการจำแนกแบบข้างต้น สามารถพิจารณาการจำแนกแบบได้อีกรูปแบบ ดังนี้

- จำแนกชั้นข้อมูลเป็น O ถ้า $z^t x' > 0$ เมื่อแบบอยู่ทางด้านบวก
- จำแนกชั้นข้อมูลเป็น X ถ้า $z^t x' < 0$ เมื่อแบบอยู่ทางด้านลบ

เนื่องจาก ฟังก์ชันสามารถเขียนให้อยู่ในรูปของ

$$f(x') = z^t x' = 0$$

สำหรับ

$$z^t = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

และ

$$x' = \begin{pmatrix} 1 \\ x \\ x^2 \end{pmatrix}$$

ตัวอย่างถัดไป แสดงการหาค่าเวกเตอร์น้ำหนัก สำหรับเซตข้อมูลที่ไม่สามารถจำแนกเชิงเส้นได้

ตัวอย่างที่ 10 พิจารณาแบบ1 มิติ ในตารางที่ 6.11 ซึ่งบรรจุชั้นข้อมูลที่ไม่สามารถจำแนกเชิงเส้นได้ ตัวอย่างนี้จะแสดงให้เห็นว่า ฟังก์ชัน f(x) = 0 เป็นขอบเขตการตัดสินใจที่สามารถแบ่งแยกชั้นข้อมูล O และ X ออกจากกันได้

ทำการแปลงแบบในตารางที่ 6.11 แล้วทำให้อยู่ในรูปปกติ จะได้แบบตาม ตารางที่ 6.12 โดยที่ความหมายของแต่ละหลักมีดังนี้

- หลักที่ 1 ได้แก่ 1 หรือ –1 แทน นิพจน์แรกของฟังก์ชัน f(x)
- หลักที่ 2 ได้แก่ x หรือ –x แทน นิพจน์กลางของฟังก์ชัน f(x)
- หลักที่ 3 ได้แก่ x^2 หรือ $-x^2$ แทน นิพจน์ท้ายของฟังก์ชัน f(x)

จากนั้น ใช้การเรียนรู้เพอร์เซ็ปตรอน เพื่อเรียนรู้น้ำหนักของเวกเตอร์ ตามขั้นตอนต่อไปนี้

ตารางที่ 6.11 แบบที่จำแนกเชิงเส้นไม่ได้

Pattern No.	x	Class
1	1	O
2	–1	O
3	2	O
4	–2	O
5	3	X
6	4	X
7	–3	X
8	–4	X

ตารางที่ 6.12 การทำให้อยู่ในรูปปกติของแบบที่จำแนกเชิงเส้นไม่ได้

Pattern No.	1	x	x^2	Bias
1	1	1	1	1
2	1	–1	1	1
3	1	2	4	1
4	1	–2	4	1
5	–1	3	–9	–1
6	–1	4	–16	–1
7	–1	–3	–9	–1
8	–1	–4	–16	–1

กำหนดค่าเริ่มต้น $w^t_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

เนื่องจาก w^t_1 จำแนก x_1 ผิดพลาด ดังนั้น ปรับปรุงค่าเวกเตอร์น้ำหนัก $w^t_2 = w^t_1 + x_1$

ทำตามขั้นตอนวิธีของการเรียนรู้เพอร์เซ็ปตรอน จนกระทั่ง w^t_{27} จำแนกแบบทั้ง 8 ถูกต้อง โดยที่

$$w^t_{27} = \begin{pmatrix} 12 \\ -1 \\ -4 \\ 12 \end{pmatrix}$$

ดังนั้น ขอบเขตการตัดสินใจ คือ

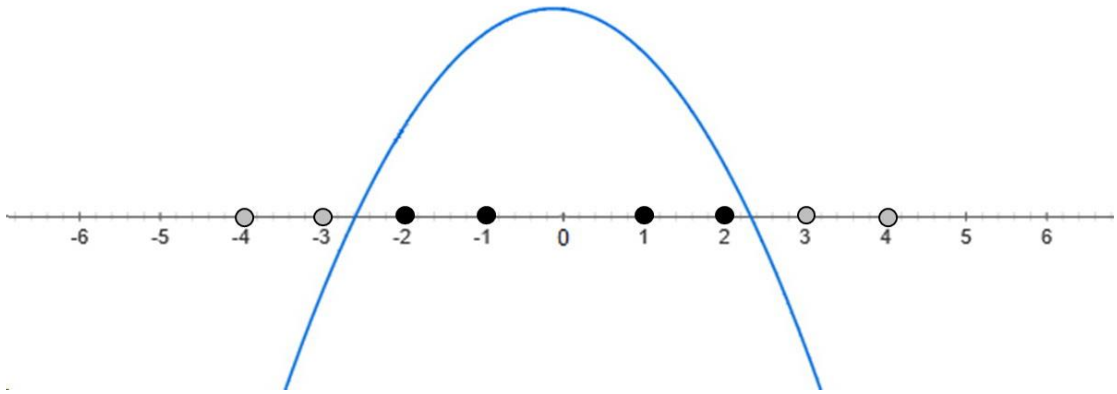
$$f(x) = 12 - x - 4x^2 + 12$$

หรือเขียนให้อยู่ในรูปแบบด้านล่างนี้ได้

$$f(x) = 24 - x - 4x^2$$

โดยที่รูปแบบนี้แสดงตามกราฟพาราโบลาในรูปที่ 6.9 โดยที่สามารถแบ่งแยกชั้นข้อมูล O (สีดำ) และ X (สีเทา) ได้อย่างถูกต้อง

ตัวอย่างนี้ได้แสดงให้เห็นถึงรูปแบบทั่วไปของการจำแนกเชิงเส้น ซึ่งสามารถอยู่ในรูปแบบของฟังก์ชันเชิงเส้นหรือฟังก์ชันไม่เชิงเส้นก็ได้ ในส่วนเพิ่มเติม เครือข่ายประสาทเป็นตัวแบบที่ได้รับความนิยมในการจำแนกแบบโดยใช้รูปแบบทั่วไปของการจำแนกเชิงเส้น รายละเอียดของตัวแบบชนิดนี้จะกล่าวถึงต่อไป ในหัวข้อที่ 5



รูปที่ 6.9 ขอบเขตการตัดสินใจไม่เชิงเส้น

4. ปัญหาหลายชั้นข้อมูล

ข้อจำกัดของฟังก์ชันจำแนกเชิงเส้นคือ ต้องใช้กับเซตข้อมูลที่มีเพียงสองชั้นข้อมูลและสามารถจำแนกเชิงเส้นได้ อย่างไรก็ตาม ในหลาย ๆ ปัญหา จำเป็นต้องยุ่งเกี่ยวกับเซตข้อมูลที่มีชั้นข้อมูลจำนวนหลากหลาย กำหนดให้เซตข้อมูลหนึ่งมี 3 ชั้นข้อมูล ได้แก่ X, O และ * ตามตารางที่ 6.13 การประยุกต์การจำแนกแบบของปัญหาสองชั้นมูลไปเป็นปัญหาหลายชั้นข้อมูลมีอยู่ 2 วิธีการ สำหรับนำเสนอในหัวข้อนี้ ซึ่งต่างก็เป็นการสร้างกลุ่มของตัวจำแนกแบบ การตัดสินใช้ เสียงส่วนใหญ่ (Majority Vote) ของการทำนายจากตัวแบบทั้งหมด

ตารางที่ 6.13 เซตข้อมูลหลายชั้นข้อมูล

Pattern No.	1	2	Class
1	0.5	3.0	X
2	1.0	3.0	X
3	0.5	2.5	X
4	6.0	6.0	O
5	6.0	6.5	O
6	7.0	6.0	O
7	10.0	0.5	*
8	10.0	1.0	*
9	11.0	1.0	*

I. หนึ่งต่อหนึ่ง (OAO: One Against One)

วิธีนี้พิจารณาทีละหนึ่งชั้นข้อมูลต่อหนึ่งชั้นข้อมูล หรือทีละคู่ของชั้นข้อมูล จำนวนครั้งของการพิจารณา เป็นไปตามสมการด้านล่างนี้

$$P = \frac{C(C-1)}{2}$$

- P คือ จำนวนคู่
- C คือ จำนวนชั้นข้อมูลในเซตข้อมูล

จำนวนคู่ของชั้นข้อมูลมีค่าเท่ากับจำนวนตัวจำแนกแบบที่ต้องเรียนรู้ เช่น เซตข้อมูลนี้มี 3 ชั้นข้อมูล แล้วต้องสร้างตัวจำแนกแบบจำนวน 3 ตัว เพื่อพิจารณา ดังต่อไปนี้

- ❖ w^t_{XO} คือ เวกเตอร์สำหรับแบ่งแยกแบบในคู่ของชั้นข้อมูล X และ ชั้นข้อมูล O
- ❖ w^t_{X*} คือ เวกเตอร์สำหรับแบ่งแยกแบบในคู่ของชั้นข้อมูล X และ ชั้นข้อมูล *
- ❖ w^t_{O*} คือ เวกเตอร์สำหรับแบ่งแยกแบบในคู่ของชั้นข้อมูล O และ ชั้นข้อมูล *

ตัวอย่างที่ 8 พิจารณาแบบในชั้นตารางที่ 6.14 จงแสดงวีธีการ OAO กำหนดให้หลัก Bias คือ มิติที่เพิ่มมาหลังจากการทำให้อยู่ในรูปปกติ

ตัวอย่างที่ 8.1 เวกเตอร์ w^t_{XO} พิจารณาเพียงแบบจากชั้นข้อมูล X และ O ตามตารางที่ 6.14 ด้านล่างนี้

ตารางที่ 6.14 แบบ X และ O

Pattern No.	1	2	Bias
1	0.5	3.0	1
2	1.0	3.0	1
3	0.5	2.5	1
4	–6.0	–6.0	–1
5	–6.0	–6.5	–1
6	–7.0	–6.0	–1

ใช้การเรียนรู้เพอร์เซ็ปตรอน เริ่มต้นจาก $w^t_1 = \begin{pmatrix}0\\0\\0\end{pmatrix}$

$w^t_1x_1 = 0$ ดังนั้น $w^t_2 = w^t_1 + x_1 = \begin{pmatrix}0.5\\3\\1\end{pmatrix}$

$w^t_2x_2 = 10.5 > 0$ และ $w^t_2x_3 = 8.75 > 0$

$w^t_2x_4 = -22 < 0$ ดังนั้น $w^t_3 = w^t_2 + x_4 = \begin{pmatrix}-5.5\\-3\\0\end{pmatrix}$

w^t_3 จำแนกแบบที่ 5 และ 6 ถูกต้อง เนื่องจาก $w^t_3x_5 > 0$ และ $w^t_3x_6 > 0$

w^t_3 จำแนกแบบที่ 1 ผิดพลาด เนื่องจาก $w^t_3x_1 = -11.75 < 0$ ดังนั้น $w^t_4 = w^t_3 + x_1 = \begin{pmatrix}-5\\0\\1\end{pmatrix}$

$w^t_4x_2 = -4 < 0$ ดังนั้น $w^t_5 = w^t_4 + x_2 = \begin{pmatrix}-4\\3\\2\end{pmatrix}$

จากการแทนค่า w^t_5 พบว่าจำแนกแบบ 3, 4, 5, 6, 1 และ 2 ได้อย่างถูกต้อง ดังนั้นเวกเตอร์น้ำหนัก $w^t_{XO} = \begin{pmatrix}-4\\3\\2\end{pmatrix}$ เป็นขอบเขตการตัดสินใจที่สามารถแบ่งแยกชั้นข้อมูล X และ O ออกจากกันได้

ตัวอย่างที่ 8.2 เวกเตอร์ w^t_{X*} พิจารณาเพียงแบบจากชั้นข้อมูล X และ $*$ ตามตารางที่ 6.15 ด้านล่างนี้

ตารางที่ 6.15 แบบ X และ $*$

Pattern No.	1	2	Bias
1	0.5	3.0	1
2	1.0	3.0	1
3	0.5	2.5	1
4	–10.0	–0.5	–1
5	–10.0	–1.0	–1
6	–11.0	–1.0	–1

ใช้การเรียนรู้เพอร์เซ็ปตรอน เริ่มต้นจาก $w^t_1 = \begin{pmatrix}0\\0\\0\end{pmatrix}$ ดังนั้น $w^t_2 = \begin{pmatrix}0.5\\3\\1\end{pmatrix}$ จากนั้น $w^t_3 = \begin{pmatrix}-9.5\\2.5\\0\end{pmatrix}$

(การคำนวณในส่วนที่ข้ามไป สามารถทำเพิ่มเติมได้จากแบบฝึกหัด ข้อ 7.1)

เมื่อคำนวณต่อไป ได้เวกเตอร์น้ำหนักสุดท้ายคือ $w^t_4 = w^t_{X*} = \begin{pmatrix}-9\\5\\1\end{pmatrix}$

ตัวอย่างที่ 8.3 เวกเตอร์ w^t_{O*} พิจารณาเพียงแบบจากชั้นข้อมูล O และ * ตามตารางที่ 6.16 ด้านล่างนี้

ตารางที่ 6.16 แบบ O และ *

Pattern No.	1	2	Bias
1	6.0	6.0	1
2	6.0	6.5	1
3	7.0	6.0	1
4	–10.0	–0.5	–1
5	–10.0	–1.0	–1
6	–11.0	–1.0	–1

ใช้การเรียนรู้เพอร์เซ็ปตรอน เริ่มต้นจาก $w^t_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ ดังนั้น คำนวณค่า $w^t_2 = \begin{pmatrix} 0.5 \\ 3 \\ 1 \end{pmatrix}$

(การคำนวณในส่วนที่ข้ามไป สามารถทำเพิ่มเติมได้จากแบบฝึกหัด ข้อ 7.2)

จากการคำนวณทั้งหมด ได้เวกเตอร์น้ำหนักสุดท้าย คือ $w^t_3 = w^t_{O*} = \begin{pmatrix} -4 \\ 5.5 \\ 0 \end{pmatrix}$

จากการสังเกต พบว่าวิธีการ OAO สร้างตัวจำแนกแบบทวิภาคจำนวน 3 ตัวแบบ ได้แก่ w^t_{XO}, w^t_{X*} และ w^t_{O*} ทดลองใช้ขอบเขตการตัดสินใจเหล่านี้ จำแนกแบบที่ผ่านการแปลงโดยการเพิ่มมิติอีก 1 แล้วเพิ่มค่า 1 ที่มิตินั้น ตารางที่ 6.17 แสดงการแปลงแบบจาก ตารางที่ 6.13 ในส่วนเพิ่มเติม การแปลผลการทำนาย อธิบายได้ดังนี้.

กำหนดค่า $w_{C1\,C2}\, x_i$ เมื่อ

- C_1 คือ ชั้นข้อมูลฝั่งบวก
- C_2 คือ ชั้นข้อมูลฝั่งลบ
- $w_{C1\,C2}$ คือ ขอบเขตการตัดสินใจสำหรับตัดสินว่าแบบ x_i อยู่ในชั้นข้อมูล C_1 หรือ C_2
- x_i คือ แบบลำดับที่ i
- $w_{C1\,C2}\, x_i$ คือ ค่าที่ใช้สำหรับแปลผลการทำนาย ดังนี้
  - ถ้า $w_{C1\,C2}\, x_i > 0$ แล้ว x_i อยู่ในชั้นข้อมูล C_1 (ฝั่งบวก)
  - ถ้า $w_{C1\,C2}\, x_i < 0$ แล้ว x_i อยู่ในชั้นข้อมูล C_2 (ฝั่งลบ)

ตารางที่ 6.17 แบบที่ผ่านการแปลง

Pattern No.	1	2	3	Class
1	0.5	3.0	1.0	X
2	1.0	3.0	1.0	X
3	0.5	2.5	1.0	X
4	6.0	6.0	1.0	O
5	6.0	6.5	1.0	O
6	7.0	6.0	1.0	O
7	10.0	0.5	1.0	*
8	10.0	1.0	1.0	*
9	11.0	1.0	1.0	*

ทดลองนำตัวจำแนกทวิภาคทั้ง 3 ทำนายแบบ ดังต่อไปนี้

(a) พิจารณาแบบที่ 1

$x_1 = (0.5, 3, 1)$

จากการแทนค่าพบว่า

$w^t_{XO}\ x_1 = 9 > 0$ ดังนั้น x_1 ถูกจำแนกให้อยู่ในชั้นข้อมูล X

$w^t_{X*}\ x_1 = 11.5 > 0$ ดังนั้น x_1 ถูกจำแนกให้อยู่ในชั้นข้อมูล X

$w^t_{O*}\ x_1 = 14.5 > 0$ ดังนั้น x_1 ถูกจำแนกให้อยู่ในชั้นข้อมูล O

เนื่องจากการทำนายอาศัยตามเสียงส่วนใหญ่ ดังนั้น x_1 ถูกจำแนกให้อยู่ใน X

(b) พิจารณาแบบที่ 5

$X_5 = (6, 6.5, 1)$

จากการแทนค่าพบว่า

$w^t_{XO}\ x_5 = -2.5 < 0$ ดังนั้น x_5 ถูกจำแนกให้อยู่ในชั้นข้อมูล O

$w^t_{X*}\ x_5 = -20.5 < 0$ ดังนั้น x_5 ถูกจำแนกให้อยู่ในชั้นข้อมูล *

$w^t_{O*}\ x_5 = 11.75 > 0$ ดังนั้น x_5 ถูกจำแนกให้อยู่ในชั้นข้อมูล O

เนื่องจากการทำนายอาศัยตามเสียงส่วนใหญ่ ดังนั้น x_5 ถูกจำแนกให้อยู่ใน O

(c) พิจารณาแบบที่ 9

$X_9 = (11, 1, 1)$

จากการแทนค่าพบว่า

$w^t_{XO}\ x_9 = -39 < 0$ ดังนั้น x_9 ถูกจำแนกให้อยู่ในชั้นข้อมูล O

$w^t_{X*}\ x_9 = -93 < 0$ ดังนั้น x_9 ถูกจำแนกให้อยู่ในชั้นข้อมูล ∗

$w^t_{O*}\ x_9 = -38.5 < 0$ ดังนั้น x_9 ถูกจำแนกให้อยู่ในชั้นข้อมูล ∗

เนื่องจากการทำนายอาศัยตามเสียงส่วนใหญ่ ดังนั้น x_9 ถูกจำแนกให้อยู่ใน ∗

II. หนึ่งต่อทั้งหมด (OAA: One Against All)

วิธีนี้เป็นการแปลงปัญหาหลายชั้นข้อมูลเป็นปัญหาสองชั้นข้อมูล สำหรับแต่ละชั้นข้อมูล c_i ให้สร้างชั้นข้อมูล $\bar{c}_i$ ซึ่งมาจากการรวมชั้นข้อมูลที่เหลือทั้งหมด ดังนั้น

$\bar{c}_i = U_{j=1}^{c}$ เมื่อ $i \neq j$ และ $j = 1, 2, 3, \ldots, C$

- C คือ จำนวนชั้นข้อมูลทั้งหมดในเซตข้อมูล

ด้วยวิธีการนี้ เป็นการพิจารณาแบบทีละชั้นข้อมูลกับแบบจากชั้นข้อมูลที่เหลือทั้งหมด จำนวนชั้นข้อมูลมีค่าเท่ากับจำนวนตัวจำแนกแบบที่ต้องเรียนรู้ เช่น เซตข้อมูลนี้มี 3 ชั้นข้อมูล แล้วต้องสร้างตัวจำแนกแบบจำนวน 3 ตัว เพื่อพิจารณา ดังต่อไปนี้

- ❖ w^t_X คือ เวกเตอร์สำหรับแบ่งแยกแบบในชั้นข้อมูล X หรือ นอกชั้นข้อมูล X
- ❖ w^t_O คือ เวกเตอร์สำหรับแบ่งแยกแบบในชั้นข้อมูล O หรือ นอกชั้นข้อมูล O
- ❖ w^t_* คือ เวกเตอร์สำหรับแบ่งแยกแบบในชั้นข้อมูล ∗ หรือ นอกชั้นข้อมูล ∗

ตัวอย่างที่ 8 พิจารณาแบบในชั้นตารางที่ 6.18 จงแสดงวีธีการ OAA กำหนดให้หลัก Bias คือ มิติที่เพิ่มมาหลังจากการทำให้อยู่ในรูปปกติ

ตัวอย่างที่ 9.1 เวกเตอร์ w^t_X พิจารณาเพียงแบบจากชั้นข้อมูล x และ $\bar{x}$ ตามตารางที่ 6.18 ต่อไปนี้

ตารางที่ 6.18 แบบ X และ แบบที่เหลือ

Pattern No.	1	2	Bias
1	0.5	3.0	1
2	1.0	3.0	1
3	0.5	2.5	1
4	–6.0	–6.0	–1
5	–6.0	–6.5	–1
6	–7.0	–6.0	–1
7	–10.0	–0.5	–1
8	–10.0	–1.0	–1
9	–11.0	–1.0	–1

ใช้การเรียนรู้เพอร์เซ็ปตรอน เริ่มต้นจาก $w^t_1 = \begin{pmatrix}0\\0\\0\end{pmatrix}$

$w^t_1x_1 = 0$ ดังนั้น $w^t_2 = w^t_1 + x_1 = \begin{pmatrix}0.5\\3\\1\end{pmatrix}$

w^t_2 จำแนก x_2, x_3 ถูกต้อง แต่จำแนก x_4 ผิดพลาด เพราะ $w^t_2x_4 = –22$ ดังนั้น $w^t_3 = w^t_2 + x_4 = \begin{pmatrix}-5.5\\-3\\0\end{pmatrix}$

w^t_3 จำแนก x_5, x_6, x_7, x_8 x_9 ถูกต้อง แต่จำแนก x_1 ผิด ดังนั้น $w^t_4 = \begin{pmatrix}-5\\0\\1\end{pmatrix}$

w^t_4 จำแนก x_2 ผิดพลาด ดังนั้น $w^t_5 = \begin{pmatrix}-4\\3\\2\end{pmatrix}$

w^t_5 จำแนก x_3, x_4, x_5, x_6, x_7, x_8 x_9, x_1, x_2 ถูกต้องหมด ดังนั้น $w^t_X = \begin{pmatrix}-4\\3\\2\end{pmatrix}$

ตัวอย่างที่ 9.2 เวกเตอร์ w^t_O พิจารณาเพียงแบบจากชั้นข้อมูล O และ $\bar{O}$ ตามตารางที่ 6.19 ต่อไปนี้

เริ่มต้นจาก $w^t_1 = \begin{pmatrix}0\\0\\0\end{pmatrix}$ จากนั้น คำนวณค่า $w^t_2 = w^t_1 + x_1 = \begin{pmatrix}0.5\\3\\1\end{pmatrix}$

(การคำนวณในส่วนที่ข้ามไป สามารถทำเพิ่มเติมได้จากแบบฝึกหัด ข้อ 7.3)

ทำตามขั้นตอนวิธีของการเรียนรู้เพอร์เซ็ปตรอน จนกระทั่งได้ขอบเขตการตัดสินใจ ได้แก่ $w^t_O = \begin{pmatrix}-1.5\\5.5\\-21\end{pmatrix}$

ตารางที่ 6.19 แบบ O และ แบบที่เหลือ

Pattern No.	1	2	Bias
1	–0.5	–3.0	–1
2	–1.0	–3.0	–1
3	–0.5	–2.5	–1
4	6.0	6.0	1
5	6.0	6.5	1
6	7.0	6.0	1
7	–10.0	–0.5	–1
8	–10.0	–1.0	–1
9	–11.0	–1.0	–1

ตัวอย่างที่ 9.2 เวกเตอร์ $w^t{}_*$ พิจารณาเพียงแบบจากชั้นข้อมูล * และ $\bar{*}$ ตามตารางที่ 6.20 ด้านล่างนี้

ตารางที่ 6.20 แบบ * และ แบบที่เหลือ

Pattern No.	1	2	Bias
1	–0.5	–3.0	–1
2	–1.0	–3.0	–1
3	–0.5	–2.5	–1
4	–6.0	–6.0	–1
5	–6.0	–6.5	–1
6	–7.0	–6.0	–1
7	10.0	0.5	1
8	10.0	1.0	1
9	11.0	1.0	1

เริ่มต้นจาก $w^t{}_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ ดังนั้น $w^t{}_2 = \begin{pmatrix} 0.5 \\ 3 \\ 1 \end{pmatrix}$

(การคำนวณในส่วนที่ข้ามไป สามารถทำเพิ่มเติมได้จากแบบฝึกหัด ข้อ 7.4)

หลังจากเวกเตอร์น้ำหนักโดนปรับปรุงค่า 4 ครั้ง $w^t{}_5 = \begin{pmatrix} 6 \\ 6.5 \\ 1 \end{pmatrix}$

เนื่องจาก $w^t{}_5$ จำแนกแบบทุกแบบได้อย่างถูกต้อง ดังนั้น $w^t{}_* = \begin{pmatrix} 6 \\ 6.5 \\ 1 \end{pmatrix}$

เช่นเดียวกันกับวิธีการก่อนหน้า จำเป็นต้องแปลงแบบโดยการเพิ่มค่า 1 ที่มิติใหม่ จากนั้นนำ 3 ตัวแบบทวิภาคที่สร้างจากวิธีการ OAA ได้แก่ w^t_X, w^t_O และ w^t_* จำแนกแบบจาก ตารางที่ 6.8 ดังนี้

(a) พิจารณาแบบ $x_1 = (0.5, 3, 1)$ ทดลองแทนค่า ดังนี้

$w^t_X x_1 = 9 > 0$ ดังนั้น x_1 ถูกจำแนกให้อยู่ในชั้นข้อมูล X

$w^t_O x_1 = –5.25 > 0$ ดังนั้น x_1 ถูกจำแนกให้อยู่นอกชั้นข้อมูล O (X หรือ $*$)

$w^t_* x_1 = –35.25 > 0$ ดังนั้น x_1 ถูกจำแนกให้อยู่นอกชั้นข้อมูล $*$ (X หรือ O)

จากการตีความผลการทำนายส่วนใหญ่ แบบ x_1 ถูกจำแนกให้อยู่ในชั้นข้อมูล X

(b) พิจารณาแบบ $X_5 = (6, 6.5, 1)$ ทดลองแทนค่า ดังนี้

$w^t_X x_5 = –2.5 < 0$ ดังนั้น x_5 ถูกจำแนกให้อยู่นอกชั้นข้อมูล X (O หรือ $*$)

$w^t_O x_5 = 5.75 > 0$ ดังนั้น x_5 ถูกจำแนกให้อยู่ในชั้นข้อมูล O

$w^t_* x_5 = –71.75 < 0$ ดังนั้น x_5 ถูกจำแนกให้อยู่นอกชั้นข้อมูล $*$ (X หรือ O)

เนื่องจากการทำนายอาศัยตามเสียงส่วนใหญ่ ดังนั้น x_5 ถูกจำแนกให้อยู่ใน O

(c) พิจารณาแบบที่ $X_9 = (11, 1, 1)$ ทดลองแทนค่า ดังนี้

$w^t_X x_9 = –39 < 0$ ดังนั้น x_9 ถูกจำแนกให้อยู่นอกชั้นข้อมูล X (O หรือ $*$)

$w^t_O x_9 = –32 < 0$ ดังนั้น x_9 ถูกจำแนกให้อยู่นอกชั้นข้อมูล O (X หรือ $*$)

$w^t_* x_9 = 14 > 0$ ดังนั้น x_9 ถูกจำแนกให้อยู่ในชั้นข้อมูล $*$

เนื่องจากการทำนายอาศัยตามเสียงส่วนใหญ่ ดังนั้น x_9 ถูกจำแนกให้อยู่ใน $*$

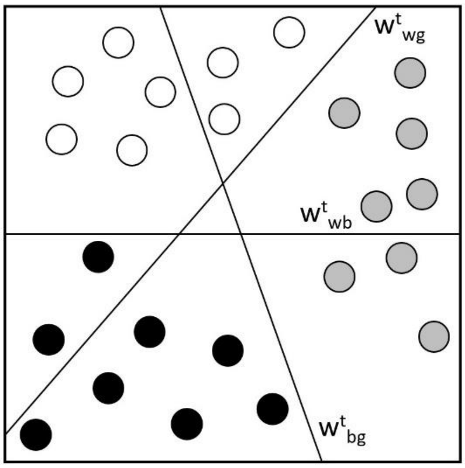
ความแตกต่างระหว่างวิธีการ OAO และ OAA แสดงให้เห็นในรูปที่ 6.10 และ 6.11 กำหนดให้เซตข้อมูลนี้บรรจุแบบ 2 มิติ จาก 3 ชั้นข้อมูล ได้แก่ ขาว (w) ดำ (b) เทา (g) นอกจากนี้ ขอบเขตการตัดสินใจแต่ละเส้นตรงมีความหมาย ดังต่อไปนี้

❖ OAO

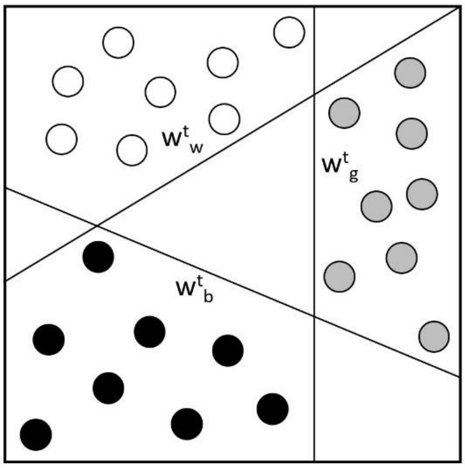
- w^t_{wb} คือ เส้นตรงที่แบ่งแยกชั้นข้อมูลขาวและดำออกจากกัน
- w^t_{wg} คือ เส้นตรงที่แบ่งแยกชั้นข้อมูลขาวและเทาออกจากกัน
- w^t_{bg} คือเส้นตรงที่แบ่งแยกชั้นข้อมูลดำและเทาออกจากกัน

- OAA
  - w^t_w คือ เส้นตรงที่แบ่งแยกชั้นข้อมูลขาวและนอกขาวออกจากกัน
  - w^t_b คือเส้นตรงที่แบ่งแยกชั้นข้อมูลดำและนอกดำออกจากกัน
  - w^t_g คือเส้นตรงที่แบ่งแยกชั้นข้อมูลเทาและนอกเทาออกจากกัน

สังเกตได้ว่า OAO สร้างขอบเขตการตัดสินใจตัดผ่านชั้นข้อมูลที่ไม่ได้พิจารณา ในขณะที่เส้นตรงของ OAA ไม่ตัดผ่านชั้นข้อมูลใด ๆ เลย เพียงแต่จะมีแบบจากต่างชั้นข้อมูลกันรวมอยู่ในขอบเขตการตัดสินใจเดียวกัน



รูปที่ 6.10 ขอบเขตการตัดสินใจ OAO



รูปที่ 6.11 ขอบเขตการตัดสินใจ OAA

การแปลงปัญหาหลายชั้นข้อมูลเป็นปัญหาสองชั้นข้อมูล ไม่ว่าจะเป็นวิธีการ OAO หรือ OAA ต่างก็เป็นการสร้างกลุ่มของตัวจำแนกแบบ เพื่อช่วยทำนายแบบทดสอบเหมือนกัน แต่เหมาะสมกับลักษณะเฉพาะตัวที่แตกต่างกันของเซตข้อมูล คำแนะนำสำหรับการนำแต่ละวิธีการ แสดงตามหัวข้อด้านล่างนี้

- OAO เหมาะกับการนำไปใช้งานจริง เนื่องจากใช้เวลาประมวลผลที่เหมาะสม
- OAA เหมาะสมกับเซตข้อมูลที่มีจำนวนชั้นข้อมูลจำนวนน้อย เพราะจะได้ลดผลกระทบเรื่องความแตกต่างของแบบจากต่างชั้นข้อมูล

นอกจากนี้ จุดแข็งและจุดอ่อนของวิธีการแต่ละประเภทแสดงตามรายการในตารางที่ 6.21

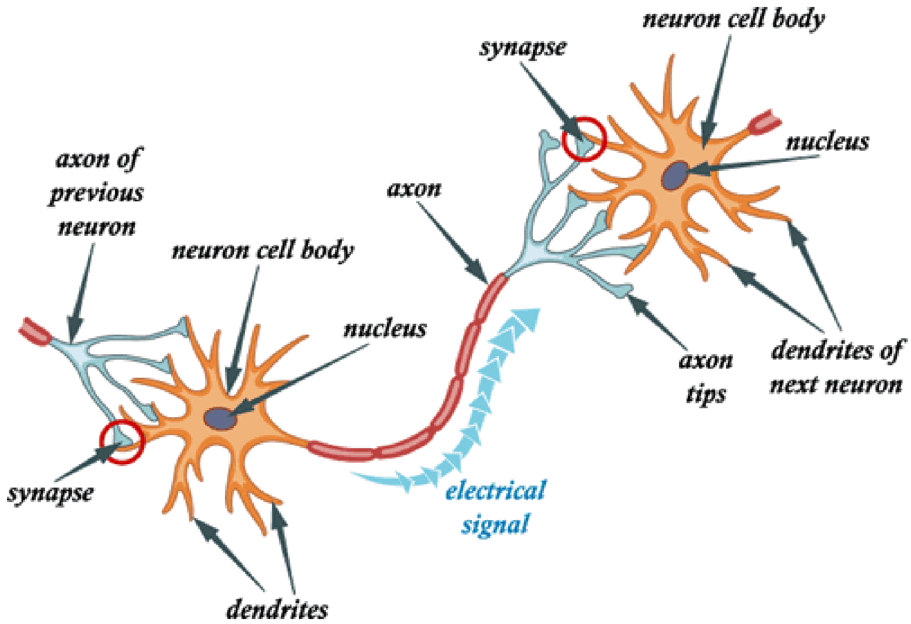
ตารางที่ 6.21 การเปรียบเทียบระหว่าง OAO และ OAA

วิธีการ / เปรียบเทียบ	หนึ่งต่อหนึ่ง	หนึ่งต่อทั้งหมด
ข้อดี	เนื่องจากไม่มีการรวมแบบจากต่างชั้นข้อมูลเดียวกัน ใช้เพียงแบบจากแต่ละชั้นข้อมูลเท่านั้น จึงมีความเหมาะสมกับกระบวนการเรียนรู้ของตัวจำแนกแบบ เนื่องจากแบบในชั้นข้อมูลเดียวกันย่อมมีความคล้ายกัน	ไม่เหมาะสมกับเซตข้อมูลที่ชั้นข้อมูลประกอบไปด้วย ข้อมูลรบกวน (Noise)
ข้อเสีย	ต้องสร้างตัวจำแนกแบบเป็นจำนวนมาก ส่งผลให้เวลาประมวลผลช้า ถ้าเซตข้อมูลมีปริมาณของชั้นข้อมูลสูง	แบบทั้งหมดนอกเหนือจากแบบที่พิจารณาจะถูกรวมเข้าด้วยกันเป็นชั้นข้อมูลเดียว ส่งผลให้ฟีเจอร์ในชั้นข้อมูลรวมนี้มีความแตกต่างกันสูงมาก ซึ่งไม่เหมาะสมกับการรู้จำแบบ

5. โครงข่ายประสาท

โครงข่ายประสาทเทียม (Artificial Neural Network) ถูกพัฒนาจากการจำลองการทำงานของสมองมนุษย์ ซึ่งมีโครงสร้างแสดงตาม รูปที่ 6.12 สมองมนุษย์ประกอบไปด้วย เซลล์ประสาท (Neuron) นับล้านเซลล์ ซึ่งเซลล์ประสาทเหล่านี้ติดต่อสื่อสารระหว่างกันโดยการส่งผ่าน สัญญาณไฟฟ้าเคมี (Electrochemical Signal) สัญญาณนี้วิ่งผ่าน จุดประสานประสาท (Synapse) โดยที่สัญญาณที่ผ่านจุดประสานตำแหน่งต่าง ๆ จะถูกผสมเข้าด้วยกันเป็นสัญญาณเดียว ถ้าสัญญาณรวมนี้มีค่าสูงกว่า ขีดแบ่ง (Threshold) เซลล์ประสาทที่รับสัญญาณนี้จะถูกกระตุ้นให้ส่งผลลัพธ์ไปยังเซลล์ประสาทอื่นผ่านทางแกนประสาท (Axon) ต่อไปในตำราฉบับนี้ จะใช้คำว่าโครงข่ายประสาทแทนโครงข่ายประสาทเทียมในความหมายที่เทียบเท่ากัน

โครงข่ายประสาทประกอบไปด้วย เซลล์ประสาทประดิษฐ์ (Artificial Neuron) ที่มีโครงสร้างคล้ายกับเซลล์ประสาทมนุษย์ ข้อมูลเข้าที่ถูกส่งไปยังเซลล์ประสาทแต่ละเซลล์จะถูกให้ค่าน้ำหนักและผสานเข้าด้วยกัน ถ้าผลรวมมีค่าสูงกว่าขีดแบ่งแล้วเซลล์ประสาทจึงจะส่งข้อมูลออก ยกตัวอย่างเช่น ข้อมูลออกคือ 1 ในกรณีที่ผลรวมมีค่าเกินขีดแบ่ง นอกเหนือจากนี้ ให้ข้อมูลออกคือ 0 เป็นต้น



รูปที่ 6.12 องค์ประกอบสมองมนุษย์

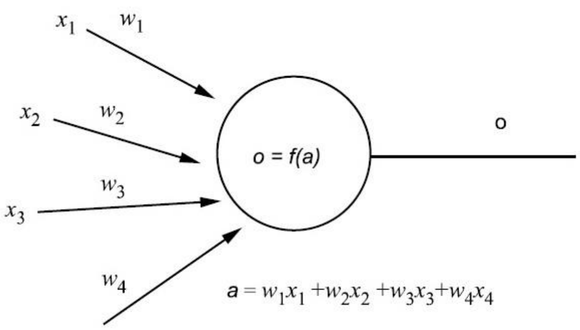
(**ที่มา:** https://simplebiologyy.blogspot.com/2014/08/conduction–of–nerve–impulse.html)

เซลล์ประสาทประดิษฐ์ที่ใช้สำหรับการจำแนกแบบ แสดงตามแผนผังในรูปที่ 6.13 โดยมีองค์ประกอบ ดังต่อไปนี้

- $x_1, x_2, x_3, \ldots, x_n$ คือ ข้อมูลเข้า ลำดับที่ 1 ถึง ลำดับที่ n
- $w_1, w_2, w_3, \ldots, w_n$ คือ น้ำหนักที่สอดคล้องกับข้อมูลเข้าในแต่ละลำดับ
- n คือ จำนวนข้อมูลเข้าทั้งหมด

ข้อมูลเข้าคือแบบในเซตข้อมูล โดยที่ x_i แทน ฟีเจอร์ลำดับที่ i ของแบบ ซึ่งสอดคล้องกับค่าน้ำหนักลำดับที่ i และ n คือจำนวนมิติของแบบ นอกจากนี้ การกระตุ้น (Activation) สามารถคำนวณได้ ดังนี้

$$a = w_1x_1 + w_2x_2 + \ldots + w_nx_n$$

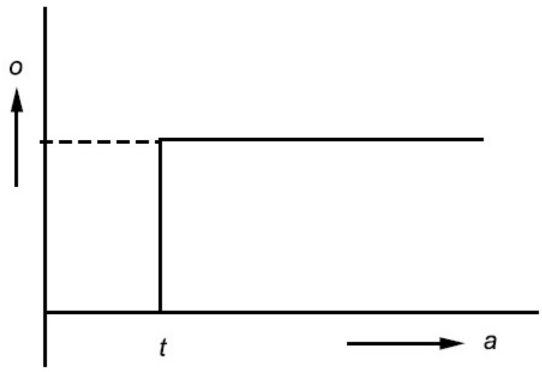


รูปที่ 6.13 เซลล์ประสาทประดิษฐ์

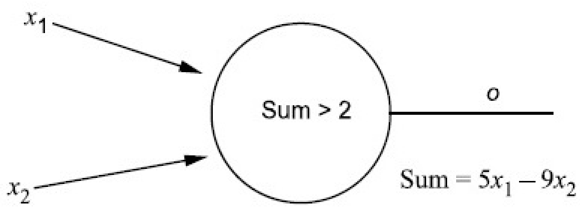
ข้อมูลออก o ได้มาจากฟังก์ชันกระตุ้นขีดแบ่ง แสดงตามรูปที่ 6.14 เมื่อ t คือ ขีดแบ่ง โดยที่

$o = 1$ ถ้า $a \geq t$

$o = 0$ ถ้า $a < t$



รูปที่ 6.14 ฟังก์ชันกระตุ้นขีดแบ่ง



รูปที่ 6.15 เพอร์เซ็ปตรอนของขอบเขตการตัดสินใจ

ฟังก์ชันกระตุ้นขีดแบ่งถูกใช้ในการเรียนรู้เพอร์เซ็ปตรอน พิจารณาขอบเขตการตัดสินใจที่อยู่ในรูปของเส้นตรง $5x_1 - 9x_2 - 2 = 0$ จากตัวอย่างที่ 5

เนื่องจาก

$$5x_1 - 9x_2 - 2 = 0$$

สามารถเขียนให้อยู่ในรูปของ

$$w_1x_1 - w_2x_2 = 2$$

โดยที่

$5x_1 - 9x_2 < 2$ แล้วแบบในชั้นข้อมูล X (ด้านลบ)

$5x_1 - 9x_2 > 2$ แล้วแบบในชั้นข้อมูล O (ด้านบวก)

พิจารณาเส้นตรงนี้ด้วยเซลล์ประสาทประดิษฐ์ตาม โดยตัวแปรต่าง ๆ มีค่าดังนี้

$w_1 = 5$ และ $w_2 = 9$

$a = 5x_1 - 9x_2$

$t = 2$

พิจารณาเซลล์ประสาทประดิษฐ์ใน รูปที่ 6.5

$a < t$ แล้ว $o = 0$

$a > t$ แล้ว $o = 1$

หรือ

$5x_1 - 9x_2 < 2$ แล้ว $o = 0$

$5x_1 - 9x_2 > 2$ แล้ว $o = 1$

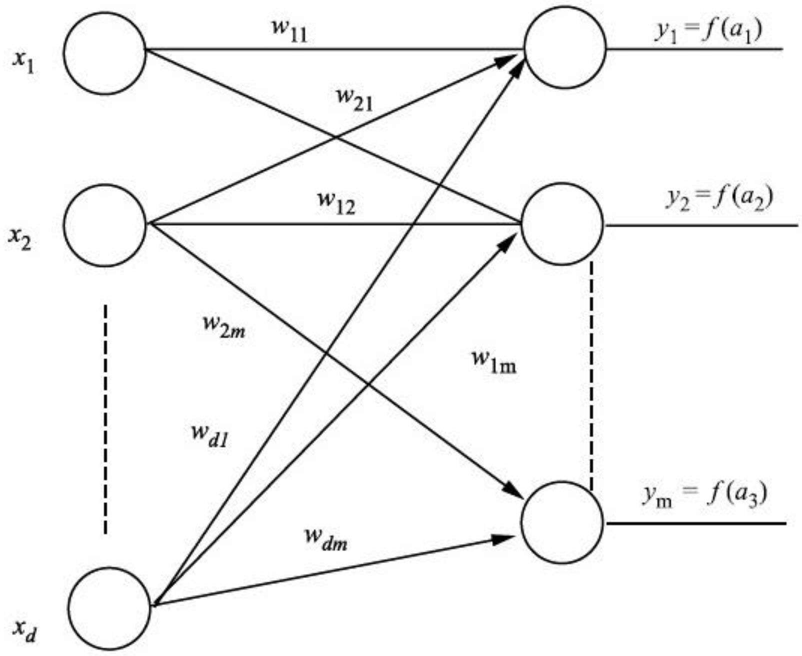
โครงข่ายการป้อนไปข้างหน้า

โครงข่ายประสาทที่ใช้แนวคิดพื้นฐานที่สุด ได้แก่ โครงข่ายการป้อนไปข้างหน้า (Feed-Forward Network) ประกอบไปด้วย หน่วยข้อมูลเข้า (Input Unit) ที่เชื่อมต่อกับ หน่วยข้อมูลออก (Output Unit) แสดงตามรูปที่ 6.16 โดยที่ องค์ประกอบต่าง ๆ มีดังต่อไปนี้

- d คือ จำนวนหน่วยข้อมูลเข้า
- m คือ จำนวนหน่วยข้อมูลออก
- $x_1, x_2, \ldots, x_d$ คือ ข้อมูลเข้า ลำดับที่ 1 ถึง ลำดับที่ d
- $y_1, y_2, \ldots, y_m$ คือ ข้อมูลออก ลำดับที่ 1 ถึง ลำดับที่ m ค่านี้ขึ้นอยู่กับค่า a ดังนี้
- $w_{11}, w_{12}, \ldots, w_{1m}, \ldots, w_{d1}, w_{d2}, \ldots, w_{dm}$ คือ น้ำหนักที่สอดคล้องกับข้อมูลเข้าและออก

ความหมายของค่าน้ำหนักเช่น w_{ij} คือ ค่าน้ำหนักที่สอดคล้องระหว่างข้อมูลเข้าลำดับที่ i และ ข้อมูลออกลำดับที่ j ยกตัวอย่าง ดังต่อไปนี้

- w_{11} คือ ค่าน้ำหนักของ ข้อมูลเข้าตัวที่ 1 และ ข้อมูลออกตัวที่ 1 ซึ่งเชื่อมต่อกัน
- w_{21} คือ ค่าน้ำหนักของ ข้อมูลเข้าตัวที่ 2 และ ข้อมูลออกตัวที่ 1 ซึ่งเชื่อมต่อกัน



รูปที่ 6.16 โครงข่ายชนิดป้อนไปข้างหน้า

พิจารณาเพอร์เซ็ปตรอนในรูปที่ 6.15 ซึ่งสามารถพิจารณาเป็นโครงข่ายการป้อนไปข้างหน้าได้ โดยกำหนดค่าตัวแปร ดังต่อไปนี้

$d = 2$

$m = 1$

$w_{11} = 5$ และ $w_{21} = -9$

$t = 2$

การเรียนรู้ของตัวแบบ

ค่าน้ำหนักเริ่มต้นของโครงข่ายประสาทถูกสุ่มขึ้นระหว่าง 0 ถึง 1 เมื่อมีข้อมูลเข้าวิ่งเข้าสู่ตัวแบบ แล้วการกระตุ้นของข้อมูลออกตัวที่ i ($1 \le i \le m$) คำนวณได้ ตามสมการด้านล่างนี้

$$a_i = w_{1i}x_i + w_{2i}x_2 + \ldots + w_{di}x_d$$

ข้อมูลออกลำดับที่ i คำนวณได้ ดังนี้

$$o_i = f(a_i)$$

กำหนดให้

$t_1, \ldots, t_m$ คือ ข้อมูลออกที่ถูกต้อง (Correct Output)

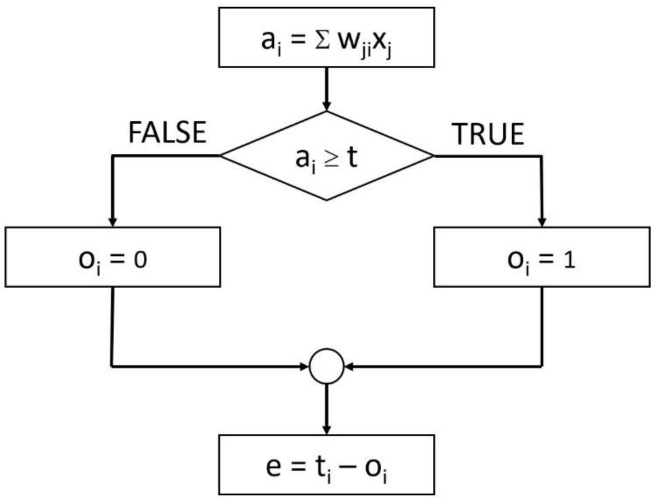
ดังนั้น ความผิดพลาดของข้อมูลออกลำดับที่ i คำนวณได้ ดังนี้

$$e_i = t_i - o_i$$

ค่าความผิดพลาดนี้ถูกใช้สำหรับ การแพร่กระจายย้อนกลับ (Back-propagation) เพื่อปรับปรุงค่าน้ำหนัก w_{ji} ระหว่าง ข้อมูลเข้าตัวที่ j และ ข้อมูลออกตัวที่ i คำนวณได้ตามสมการด้านล่างนี้

$$w_{ji} = w_{ji} + \alpha \times x_j \times e_i$$

การเรียนรู้ของโครงข่ายประสาท แสดงตามผังงานในรูปที่ 6.17 และ ตัวอย่างต่อไปนี้



รูปที่ 6.17 ผังงานการเรียนรู้ของโครงข่ายประสาท

ตัวอย่างที่ 10 กำหนดแบบฝึกฝนจำนวน 4 แบบ ตามตารางที่ 6.22 แต่ละแบบมี 4 มิติ (d = 4) และข้อมูลออก 2 ตัว (m = 2) กำหนดค่าเริ่มต้น ดังนี้ น้ำหนัก w_{ji} = 0.5 สำหรับ $1 \leq j \leq 4$ เมื่อ i = 1, 2 ขีดแบ่ง t = 0 และ α = 0.5

เนื่องจาก d = 4 และ m = 2 แล้ว จำนวนน้ำหนักทั้งหมดคือ 8 ตัว ดังนี้

w_{11}, w_{12} คือ น้ำหนักที่เชื่อมต่อระหว่าง ข้อมูลเข้าตัวที่ 1 และ ข้อมูลออกตัวที่ 1, 2 ตามลำดับ

w_{21}, w_{22} คือ น้ำหนักที่เชื่อมต่อระหว่าง ข้อมูลเข้าตัวที่ 2 และ ข้อมูลออกตัวที่ 1, 2 ตามลำดับ

w_{31}, w_{32} คือ น้ำหนักที่เชื่อมต่อระหว่าง ข้อมูลเข้าตัวที่ 3 และ ข้อมูลออกตัวที่ 1, 2 ตามลำดับ

w_{41}, w_{42} คือ น้ำหนักที่เชื่อมต่อระหว่าง ข้อมูลเข้าตัวที่ 4 และ ข้อมูลออกตัวที่ 1, 2 ตามลำดับ

ตารางที่ 6.22 แบบ 4 มิติ และ หน่วยข้อมูลออก 2 หน่วย

Pattern No.	x_1	x_2	X_3	X_4	t_1	t_2
1	1	0	0	0	1	0
2	0	1	0	0	1	0
3	0	0	0	1	0	1
4	0	0	1	0	0	1

เนื่องจาก d = 4 และ m = 2 แล้ว จำนวนน้ำหนักทั้งหมดคือ 8 ตัว ดังนี้

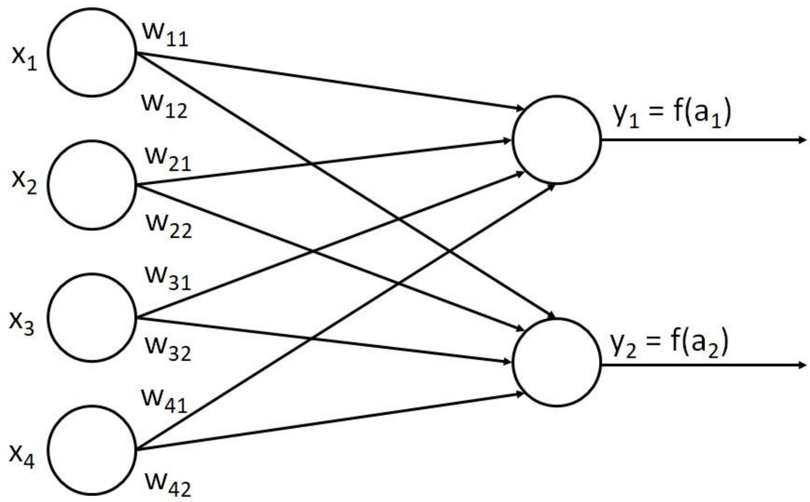
w_{11}, w_{12} คือ น้ำหนักที่เชื่อมต่อระหว่าง ข้อมูลเข้าตัวที่ 1 และ ข้อมูลออกตัวที่ 1, 2 ตามลำดับ

w_{21}, w_{22} คือ น้ำหนักที่เชื่อมต่อระหว่าง ข้อมูลเข้าตัวที่ 2 และ ข้อมูลออกตัวที่ 1, 2 ตามลำดับ

w_{31}, w_{32} คือ น้ำหนักที่เชื่อมต่อระหว่าง ข้อมูลเข้าตัวที่ 3 และ ข้อมูลออกตัวที่ 1, 2 ตามลำดับ

w_{41}, w_{42} คือ น้ำหนักที่เชื่อมต่อระหว่าง ข้อมูลเข้าตัวที่ 4 และ ข้อมูลออกตัวที่ 1, 2 ตามลำดับ

รูปที่ 6.18 แสดงตำแหน่งของน้ำหนักที่การเชื่อมต่อแต่ละคู่ของข้อมูลเข้าและข้อมูลออก



รูปที่ 6.18 ค่าน้ำหนักในแต่ละการเชื่อมต่อ

1. พิจารณาแบบที่ 1 การกระตุ้นคำนวณได้ ตามสมการด้านล่างนี้

$a_1 = w_{11}x_1 + w_{21}x_1 + w_{31}x_1 + w_{41}x_1$

$a_2 = w_{12}x_2 + w_{22}x_2 + w_{32}x_2 + w_{42}x_2$

แทนค่าตัวแปร ได้ค่าดังต่อไปนี้

$a_1 = 0.5\times1 + 0.5\times0 + 0.5\times0 + 0.5\times0 = 0.5$

$a_2 = 0.5\times1 + 0.5\times0 + 0.5\times0 + 0.5\times0 = 0.5$

เนื่องจาก $a_1 > t$ ดังนั้น ข้อมูลออก $o_1 = 1$ และ ความผิดพลาด คำนวณได้ ดังนี้

$e_1 = 1 - 1 = 0$

เนื่องจาก $a_2 > t$ ดังนั้น ข้อมูลออก $o_2 = 1$ และ ความผิดพลาด คำนวณได้ ดังนี้

$e_2 = 0 - 1 = -1$

2. ปรับปรุงค่าน้ำหนัก ตามสมการด้านล่างนี้

$w_{11} = w_{11} + \alpha \times x_1 \times e_1;$ $w_{12} = w_{12} + \alpha \times x_1 \times e_2$

$w_{21} = w_{21} + \alpha \times x_2 \times e_1;$ $w_{22} = w_{22} + \alpha \times x_2 \times e_2$

$w_{31} = w_{31} + \alpha \times x_3 \times e_1;$ $w_{32} = w_{32} + \alpha \times x_3 \times e_2$

$w_{41} = w_{41} + \alpha \times x_4 \times e_1;$ $w_{42} = w_{42} + \alpha \times x_4 \times e_2$

แทนค่าตัวแปร ได้ค่าดังต่อไปนี้

$w_{11} = 0.5 + 0.5 \times 1 \times 0 = 0.5;$ $w_{12} = 0.5 + 0.5 \times 1 \times (-1) = 0.0$

$w_{21} = 0.5 + 0.5 \times 0 \times 0 = 0.5;$ $w_{22} = 0.5 + 0.5 \times 0 \times (-1) = 0.5$

$w_{31} = 0.5 + 0.5 \times 0 \times 0 = 0.5;$ $w_{32} = 0.5 + 0.5 \times 0 \times (-1) = 0.5$

$w_{41} = 0.5 + 0.5 \times 0 \times 0 = 0.5;$ $w_{42} = 0.5 + 0.5 \times 0 \times (-1) = 0.5$

3. พิจารณาแบบที่ 2 การกระตุ้น มีค่าดังนี้

$a_1 = 0.5\times0 + 0.5\times1 + 0.5\times0 + 0.5\times0 = 0.5$

$a_2 = 0.0\times0 + 0.5\times1 + 0.5\times0 + 0.5\times0 = 0.5$

เนื่องจาก $a_1 > t$ ดังนั้น ข้อมูลออก $o_1 = 1$ แล้ว ความผิดพลาด คำนวณได้ ดังนี้

$e_1 = 1 - 1 = 0$

เนื่องจาก $a_2 > t$ ดังนั้น ข้อมูลออก $o_2 = 1$ แล้ว ความผิดพลาด คำนวณได้ ดังนี้

$e_2 = 0 - 1 = -1$

4. ปรับปรุงค่าน้ำหนัก ได้ดังนี้

$w_{11} = 0.5 + 0.5 \times 0 \times 0 = 0.5$; $w_{12} = 0.0 + 0.5 \times 0 \times (-1) = 0.0$

$w_{21} = 0.5 + 0.5 \times 1 \times 0 = 0.5$; $w_{22} = 0.5 + 0.5 \times 1 \times (-1) = \underline{0.0}$

$w_{31} = 0.5 + 0.5 \times 0 \times 0 = 0.5$; $w_{32} = 0.5 + 0.5 \times 0 \times (-1) = 0.5$

$w_{41} = 0.5 + 0.5 \times 0 \times 0 = 0.5$; $w_{42} = 0.5 + 0.5 \times 0 \times (-1) = 0.5$

สังเกตได้ว่า w_{22} มีการเปลี่ยนแปลงค่า

5. พิจารณาแบบที่ 3 และ 4 จบด้วยการปรับปรุงค่าน้ำหนัก ได้ดังนี้

$w_{11} = 0.5$; $w_{12} = 0.0$

$w_{21} = 0.5$; $w_{22} = 0.0$

$w_{31} = 0.0$; $w_{32} = 0.5$

$w_{41} = 0.0$; $w_{42} = 0.5$

จากค่าน้ำหนักล่าสุด แบบทั้งหมดถูกจำแนกได้อย่างถูกต้อง เนื่องจาก $e_1 = e_2 = 0$ ดังนั้น น้ำหนักทั้งหมดจะไม่ถูกปรับปรุงค่า และ การเรียนรู้ของตัวแบบจะสิ้นสุดลง

ตัวอย่างที่ 11 กำหนดแบบฝึกฝน 4 มิติ จำนวน 4 แบบ และ ข้อมูลออก 2 ตัว กำหนดค่าเริ่มต้นดังนี้ น้ำหนัก $w_{ji} = 0.5$ สำหรับ $1 \leq j \leq 4$ และ $1 \leq i \leq 2$ ขีดแบ่ง $t = 0$ และ $\alpha = 0.5$ ในส่วนเพิ่มเติม จงแสดงการคำนวณโดยใช้โครงข่ายการป้อนไปข้างหน้า ในการหาค่าน้ำหนักทั้งหมด เมื่อฟังก์ชันการกระตุ้นขีดแบ่ง แสดงตามเงื่อนไขด้านล่างนี้

$$o = 1 \text{ ถ้า } a > t$$

$$o = 0 \text{ ถ้า } a \leq t$$

ตารางที่ 6.23 แบบ 4 มิติ และ หน่วยข้อมูลออก 2 หน่วย 2

Pattern No.	x_1	x_2	x_3	x_4	t_1	t_2
1	0	0	0	1	1	0
2	0	0	1	0	1	0
3	0	1	0	0	0	1
4	1	0	0	0	0	1

1. พิจารณาแบบที่ 1 การกระตุ้น คำนวณได้ ดังนี้

$$a_1 = 0.5\times0 + 0.5\times0 + 0.5\times0 + 0.5\times1 = 0.5$$

$$a_2 = 0.5\times0 + 0.5\times0 + 0.5\times0 + 0.5\times1 = 0.5$$

เนื่องจาก $a_1 > t$ ดังนั้น ข้อมูลออก $o_1 = 1$ แล้ว ความผิดพลาด คำนวณได้ ดังนี้

$$e_1 = 1 - 1 = 0$$

เนื่องจาก $a_2 > t$ ดังนั้น ข้อมูลออก $o_2 = 1$ แล้ว ความผิดพลาด คำนวณได้ ดังนี้

$$e_2 = 0 - 1 = -1$$

2. ปรับปรุงค่าน้ำหนัก ได้ดังนี้

$w_{11} = 0.5 + 0.5 \times 0 \times 0 = 0.5$; $w_{12} = 0.5 + 0.5 \times 0 \times (-1) = 0.5$

$w_{21} = 0.5 + 0.5 \times 0 \times 0 = 0.5$; $w_{22} = 0.5 + 0.5 \times 0 \times (-1) = 0.5$

$w_{31} = 0.5 + 0.5 \times 0 \times 0 = 0.5$; $w_{32} = 0.5 + 0.5 \times 0 \times (-1) = 0.5$

$w_{41} = 0.5 + 0.5 \times 1 \times 0 = 0.5$; $w_{42} = 0.5 + 0.5 \times 1 \times (-1) = \underline{0.0}$

สังเกตได้ว่า w_{42} มีการเปลี่ยนแปลงค่า

3. พิจารณาแบบที่ 2 การกระตุ้น มีค่า ดังนี้

$$a_1 = 0.5\times0 + 0.5\times0 + 0.5\times1 + 0.5\times0 = 0.5$$

$$a_2 = 0.5\times0 + 0.5\times0 + 0.5\times1 + 0.0\times0 = 0.5$$

เนื่องจาก $a_1 > t$ ดังนั้น ข้อมูลออก $o_1 = 1$ แล้ว ความผิดพลาด คำนวณได้ ดังนี้

$$e_1 = 1 - 1 = 0$$

เนื่องจาก $a_2 > t$ ดังนั้น ข้อมูลออก $o_2 = 1$ แล้ว ความผิดพลาด คำนวณได้ ดังนี้

$$e_2 = 0 - 1 = -1$$

4. ปรับปรุงค่าน้ำหนัก ได้ดังนี้

$w_{11} = 0.5 + 0.5 \times 0 \times 0 = 0.5$; $w_{12} = 0.5 + 0.5 \times 0 \times (-1) = 0.5$

$w_{21} = 0.5 + 0.5 \times 0 \times 0 = 0.5$; $w_{22} = 0.5 + 0.5 \times 0 \times (-1) = 0.5$

$w_{31} = 0.5 + 0.5 \times 1 \times 0 = 0.5$; $w_{32} = 0.5 + 0.5 \times 1 \times (-1) = \underline{0.0}$

$w_{41} = 0.5 + 0.5 \times 0 \times 0 = 0.5$; $w_{42} = 0.0 + 0.5 \times 0 \times (-1) = 0.0$

5. พิจารณาแบบที่ 3 การกระตุ้น มีค่า ดังนี้

$a_1 = 0.5\times0 + 0.5\times1 + 0.5\times0 + 0.5\times0 = 0.5$

$a_2 = 0.5\times0 + 0.5\times1 + 0.0\times0 + 0.0\times0 = 0.5$

เนื่องจาก $a_1 > t$ ดังนั้น ข้อมูลออก $o_1 = 1$ แล้ว ความผิดพลาด คำนวณได้ ดังนี้

$e_1 = 0 - 1 = -1$

เนื่องจาก $a_2 > t$ ดังนั้น ข้อมูลออก $o_2 = 1$ แล้ว ความผิดพลาด คำนวณได้ ดังนี้

$e_2 = 1 - 1 = 0$

6. ปรับปรุงค่าน้ำหนัก ได้ดังนี้

$w_{11} = 0.5 + 0.5 \times 0 \times (-1) = 0.5;$ $w_{12} = 0.5 + 0.5 \times 0 \times 0 = 0.5$

$w_{21} = 0.5 + 0.5 \times 1 \times (-1) = \underline{0.0};$ $w_{22} = 0.5 + 0.5 \times 1 \times 0 = 0.5$

$w_{31} = 0.5 + 0.5 \times 0 \times (-1) = 0.5;$ $w_{32} = 0.0 + 0.5 \times 0 \times 0 = 0.0$

$w_{41} = 0.5 + 0.5 \times 0 \times (-1) = 0.5;$ $w_{42} = 0.0 + 0.5 \times 0 \times 0 = 0.0$

7. พิจารณาแบบที่ 4 การกระตุ้น มีค่า ดังนี้

$a_1 = 0.5\times1 + 0.0\times0 + 0.5\times0 + 0.5\times0 = 0.5$

$a_2 = 0.5\times1 + 0.5\times0 + 0.0\times0 + 0.0\times0 = 0.5$

เนื่องจาก $a_1 > t$ ดังนั้น ข้อมูลออก $o_1 = 1$ แล้ว ความผิดพลาด คำนวณได้ ดังนี้

$e_1 = 0 - 1 = -1$

เนื่องจาก $a_2 > t$ ดังนั้น ข้อมูลออก $o_2 = 1$ แล้ว ความผิดพลาด คำนวณได้ ดังนี้

$e_2 = 1 - 1 = 0$

8. ปรับปรุงค่าน้ำหนัก ได้ดังนี้

$w_{11} = 0.5 + 0.5 \times 1 \times (-1) = \underline{0.0};$ $w_{12} = 0.5 + 0.5 \times 1 \times 0 = 0.5$

$w_{21} = 0.0 + 0.5 \times 0 \times (-1) = 0.0;$ $w_{22} = 0.5 + 0.5 \times 0 \times 0 = 0.5$

$w_{31} = 0.5 + 0.5 \times 0 \times (-1) = 0.5;$ $w_{32} = 0.0 + 0.5 \times 0 \times 0 = 0.0$

$w_{41} = 0.5 + 0.5 \times 0 \times (-1) = 0.5;$ $w_{42} = 0.0 + 0.5 \times 0 \times 0 = 0.0$

9. พิจารณาแบบที่ 1 การกระตุ้น คำนวณได้ ดังนี้

$$a_1 = 0.0\times0 + 0.0\times0 + 0.5\times0 + 0.5\times1 = 0.5$$

$$a_2 = 0.5\times0 + 0.5\times0 + 0.0\times0 + 0.0\times1 = 0.0$$

เนื่องจาก $a_1 > t$ ดังนั้น ข้อมูลออก $o_1 = 1$ แล้ว ความผิดพลาด คำนวณได้ ดังนี้

$$e_1 = 1 - 1 = 0$$

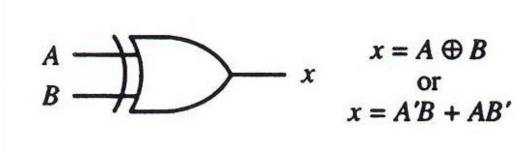
เนื่องจาก $a_2 \leq t$ ดังนั้น ข้อมูลออก $o_2 = 0$ แล้ว ความผิดพลาด คำนวณได้ ดังนี้

$$e_2 = 0 - 0 = 0$$

จากค่าน้ำหนักล่าสุด แบบทั้งหมดถูกจำแนกได้อย่างถูกต้อง เนื่องจาก $e_1 = e_2 = 0$ ดังนั้น น้ำหนักทั้งหมดจะไม่ถูกปรับปรุงค่า และ การเรียนรู้ของตัวแบบจะสิ้นสุดลง

โครงข่ายประสาทในหัวข้อนี้สามารถใช้งานได้ในเซตข้อมูลที่จำแนกเชิงเส้นได้เท่านั้น อย่างไรก็ตาม ในกรณีที่ชั้นข้อมูลไม่สามารถจำแนกเชิงเส้นได้ ตัวแบบชนิดนี้จะไม่สามารถสร้างขอบเขตการตัดสินใจไม่เชิงเส้นได้ ตามบทพิสูจน์ในตัวอย่างถัดไป

ตัวอย่างที่ 11 พิจารณาลอจิกเกตพื้นฐาน Exclusive OR (XOR) ตามรูปที่ 6.19 ซึ่งมีตารางค่าความจริงตามตารางที่ 6.23 โดยที่ โครงข่ายการป้อนไปข้างหน้าประกอบไปด้วย 2 ข้อมูลเข้า (d = 2) และ 1 ข้อมูลออก (m = 1) จงพิสูจน์ให้เห็นว่าเซตข้อมูลนี้ไม่สามารถจำแนกเชิงเส้นได้ โดยใช้ การพิสูจน์โดยข้อขัดแย้ง (Contradiction Proof)



รูป Exclusive OR Gate

ตารางที่ 6.23 เซตข้อมูล Exclusive OR

A	B	x
0	0	–1
0	1	1
1	0	1
1	1	–1

พิสูจน์

สมมติว่ามีขีดแบ่ง c ตามเงื่อนไขด้านล่าง โดยที่ $w_1 = a$ และ $w_2 = b$ เมื่อ a และ b เป็น จำนวนจริงใด ๆ

$ax_1 + bx_2 > c$ สำหรับชั้นข้อมูล 1

$ax_1 + bx_2 \leq c$ สำหรับชั้นข้อมูล 0

ดังนั้น แบบทั้งสี่ในตารางค่าความจริง มีความเกี่ยวข้องกับอสมการทั้งสี่ ดังต่อไปนี้

$a \cdot 0 + b \cdot 0 \leq c$ *{อสมการ 6.1}*

$a \cdot 0 + b \cdot 1 > c$ *{อสมการ 6.2}*

$a \cdot 1 + b \cdot 0 > c$ *{อสมการ 6.3}*

$a \cdot 1 + b \cdot 1 \leq c$ *{อสมการ 6.4}*

จาก อสมการ 6.1 ได้ค่า $c \geq 0$ ดังนั้น c มีค่าไม่ติดลบ

จาก อสมการ 6.2 ได้ค่า $c < b$ ดังนั้น c มีค่าน้อยกว่า b

จาก อสมการ 6.3 ได้ค่า $c < a$ ดังนั้น c มีค่าน้อยกว่า a

จาก อสมการ 6.4 ได้ค่า $c \geq a + b$ ดังนั้น c มีค่ามากกว่า a + b

เนื่องจาก c มีค่าน้อยกว่า a และ b จากอสมการที่ 6.2 และ 6.3 ดังนั้น c ต้องมีค่าน้อยกว่าผลรวมของ a และ b ซึ่งทำให้กลุ่มอสมการขัดแย้งกันเอง สิ่งที่สมมติไว้ตอนแรกจึงไม่เป็นจริง พิสูจน์ได้ว่า ไม่สามารถหาค่าขีดแบ่ง c ที่แบ่งแยกชั้นข้อมูลออกจากกันได้ หรือ ชั้นข้อมูลไม่สามารถจำแนกเชิงเส้นได้ □

6. เครื่องกลเวกเตอร์สนับสนุน

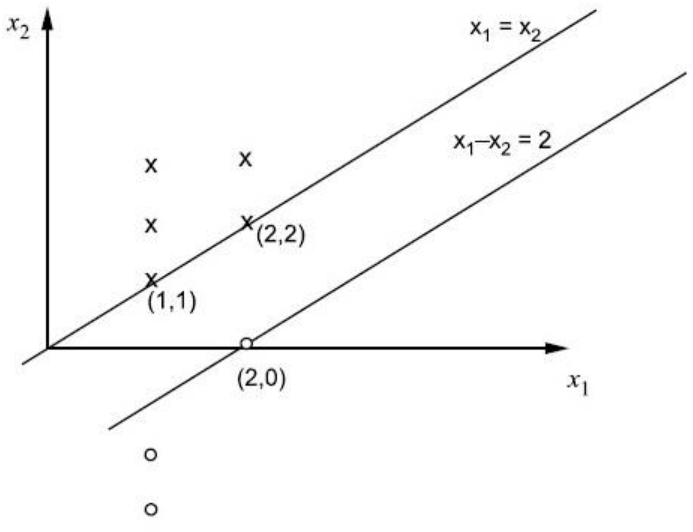
เครื่องกลเวกเตอร์สนับสนุน หรือ เอสวีเอ็ม (SVM: Support Vector Machine) เป็นตัวจำแนกที่สร้างขอบเขตการตัดสินใจบนปริภูมิหลายมิติ ในทางเรขาคณิต เวกเตอร์สนับสนุน (Support Vector) คือ เซตของแบบฝึกฝนที่ใกล้ขอบเขตการตัดสินใจมากที่สุด การศึกษาพฤติกรรมของเครื่องกลเวกเตอร์สนับสนุน แสดงตามตัวอย่างด้านล่างนี้

ตัวอย่างที่ 12 พิจารณาแบบ 2 มิติ ทั้ง 3 ในรูปที่ 6.20 ได้แก่ (1, 1) ; (2, 2) จากชั้นข้อมูล X และ (2, 0) จากชั้นข้อมูล O โดยที่ เส้นตรง $x_1 - x_2 = 0$ และ $x_1 - x_2 = 2$ มีคุณสมบัติเป็นขอบเขตของชั้นข้อมูล X และ O ตามลำดับ เส้นตรงทั้งหมดนี้ ถูกเรียกว่า เส้นตรงสนับสนุน (Support Line) และ แบบทั้งหมดนี้ ถูกเรียกว่า เวกเตอร์สนับสนุน (Support Vector)

พิจารณาแบบเพิ่มเติม ได้แก่ (1, 2) ; (1, 3) ; (2, 3) จากชั้นข้อมูล X และ (2, –1) ; (1, –2) ; (1, –3) จากชั้นข้อมูล O ซึ่งแบบเหล่านี้สามารถถูกจำแนกได้อย่างถูกต้องด้วยเส้นตรงสนับสนุน กล่าวคือ แบบจากชั้นข้อมูล X เป็นไปตามเงื่อนไข $x_1 - x_2 < 0$ ในทำนองเดียวกัน แบบจากชั้นข้อมูล O เป็นไปตามเงื่อนไข $x_1 - x_2 > 2$ ระยะห่างระหว่างเส้นตรงสนับสนุน 2 เส้นนี้เรียกว่า มาร์จิ้น (Margin) เนื่องจากเส้นตรงสนับสนุนต้องถูกสร้างโดยให้ระยะห่างนี้มีค่าสูงที่สุด ดังนั้นเส้นตรงสนับสนุนต้องอยู่ห่างไกลกันให้มากที่สุดเท่าที่จะเป็นไปได้

เส้นตรง $x_1 - x_2 = 1$ ห่างจากเส้นตรงการตัดสินใจทั้ง 2 เส้น ข้างต้นเท่ากัน และ สามารถใช้เป็นขอบเขตการตัดสินใจระหว่าง 2 ชั้นข้อมูลได้ ซึ่งแบบที่เป็นไปตามเงื่อนไข $x_1 - x_2 < 1$ จะอยู่ในชั้นข้อมูล X และ แบบที่เป็นไปตามเงื่อนไข $x_1 - x_2 > 1$ จะอยู่ในชั้นข้อมูล O การอภิปรายข้างต้นได้ข้อสังเกตดังนี้

1. เครื่องกลเวกเตอร์สนับสนุน คือตัวจำแนกทวิภาคที่สร้างขอบเขตการตัดสินใจเชิงเส้น สำหรับจำแนกแบบจากชั้นข้อมูลทั้งสอง
2. เวกเตอร์สนับสนุน คือเวกเตอร์บางส่วนจากเวกเตอร์ที่อยู่บน ระนาบสนับสนุน (Support Plane) ในปริภูมิ d มิติ ($d \geq 3$) หรืออยู่บน เส้นตรงสนับสนุน (Support Line) ในปริภูมิ 2 มิติ ($d = 2$)
3. เครื่องกลเวกเตอร์สนับสนุน เรียนรู้แบบฝึกฝนแล้วสร้างการจำแนกเชิงเส้นในรูปแบบ $w^t + b$ (w คือ น้ำหนักของเวกเตอร์ และ b คือ ค่าของขีดแบ่ง) โดยพยายามให้มาร์จิ้นมีค่าสูงที่สุด ซึ่งแตกต่างจากการเรียนรู้เพอร์เซ็ปตรอนที่ต้องการเพียงแบ่งชั้นข้อมูลออกจากกันเท่านั้น
4. ถ้าแบบในชั้นข้อมูลสามารถถูกจำแนกเชิงเส้นได้ แล้วจะสามารถหาขอบที่มีค่ามากที่สุดได้



รูปที่ 6.20 เวกเตอร์สนับสนุนและเส้นตรงสนับสนุน

กรณีจำแนกเชิงเส้น

พิจารณาตัวอย่างจากแบบ 2 มิติ จากตัวอย่างต่อไปนี้

ตัวอย่างที่ 13 พิจารณาแบบ (1, 1) ; (2, 2) จากชั้นข้อมูล X และ (2, 0) จากชั้นข้อมูล O ในรูปที่ 6.19 อีกครั้ง เนื่องจากชั้นข้อมูลทั้งสองนี้สามารถจำแนกเชิงเส้นได้ ดังนั้น จึงมีจำนวนของเส้นตรงเป็นอนันต์ที่สามารถแบ่งแยกชั้นข้อมูลเหล่านี้ออกจากกันได้

ถ้าใช้การเรียนรู้เพอร์เซ็ปตรอน ขอบเขตการตัดสินใจที่เรียนรู้ได้คือเส้นตรง $x_1 - 3x_2 - 1 = 0$ (อ้างอิงถึง ตัวอย่างที่ 9) นอกจากเส้นตรงนี้แล้ว เส้นตรง $x_1 - 2x_2 = 0$ และ $x_1 - x_2 = 1$ ก็สามารถแบ่งแยกชั้นข้อมูลเหล่านี้ได้เช่นกัน จากนี้ไปจะมีการเพิ่มข้อจำกัดสำหรับขอบเขตการตัดสินใจเชิงเส้น โดยเส้นตรงเส้นนี้ต้องถูกยึดอยู่กับที่ด้วยเงื่อนไขบางอย่าง

สำหรับ SVM แล้ว เส้นตรงที่ทำให้มาร์จิ้นมีค่าสูงที่สุดจะถูกเลือกเป็นขอบเขตการตัดสินใจ โดยทั่วไปแล้ว เส้นตรงเส้นนี้จะถูกเรียนรู้จากตัวแบบ ถ้าชั้นข้อมูลสามารถจำแนกเชิงเส้นได้ ยกตัวอย่างจากแบบทั้งสามในตัวอย่างนี้

- ❖ แบบ (1, 1) และ (2, 2) จากชั้นข้อมูล X อยู่บนเส้นตรงสนับสนุน $x_1 - x_2 = 0$
- ❖ แบบ (2, 0) จากชั้นข้อมูล O อยู่บนเส้นตรงสนับสนุน $x_1 - x_2 = 2$

เส้นตรงสนับสนุนสองเส้นนี้ก่อให้เกิดมาร์จิ้น ดังนั้นขอบเขตการตัดสินใจที่อยู่กึ่งกลางระหว่างเส้นตรงสองเส้นนี้ ได้แก่ เส้นตรง $x_1 - x_2 = 1$

เนื่องจากแบบ (1, 0) และ (2, 1) จากชั้นข้อมูล X บนเส้นตรง $x_1 - x_2 = 1$ ซึ่งเป็นขอบเขตการตัดสินใจ ดังนั้น ตัวจำแนกเชิงเส้นสามารถอนุพันธ์ให้อยู่ในรูปของ $f(x) = w^t x + b$ จากการพิจารณาเวกเตอร์น้ำหนัก $w^t = \binom{w_1}{w_2}$ ที่เกี่ยวข้องกับแต่ละแบบ ดังต่อไปนี้

❖ **แบบ (1, 0)**

$$w^t x + b = \binom{w_1}{w_2} \cdot \binom{1}{0} + b = w_1 + b$$

ดังนั้น

$$w_1 + b = 0 \qquad \textit{\{สมการที่ 6.1\}}$$

❖ **แบบ (2, 1)**

$$w^t x + b = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 1 \end{pmatrix} = 2w_1 + w_2 + b$$

ดังนั้น

$$2w_1 + w_2 + b = 0 \qquad \textit{\{สมการที่ 6.2\}}$$

{สมการที่ 6.2} – {สมการที่ 6.1}

$(2w_1 + w_2 + b) – (w_1 + b) = 0$

$w_1 + w_2 = 0$

เนื่องจาก $w_1 = –w_2$ ดังนั้น

$$w^t = \begin{pmatrix} a \\ -a \end{pmatrix}$$

แทนค่า w_1 ลงใน *{สมการที่ 6.1}* ได้ดังนี้

$a = –b$

โดยทั่วไป a เป็นค่าคงที่ ยกตัวอย่างการเลือกค่า a = 1 บนเส้นตรงแต่ละเส้น ดังนี้

1. เส้นตรง $x_1 – x_2 = 1$ สามารถเขียนให้อยู่ในรูปของ $w^t x + b$ เมื่อ a = 1 ได้ดังนี้

$$\begin{aligned} x_1 - x_2 &= 1 \\ (x_1 - x_2) + (-1) &= 1 - 1 \\ \begin{pmatrix} 1 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + b &= 0 \end{aligned}$$

ดังนั้น

$$w^t x + b = 0$$

2. เส้นตรง $x_1 – x_2 = 2$ สามารถเขียนให้อยู่ในรูปของ $w^t x + b$ เมื่อ a = 1 ได้ดังนี้

$$\begin{aligned} x_1 - x_2 &= 2 \\ (x_1 - x_2) + (-1) &= 2 - 1 \end{aligned}$$

$\begin{pmatrix} 1 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + b \quad = \quad 1$

ดังนั้น

$w^t x + b = 1$

3. เส้นตรง $x_1 - x_2 = 0$ สามารถเขียนให้อยู่ในรูปของ $w^t x + b$ เมื่อ $a = 1$ ได้ดังนี้

$x_1 - x_2 \quad = \quad 0$

$(x_1 - x_2) + (-1) \quad = \quad 0 - 1$

$\begin{pmatrix} 1 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + b \quad = \quad -1$

ดังนั้น

$w^t x + b = -1$

จากเส้นตรงทั้งสามเส้นข้างต้น แสดงตามรูปที่ 6.21 และสรุปได้ดังนี้

$w^t x + b = 0$	สำหรับแบบจากขอบเขตการตัดสินใจ	$x_1 - x_2 = 1$
$w^t x + b = 1$	สำหรับแบบจากเส้นตรงสนับสนุน	$x_1 - x_2 = 2$
$w^t x + b = -1$	สำหรับแบบจากเส้นตรงสนับสนุน	$x_1 - x_2 = 0$

เส้นตรงทั้งหมดนี้ มาจากการให้ค่า $a = 1$ ดังนั้น

$w^t = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

$b = -1$

พิจารณาการคำนวณระยะทางปกติระหว่างจุดใด ๆ บนระนาบหลายมิติ $w^t x + b = 1$ และ ขอบเขตการตัดสินใจ $w^t x + b = 0$ ในปริภูมิ d มิติ โดยใช้สมการการคำนวณระยะทางปกติจาก ตัวอย่างที่ 5 ดังนี้

$$\frac{w^t x + b}{\|w^t\|}$$

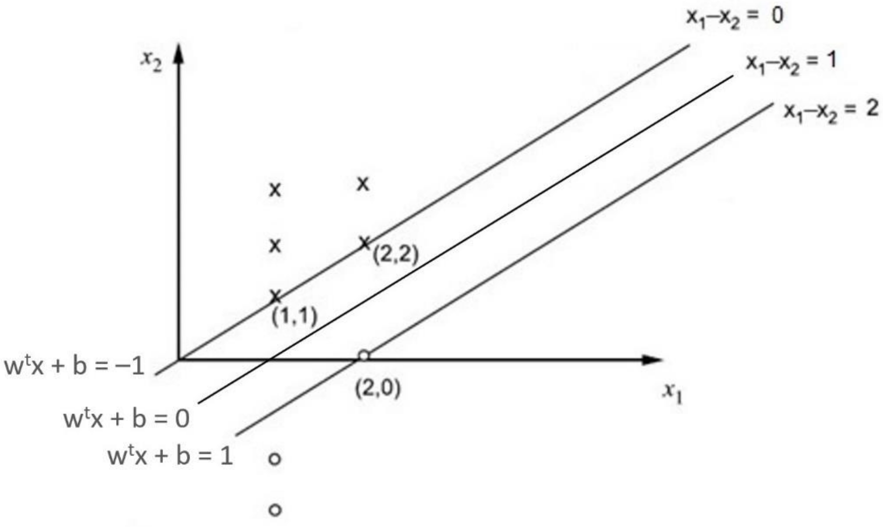
ด้วยตำแหน่งของ x จะได้ว่า $w^t x + b$ ในสมการข้างต้นมีค่าเท่ากับ 1

ดังนั้น ระยะทางปกติระหว่าง $w^t x + b = 1$ และ $w^t x + b = 0$ คือ $\frac{1}{\|w^t\|}$

ในทำนองเดียวกัน ระยะทางปกติ ระหว่าง $w^tx + b = -1$ และ $w^tx + b = 0$ คือ $\frac{1}{\|w^t\|}$

ท้ายที่สุดแล้ว ระยะทางปกติ ระหว่าง $w^tx + b = 1$ และ $w^tx + b = -1$ คือ $\frac{2}{\|w^t\|}$

ค่านี้เรียกว่า มาร์จิ้น สำหรับรายละเอียดเพิ่มเติมจะกล่าวถึงในตัวอย่างถัดไป



รูปที่ 6.21 เวกเตอร์สนับสนุนและเส้นตรงสนับสนุน

ตัวอย่างที่ 14 พิจารณาปัญหาการจำแนกเชิงเส้นในตัวอย่างที่ผ่านมา เนื่องจากระยะทางระหว่างเส้นตรงสนับสนุนทั้ง 2 เส้น มีค่าเท่ากับ $\frac{2}{\|w^t\|}$ เมื่อ $w^t = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ ดังนั้น มาร์จิ้น (p) คำนวณได้ตามสมการด้านล่างนี้

$$p = \frac{2}{\|w^t\|}$$

$$= \frac{2}{\sqrt{1^2+(-1)^2}}$$

$$= \sqrt{2}$$

มาร์จิ้นมีค่าสูงสุดก็ต่อเมื่อฟังก์ชัน $\frac{\|w^t\|}{2}$ มีค่าต่ำสุด เนื่องจากแบบในแต่ละชั้นข้อมูลมีเงื่อนไขที่เกี่ยวข้องกับค่า w^t ดังนี้

$w^tx + b \leq -1$ สำหรับแบบจากชั้นข้อมูล X

$w^tx + b \geq 1$ สำหรับแบบจากชั้นข้อมูล O

ปัญหานี้จัดเป็น ปัญหาการหาค่าเหมาะสมที่สุด (Optimization Problem) โดยวิธีการแก้ปัญหาจะกล่าวถึงในตัวอย่างถัดไป

ตัวอย่างที่ 15 พิจารณแบบ 1 มิติ ในตารางที่ 2.4 สืบเนื่องจากตัวอย่างก่อนหน้า ฟังก์ชัน $\frac{\|w^t\|}{2}$ ต้องมีค่าต่ำสุด จึงพิจารณาขอบเขตการตัดสินใจสำหรับแต่ละแบบ ดังต่อไปนี้

ตารางที่ 6.24 SVM สำหรับข้อมูล 1 มิติ

Pattern No.	x	Class
1	1	X
2	2	X
3	4	O

$w^t x + b \le -1$ สำหรับแบบ x = 1 จากชั้นข้อมูล X

$2w^t x + b \le -1$ สำหรับแบบ x = 2 จากชั้นข้อมูล X

$4w^t x + b \ge 1$ สำหรับแบบ x = 4 จากชั้นข้อมูล O

ในตัวอย่างนี้ได้ใช้ ฟังก์ชันลากรองจ์ (Lagrange Function) เพื่อแก้ปัญหาการหาค่าเหมาะสมที่สุด ตามขั้นตอน ดังต่อไปนี้

หาค่าต่ำสุด $\frac{\|w^t\|}{2}$

ภายใต้เงื่อนไข $w^t x + b + 1 \le 0$

$2w^t x + b + 1 \le 0$

$-4w^t x - b + 1 \le 0$

สร้างฟังก์ชันใหม่ เมื่อ α_1, α_2, α_3 คือ ตัวคูณลากรองจ์ มีทั้งหมด 3 ตัว ๆ ละ 1 เงื่อนไข ได้ดังนี้

$$J(w) = \frac{w^2}{2} - \alpha_1(-w - b - 1) - \alpha_2(-2w - b - 1) - \alpha_3(4w + b - 1)$$ *{สมการ 6ก}*

จากนั้นทำการแก้สมการเพื่อหาค่าตัวแปรต่าง ๆ โดยการคำนวณหา อนุพันธ์ย่อย (Partial Derivative) ของฟังก์ชัน *J*(*w*) เมื่อตัวแปรอิสระคือ w และ b ตามสมการต่อไปนี้

$\frac{\delta j}{\delta w} = 0$

$w = -\alpha_1 - 2\alpha_2 + 4\alpha_3$ *{สมการ 6ข}*

$\frac{\delta j}{\delta b} = 0$

$0 = \alpha_1 + \alpha_2 - \alpha_3$ *{สมการ 6ค}*

ในทำนองเดียวกัน หาอนุพันธ์ย่อยของฟังก์ชัน *J*(*w*) เมื่อตัวแปรอิสระคือ α_1, α_2, α_3 ตามสมการต่อไปนี้

–w – b – 1 = 0 *{สมการ 6ง}*

–2w – b – 1 = 0 *{สมการ 6จ}*

4w + b – 1 = 0 *{สมการ 6ฉ}*

เนื่องจาก สมการ 6ง และ 6จ ขัดแย้งกันเอง จึงเลือก สมการ 6จ และ 6ฉ เพื่อแก้สมการ ได้ค่าดังนี้

w = 1 และ b = –3

จากเงื่อนไขของการหาอนุพันธ์ย่อยแล้ว ค่าของ $\alpha_1(-w-b-1)$ ต้องมีนิพจน์ใดนิพจน์หนึ่งเป็น ศูนย์ ดังนั้น เลือกให้นิพจน์แรกเป็น 0 ดังนั้นจะได้ความสัมพันธ์ ดังต่อไปนี้

$-w - b - 1 \neq 0$

$\alpha_1 = 0$

แทนค่า α_1 ลงในสมการ 6ค ดังนั้น

$\alpha_2 = \alpha_3$

แทนค่า *w* และ α_2 และ ลงในสมการ 6ข ดังนั้น

$1 = -2\alpha_3 + 4\alpha_3$

$\alpha_3 = \frac{1}{2}$

$\alpha_2 = \frac{1}{2}$

เนื่องจาก ขอบเขตการตัดสินใจอยู่ในรูปแบบของ wx + b ดังนั้น เส้นตรงที่แบ่งแยกชั้นข้อมูล X และ O ออกจากกัน ได้แก่

x – 3 = 0 หรือ x = 3

ถึงแม้ว่าตัวอย่างที่ผ่านมา แบบมีเพียง 1 มิติ อย่างไรก็ตาม วิธีการแก้ปัญหานี้เป็นวิธีการแก้ปัญหาทั่วไป ที่สามารถนำไปใช้ประยุกต์ใช้กับแบบ 2 มิติ ได้ ซึ่งวิธีทำจะแสดงให้เห็นในตัวอย่างถัดไป

ตัวอย่างที่ 16 พิจารณแบบ 2 มิติ จากตัวอย่างที่ 13 ซึ่งประกอบไปด้วย แบบ (1, 1) และ (2, 2) จากชั้นข้อมูล X และ (2, 0) จากชั้นข้อมูล O กำหนดให้ $w^t = \binom{w_1}{w_2}$ เป็นเวกเตอร์ 2 มิติ ดังนั้น ฟังก์ชันจุดประสงค์ และเงื่อนไข คือ

หาค่าต่ำสุด $\frac{\|w^t\|}{2}$

ภายใต้เงื่อนไข $w^tx + b + 1 \leq 0$ $\forall$x ใน X

$w^tx + b + 1 \geq 0$ $\forall$x ใน O

จากเงื่อนไขแรก เมื่อกำหนดให้ x คือ (1, 1) และ (2, 2) จะได้เงื่อนไขที่เทียบเท่ากัน ได้แก่

$w_1 + w_2 + b \leq -1$

$2w_1 + 2w_2 + b \leq -1$

จากเงื่อนไขแรก เมื่อกำหนดให้ x คือ (2, 0) จะได้เงื่อนไขที่เทียบเท่ากัน ได้แก่

$-2w_1 - b \leq -1$

สร้างฟังก์ชันใหม่ จากตัวคูณลากรองจ์ ได้ดังนี้

$$J(w) = \frac{\|w\|^2}{2} - \alpha_1(-w_1 - w_2 - b - 1) - \alpha_2(-2w_1 - 2w_2 - b - 1) - \alpha_3(-2w_1 + b - 1) \qquad \text{\{สมการ 6a\}}$$

หรือ

$$J(w) = \frac{\|w\|^2}{2} - \alpha_1\left(w^t \cdot \binom{-1}{-1} - b - 1\right) - \alpha_2\left(w^t \cdot \binom{2}{2} - b - 1\right) - \alpha_3\left(w^t \cdot \binom{2}{0} + b - 1\right) \qquad \text{\{สมการ 6A\}}$$

คำนวณหา อนุพันธ์ย่อย (Partial Derivative) ของฟังก์ชัน J(w) เมื่อตัวแปรอิสระคือ w ได้ดังนี้ และแบบ (1, 1) ; (2, 2) และ (2, 0) คือ เวกเตอร์สนับสนุน

$$w^t + \alpha_1\binom{1}{1} + \alpha_2\binom{2}{2} - \alpha_3\binom{2}{0} = 0 \qquad \text{\{สมการ 6b\}}$$

หรือ

$$\begin{pmatrix} w_1 & +\alpha_1 + 2\alpha_2 - 2\alpha_3 \\ w_2 & +\alpha_1 + 2\alpha_2 + 0\alpha_3 \end{pmatrix} = 0$$ *{สมการ 6A}*

จากสมการข้างต้น

$$w_1 = -\alpha_1 - 2\alpha_2 + 2\alpha_3$$ *{สมการ 6c}*

$$w_2 = -\alpha_1 - 2\alpha_2$$ *{สมการ 6d}*

คำนวณหา อนุพันธ์ย่อย (Partial Derivative) ของฟังก์ชัน J(w) เมื่อตัวแปรอิสระคือ b ได้ดังนี้

$$\alpha_1 + \alpha_2 - \alpha_3 = 0$$ *{สมการ 6e}*

คำนวณหา อนุพันธ์ย่อย (Partial Derivative) ของฟังก์ชัน J(w) เมื่อตัวแปรอิสระคือ α_1, α_2, α_3 ได้ดังนี้

$$-w_1 - w_2 - b - 1 = 0$$ *{สมการ 6f}*

$$-2w_1 - 2w_2 - b - 1 = 0$$ *{สมการ 6g}*

$$2w_1 + b - 1 = 0$$ *{สมการ 6h}*

จากสมการ 6g และ 6h ได้ความสัมพันธ์ของ w_2 ดังนี้

$$-2w_2 - 2 = 0$$

$$w_2 = -1$$

จากสมการ 6f และ 6h ได้ความสัมพันธ์ของ w_1 ดังนี้

$$w_1 + w_2 = 0$$

$$w_1 = 1$$

จากสมการ 6h ได้ค่า b ดังนี้

$$b = -1$$

ดังนั้น ขอบเขตการตัดสินใจซึ่งอยู่ในรูปแบบของ wx + b คือ

$$\begin{pmatrix} 1 \\ -1 \end{pmatrix} - 1$$

จากสมการ 6c 6d 6e แก้สมการหาค่า ได้ดังต่อไปนี้

$$\alpha_1 = 1$$

$$\alpha_2 = 0$$

$$\alpha_3 = 1$$

สำหรับรูปแบบทั่วไปของเวกเตอร์น้ำหนัก เป็นไปตามความสัมพันธ์ด้านล่างนี้ เมื่อ x_i คือ เวกเตอร์สนับสนุนตัวที่ i

$$w = \sum_i \alpha_i y_i x_i$$

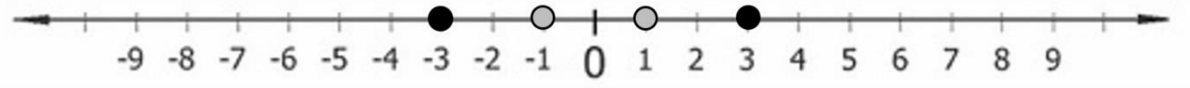
- y_i = +1 สำหรับแบบฝั่งบวก (ชั้นข้อมูล O)
- y_i = –1 สำหรับแบบฝั่งลบ (ชั้นข้อมูล X)

กรณีจำแนกไม่เชิงเส้น

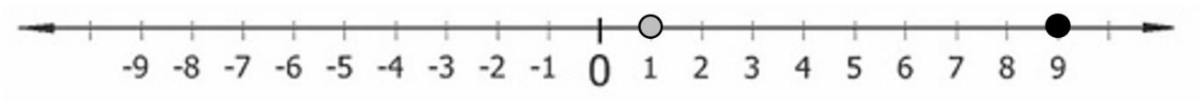
ตัวอย่างที่ 16 พิจารณากลุ่มของแบบ 1 มิติ ที่ประกอบไปด้วย แบบ –3 และ 3 จากชั้นข้อมูล O และแบบ –1 และ 1 จากชั้นข้อมูล X ชั้นข้อมูลเหล่านี้ไม่สามารถจำแนกเชิงเส้นได้ อย่างไรก็ตาม ถ้าแบบเหล่านี้ผ่าน การแปลง (Mapping) ด้วยฟังก์ชัน $f(x) = x^2$ พบว่า ข้อมูลออกที่ได้คือ 9 และ 9 จากชั้นข้อมูล O และ แบบ 1 และ 1 จากชั้นข้อมูล X ซึ่งแบบใหม่เหล่านี้สามารถจำแนกเชิงเส้นได้

หรืออีกหนึ่งวิธีคือใช้การแปลงแบบจากหนึ่งมิติไปสองมิติใช้การแปลงด้วยฟังก์ชัน $f(x) = (x, x^2)$ ข้อมูลออก คือ (–3, 9) และ (3, 9) สำหรับชั้นข้อมูล X และ (–1, 1) และ (1, 1) สำหรับชั้นข้อมูล O ซึ่งสามารถจำแนกเชิงเส้นได้ จากทั้งสองวิธีที่กล่าวมาข้างต้นนี้ แสดงให้เห็นด้วยรูปที่ 6.21 ถึง 6.23 เมื่อวงกลมสีดำแทนชั้นข้อมูล O และ วงกลมสีเทาแทนชั้นข้อมูล X

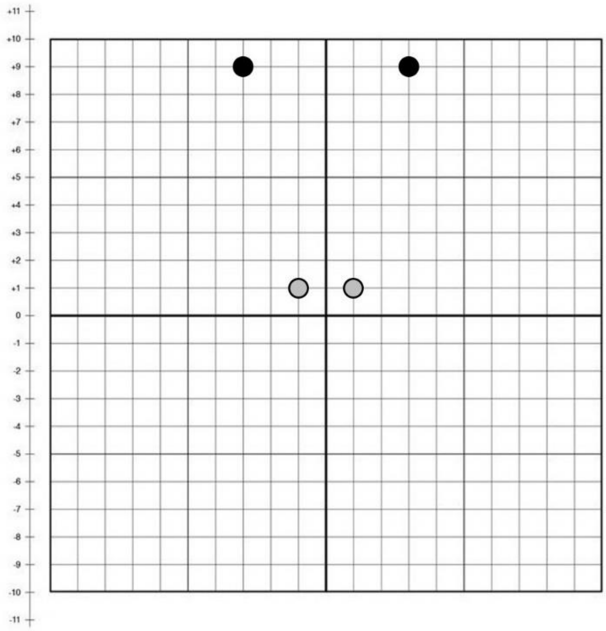
นิยามของ ผลคูณแบบดอท (Dot Product) ของสองเวกเตอร์ คือ $\binom{p_1}{p_2} \cdot \binom{q_1}{q_2} = p_1 q_1 + p_2 q_2$ โดยทั่วไปแล้ว แบบใน d มิติ อาจถูกแปลงเป็น แบบใน D มิติ (D > d) เพื่อจุดมุ่งหมายในการทำให้แบบสามารถจำแนกเชิงเส้นได้ ดังที่จะกล่าวถึงในตัวอย่างถัดไป



รูปที่ 6.21 แบบ 1 มิติ



รูปที่ 6.22 แบบ 1 มิติ ที่ผ่านการแปลงจากฟังก์ชัน $f(x) = x^2$



รูปที่ 6.23 แบบ 1 มิติ ที่ผ่านการแปลงจากฟังก์ชัน $f(x) = (x, x^2)$ กลายเป็น แบบ 2 มิติ

ตัวอย่างที่ 17 พิจารณาตารางค่าความจริงของฟังก์ชัน f ในตารางที่ 6.25 จงพิสูจน์ว่าชั้นข้อมูล X และ O ไม่สามารถจำแนกเชิงเส้นได้ เมื่อแต่ละค่าของฟังก์ชันแทนชั้นข้อมูล ดังต่อไปนี้

$f(x_1, x_2) = 0$ แทน ชั้นข้อมูล X

$f(x_1, x_2) = 1$ แทน ชั้นข้อมูล O

ตารางที่ 6.25 ตารางค่าความจริงสำหรับ $f(x_1, x_2)$

x_1	x2	$f(x_1, x_2)$
0	0	1
0	1	0
1	0	0
1	1	1

พิสูจน์

สมมติไปก่อนว่าแบบทั้งสองสามารถจำแนกเชิงเส้นได้ แสดงว่ามีเส้นตรง $\alpha_1 x_1 + \alpha_2 x_2 + c = 0$ ซึ่งแบ่งแยกแบบจากชั้นข้อมูล X และ O ออกจากกัน หรือกล่าวแบบเจาะจงได้ ดังนี้

$\alpha_1 x_1 + \alpha_2 x_2 + c > 0$ สำหรับแบบ (x_1, x_2) จากชั้นข้อมูล O ได้แก่แบบ 1 และ 4

$\alpha_1 x_1 + \alpha_2 x_2 + c < 0$ สำหรับแบบ (x_1, x_2) จากชั้นข้อมูล X ได้แก่แบบ 2 และ 3

แบบ 1 ถึง แบบ 4 มีอสมการที่เกี่ยวข้อง ตามลำดับ ดังต่อไปนี้

$$c > 0$$

$$\alpha_1 + \alpha_2 + c > 0$$

$$\alpha_2 + c < 0$$

$$\alpha_1 + c < 0$$

นำ 2 อสมการแรกบวกกัน ได้ดังนี้

$$\alpha_1 + \alpha_2 + 2c > 0$$

นำ 2 อสมการหลังบวกกัน ได้ดังนี้

$$\alpha_1 + \alpha_2 + c > 0$$

พบว่า สิ่งที่สมมติไว้ตอนแรกไม่เป็นจริง ดังนั้น ชั้นข้อมูลเหล่านี้ไม่สามารถจำแนกเชิงเส้นได้ □

อย่างไรก็ตาม ถึงแม้ว่าจะไม่สามารถหาเส้นตรงที่แบ่งแยกชั้นข้อมูลออกจากกันได้ แต่จากการขยายมิติของแบบจาก 2 มิติ เป็น 3 มิติ ส่งผลให้ชั้นข้อมูลทั้งสองสามารถจำแนกเชิงเส้นได้ แสดงให้เห็นตามตัวอย่างสุดท้ายนี้

ตัวอย่างที่ 18 จากตารางค่าความจริงของฟังก์ชัน f ในตารางที่ 6.25 ทำการเพิ่มฟีเจอร์ $x_1 \wedge x_2$ เพื่อขยายมิติของแบบจาก 2 มิติ เป็น 3 มิติ จะได้ผลลัพธ์เป็นแบบตามตารางที่ 6.25 ซึ่งสามารถจำแนกชั้นข้อมูลออกจากกันได้

ตารางที่ 6.26 ตารางค่าความจริงสำหรับ $f(x_1, x_2, x_1 \wedge x_2)$

x_1	x_2	$x_1 \wedge x_2$	$f(x_1, x_2, x_1 \wedge x_2)$
0	0	0	1
0	1	0	0
1	0	0	0
1	1	1	1

รูปที่ 6.24 และ 6.25 แสดงการเปรียบเทียบ แบบในปริภูมิ 2 มิติ ของฟังก์ชัน $f(x_1, x_2)$ และแบบในปริภูมิ 3 มิติ ของฟังก์ชัน $f(x_1, x_2, x_1 \wedge x_2)$ เมื่อสัญลักษณ์ในกราฟเป็นตัวแทนของชั้นข้อมูลดังต่อไปนี้

- ❖ วงกลมสีน้ำเงิน แทนแบบในชั้นข้อมูล X ที่มีผลลัพธ์ของฟังก์ชันเป็น 0
- ❖ วงกลมสีแดง แทนแบบในชั้นข้อมูล O ที่มีผลลัพธ์ของฟังก์ชันเป็น 1

ตัวอย่างที่ 19 จงแสดงให้เห็นว่า SVM สามารถสร้างระนาบเพื่อแบ่งแยกชั้นข้อมูลของแบบในปริภูมิ 3 มิติ ที่ถูกแปลงมาจากแบบในปริภูมิ 2 มิติ ออกจากกัน โดยอาศัยผลคูณแบบดอทในปริภูมิ D มิติ ที่ถูกดำเนินการมาจากผลคูณแบบดอดในปริภูมิ d มิติ (D > d) เมื่อ

แบบในปริภูมิ 2 มิติ คือ $(p,\ q)$

แบบในปริภูมิ 3 มิติ คือ $(p^2, q^2, \sqrt{2}pq)$

กำหนดตัวแปรต่าง ๆ ดังต่อไปนี้

$u^t \cdot v^t$ คือ ผลคูณแบบดอทในปริภูมิ 2 มิติ

$u^{t\prime} \cdot v^{t\prime}$ คือ ผลคูณแบบดอทในปริภูมิ 3 มิติ

เมื่อ

u^t และ v^t คือ เวกเตอร์ของแบบในปริภูมิ 2 มิติ โดยที่

$$u^t = \binom{p_1}{q_1}$$

$$v^t = \binom{p_2}{q_2}$$

$u^{t\prime}$ และ $u^{t\prime}$ คือ เวกเตอร์ของแบบในปริภูมิ 2 มิติ โดยที่

$$u^{t\prime} = \begin{pmatrix} p_1^2 \\ q_1^2 \\ \sqrt{2}p_1q_1 \end{pmatrix}$$

$$v^{t\prime} = \begin{pmatrix} p_2^2 \\ q_2^2 \\ \sqrt{2}p_2q_2 \end{pmatrix}$$

จากนิยามของผลคูณแบบดอท

$$u^{t\prime} \cdot v^{t\prime} = \begin{pmatrix} p_1^2 \\ q_1^2 \\ \sqrt{2}p_1q_1 \end{pmatrix} \cdot \begin{pmatrix} p_2^2 \\ q_2^2 \\ \sqrt{2}p_2q_2 \end{pmatrix}$$

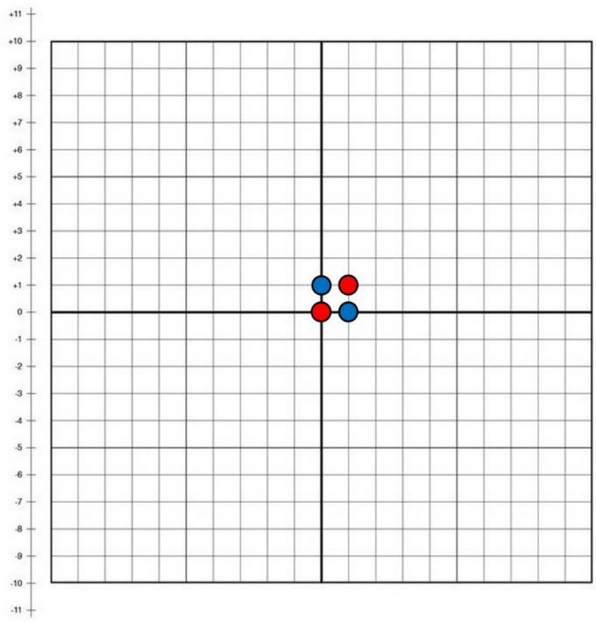
$$= p_1^2p_2^2 + q_1^2q_2^2 + 2 \cdot p_1q_2p_2q_2$$

$$= (p_1p_2)^2 + 2 \cdot (p_1p_2)(q_1q_2) + (q_1q_2)^2$$

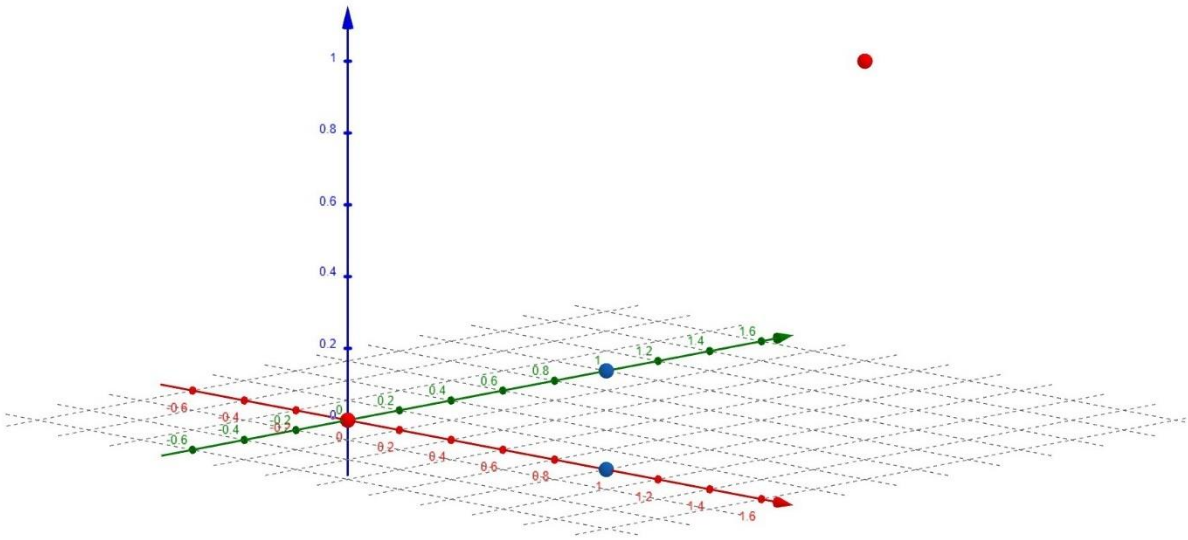
$$= (p_1p_2 + q_1q_2)^2$$

ดังนั้น

$$u^{t\prime} \cdot v^{t\prime} = (u^t \cdot v^t)^2$$



รูปที่ 6.24 ข้อมูลออกของ $f(x_1, x_2)$



รูปที่ 6.25 ข้อมูลออกของ $f(x_1, x_2, x_1 \wedge x_2)$

อภิปราย

ถึงแม้ว่าตัวแบบในบทนี้ อันได้แก่ การเรียนรู้เพอร์เซ็ปตรอน และ โครงข่ายประสาท มีข้อจำกัดคือสามารถจำแนกแบบจากเซตข้อมูลที่บรรจุเพียงสองชั้นข้อมูลก็ตาม แต่สามารถประยุกต์เพื่อนำไปทำนายแบบจากเซตข้อมูลที่มีหลายชั้นข้อมูลได้ โดยลักษณะเฉพาะตัวของการเรียนรู้เพอร์เซ็ปตรอนแล้วขั้นตอนวิธีนี้ทำการเรียนรู้และปรับเวกเตอร์น้ำหนักของตัวจำแนกแบบ จนกระทั่งสามารถทำนายแบบได้อย่างแม่นยำที่สุด โครงข่ายประสาทมีงานวิจัยที่เกี่ยวข้องจำนวนมาก และตัวทำนายชนิดนี้ยังคงถูกพัฒนาอย่างต่อเนื่อง เอสวีเอ็มเป็นตัวจำแนกแบบที่ใช้พื้นฐานของขอบเขตการตัดสินใจเชิงเส้นและทำงานได้ดีบนปริภูมิที่มีมิติสูงมาก

ตัวแบบทั้งหมดที่กล่าวถึงในบทนี้จัดได้ว่ามีประสิทธิภาพสูงสุด เมื่อเปรียบเทียบกับตัวแบบในบทก่อนหน้านี้ทั้งหมดแล้ว ตัวแบบในบทนี้มีความแม่นยำมากกว่าในการทำนายแบบทดสอบ แต่ก็ต้องใช้เวลาในการประมวลผลที่ยาวนานกว่าสำหรับขั้นตอนการเรียนรู้ของตัวแบบ อย่างไรก็ตาม การตั้งค่าพารามิเตอร์เพื่อลดจำนวนรอบในการเรียนรู้ส่งผลให้การประมวลผลเร็วขึ้นแต่ก็แลกด้วยความแม่นยำที่ลดลง ข้อเสียที่ชัดเจนของตัวจำแนกแบบทั้งหมดในบทนี้ ได้แก่ การอธิบายการทำนายแบบเป็นภาษามนุษย์ทำได้ยาก เนื่องจากใช้ฟังก์ชันทางคณิตศาสตร์ที่ซับซ้อน ซึ่งแตกต่างจากต้นไม้การตัดสินใจที่ตีความผลการทำนายได้ง่ายกว่า

สรุป

กล่าวได้ว่า แบบในเซตข้อมูลจะถูกจำแนกเชิงเส้นได้ ถ้าสามารถหาขอบเขตการตัดสินใจสำหรับแบ่งแยกแบบจากต่างชั้นข้อมูลกันให้ออกจากกันได้ ในกรณีของปริภูมิสองมิติ ขอบเขตการตัดสินใจได้แก่เส้นตรง อย่างไรก็ตาม ในกรณีของปริภูมิสามมิติ ขอบเขตการตัดสินใจนี้ได้แก่ระนาบ ในขณะที่ขอบเขตการตัดสินใจมีลักษณะเป็นระนาบหลายมิติสำหรับปริภูมิหลายมิติ (มิติมากกว่าสาม) นอกจากนี้ขอบเขตการตัดสินใจยังแบ่งออกได้เป็น 2 รูปแบบ ได้แก่ รูปแบบเอกพันธุ์ และ รูปแบบไม่เอกพันธุ์ อย่างไรก็ตาม รูปแบบไม่เอกพันธ์สามารถแปลงเป็นรูปแบบเอกพันธุ์ได้

การเรียนรู้เพอร์เซ็ปตรอนใช้กับแบบที่ผ่านการแปลงและทำให้อยู่ในรูปปกติ โดยมีข้อจำกัดว่าชั้นข้อมูลต้องสามารถจำแนกเชิงเส้นได้ การทำงานของการเรียนรู้นี้จะทำการปรับเวกเตอร์น้ำหนักจนกระทั่งสามารถจำแนกแบบทั้งหมดได้อย่างถูกต้อง อย่างไรก็ตาม ในกรณีที่แบบไม่สามารถจำแนกเชิงเส้นได้ การเรียนรู้เพอร์เซ็ปตรอนอาจไม่มีทางจบการทำงานได้ เนื่องจากจำนวนรอบในการปรับค่าเวกเตอร์น้ำหนักเป็นอนันต์

เนื่องจากการเรียนรู้เพอร์เซ็ปตรอน สามารถใช้ได้กับเซตข้อมูลที่มีเพียงสองชั้นข้อมูลเท่านั้น อย่างไรก็ตาม ถ้าเซตข้อมูลมีจำนวนชั้นข้อมูลมากกว่านี้ จำเป็นต้องแปลงปัญหาหลายชั้นข้อมูลเป็นปัญหาชั้นข้อมูลทวิภาค ซึ่งทำได้สองวิธีได้แก่ OAO และ OAA วิธีการแรกคือการพิจารณาทีละคู่ของชั้นข้อมูล ซึ่งต้องใช้จำนวนครั้งในการจับคู่เป็นจำนวนมาก ในขณะที่ วิธีหลังเป็นการพิจารณาคู่ของชั้นข้อมูลเช่นกัน แต่จับคู่ชั้นข้อมูลที่กำลังพิจารณาและชั้นข้อมูลที่เหลือทั้งหมด จำนวนครั้งที่ใช้ในการจับคู่ทั้งหมดจะน้อยกว่าวิธีแรก การสรุปผลการทำนายของทั้งสองวิธีข้างต้นให้ใช้เสียงส่วนใหญ่ของกลุ่มตัวจำแนกแบบที่สร้างขึ้นเป็นตัวตัดสิน

โครงข่ายประสาทคือตัวทำนายที่จำลองทำงานของสมองของมนุษย์ โดยประกอบไปด้วยเซลล์ประสาทประดิษฐ์จำนวนมากที่ทำหน้าที่รับส่งข้อมูลหากัน โครงข่ายการป้อนไปข้างหน้าจัดเป็นโครงข่ายประสาทพื้นฐาน ตัวแบบชนิดนี้มีการทำงานคล้ายกับการเรียนรู้เพอร์เซ็ปตรอนในการสร้างฟังก์ชันการจำแนกเชิงเส้น สำหรับการเรียนรู้เพื่อปรับค่าน้ำหนักเช่นกัน

เครื่องกลเวกเตอร์สนับสนุนคือตัวแบบที่ค้นหาเวกเตอร์สนับสนุน จากแบบฝึกฝนที่เหมาะสมในเซตข้อมูล เพื่อนำมาสร้างขอบเขตการตัดสินใจสำหรับแบ่งแยกชั้นข้อมูลออกจากกัน อย่างไรก็ตาม ถ้าแบบในปริภูมิของมิติที่ต่ำกว่าไม่สามารถจำแนกเชิงเส้นได้ อาจต้องแปลงแบบให้อยู่ในปริภูมิของมิติที่สูงกว่า เพื่อจุดประสงค์ให้แบบจากต่างชั้นข้อมูลมีคุณสมบัติในการจำแนกเชิงเส้นได้

เนื้อหาตั้งแต่บทที่สามจนถึงบทก่อนหน้า กล่าวถึงตัวจำแนกแบบที่ประยุกต์แนวคิดพื้นฐานแตกต่างกันออกไป อย่างไรก็ตาม ตัวแบบทุกประเภทจำเป็นต้องเรียนรู้จากแบบจากเซตฝึกฝน เพื่อให้สามารถรู้จำแบบในเซตทดสอบได้อย่างแม่นยำ อย่างไรก็ตาม ยังมีวิธีเพิ่มประสิทธิภาพการทำนายอีกชนิดหนึ่งที่ไม่ใช่การเรียนรู้ของตัวจำแนกแบบ แต่เป็นการปรับสัดส่วนของแต่ละชั้นข้อมูลใหม่ก่อนการส่งให้ตัวแบบเรียนรู้ การปรับนี้อาจปรับลดสัดส่วนหรือปรับเพิ่มสัดส่วนในแต่ละชั้นข้อมูลก็ได้ วิธีนี้เรียกว่า การสุ่ม ดังที่จะได้กล่าวในบทต่อไป

แบบฝึกหัด

1. จงแสดงให้เห็นว่า เส้นตรง 3 เส้นด้านล่างนี้ ขนานกัน

   - $x_1 - x_2 = 0$
   - $2x_1 - 2x_2 = -1$
   - $2x_1 - 2x_2 = 1$

2. จงแสดงให้เห็นว่าเวกเตอร์น้ำหนัก w^t ตั้งฉากกับขอบเขตการตัดสินใจในปริภูมิ d มิติ เมื่อ $d \geq 2$

3. กำหนดให้ θ แทนมุมระหว่าง w^t และ x จงแสดงให้เห็นว่า $\cos\theta$ มีค่าเป็นบวก ถ้า $w^t x > 0$

4. พิจารณาแบบใน 2 มิติ ในรูปที่ 6.1 และ ขอบเขตการตัดสินใจ $x_1 - x_2 = 1$ จงแสดงให้เห็นว่า ระยะทางจากจุด (α, α) เมื่อ α จำนวนจริงใด ๆ ไปยังขอบเขตการตัดสินใจเท่ากับ $\frac{-1}{\sqrt{2}}$

5. พิจารณาแบบ 2 มิติ ได้แก่ แบบ (1, 1) ; (2, 2) จากชั้นข้อมูล X และ แบบ (2, 0) จากชั้นข้อมูล O จงใช้วิธีการเรียนรู้เพอร์เซ็ปตรอน เพื่อแสดงให้เห็นว่าขอบเขตการตัดสินใจที่ได้คือเวกเตอร์น้ำหนัก $x - 3y - 1 = 0$

6. จากเซตข้อมูลในตารางที่ 6.13 สำหรับปัญหาหลายชั้นข้อมูล จงแสดงให้เห็นว่าเวกเตอร์แต่ละตัวมีค่าดังต่อไปนี้

   6.1 $w_{X*} = (-9, 5, 1)^t$

   6.2 $w_{O*} = (-4, 5.5, 0)^t$

   6.3 $w_O = (-1.5, 5.5, -21)^t$

   6.4 $w_* = (2.5, -11.5, -2)^t$

7. จงแสดงการคำนวณโดยใช้วิธีการเรียนรู้เพอร์เซ็ปตรอน การคำนวณในส่วนที่ข้ามไปในตัวอย่างที่ 7

8. พิจารณาแบบ (1, 1) ; (2, 2) จากชั้นข้อมูล X และ (2, 0) จากชั้นข้อมูล O จงแสดงให้เห็นว่า ขอบเขตการตัดสินใจของการเรียนรู้เพอร์เซ็ปตรอน คือ $x_1 - 3x_2 - 1 = 0$

9. จงแสดงการคำนวณโดยใช้วิธีการเรียนรู้เพอร์เซ็ปตรอน จากเซตฝึกฝน ในตัวอย่างที่ 6 ของบทที่ 4 ถ้าคำนวณครบ 10 รอบแล้ว เงื่อนไขหยุดการทำงานไม่เป็นจริง ให้นำค่า w_{10} มาใช้

10. กำหนดแบบฝึกฝนจำนวน 4 แบบ แต่ละแบบมี 4 มิติ และ ข้อมูลออก 2 ตัว ตามตารางที่ 6.27 กำหนดค่าเริ่มต้น ดังนี้ น้ำหนัก w_{ji} = 0.5 สำหรับ $1 \leq j \leq 4$ และ i = 1, 2 ขีดแบ่ง t = 0 และ α = 0.5 ในส่วนเพิ่มเติมฟังก์ชันการกระตุ้นขีดแบ่ง คือ

$$o = 1 \text{ ถ้า } a > t$$

$$o = 0 \text{ ถ้า } a \leq t$$

จงแสดงการคำนวณโดยใช้ โครงข่ายการป้อนไปข้างหน้า ในการหาค่าน้ำหนักทั้งหมด ถ้าคำนวณครบ 4 รอบ แต่ $e_1 \neq 0$ และ $e_2 \neq 0$ ให้หยุดทันที แล้วตอบค่าน้ำหนักล่าสุด

ตารางที่ 6.27 แบบ 4 มิติ และ หน่วยข้อมูลออก 2 หน่วย 2

Pattern No.	x_1	x_2	x_3	x_4	t_1	t_2
1	1	0	0	0	0	1
2	0	1	0	0	0	1
3	0	0	1	0	1	0
4	0	0	0	1	1	0

11. พิจารณาตารางค่าความจริงของฟังก์ชัน f ใน ตัวอย่างที่ 18 กำหนดให้ข้อมูลออกเป็น 0 สัมพันธ์กับชั้นข้อมูล X และ ข้อมูลออกเป็น 1 สัมพันธ์กับชั้นข้อมูล O จงแสดงให้เห็นว่า สองชั้นข้อมูลนี้สามารถจำแนกเชิงเส้นได้

บทที่ 7
ปัญหาชั้นข้อมูลอสมดุล

เนื้อหาในบทสุดท้ายกล่าวถึงปัญหาที่มีความท้าทายมากปัญหาหนึ่ง และพบได้จริงในทางปฏิบัติ ได้แก่ ปัญหาชั้นข้อมูลอสมดุลซึ่งเกิดขึ้นในโปรแกรมประยุกต์หลายโปรแกรม ปัญหานี้ไม่สามารถแก้ไขได้โดยการปรับแต่งค่าพารามิเตอร์ของตัวจำแนกแบบ แต่จำเป็นต้องปรับสัดส่วนของชั้นข้อมูลในเซตข้อมูลให้มีความสมดุลมากขึ้น เทคนิคที่ใช้รับมือปัญหานี้มีหลากหลายเทคนิค ทั้งการเพิ่มขนาดของชั้นข้อมูลที่มีขนาดเล็ก หรือลดขนาดของชั้นข้อมูลที่มีขนาดใหญ่ รายละเอียดของปัญหาและเทคนิคต่าง ๆ กำลังจะอธิบาย ดังหัวข้อต่อไปนี้ เป็นการจบเนื้อหาของตำราฉบับนี้โดยสมบูรณ์

จุดประสงค์การเรียนรู้

- นิยามปัญหาชั้นข้อมูลอสมดุลได้
- แยกความแตกต่างของสโมทแต่ละเทคนิคได้
- ปรับแต่งพารามิเตอร์ในมิวท์ได้
- สามารถประยุกต์เซฟเลเวลกราฟในการเลือกสโมทหรือมิวท์ที่เหมาะสมได้

1. นิยามของปัญหา

ปัญหาชั้นข้อมูลอสมดุล (Class Imbalance Problem) ปรากฏในเซตข้อมูลที่สัดส่วนของชั้นข้อมูลมีความแตกต่างกันสูงมาก ดังนั้นผลเสียที่เกิดขึ้นคือตัวจำแนกแบบมี อคติ (Bias) ต่อชั้นข้อมูลที่มีสัดส่วนต่ำกว่า ได้แก่ ชั้นด้อย (Minority Class) หรือ ชั้นบวก (Positive Class) มากกว่าชั้นข้อมูลที่มีสัดส่วนสูงกว่า ได้แก่ ชั้นเด่น (Majority Class) หรือ ชั้นลบ (Negative Class) ทำให้ตัวแบบที่ได้ขาดประสิทธิภาพในการทำนายชั้นด้อย หรือกล่าวได้ว่ามีค่าความแม่นยำในการทำนายชั้นด้อยต่ำจนไม่เป็นที่น่าพอใจ สำหรับปัญหาประเภทนี้ ชั้นด้อยที่มีจำนวนแบบน้อยกว่าจัดเป็นข้อมูลที่มีความน่าสนใจและความสำคัญมากกว่าชั้นเด่นที่มีจำนวนแบบมากกว่า

ในเซตข้อมูลของคนไข้ที่เป็นโรคมะเร็ง ประชากรที่เป็นโรคมะเร็งจัดเป็นชั้นด้อยเพราะมีปริมาณน้อยมากเมื่อเปรียบเทียบกับประชากรสุขภาพดีที่จัดเป็นชั้นเด่น ด้วยขนาดที่เล็กมากของชั้นด้อยส่งผลให้ตัวจำแนกแบบไม่มีประสิทธิภาพในการทำนายประชากรที่เป็นโรคมะเร็งจากกลุ่มตัวอย่าง ทำให้มีผลร้ายต่อประชากรเหล่านั้น เพราะอาจไม่ได้เข้ารับการรักษาอย่างทันท่วงทีและอาจเสียชีวิตได้ในเวลา

ต่อมา ในทางกลับกัน การทำนายประชากรสุขภาพดีว่าเป็นโรคมะเร็ง ส่งผลกระทบเพียงค่าบริการทางการแพทย์ในการตรวจหาโรคมะเร็งเท่านั้น ดังนั้น ความเสียหายในการทำนายแบบในชั้นด้อยผิดพลาดมีค่าสูงกว่าการทำนายแบบในชั้นเด่นผิดพลาด

ในเซตข้อมูล การบุกรุก (Intrusion) ในเครือข่ายคอมพิวเตอร์ ชั้นข้อมูลด้อยได้แก่แพ็กเกจการบุกรุก ส่วนชั้นข้อมูลเด่นได้แก่แพ็กเกจปกติของผู้ใช้ แพ็กเกจการบุกรุกจัดอยู่ในชั้นข้อมูลด้อยเพราะชั้นข้อมูลนี้มีขนาดเล็กมากเมื่อเปรียบเทียบกับชั้นข้อมูลของแพ็กเกจปกติของผู้ใช้ การจำแนกแบบถูกนำไปประยุกต์ใช้กับไฟร์วอลล์ไม่ว่าจะเป็นซอร์ฟแวร์หรือฮาร์ดแวร์ เพื่อปกป้องระบบเครือข่ายคอมพิวเตอร์ไม่ให้ถูกโจมตี การทำนายแพ็กเกจบุกรุกพลาดจะส่งผลร้ายต่อทรัพยากรมหาศาล เนื่องจากอาจโดนเจาะระบบเพื่อขโมยข้อมูลสำคัญเช่นข้อมูลลูกค้าและข้อมูลบัตรเครดิตเป็นต้น

2. หน่วยวัดสมรรถนะ

เมทริกซ์สับสน (Confusion Matrix) ในตารางที่ 7.1 ถูกใช้สำหรับคำนวณ หน่วยวัดสมมรถนะ (Performance Measure) เพื่อวัดประสิทธิภาพของตัวจำแนกแบบ สัญลักษณ์ต่าง ๆ ในตาราง ถูกอธิบายดังต่อไปนี้

Actual Negative คือ ชั้นข้อมูลจริงที่บรรจุแบบลบ (ชั้นข้อมูลเด่น)

Actual Positive คือ ชั้นข้อมูลจริงที่บรรจุแบบบวก (ชั้นข้อมูลด้อย)

Predicted Negative คือ ชั้นข้อมูลที่ถูกทำนายเป็นชั้นลบ

Predicted Positive คือ ชั้นข้อมูลจริงที่ถูกทำนายเป็นชั้นบวก

TN (True Negative) คือ จำนวนแบบในชั้นลบที่ถูกทำนายถูกต้อง (ทายชั้นลบเป็นชั้นลบ)

FP (False Positive) คือ จำนวนแบบในชั้นลบที่ถูกทำนายผิด (ทายชั้นลบเป็นชั้นบวก)

FN (False Negative) คือ จำนวนแบบในชั้นบวกที่ถูกทำนายผิด (ทายชั้นบวกเป็นชั้นลบ)

TP (True Positive) คือ จำนวนแบบในชั้นบวกที่ถูกทำนายถูกต้อง (ทายชั้นบวกเป็นชั้นบวก)

ตารางที่ 7.1 เมทริกซ์สับสน

	Predicted Negative	Predicted Positive
Actual Negative	TN	FP
Actual Positive	FN	TP

หน่วยวัดสมรรถนะทั้งหมด ได้แก่ ความแม่นยำ (Accuracy) ความผิดพลาด (Error Rate) รีคอล (Recall) พรีซิชัน (Precision) และ หน่วยวัดเอฟ (F-measure) คำนวณได้ ดังนี้

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{FP} + \text{FN} + \text{TP}}$$

$$\text{Error Rate} = \frac{\text{FP} + \text{FN}}{\text{TN} + \text{FP} + \text{FN} + \text{TP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{FN} + \text{TP}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{FP} + \text{TP}}$$

$$\text{F–measure} = \frac{(1+\beta)^2 \cdot \text{Recall} \cdot \text{Precision}}{\beta^2 \cdot \text{Recall} + \text{Precision}}$$

ตัวอย่างที่ 1 เซตข้อมูลมะเร็งเต้านมจาก KDD Cup 2008 เป็นเซตข้อมูลอสมดุลที่บรรจุแบบทั้งหมด จำนวน 102,294 แบบ เซตข้อมูลนี้มี 2 ชั้นข้อมูล ได้แก่ ไม่มีเนื้อร้าย (ชั้นเด่น) จำนวน 101,671 แบบ และ มีเนื้อร้าย (ชั้นด้อย) จำนวน 623 แบบ ถ้าตัวจำแนกแบบประเภทหนึ่งทำนายแบบทั้งหมดเป็น ไม่มีเนื้อร้าย ไม่ว่าแบบนั้นจะอยู่ในชั้นข้อมูลใดก็ตาม ส่งผลให้ความแม่นยำและความผิดพลาดคำนวณได้ ดังนี้

Accuracy = 101,671 / 102,294 = 99.39%

Error Rate = 623 / 102,294 = 0.61%

จากการสังเกต ถึงแม้ว่าความแม่นยำมีค่าสูงมากคือเกือบ 100% แต่แบบจากชั้นข้อมูลมีเนื้อร้ายถูกทำนายผิดหมดทุกแบบใกล้เคียง 0% ดังนั้นหน่วยวัดนี้ไม่เหมาะสมในการใช้กับเซตข้อมูลนี้ เนื่องจากขาดประสิทธิภาพในการจำแนกแบบจากชั้นด้อย (มีเนื้อร้าย) ซึ่งเป็นชั้นข้อมูลที่มีความสำคัญกว่า

หน่วยวัดสมรรถนะที่เหมาะสมกับเซตข้อมูลอสมดุล ได้แก่ รีคอล และ พรีซิชัน เนื่องจากมีการเจาะจงไปที่แบบจากชั้นด้อยโดยเฉพาะ ในส่วนเพิ่มเติม หน่วยวัดเอฟ คือ หน่วยวัดที่รวมทั้ง รีคอล และ พรีซิชัน เป็นเทอมเดียว โดยที่ β คือ ค่าคงที่ โดยมากกำหนดค่าเป็น 1 ซึ่งหมายความว่าให้ความสำคัญกับ รีคอล และ พรีซิชัน เท่ากัน

ตัวอย่างที่ 2 กำหนด เมทริกซ์สับสน ที่ได้จากโปรแกรมเวก้า แสดงตามด้านล่างนี้

```
  a   b   <-- classified as
193   8 |   a = no-recurrence-events
 68  17 |   b = recurrence-events
```

จากเมทริกซ์สับสนข้างต้นสรุปได้ว่า

แบบจากชั้นข้อมูล a (no-recurrence-events) มีจำนวน 193 + 8 = 201 แบบ

แบบจากชั้นข้อมูล b (recurrence-events) มีจำนวน 68 + 17 = 85 แบบ

จำนวนแบบทั้งหมด มีจำนวน 201 + 85 = 286 แบบ

กำหนดให้แบบ a คือ แบบจากชั้นลบ และ b คือ แบบจากชั้นบวก ดังนั้น

TN = 193 (ทำนายแบบจากชั้นลบ a ถูกต้อง)

FP = 8 (ทำแบบแบบจากชั้นลบ a ผิดพลาด)

FN = 68 (ทำนายแบบจากชั้นบวก b ผิดพลาด)

TP = 17 (ทำนายแบบจากชั้นบวก b ถูกต้อง)

Accuracy = $\left(\frac{193+17}{286}\right)$ = 0.7343

Error Rate = = $\left(\frac{8+68}{286}\right)$ = 0.2657

Accuracy ของชั้นข้อมูล no-recurrence-events = $\left(\frac{193}{193+8}\right)$ = 0.9602

Accuracy ของชั้นข้อมูล recurrence-events = $\left(\frac{17}{68+17}\right)$ = 0.2

Recall = $\left(\frac{\mathrm{TP}}{\mathrm{FN+TP}}\right)$ = $\left(\frac{17}{68+17}\right)$ = 0.2

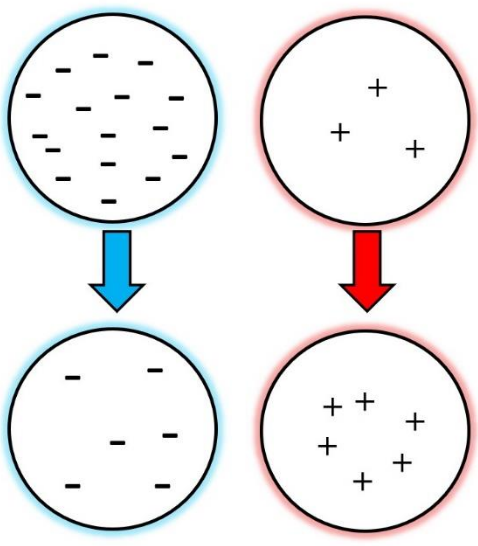
Precision = $\left(\frac{\mathrm{TP}}{\mathrm{FP+TP}}\right)$ = $\left(\frac{17}{8+17}\right)$ = 0.68

F-measure = $\frac{(1+\beta)^2\cdot\mathrm{Recall}\cdot\mathrm{Precision}}{\beta^2\cdot\mathrm{Recall}+\mathrm{Precision}}$ = $\frac{(1+1)^2\cdot 0.2\cdot 0.68}{1^2\cdot 0.2+0.68}$ = 0.6182

3. การสุ่มเพิ่มและการสุ่มลด

การสุ่ม (Re-sampling) เป็นเทคนิคที่ใช้กันอย่างแพร่หลายในการแก้ปัญหาชั้นข้อมูลอสมดุล แบ่งออกเป็น 2 ประเภท ได้แก่ การสุ่มเพิ่ม (Over-sampling) และ การสุ่มลด (Under-sampling) เทคนิคแรกเป็นการสร้างแบบด้อยเข้าไปในเซตข้อมูลเพื่อขยายขนาดของชั้นด้อยให้ใหญ่ขึ้น เทคนิคหลังเป็นการกำจัดแบบเด่นออกจากเซตข้อมูลเพื่อลดขนาดของชั้นเด่นให้เล็กลง เทคนิคทั้งสองจะทำการปรับขนาดของชั้นเด่นและชั้นด้อยให้มีความใกล้เคียงกัน แสดงตามรูปที่ 7.1 เพื่อให้ตัวจำแนกแบบสามารถทำนายแบบในชั้นด้อยได้อย่างมีประสิทธิภาพ อย่างไรก็ตาม ข้อเสียของเทคนิคแรกคือทำให้ขนาดของเซตข้อมูลใหญ่ขึ้น ส่งผลให้ใช้เวลาประมวลผลยาวนานกว่าเดิม นอกจากนี้ อาจมีการสุ่มเพิ่มใน บริเวณซ้อนทับ (Over-lapping Region) ระหว่างชั้นเด่นและชั้นด้อย ส่งผลให้ตัวจำแนกแบบมีความสับสนในการทำนายแบบในบริเวณดังกล่าว ส่วนข้อเสียของเทคนิคหลังคืออาจกำจัดข้อมูลที่มี ความสำคัญ (Significant) ในเซตข้อมูลให้สูญหายไป

การสุ่มชนิดพื้นฐาน ได้แก่ การสุ่มเพิ่มแบบสุ่ม (Random Over-sampling) ที่เพิ่มขนาดชั้นด้อยโดยการสุ่มเลือกแบบด้อยจากเซตทดสอบ แล้วคัดลอกลงเซตข้อมูลโดยไม่มีการเปลี่ยนแปลงค่าใด ๆ ดังนั้น แบบที่เกิดจากการคัดลอกนี้มีคุณลักษณะที่ไม่หลากหลาย จึงก่อให้เกิด ปัญหาแน่นเกินไป (Over-fitting Problem) ที่ทำให้ตัวจำแนกแบบทำนายแบบที่มีค่าแตกต่างกันจากแบบทดสอบได้ไม่ดีเท่าที่ควร การสุ่มชนิดพื้นฐานอีกวิธี ได้แก่ การสุ่มลดแบบสุ่ม (Random Under-sampling) ที่ลดขนาดชั้นเด่น โดยการสุ่มเลือกแบบเด่นแล้วลบออกจากเซตข้อมูล ที่อาจทำให้เซตข้อมูลสูญเสียข้อมูลที่สำคัญออกไป



รูปที่ 7.1 การสุ่มลด (รูปซ้าย) และ การสุ่มเพิ่ม (รูปขวา)

4. สโมท

สโมท (SMOTE: Synthetic Minority Over-sampling Technique) คือเทคนิคการสุ่มเพิ่มที่อาศัยแนวคิดของเพื่อนบ้านใกล้ที่สุดในการสังเคราะห์แบบด้อยเข้าไปในเซตข้อมูล แบบสังเคราะห์ที่ได้นี้มีความหลากหลายของค่าในฟีเจอร์มากกว่าแบบที่ถูกสร้างจากการคัดลอกข้อมูล สโมทพื้นฐานถูกออกแบบให้ประมวลผลได้บนเซตข้อมูลที่บรรจุฟีเจอร์ชนิด ต่อเนื่อง (Continuous) ได้แก่ จำนวนเต็ม และ จำนวนจริง ขั้นตอนของการคำนวณค่าแบบด้อยสังเคราะห์ แสดงตามลำดับด้านล่างนี้

สำหรับแต่ละคุณลักษณะต่อเนื่อง

1. หาค่าผลต่างระหว่างคุณลักษณะของแบบด้อยและแบบที่สุ่มจากเพื่อนบ้าน k ตัว
2. คูณค่าผลต่างนี้ด้วยตัวเลขสุ่มระหว่าง 0 ถึง 1
3. บวกค่าผลคูณนี้กับฟีเจอร์ของแบบด้อย เพื่อสร้างฟีเจอร์ของแบบสังเคราะห์

ตัวอย่างที่ 3 กำหนด แบบด้อย (6, 4) และ เพื่อนบ้านใกล้ที่สุด (4, 3)

ตัวแปรต่าง ๆ อธิบาย ดังนี้

f1_1 คือ ค่าคุณลักษณะแรกของแบบด้อย

f1_2 คือ ค่าคุณลักษณะหลังของแบบด้อย

f2_1 คือ ค่าคุณลักษณะแรกของเพื่อนบ้านใกล้ที่สุด

f2_2 คือ ค่าคุณลักษณะหลังของเพื่อนบ้านใกล้ที่สุด

ดังนั้น

f1_1 = 6 ; f1_2 = 4

f2_1 = 4 ; f2_2 = 3

f2_1 – f1_1 = 4 – 6 = –2 ; f2_2 – f1_2 = 3 – 4 = –1

ถ้าสมมติว่าตัวเลขสุ่มได้แก่ 0.5 แล้วแบบสังเคราะห์ใหม่ คำนวณได้ ดังนี้

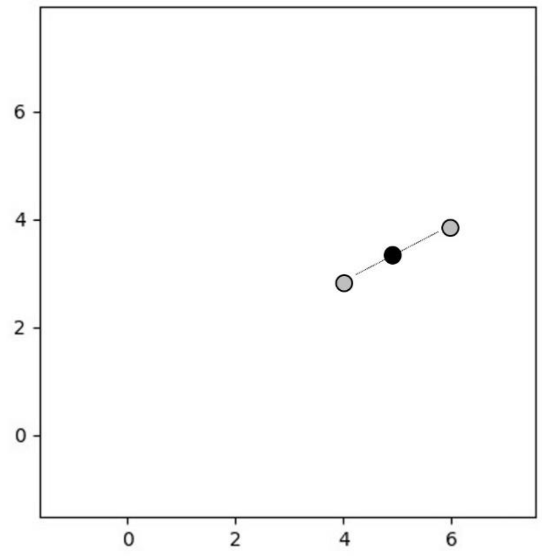
(f1', f2') = (6, 4) + rand(0-1) * (–2, –1)

= (6, 4) + 0.5 * (–2, –-1)

= (6, 4) + (–1, –0.5)

= (5, 3.5)

ข้อมูลสังเคราะห์ที่ได้ แสดงตามรูปที่ 7.3 เมื่อวงกลมสีเทาแทนแบบด้อยและเพื่อนบ้านใกล้ที่สุด ในขณะที่วงกลมสีดำแทนแบบสังเคราะห์ สังเกตได้ว่าแบบสังเคราะห์จะอยู่ในแนวส่วนของเส้นตรงระหว่างแบบด้อยและเพื่อนบ้านใกล้ที่สุด



รูปที่ 7.3 แบบสังเคราะห์

ขั้นตอนวิธีสโมทแสดงตามรหัสเทียมในหน้าถัดไป ตัวแปรต่าง ๆ อธิบาย ดังต่อไปนี้

- *T* คือ จำนวนแบบในชั้นด้อย **(ข้อมูลเข้า)**
- *N* คือ จำนวนเปอร์เซ็นต์ของแบบสังเคราะห์ที่ต้องการสร้าง **(ข้อมูลเข้า)**
- *k* คือ จำนวนเพื่อนบ้านใกล้ที่สุด **(ข้อมูลเข้า)**
- numattrs คือ จำนวนฟีเจอร์
- *Sample* คือ ตัวแปรชุด 2 มิติ ของแบบด้อย เมื่อมิติแรกแทนอินเด็กซ์ มิติหลังแทนฟีเจอร์
- *nextindex* คือ จำนวนแบบสังเคราะห์ ค่าเริ่มต้นคือ 0
- *Synthetic* คือ ตัวแปรชุด 2 มิติ ของแบบสังเคราะห์
- *i* คือ ค่าตั้งแต่ 1 จนถึง *T*
- *nnarray* คือ อินเด็กซ์ของเพื่อนบ้านใกล้ที่สุด *k* ตัว
- *nn* คือ ตัวเลขสุ่มในช่วง 1 ถึง *k*
- *attr* คือ ลำดับที่ของฟีเจอร์ มีค่าตั้งแต่ 1 ถึง *numattrs*
- *dif* คือ ผลต่างของค่าในฟีเจอร์ระหว่างแบบด้อยและเพื่อนบ้านใกล้ที่สุด
- *gap* คือ ตัวเลขสุ่มในช่วง 0 ถึง 1

```
Algorithm SMOTE(T, N, k)
Input: Number of minority class samples T; Amount of SMOTE N%
          Number of nearest neighbors k
Output: (N/100) * T synthetic minority class samples

1. (* If N is less than 100%, randomize the minority class samples
      as only a random percent of them will be SMOTEd.*)
2. if N < 100
3.    then Randomize the T minority class samples
4.         T = (N/100)*T
5.         N = 100
6. endif
7. N = (int)(N/100)
   (* The amount of SMOTE is assumed to be in integral multiples of 100. *)
8. k = Number of nearest neighbors
9. numattrs = Number of attributes
10. Sample[ ][ ]: array for original minority class samples
11. newindex: keeps a count of number of synthetic samples
      generated, initialized to 0
12. Synthetic[ ][ ]: array for synthetic samples
    (* Compute k nearest neighbors for each minority class sample only. *)
13. for i ← 1 to T
14.        Compute k nearest neighbors for i,
               and save the indices in the nnarray
15.        Populate(N, i, nnarray)
16. endfor

Populate(N, i, nnarray)
(* Function to generate the synthetic samples. *)
17. while N ≠ 0 do
18.        Choose a random number between 1 and k, call it nn.
           This step chooses one of the k nearest neighbors of i.
19.        for attr ← 1 to numattrs
20.             dif = Sample[nnarray[nn]][attr] - Sample[i][attr]
21.             gap = random number between 0 and 1
22.             Synthetic[newindex][attr] = Sample[i][attr] + gap * dif
23.        endfor
24.        newindex++
25.        N = N - 1
26. endwhile
27. return (* End of Populate.*)

End of Pseudo-Code.
```

SMOTE รับข้อมูลเข้าเป็น ขนาดของชั้นด้อย จำนวนของเพื่อนบ้านใกล้ที่สุดที่ต้องการหา และเปอร์เซ็นต์ของแบบสังเคราะห์ที่ต้องการสร้าง หลังจากประมวลผลแล้วสโมทให้ข้อมูลออกเป็น แบบสังเคราะห์ตามจำนวนเปอร์เซ็นต์ที่ผู้ใช้ต้องการ คำอธิบายของรหัสเทียมสโมทมีดังต่อไปนี้

บรรทัดที่ 1 ถึง 6 ถ้าเปอร์เซ็นต์มีค่าน้อยกว่า 100% ให้ทำการสุ่มแบบด้อย

บรรทัดที่ 7 จำนวนแบบที่จะสโมทต้องอยู่ในรูปผลคูณของ 100

บรรทัดที่ 13 ถึง 16 ค้นหาเพื่อนบ้านใกล้ที่สุดของแบบด้อย จากนั้นเรียกโปรแกรมย่อย `Populate`

โปรแกรมย่อย Populate

บรรทัดที่ 17 ถึง 26 วนซ้ำสร้างแบบสังเคราะห์ให้เท่ากับจำนวนเปอร์เซ็นต์

บรรทัดที่ 19 วนซ้ำทีละฟีเจอร์

บรรทัดที่ 20 คำนวณค่าผลต่างของฟีเจอร์ระหว่างแบบและเพื่อนบ้านใกล้ที่สุด

บรรทัดที่ 21 สร้างตัวเลขสุ่มในช่วง 0 ถึง 1

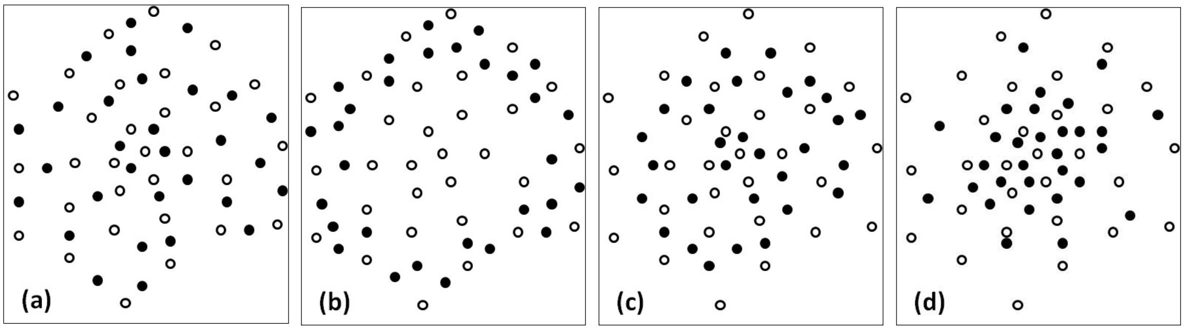
บรรทัดที่ 22 สร้างแบบสังเคราะห์ ขั้นตอนการสร้างอยู่ในตัวอย่างที่ 3

บรรทัดที่ 27 จบการทำงานของโปรแกรมย่อย *Populate*

ครอบครัวสโมท (SMOTE Family) ประกอบไปด้วยเทคนิคสโมทประเภทต่าง ๆ ได้แก่ สโมทต้นตำรับ บอร์เดอร์ไลน์สโมท (Borderline-SMOTE) เซฟเลเวลสโมท (Safe-Level-SMOTE) และดีบีสโมท (DBSMOTE) เทคนิคเหล่านี้สร้างแบบสังเคราะห์ในบริเวณที่แตกต่างกัน ดังต่อไปนี้

- **สโมท** คือ เทคนิคสุ่มเพิ่มที่ไม่เจาะจงสร้างแบบสังเคราะห์ในบริเวณใดบริเวณหนึ่งของชั้นด้อย แต่กระจายแบบสังเคราะห์ให้สม่ำเสมอทั่วทุกบริเวณ
- **บอร์เดอร์ไลน์สโมท** คือ สโมทที่เน้นการสังเคราะห์ไปที่บริเวณขอบของชั้นด้อยเท่านั้น
- **เซฟเลเวลสโมท** คือ สโมทที่ไม่เน้นการสังเคราะห์ไปที่ขอบของชั้นด้อย แต่พยายามเจาะจงตำแหน่งของแบบสังเคราะห์ให้ห่างจากขอบดังกล่าว
- **ดีบีสโมท** คือ สโมทที่เน้นการสังเคราะห์แบบด้อยไปที่บริเวณแกนของชั้นด้อย เพื่อให้แกนมีความหนาแน่นมากยิ่งขึ้น

รูปที่ 7.3 แสดงการกระจายตัวของ ชั้นด้อยสุ่มเพิ่ม (Over-sampling Minority Class) ที่ประกอบไปด้วย แบบด้อยต้นฉบับ (Original Minority Pattern) แสดงแทน วงกลมสีขาว และ แบบด้อยสังเคราะห์ (Synthetic Minority Pattern) แสดงแทน วงกลมสีดำ



รูปที่ 7.3 ชั้นด้อยสุ่มเพิ่มของ

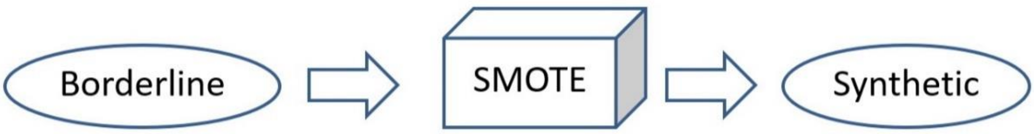
(a) SMOTE (b) Borderline-SMOTE (c) Safe-Level-SMOTE (d) DBSMOTE

บอร์เดอร์ไลน์สโมท

เทคนิคบอร์เดอร์ไลน์สโมทประยุกต์ใช้ขั้นตอนวิธีของสโมทดั้งเดิม เพียงแต่เปลี่ยนแปลงข้อมูลเข้าจากแบบด้อยทั้งหมดเป็นแบบด้อยที่อยู่บริเวณขอบของชั้นด้อยเท่านั้นแทน แบบในบริเวณนี้เรียกว่า บอร์เดอร์ไลน์ (Borderline) แสดงตามรูปที่ 7.4

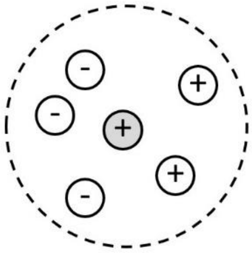
การตรวจสอบแบบด้อยที่เป็นบอร์เดอร์ไลน์ ทำได้โดยการคำนวณค่าเพื่อนบ้านใกล้ที่สุด k ตัว จากนั้นพิจารณาว่าจากแบบทั้งหมดมีเพื่อนบ้านกี่ตัวอยู่ในชั้นเด่น ถ้าเพื่อนบ้านจากชั้นเด่นมีปริมาณไม่น้อยกว่าครึ่งหนึ่งของเพื่อนบ้านทั้งหมด สรุปได้ว่าเป็นบอร์เดอร์ไลน์ เงื่อนไขของแบบประเภทนี้แสดงตามอสมการด้านล่าง เมื่อ n คือ จำนวนแบบจากจากชั้นเด่น

$$\tfrac{1}{2} k \leq n < k$$



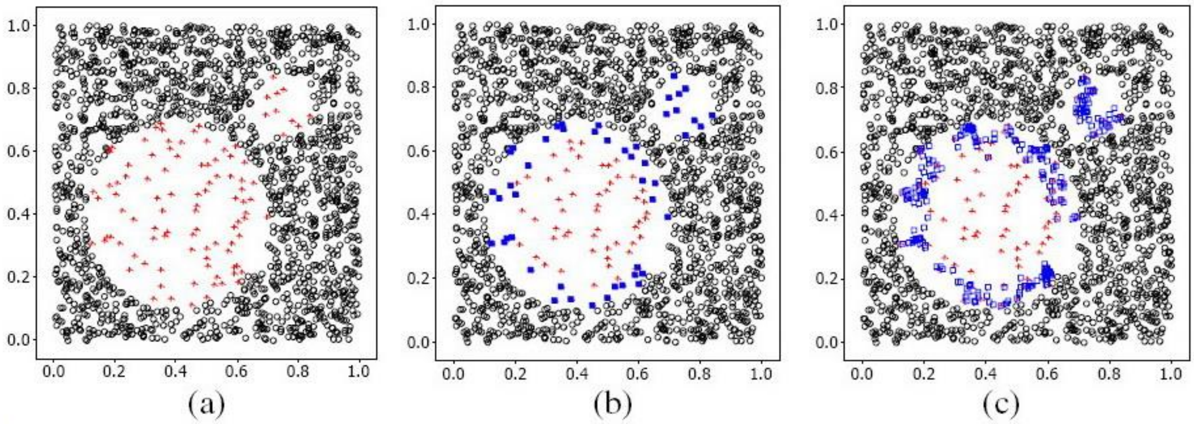
รูปที่ 7.4 บอร์เดอร์ไลน์สโมท

ตัวอย่างของบอเดอร์ไลน์แสดงตามรูปที่ 7.5 เมื่อ n = 3 และ k = 5 ซึ่งเป็นไปตามเงื่อนไข $2.5 \leq n < 5$



รูปที่ 7.5 บอร์เดอร์ไลน์ของชั้นด้อย

บอร์เดอร์ไลน์สโมทมีความเหมาะสมกับเซตข้อมูลที่รูปร่างของชั้นด้อยมีลักษณะเป็น ทรงกลม (Spherical) และเกิดการซ้อนทับกับแบบจากชั้นข้อมูลเด่นที่บริเวณขอบเป็นจำนวนมาก แสดงตามรูปที่ 7.6 เมื่อสัญลักษณ์สีแดงแทนแบบด้อย สัญลักษณ์สีดำแทนแบบเด่น สัญลักษณ์สีน้ำเงินทึบแสดงบอร์เดอร์ไลน์ สัญลักษณ์สีน้ำเงินกลวงแทนแบบสังเคราะห์ เนื่องจากบริเวณของบอร์เดอร์ไลน์อยู่ที่ขอบ ดังนั้นเมื่อทำการสโมทแล้วบริเวณขอบจะมีความหนาแน่นของชั้นด้อยมากยิ่งขึ้น ส่งผลให้ตัวจำแนกแบบทำนายแบบด้อยในบริเวณซ้อนทับได้อย่างมีประสิทธิภาพมากยิ่งขึ้น



รูปที่ 7.6 ชั้นข้อมูลรูปทรงกลม

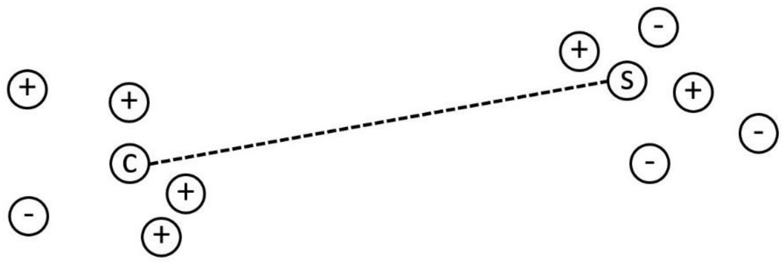
(a) ก่อนการสังเคราะห์ (b) ตรวจจับบอร์เดอร์ไลน์ (c) หลังการสังเคราะห์

เซฟเลเวลสโมท

เทคนิคเซฟเลเวลสโมทมีพื้นฐานอยู่บนสโมทดั้งเดิม ที่สร้างแบบสังเคราะห์บนส่วนของเส้นตรงระหว่างแบบด้อยและเพื่อนบ้านใกล้ที่สุด อย่างไรก็ตาม ในเทคนิคใหม่นี้ความยาวของส่วนของเส้นตรงจะอยู่ขึ้นกับ ระดับความปลอดภัย (SL: Safe Level) ของแบบและเพื่อนบ้าน โดยที่แบบสังเคราะห์จะอยู่ในตำแหน่งที่ใกล้กับแบบที่มีค่าระดับความปลอดภัยสูงกว่า ระดับความปลอดภัยคำนวณได้จากสมการด้านล่างนี้

SL = จำนวนแบบด้อยท่ามกลางเพื่อนบ้านใกล้ที่สุด k ตัว

ยกตัวอย่างเช่น แบบ c จากรูปที่ 7.7 มีเพื่อนบ้านใกล้ที่สุดเคตัวแสดงตามรูปวงกลมที่ล้อมรอบ โดยที่วงกลมบวกแทนแบบด้อยและวงกลมลบแทนแบบเด่น ดังนั้นระดับความปลอดภัยของแบบ c เท่ากับ 4 ในทำนองเดียวกัน ระดับความปลอดภัยของแบบ s เท่ากับ 2 ระดับความปลอดภัยของแบบด้อยยิ่งมีค่าสูง หมายความว่าปริภูมิรอบแบบนั้นมีความปลอดภัยในการสร้างแบบสังเคราะห์ เนื่องจากแบบสังเคราะห์จะอยู่ท่ามกลางแบบด้อยด้วยกัน ในทางกลับกัน ถ้าระดับความปลอดภัยมีค่าต่ำใกล้เคียง 0 แบบนั้นจะถูกพิจารณาเป็น นอยซ์ (Noise) และไม่เหมาะสมที่จะสร้างแบบสังเคราะห์ในบริเวณนั้น เพราะมีโอกาสซ้อนทับกับแบบเด่น



รูปที่ 7.7 แบบด้อย c และ เพื่อนบ้านใกล้ที่สุด s

อัตราส่วนระดับความปลอดภัย (SLR: Safe Level Ratio) คืออัตราส่วนระหว่างระดับความปลอดภัยของแบบด้อย c และเพื่อนบ้านใกล้ที่สุด s ยกตัวอย่างเช่น จากรูปที่ 7.7 อัตราส่วนระดับความปลอดภัยมีค่าเท่ากับ 2 นอกจากนี้ SLR ถูกใช้ในการกำหนดความยาวและตำแหน่งของส่วนของเส้นตรงที่จะสร้างแบบสังเคราะห์ โดยที่อัตตราส่วนนี้คำนวณได้จากสมการด้านล่าง

$$SLR = \frac{SL_c}{SL_s}$$

จากค่าระดับความปลอดภัยของแบบด้อยและเพื่อนบ้านใกล้ที่สุดที่คำนวณได้ นำมาคำนวณค่าอัตราส่วนความปลอดภัย พบว่าสามารถแบ่งแยกออกได้เป็น 5 กรณีที่แตกต่างกัน แสดงตามรูปที่ 7.8 ในแต่ละกรณีจะใช้การสุ่มเพิ่มที่แตกต่างกัน คำอธิบายในแต่ละกรณี มีดังต่อไปนี้

กรณีที่ 1

เงื่อนไข $SL_c = SL_s = 0$

การสุ่มเพิ่ม เนื่องจากทั้ง c และ s ถูกพิจารณาเป็นนอยซ์ ดังนั้น กรณีนี้จะไม่มีการสร้างแบบสังเคราะห์ เนื่องจาก เซฟเลเวลสโมทไม่ให้ความสำคัญกับบริเวณที่มีนอยซ์บรรจุอยู่

กรณีที่ 2

เงื่อนไข $SL_c \neq 0$ and $SL_s = 0$

การสุ่มเพิ่ม มีเพียงแค่ s ที่เป็นนอยซ์ ดังนั้น แบบสังเคราะห์ถูกสร้างบนส่วนของเส้นตรงให้ไกล s มากที่สุด โดยการคัดลอก c กรณีนี้เป็นกรณีเดียวที่ทำการคัดลอกแบบแทนการสังเคราะห์แบบ

กรณีที่ 3

เงื่อนไข $SL_c = SL_s \neq 0$

การสุ่มเพิ่ม ไม่มีแบบใดเป็นนอยซ์ ดังนั้น สร้างแบบสังเคราะห์บนส่วนของเส้นตรงระหว่างแบบด้อยและเพื่อนบ้านใกล้ที่สุด กรณีนี้เหมือนสโมท

กรณีที่ 4

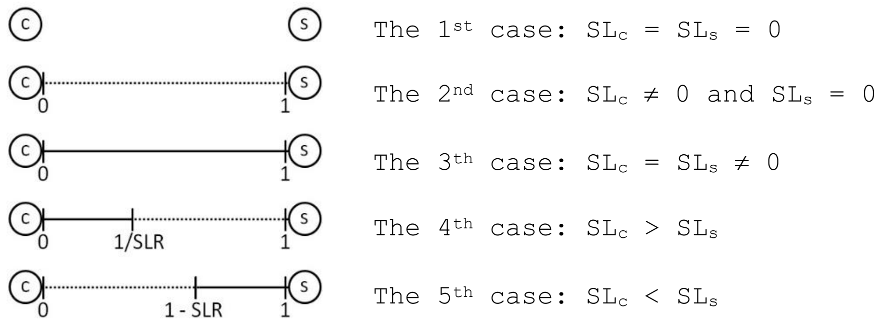
เงื่อนไข $SL_c > SL_s$

การสุ่มเพิ่ม ระดับความปลอดภัยของ c มากกว่า s ดังนั้น แบบสังเคราะห์ถูกสร้างใกล้ c มากกว่า s เนื่องจาก ตำแหน่งของส่วนของเส้นตรงค่อนมาทาง c มากกว่าทาง s

กรณีที่ 5

เงื่อนไข $SL_c < SL_s$

การสุ่มเพิ่ม ระดับความปลอดภัยของ c น้อยกว่า s ดังนั้น แบบสังเคราะห์ถูกสร้างใกล้ s มากกว่า c เนื่องจาก ตำแหน่งของส่วนของเส้นตรงค่อนมาทาง s มากกว่าทาง c



รูปที่ 7.8 กรณีทั้ง 5 ของการสุ่มเพิ่ม

เซฟเลเวลสโมทสามารถรับมือได้ดีกับ ปัญหาซ้อนทับ (Over-lapping Problem) เนื่องจากแบบสังเคราะห์ถูกสร้างให้อยู่ใกล้กับกลุ่มของแบบด้อยมากกว่ากลุ่มของแบบเด่น ดังนั้น ตัวจำแนกแบบสามารถแบ่งแยกแบบจากชั้นข้อมูลต่างกันได้ อย่างไรก็ตามเทคนิคนี้ไม่ได้เน้นบริเวณที่แบบด้อยมีความหนาแน่นสูงมาก จึงอาจเป็นไปได้ที่ตัวจำแนกแบบทำนายแบบจากบริเวณนี้ผิดพลาดได้

ดีบีสโมท

ดีบีสโมทมีความแตกต่างจากสโมทพื้นฐานค่อนข้างมาก เนื่องจากไม่ได้สร้างแบบสังเคราะห์บนส่วนของเส้นตรงระหว่างแบบด้อยและเพื่อนบ้านใกล้ที่สุด แต่ดำเนินการบนเส้นทางระหว่างแบบในชั้นด้อยไปยังจุดศูนย์กลางของชั้นด้อยแทน วิธีการนี้เป็นการเน้นบริเวณแกนกลางของชั้นด้อยให้มีความหนาแน่นมากยิ่งขึ้น นิยามต่าง ๆ ที่เกี่ยวข้อง มีดังต่อไปนี้

นิยาม 1: เพื่อนบ้านอีพีเอส (Eps-Neighborhood)

กำหนดให้ D แทนเซตข้อมูล เพื่อนบ้านอีพีเอส ของแบบ p ใช้สัญลักษณ์ $N_{Eps}(p)$ มีนิยามคือ

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

นิยาม 2 : การเข้าถึงความหนาแน่นโดยตรง (Directly Density-reachable)

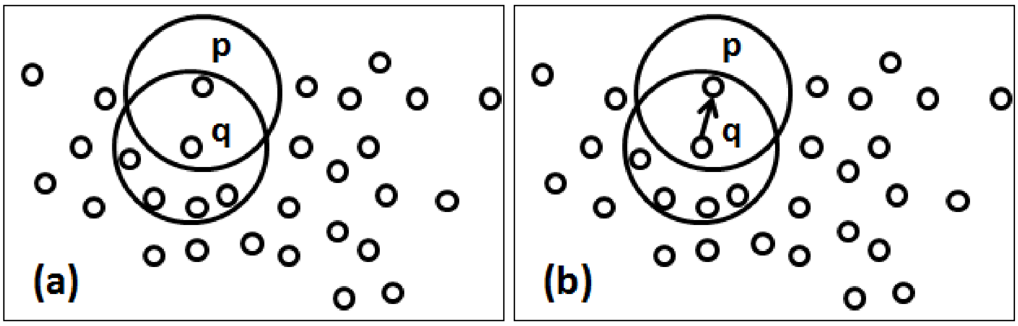
แบบ p ถูกเข้าถึงความหนาแน่นโดยตรงจากแบบ q ที่ขึ้นอยู่กับค่า Eps และ MinPts ถ้า

1. $p \in N_{Eps}(q)$ และ
2. $N_{Eps}(q)| \geq MinPts$ (เงื่อนไขแบบแกน)

จากนิยาม 1 ฟังก์ชัน dist คืนค่าระยะทางระหว่างแบบ p และ q

จากนิยาม 2 แบบแกน (Core Instance) เป็นไปตามเงื่อนไขที่ 2 ในขณะที่ แบบขอบ (Border Instance) ไม่เป็นไปตามเงื่อนไขนี้ นอกจากนี้ การเข้าถึงควาหนาแน่นโดยตรงต้องมาจากแบบแกนเท่านั้น

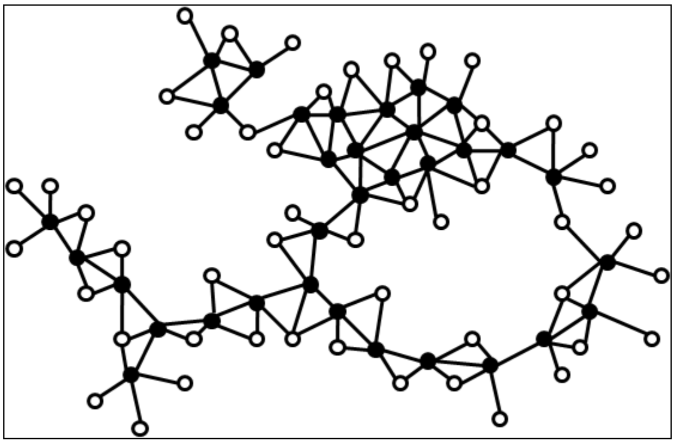
ในรูปที่ 7.9 กำหนดให้ MinPts = 5 จุดในรูปแทนแบบในเซตข้อมูล วงกลมแทนอาณาเขตที่ครอบคลุมจากจุดศูนย์กลาง เมื่อรัศมีของวงกลมมีความยาว Eps ลูกศรแสดงทิศทางของการเข้าถึงความหนาแน่นโดยตรง ในรูปย่อย (a) แบบ p เป็นสมาชิกของเพื่อนบ้านอีพีเอสของแบบ q ในรูปย่อย (b) แบบ p ถูกเข้าถึงความหนาแน่นโดยตรงจากแบบ q แต่เนื่องจากแบบ q ไม่ใช่แบบแกน ดังนั้นแบบ p จึงไม่ถูกเข้าถึงความหนาแน่นโดยตรงจากแบบ q



รูปที่ 7.9 (a) เพื่อนบ้านอีพีเอส (b) การเข้าถึงความหนาแน่นโดยตรง

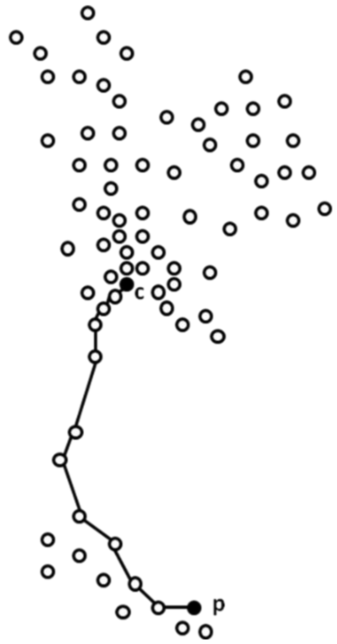
นิยาม 3: กราฟการเข้าถึงความหนาแน่นโดยตรง (Directly density-reachable graph)
กราฟการเข้าถึงความหนาแน่นโดยตรงของกลุ่มข้อมูล C ที่ถูกจัดกลุ่มข้อมูลโดยวิธีการ DBSCAN ใช้สัญลักษณ์ G(C) = (V, E) เมื่อ V คือเซตของโหนดที่แทนแบบใน C และ E คือเซตของเส้นเชื่อม โดยที่ E ถูกนิยามดังนี้ E = {(v_1, v_2) $\in$ V×V | จากค่า Eps และ MinPts แบบ v_1 ถูกเข้าถึงความหนาแน่นโดยตรงจากแบบ v_2 หรือ v_2 ถูกเข้าถึงโดยตรงจากแบบ v_1} กำหนดให้ w: E $\longrightarrow$ R แทนฟังก์ชันน้ำหนักเมื่อ w(v_1, v_2) มีค่าเท่ากับระยะทางระหว่างโหนด v_1, v_2 $\in$ V

กราฟการเข้าถึงความหนาแน่นโดยตรงแสดงตามรูปที่ 7.10 เมื่อจุดสีดำแทนแบบแกน จุดสีขาวแทนแบบขอบ เส้นเชื่อมแทนการเข้าถึงโดยตรง สำหรับแต่ละเส้นเชื่อมต้องมีอย่างน้อยหนึ่งโหนดที่เป็นแบบแกน เนื่องจากการเข้าถึงโดยตรงไม่สามารถเกิดขึ้นระหว่างแบบขอบสองแบบได้ ซึ่งเป็นไปตามนิยาม 2



รูปที่ 7.10 กราฟการเข้าถึงความหนาแน่นโดยตรง

ขั้นตอนวิธีดีบีสโมทสุ่มเพิ่มแบบสังเคราะห์ตาม เส้นทางที่สั้นสุด (Shortest Path) ระหว่างแต่ละแบบด้อยไปยัง เซนทรอยด์เทียม (Pseudo-centroid) โดยที่ เซนรรอยด์เทียมคือแบบด้อยที่ใกล้เซนทรอยด์จริงมากที่สุด เมื่อการสุ่มเพิ่มของแบบด้อยทุกแบบสิ้นสุดลง กลุ่มของแบบสังเคราะห์จะมีความหนาแน่นบริเวณเซนทรอยด์เทียม และเบาบางในบริเวณที่ห่างไกลจากเซนทรอยด์เทียมนี้ ผลที่ได้คือตัวจำแนกแบบสามารถดักจับแบบด้อยบริเวณแกนของชั้นด้อยได้ดีขึ้น รูปที่ 7.10 แสดงเส้นทางสั้นที่สุดระหว่างแบบด้อย p ไปยังเซนทรอยด์เทียมในกราฟการเข้าถึงความหนาแน่นโดยตรง สำหรับรูปนี้ละเว้นการแสดงเส้นเชื่อมของกราฟ แสดงเพียงตัวอย่างของเส้นทางสั้นที่สุดเท่านั้น



รูปที่ 7.10 เส้นทางสั้นที่สุดในกราฟการเข้าถึงความหนาแน่นโดยตรง

5. มิวท์

มิวท์ (MUTE: Majority Under-sampling Technique) คือเทคนิคการสุ่มลดสำหรับกำจัดแบบเด่นประเภทต่าง ๆ ที่มีผลกระทบต่อประสิทธิภาพการจำแนกแบบ ปริภูมิที่ทำการลบแบบเด่นออกจากเซตข้อมูลอสมดุลนี้ อาจเจาะจงในบริเวณใดบริเวณหนึ่งหรือหลายบริเวณก็ได้ เทคนิคมิวท์ประยุกต์ใช้ค่าความปลอดภัยตามสมการที่ 7.1 จากเทคนิคเซฟเลเวลสโมท อย่างไรก็ตาม เซฟเลเวลสโมทคำนวณค่าความปลอดภัยของแบบด้อย แต่มิวท์คำนวณค่านี้จากแบบเด่นแทน ดังนั้นการตีความค่านี้จึงมีความแตกต่างกัน กล่าวคือ ถ้าค่าความปลอดภัยของแบบเด่นมีค่าสูง หมายความว่าแบบเด่นอยู่ท่ามกลางแบบด้อยซึ่งทำให้เกิดปัญหาซ้อนทับ และแบบเด่นนี้จึงควรถูกกำจัดออกไป ในทางกลับกัน ถ้าค่าความปลอดภัยมีค่าต่ำ แสดงว่าแบบเด่นอยู่ท่ามกลางแบบจากชั้นข้อมูลเดียวกัน สามารถเก็บไว้ในเซตข้อมูลได้ โดยไม่เกิดผลกระทบต่อตัวจำแนกแบบ

กำหนดให้ D คือ เซตข้อมูลอสมดุล n คือ สมาชิกจากชั้นเด่น และ SL_n คือ ระดับความปลอดภัยของแบบเด่น n ดังนั้น เซตของแบบจากชั้นเด่นแต่ละประเภท นิยามได้ดังนี้

$$NOISE = \{n \in D \mid SL_n = k\}$$

$$BODERLINE = \{n \in D \mid ½\, k \leq SL_n < k\}$$

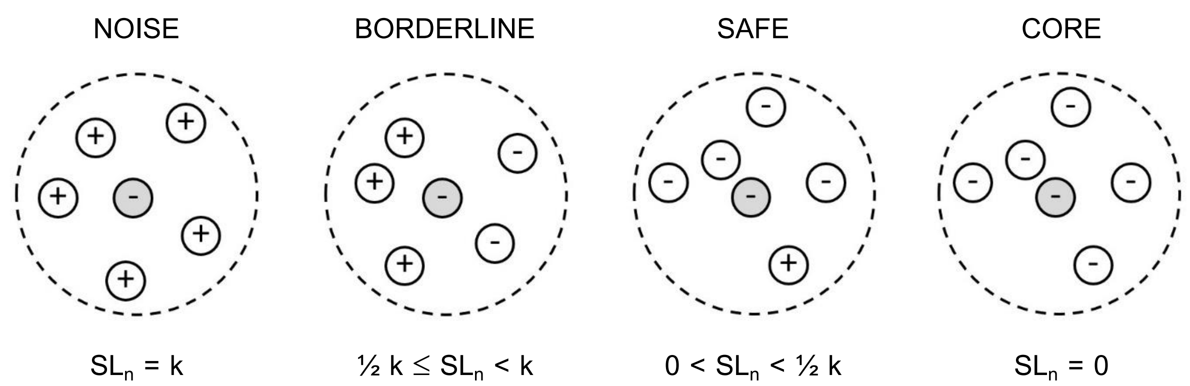
$$SAFE = \{n \in D \mid 0 < SL_n < ½\, k\}$$

$$CORE = \{n \in D \mid SL_n = 0\}$$

แบบเด่น นอยซ์ (Noise) บอร์เดอร์ไลน์ (Borderline) เซฟ (Safe) และ คอร์ (Core) มีรายละเอียดดังนี้

- **นอยซ์** (แบบเด่นนอยซ์) คือ แบบเด่นที่มีความแตกต่างจากแบบเด่นส่วนใหญ่ โดยจัดเป็นสิ่งรบกวนในชั้นเด่น เนื่องจากมีความคล้ายกับแบบด้อยมากกว่า
- **บอร์เดอร์ไลน์** (แบบเด่นบอร์เดอร์ไลน์) คือ แบบเด่นที่อยู่บริเวณขอบของชั้นเด่น การซ้อนทับกับแบบจากชั้นด้อยจะเกิดขึ้นมากที่บริเวณนี้
- **เซฟ** (แบบเด่นเซฟ) คือ แบบเด่นที่อยู่ภายในขอบ อยู่ในบริเวณที่มีความปลอดภัยสูงมากในกระบวนการเรียนรู้ของตัวจำแนกแบบ
- **คอร์** (แบบเด่นคอร์) คือ แบบเด่นที่อยู่ที่แกนของชั้นเด่น บริเวณนี้มีความปลอดภัยสูงสุดในการเรียนรู้ของตัวจำแนกแบบ

ตัวอย่างของแบบเด่นประเภทต่าง ๆ แสดงตามรูปที่ 7.11 ด้านล่างนี้

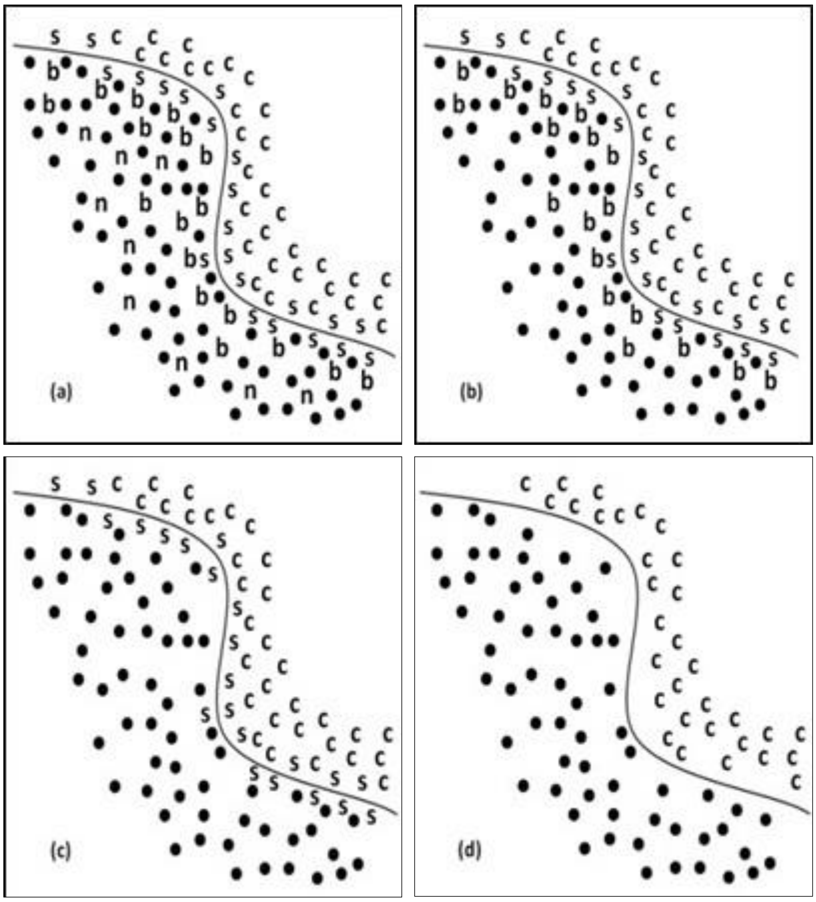


รูปที่ 7.11 นอยซ์ บอร์เดอร์ไลน์ เซฟ และ คอร์ ของชั้นเด่น

เทคนิคมิวท์แบ่งออกเป็น 3 เทคนิคย่อย ดังนี้

- MUTE(D, k) คือ เทคนิคสุ่มลดที่กำจัดแบบเด่นที่มีค่า $SL_n \geq k$ (เฉพาะนอยซ์เพียงประเภทเดียวเท่านั้น) เทคนิคนี้เน้นการกำจัดสิ่งแปลกปลอมในชั้นด้อย
- MUTE(D, ½k) คือ เทคนิคสุ่มลดที่กำจัดแบบเด่นที่มีค่า $SL_n \geq ½k$ (ประเภทนอยซ์และบอร์เดอร์ไลน์) เทคนิคนี้เน้นการกำจัดแบบเด่นที่มีการซ้อนทับกับแบบด้อย
- MUTE(D, 1) คือ เทคนิคสุ่มลดที่กำจัดแบบเด่นที่มีค่า $SL_n \geq 1$ (ทุกประเภทยกเว้นคอร์) เทคนิคนี้ต้องการเก็บไว้แต่เพียงแบบเด่นที่มีความบริสุทธิ์มากเท่านั้น

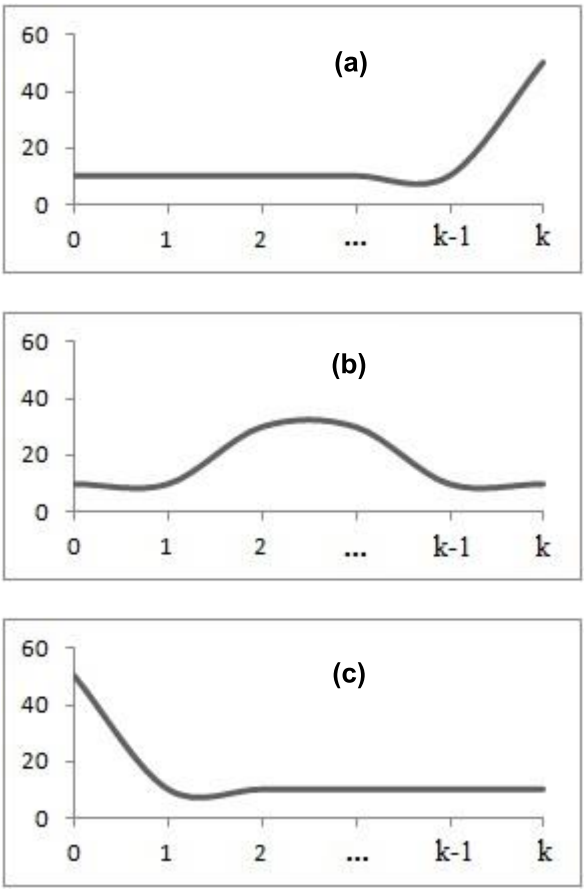
การกระจายตัวของแบบเด่นแต่ละประเภทในบริเวณซ้อนทับแสดงตามรูปที่ 7.12 โดยที่ วงกลมสีดำแทนแบบด้อย ตัวอักษร n b s c แทนแบบเด่นประเภท นอยซ์ บอร์เดอร์ไลน์ เซฟ คอร์ ตามลำดับ รูปย่อย (a) คือบริเวณที่ยังไม่ผ่านการสุ่มลดใด ๆ ใน เซตต้นฉบับ (Original Dataset) รูปย่อย (b)-(d) คือผลลัพธ์ที่เกิดจากการใช้มิวท์แต่ละเทคนิค



รูปที่ 7.12 บริเวณซ้อนทับของ (a) Original (b) MUTE(D, k) (c) MUTE(D, ½k) (d) MUTE(D, 1)

6. เซฟเลเวลกราฟ

เซฟเลเวลกราฟ (**Safe Level Graph**) คือ กราฟสำหรับอธิบายการกระจายของแบบด้อย โดยที่แกนนอนแทนค่าความปลอดภัย และแกนตั้งแทนร้อยละของแบบด้อย เซฟเลเวลกราฟมีการกระจาย 3 ลักษณะ ได้แก่ เบ้ซ้าย (Skewed to the Left) ปกติ (Normal) และ เบ้ขวา (Skewed to the Right) แสดงตาม รูปที่ 7.13



รูปที่ 7.13 เซฟเลเวลกราฟ (a) เบ้ซ้าย (b) ปกติ (c) เบ้ขวา

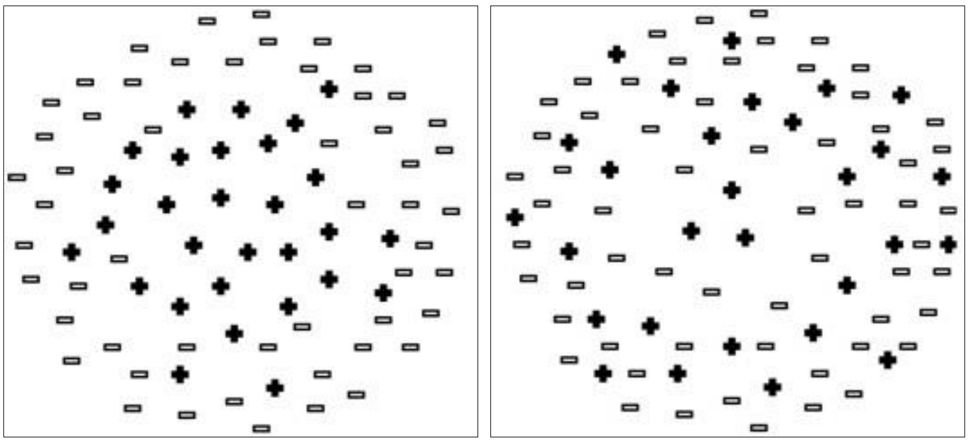
การซ้อนทับของแบบเด่นและแบบด้อยแสดงตามรูปที่ 7.14 โดยที่ สัญลักษณ์ – แสดงแทนแบบจากชั้นเด่น และ สัญลักษณ์ + แสดงแทนแบบจากชั้นด้อย รูปย่อยด้านซ้ายคือลักษณะเบ้ซ้าย และรูปย่อยด้านขวาคือลักษณะเบ้ขวา

เซฟเลเวลกราฟยังถูกใช้เป็นเครื่องมือ ในการแนะนำการเลือกเทคนิคสโมทหรือเทคนิคมิวท์ที่เหมาะสมสำหรับเซตข้อมูลลอสมดุล ตามคำแนะนำของ ตารางที่ 7.2 จากการพิจารณาลักษณะเส้นโค้งของกราฟใน 3 กรณี ได้แก่ เบ้ซ้าย เบ้ขวา ปกติ โดยมีรายละเอียด ดังต่อไปนี้

- ในกรณีที่เซฟเลเวลกราฟมีลักษณะเบ้ซ้าย การกระจายของแบบจากชั้นเด่นและชั้นด้อยแสดงตามรูปย่อยด้านซ้ายของ รูปที่ 7.14
  - เซฟเลเวลสโมทควรถูกใช้ในการเพิ่มความหนาแน่นที่คอร์ของชั้นด้อย เนื่องจากแบบด้อยหนาแน่นบริเวณแกนมากกว่าบริเวณขอบ
  - มิวท์ (D, k) ควรถูกใช้ในการลบนอยซ์ของแบบเด่นเท่านั้น เนื่องจากแบบด้อยซ้อนทับกับแบบเด่นที่บริเวณขอบของชั้นด้อย

- ถ้าเซฟเลเวลกราฟมีลักษณะเบ้ขวา แสดงตามรูปย่อยด้านขวาของ รูปที่ 7.14 เมื่อเปรียบเทียบกับลักษณะเบ้ขวาแล้ว แบบด้อยมีการซ้อนทับกับแบบเด่นมากกว่า และแบบด้อยมีความหนาแน่นบริเวณขอบสูงกว่า
  - บอร์เดอร์ไลน์สโมทควรถูกใช้ในการเพิ่มความหนาแน่นที่ขอบของชั้นด้อย เนื่องจากแบบด้อยเบาบางที่แกนแต่กระจุกตัวที่ขอบ
  - มิวท์ (D, 1) ควรถูกใช้ในการลบแบบเด่นทั้งหมดยกเว้นคอร์ เนื่องจากแบบเด่นมีการซ้อนทับตั้งแต่บริเวณขอบจนถึงแกนของชั้นด้อย

- ถ้าเซฟเลเวลกราฟมีลักษณะปกติ แบบด้อยมีการเกาะตัวที่แกนแต่ไม่หนาแน่นเท่าแกนแบบเบ้ซ้าย อย่างไรก็ตาม แบบด้อยมีการกระจายตัวที่ขอบบ้างประปราย แต่เบาบางกว่าขอบแบบเบ้ขวา
  - มิวท์ MUTE(D, ½k) ควรถูกใช้ในการลบเพียงนอยซ์และบอร์เดอร์ไลน์ของแบบเด่น เนื่องจาก แบบเด่นทั้ง 2 ประเภทนี้ อยู่ในบริเวณซ้อนทับกับชั้นด้อย ทำให้ตัวจำแนกมีความสับสนในการแยกชั้นข้อมูลทั้ง 2 ออกจากกัน



รูปที่ 7.14 ชั้นเด่นและชั้นด้อยกรณีเบ้ซ้ายและเบ้ขวา

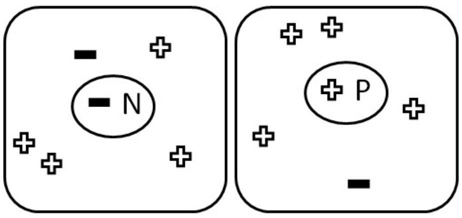
ตารางที่ 7.2 สโมทและมิวท์ที่เหมาะสม

Curve	SMOTE	MUTE
Skewness to the Left	Safe-level-SMOTE	MUTE(D, k)
Normal	No Guideline	MUTE(D, ½k)
Skewness to the Right	Borderline-SMOTE	MUTE(D, 1)

7. คอร์

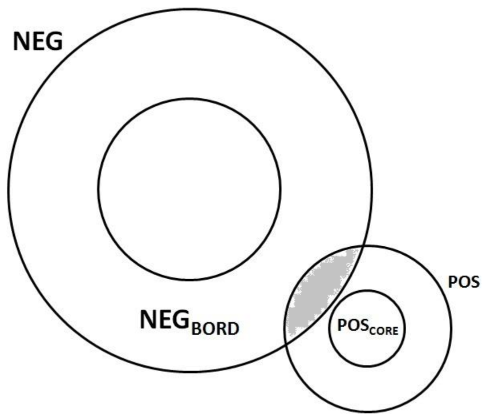
คอร์ (CORE: Core-based Synthetic Minority Over-sampling and Borderline Majority Under-sampling Technique) คือ เทคนิคที่ผสมผสานการสุ่มลดและการสุ่มเพิ่มเข้าด้วยกัน โดยแบ่งหน้าที่การทำงานดังนี้

- คอร์สุ่มลด แบบเด่นบอร์เดอร์ไลน์ ซึ่งแสดงด้วยสัญลักษณ์ N ในรูปที่ 7.15 โดยที่ เครื่องหมาย – แสดงแทนแบบเด่น และเครื่องหมาย + แสดงแทนแบบด้อย แบบเด่นประเภทนี้อยู่ท่ามกลางแบบจากต่างชั้นข้อมูล และเป็นแบบที่ควรถูกกำจัด เนื่องจากอาจรบกวนการเรียนรู้ของตัวจำแนก ในการรู้จำชั้นด้อยให้แย่ลง
- คอร์สุ่มเพิ่ม แบบด้อยคอร์ ซึ่งแสดงด้วยสัญลักษณ์ P ในรูปที่ 7.5 โดยที่ แบบด้อยประเภทนี้อยู่ท่ามกลางแบบจากชั้นข้อมูลเดียวกัน และเป็นแบบที่ควรถูกผลิต เนื่องจากอยู่ในบริเวณที่เหมาะสมให้ตัวจำแนกเรียนรู้ ในการรู้จำชั้นด้อยให้ดีขึ้น



รูปที่ 7.15 บอร์เดอร์ไลน์จากชั้นเด่น และ คอร์จากชั้นด้อย

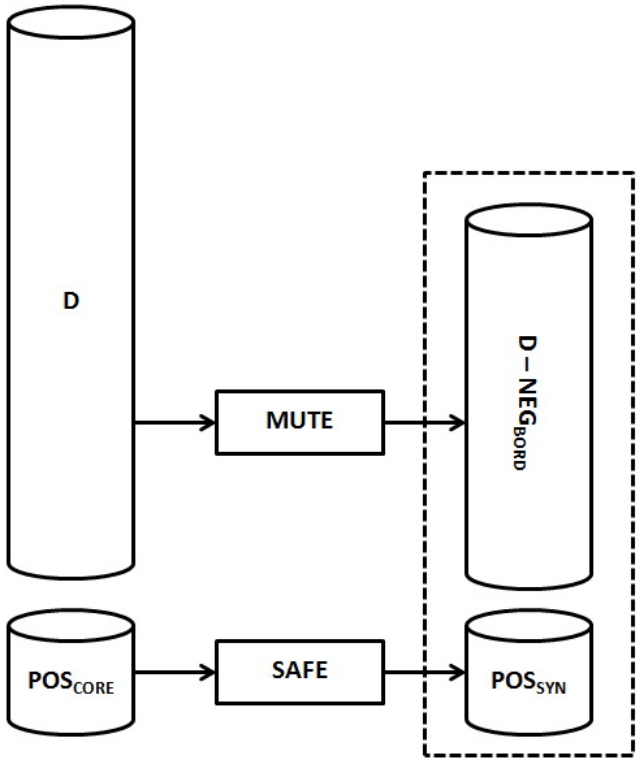
บริเวณซ้อนทับของแบบจากชั้นเด่นและจากชั้นด้อย แสดงตาม ส่วนร่วม (Intersection) ที่ถูกแรเงาในรูปที่ 7.16 โดยที่ NEG คือชั้นเด่น POS คือชั้นด้อย NEG_{BORD} คือแบบเด่นบอร์เดอร์ไลน์ POS_{CORE} คือแบบด้อยคอร์ จากรูปเห็นได้ว่า หลักการของคอร์คือการเพิ่มความหนาแน่นให้กับแกนของชั้นด้อย ทำให้แบบเด่นเบาบางลงบริเวณขอบของชั้นด้อย เพื่อบรรลุวัตถุประสงค์ของคอร์ที่ต้องการให้ชั้นด้อยโดดเด่นขึ้น และสามารถถูกรู้จำได้อย่างมีประสิทธิภาพจากตัวจำแนก



รูปที่ 7.16 บริเวณซ้อนทับระหว่างชั้นเด่นและชั้นด้อย

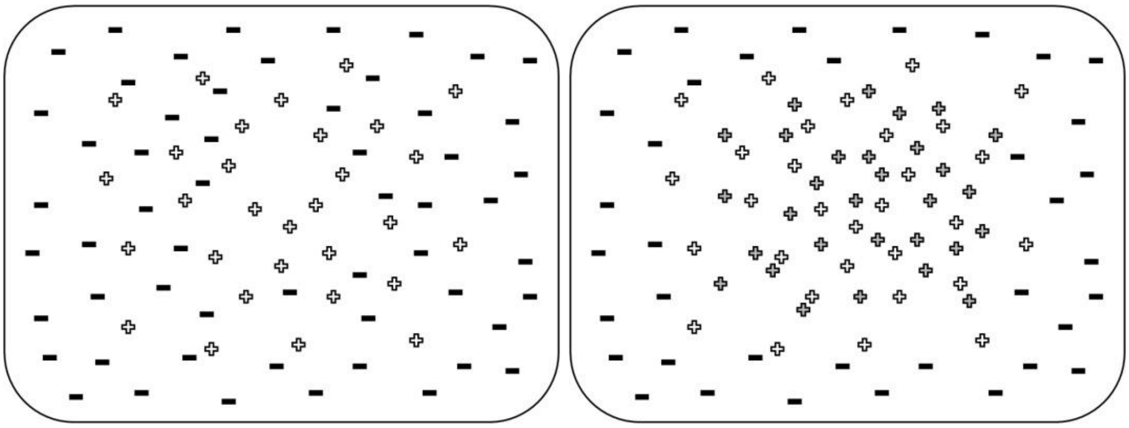
กรอบการทำงานคอร์แสดงตามรูปที่ 7.17 เซฟเลเวลสโมทคือการสุ่มเพิ่มที่ถูกเลือกใช้ และมิวท์คือการสุ่มลดที่ถูกเลือกใช้ นอกจากนี้ สัญลักษณ์ที่เกี่ยวข้องเพิ่มเติม ได้แก่ D คือ เซตข้อมูลอสมดุล และ POS_{SYN} คือ แบบด้อยสังเคราะห์ กรอบการทำงานคอร์ ถูกอธิบาย ดังนี้

1. มิวท์ ลบแบบเด่นแบบเด่นบอร์เดอร์ไลน์ออกจากเซตข้อมูลอสมดุล
2. เซฟเลเวลสโมท สร้างแบบด้อยคอร์เพิ่มเข้าไปในเซตข้อมูลอสมดุล
3. คอร์ ส่งคืน เซตข้อมูลอสมดุลที่ปราศจากแบบเด่นบอร์เดอร์ไลน์ แต่อุดมไปด้วยแบบด้อยคอร์



รูปที่ 7.17 กรอบการทำงานคอร์

ผลลัพธ์ของคอร์แสดงตาม รูปที่ 7.18 สัญลักษณ์ที่เกี่ยวข้องเพิ่มเติม ได้แก่ + แรเงาแสดงแทนแบบด้อยสังเคราะห์ รูปย่อยด้านซ้ายคือบริเวณชั้นด้อยในเซตข้อมูลอสมดุลดั้งเดิม พบว่ามีแบบเด่นกระจายตัวไปทั่วบริเวณตั้งแต่ขอบจนถึงแกนของชั้นด้อย และรูปย่อยด้านขวาคือบริเวณชั้นด้อยในเซตข้อมูลอสมดุลที่ผ่านการสุ่มจากคอร์เรียบร้อยแล้ว พบว่าแกนของชั้นด้อยมีความหนาแน่นเพิ่มขึ้น และขอบของชั้นด้อยสะอาดมากขึ้น



รูปที่ 7.18 บริเวณชั้นด้อยก่อนและหลังการสุ่มโดยคอร์

อภิปราย

ความท้าทายของปัญหาชั้นข้อมูลอสมดุล คือการระบุว่าจำนวนแบบสังเคราะห์ในการสุ่มเพิ่มควรมีปริมาณเท่าไหร่ เพื่อให้ตัวจำแนกแบบมีความแม่นยำในการทำนายชั้นด้อยได้อย่างมีประสิทธิภาพมากที่สุด ในทางกลับกัน จำนวนแบบจากชั้นเด่นที่ควรถูกกำจัดในการสุ่มลดควรมีเท่าไหร่ ตัวเลขเหล่านี้ส่วนใหญ่ได้จากการทดลอง ในส่วนเพิ่มเติม เซฟเลเวลกราฟคือเทคนิคที่ใช้สร้างกราฟสำหรับชั้นข้อมูลเด่นและกราฟสำหรับชั้นข้อมูลด้อย เพื่อให้ผู้ใช้สามารถวิเคราะห์ถึงปริมาณแบบที่เหมาะสมในการสุ่มเพิ่มโดยสโมท และปริมาณแบบที่เหมาะสมในการสุ่มลดโดยมิวท์ได้

เทคนิคสโมททั้ง 4 ประกอบไปด้วย สโมทต้นตำรับ บอร์เดอร์ไลน์สโมท เซฟเลเวลสโมท และดีบีสโมท เทคนิคทั้งหมดมีจุดแข็งและจุดอ่อนที่แตกต่างกัน การอภิปรายสมาชิกในครอบครัวสโมทแสดงตามตารางต่อไปนี้

ตารางที่ 7.3 อภิปรายครอบครัวสโมท

สโมท	แนวคิด	ข้อดี	ข้อเสีย
สโมทดั้งเดิม	การสุ่มเพิ่มเกิดขึ้นอย่างทั่วถึงทุกบริเวณของชั้นด้อย สร้างแบบสังเคราะห์บนส่วนของเส้นตรงที่เชื่อมระหว่างแบบด้อยและเพื่อนบ้านใกล้ที่สุดที่ถูกสุ่มเลือก การกระจายตัวของแบบสังเคราะห์ไม่ได้หนาแน่นไปที่บริเวณใดบริเวณหนึ่งของชั้นด้อย	สโมทถูกบรรจุอยู่ในซอฟต์แวร์ทางด้านการทำเหมืองข้อมูลหลายผลิตภัณฑ์ เช่น เวก้า มีความง่ายในการเขียนโปรแกรม ใช้เวลาประมวลผลเร็วกว่าสโมทประเภทอื่น	ล้มเหลวในการรับมือปัญหาซ้อนทับ เนื่องจากสโมทอาจสร้างแบบสังเคราะห์ในบริเวณที่มีแบบจากชั้นเด่นได้ แบบสังเคราะห์ที่สร้างจากสโมทอาจจะทับซ้อนกับแบบจากชั้นข้อมูลเด่นได้
บอร์เดอร์ไลน์สโมท	การสุ่มเพิ่มเกิดขึ้นกับบอร์เดอร์ไลน์ในบริเวณซ้อนทับเท่านั้น สร้างแบบสังเคราะห์โดยการป้อนบอร์เดอร์ไลน์เป็นข้อมูลเข้าของขั้นตอนวิธีสโมท การกระจายตัวของแบบสังเคราะห์หนาแน่นที่ขอบ	รีคอลมีค่าสูง เพราะว่าแบบด้อยในปริภูมิซ้อนทับที่ถูกทำนายผิดพลาดมีปริมาณน้อย ใช้เวลาประมวลผลเทียบเท่าสโมทพื้นฐาน	พรีซิชันไม่เป็นที่น่าพอใจ เนื่องจากแบบด้อยในบริเวณแกนกลางของชั้นด้อยอาจถูกทำนายผิดพลาดได้มาก หน่วยวัดเอฟไม่เป็นที่น่าพอใจ เนื่องจากพรีซิชันมีค่าต่ำ

ตารางที่ 7.3 อภิปรายครอบครัวสโมท (ต่อ)

ครอบครัวสโมท	แนวคิด	ข้อดี	ข้อเสีย
เซฟเลเวล สโมท	แต่ละแบบด้อยมีค่าระดับความปลอดภัยคำนวณจากจำนวนของแบบจากชั้นด้อยท่ามกลางเพื่อนบ้านใกล้ที่สุดเคตัว แต่ละแบบสังเคราะห์ถูกสุ่มเพิ่มบนส่วนของเส้นตรงเดียวกันกับสโมท แต่ถูกสร้างให้อยู่ใกล้แบบที่มีค่าระดับความปลอดภัยสูงกว่า ความหนาแน่นของแบบสังเคราะห์มีลักษณะใกล้เคียงกับของสโมท	สังเคราะห์ถูกสร้างให้เข้าใกล้แบบด้อยมากกว่าแบบเด่น	กรณีที่ชั้นเด่นไม่เกิดการซ้อนทับกับชั้นด้อยผลลัพธ์ที่ได้แทบไม่แตกต่างจากสโมท แต่ต้องใช้เวลาประมวลผลสูงกว่าสโมท
ดีบีสโมท	แบบสังเคราะห์ถูกสร้างจากกราฟการเข้าถึงความหนาแน่นโดยตรงท่ามกลางเส้นทางสั้นที่สุดจากแต่ละแบบด้อยไปยังเซนทรอยด์เทียมของชั้นด้อย การกระจายตัวของแบบสังเคราะห์มีความหนาแน่นบริเวณเซนทรอยด์เทียมและเบาบางในบริเวณที่ไกลจากเซนทรอยด์เทียม รักษารูปทรงของชั้นข้อมูลไว้ได้เหมือนเดิม	รับมือปัญหาซ้อนทับได้อย่างมีประสิทธิภาพ เนื่องจากแบบสังเคราะห์มีแนวโน้มที่จะเข้าใกล้แกนกลางของชั้นด้อยมากกว่าขอบของชั้นด้อย ดังนั้นแบบสังเคราะห์จึงไม่เกิดขึ้นบ่อยครั้งในปริภูมิซ้อนทับ	กรณีที่ชั้นเด่นไม่เกิดการซ้อนทับกับชั้นด้อยผลลัพธ์ที่ได้แทบไม่แตกต่างจากสโมท แต่ต้องใช้เวลาประมวลผลสูงกว่าสโมท

สรุป

เซตข้อมูลมีลักษณะอสมดุลก็ต่อเมื่อชั้นข้อมูลมีสัดส่วนที่แตกต่างกันอย่างเห็นได้ชัดเจน ส่งผลให้ตัวจำแนกมีอคติต่อชั้นด้อย กล่าวคือไม่สนใจค่าความแม่นยำของชั้นข้อมูลนี้ ดังนั้นต้องใช้ตัววัดอื่นได้แก่ รีคอล และ พรีซิชัน ซึ่งเจาะจงไปที่ชั้นข้อมูลด้อย หน่วยวัดสมรรถนะเหล่านี้ได้มาจากเมทริกซ์สับสน เทคนิคที่ใช้ในการแก้ปัญหาชั้นข้อมูลอสมดุลแบ่งออกเป็น 2 ประเภท ได้แก่ การเพิ่มที่ขยายขนาดชั้นด้อย และ การสุ่มลดที่ลดขนาดของชั้นเด่น

เทคนิคที่ได้รับความนิยมในการใช้งานสูงสุด ได้แก่ สโมทที่สุ่มเพิ่มโดยอาศัยการสร้างข้อมูลสังเคราะห์ที่มีความคล้ายกับข้อมูลจริง สโมทได้รับการพัฒนาปรับปรุงให้มีประสิทธิภาพสูงขึ้นเป็นเทคนิคต่าง ๆ ที่แตกต่างกัน ได้แก่ บอร์เดอร์ไลน์สโมทที่สร้างข้อมูลสังเคราะห์บริเวณขอบของชั้นด้อยเท่านั้น เซฟเลเวลสโมทที่ไม่สร้างข้อมูลบริเวณขอบดังกล่าวแต่สร้างในบริเวณอื่นที่เหลือ ดีบีสโมทที่สร้างข้อมูลบริเวณแกนกลางของชั้นด้อยเท่านั้น เทคนิคที่ทำงานตรงข้ามกับสโมธ ได้แก่ มิวท์ที่สุ่มลดโดยการกำจัดข้อมูลประเภท นอยซ์ บอร์เดอร์ไลน์ เซฟ หรือ คอร์ ในชั้นเด่น โดยมิวท์มีหลายรุ่นให้เลือกใช้ในการลบข้อมูลที่แตกต่างกันออกไป

เซฟเลเวลกราฟถูกใช้ในการอธิบายลักษณะเฉพาะของเซตข้อมูลอสมดุล โดยที่ลักษณะของกราฟแบ่งออกได้เป็น 3 ลักษณะ ได้แก่ เบ้ซ้าย ปกติ และ เบ้ขวา ซึ่งแสดงคุณลักษณะของชั้นข้อมูลที่แตกต่างกันออกไป นอกจากนี้ เซฟเลเวลกราฟสามารถใช้เป็นเครื่องมือในเลือกเทคนิคสโมทหรือมิวท์ที่เหมาะสมได้ คอร์ คือ เทคนิคการสุ่มที่รวมการทำงานของเซฟเลเวลสโมทและมิวท์เข้าด้วยกัน

แบบฝึกหัด

1. จงยกตัวอย่างเซตข้อมูลอสมดุล แล้วให้ระบุ ชั้นเด่น และ ชั้นด้อย รวมถึง เปรียบเทียบความเสียหายของการทายผิด จากแบบเด่นเป็นแบบด้อย และ จากแบบด้อยเป็นแบบเด่น

2. จากเซตข้อมูลมะเร็งเต้านมในตัวอย่างที่ 1 ถ้าตัวจำแนกชนิดหนึ่งทำนายแบบด้อยถูกต้องจำนวน 512 แบบ และ ทำนายแบบเด่นถูกต้องจำนวน 90,360 แบบ จงเขียนเมทริกซ์สับสน แล้วคำนวณค่ารีคอล พรีซิชัน หน่วยวัดเอฟ เมื่อกำหนดให้ β มีค่าเท่ากับ 1

3. กำหนด แบบด้อย (62, -77, -84) และ เพื่อนบ้านใกล้ที่สุด (10, 19, -98) จงแสดงวิธีการสร้างแบบสังเคราะห์ด้วยเทคนิคสโมท เมื่อกำหนดให้ตัวเลขสุ่มมีค่าเท่ากับ 0.25

4. MUTE(D, 0) คือ เทคนิคสุ่มลดที่กำจัดแบบเด่นทุกประเภทในเซตข้อมูลอสมดุล จงอธิปราย มิวท์ประเภทนี้มีข้อดีหรือข้อเสียอย่างไร

5. จงปรับปรุงคอร์ โดยการรวมการสุ่มเพิ่มที่ไม่ใช่เซฟเลเวลสโมทและการสุ่มลดที่ไม่ใช่มิวท์ แล้วอธิบายกรอบการทำงาน นอกจากนี้ ให้อธิปราย ประสิทธิภาพของคอร์ใหม่เปรียบเทียบกับคอร์ดั้งเดิม

บรรณานุกรม

[1] Blake, C. L., Merz, C. J.: The UC Irvine Machine Learning Repository. http://archive.ics.uci.edu/ml/. University of California, Irvine, CA (1998)

[2] Bunkhumpornpat, C., Sinapiromsaran, K.: CORE: Core-based Synthetic Minority Over-sampling and Borderline Majority Under-sampling Technique. International Journal of Data Mining and Bioinformatics (IJDMB), vol. 12, no. 1, pp. 44-58 (2015)

[3] Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C.: DBSMOTE: Density-Based Synthetic Minority Over-sampling TEchnique. Applied Intelligence (APIN), vol. 36, no. 3, pp. 664-684 (2012)

[4] Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C.: MUTE: Majority Under-sampling Technique. In: 8th International Conference on Information, Communications, and Signal Processing (ICICS), pp. 1-4. Singapore (2011)

[5] Bunkhumpornpat, C., Sinapiromsaran, K.: Safe Level Graph for Majority Under-sampling Techniques. Chiang Mai Journal of Science (CMJS), vol. 41, no. 5/2, pp. 1419-1428 (2014)

[6] Bunkhumpornpat, C., Subpaiboonkit, S.: Safe Level Graph for Synthetic Minority Over-sampling Techniques. In: 13th International Symposium on Communications and Information Technologies (ISCIT), pp. 570-575. Samui Island, Thailand (2013)

[7] Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C.: Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-sampling TEchnique for handling the class imbalanced problem. In: Theeramunkong, T., Kijsirikul, B., Cercone, N., Ho, T.-B. (eds.) 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD). Bangkok, Thailand. Lecture Notes in Artificial Intelligence (LNAI), vol. 5476, pp. 475-482. Springer, Heidelberg (2009)

[8] Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P.: SMOTE: Synthetic Minority Over-Sampling TEchnique. Journal of Artificial Intelligence Research, vol. 16, pp. 341-378 (2002)

[9] Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) the 2005 International Conference on Intelligent Computing, Hefei, China. Lecture Notes in Computer Science, vol. 3644, pp. 878-887. Springer, Heidelberg (2005)

[10] Han, J., Kamber, M., Pei, J.: *Data Mining: Concepts and Techniques*, 3rd Edition. Morgan Kaufman (2011)

[11] He, H., Ma, Y.: *Imbalanced Learning: Foundations, Algorithms, and Applications*. Wiley (2013)

[12] Kaukoranta, T., Smed, J., Hakonen, H.: *Role of Pattern Recognition in Computer Games*, Proceedings of the 2nd International Conference on Application and Development of Computer Games, pp. 189–94, Hong Kong SAR, China (2003)

[13] Moris Mano, M., Computer System Architecture, 3rd ed. NJ: Prentice Hall (1992)

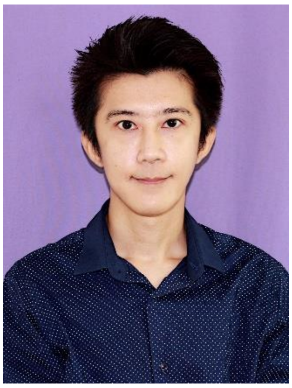
[14] Murty, M. N., Devi, V. S.: *Pattern Recognition: An Algorithmic Approach (Undergraduate Topics in Computer Science)*. Springer (2012)

[15] Tortora, G. J., Derrickson, B.: *Introduction to Human Body: The Essential of Anatomy and Physiology*, 8th Edition. Hoboken, NJ: John Wiley & Sons, Inc. (2010)

[16] Witten, I. H., Frank, E., Hall, M. A.: *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd Edition. Morgan Kaufman (2011)

ประวัติผู้แต่ง

ผู้แต่งสำเร็จการศึกษาปริญญาตรีสาขาวิศวกรรมคอมพิวเตอร์จากมหาวิทยาลัยรังสิต และปริญญาโทสาขาวิทยาการคอมพิวเตอร์จากมหาวิทยาลัยเชียงใหม่ แล้วเริ่มต้นทำงานตำแหน่งอาจารย์ในปีเดียวกัน ณ ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่ หลังจากนั้น ลาศึกษาต่อระดับปริญญาเอกจนกระทั่งสำเร็จการศึกษา สาขาวิทยาการคอมพิวเตอร์จากจุฬาลงกรณ์มหาวิทยาลัย ต่อมาได้รับการแต่งตั้งให้ดำรงตำแหน่งผู้ช่วยศาสตราจารย์ในปี พ.ศ. ๒๕๕๗ นอกจากนี้ ผู้แต่งมีประสบการณ์สอนและความเชี่ยวชาญในงานสอนและงานาวิจัย ทางด้านการรู้จำแบบและวิทยาการข้อมูล ผู้แต่งมีผลงานวิจัยตีพิมพ์ในวารสารวิชาการนานาชาติจำนวนหลายฉบับ โดยได้รับทุนจาก สำนักงานกองทุนสนับสนุนการวิจัย (สกว.) สำนักงานการวิจัยแห่งชาติ (วช.) ทุนโครงการปริญญาเอกกาญจนาภิเษก (คปก.) และจากหลายหน่วยงาน



ชุมพล บุญคุ้มพรภัทร