บทที่ 3
ซัพพอร์ตเวกเตอร์แมทชีน

เนื้อหาในบทนี้จะกล่าวถึงวิวัฒนาการของโครงข่ายประสาทเทียม และการพัฒนาจนกลายมาเป็นซัพพอร์ตเวกเตอร์แมทชีน ทฤษฎีของซัพพอร์ตเวกเตอร์แมทชีน ชนิดของซัพพอร์ตเวกเตอร์แมท ชีนและการประยุกต์ใช้งานซัพพอร์ตเวกเตอร์แมทชีนในงานด้านต่างๆ ที่เกี่ยวข้อง

3.1 วิวัฒนาการของโครงข่ายประสาทเทียม

โดยทั่วไป การใช้งานโครงข่ายประสาทเทียม (Artificial Neural Network) [20] จะต้องมีการกำหนดพารามิเตอร์หลายตัว เช่น จำนวนของชั้นซ่อน (Hidden Layer) จำนวนโหนดของแต่ละชั้นซ่อน (Hidden Node) หรืออัตราการเรียนรู้ (Learning Rate) เป็นต้น ทำให้การเลือกใช้งานแบบจำลองโครงข่ายประสาทเทียมที่ดีนั้นจะต้องอาศัยประสบการณ์จากผู้ใช้งานที่มีความชำนาญทางด้านนี้โดยเฉพาะ ซึ่งแบบจำลองของโครงข่ายประสาทเทียมมีให้เลือกใช้อยู่หลายแบบด้วยกัน ในที่นี้จะขอยกตัวอย่างการใช้งานของแบบจำลองโครงข่ายประสาทแบบแพร่กลับค่าผิดพลาด (Error Backpropagation Neural Network) โดยเริ่มจากเลือกแบบจำลองเริ่มต้นขึ้นมาโดยการประมาณจากประสบการณ์ จากนั้นทำการทดสอบแบบจำลองเริ่มต้นที่สร้างขึ้นโดยการวัดค่าสมรรถนะ (Performance) ของระบบ ทั้งนี้จะนิยมใช้ค่าผิดพลาดกำลังสองเฉลี่ย (Mean Square Error) ของค่าเอาต์พุตที่ต้องการกับค่าเอาต์พุตที่ได้จากชุดข้อมูลที่นำมาฝึกสอนทั้งหมด และจะต้องทำการทดลองปรับแต่งแบบจำลองไปเรื่อยๆ จนกระทั่งได้ค่าผิดพลาดกำลังสองเฉลี่ยที่น้อยที่สุดจึงจะได้แบบจำลองที่เหมาะที่สุด (Optimal) ซึ่งหมายถึงโครงสร้างของแบบจำลองที่ได้ มีความเหมาะสมเฉพาะชุดของข้อมูลที่ได้เตรียมมาฝึกสอนเท่านั้น

3.2 วิวัฒนาการของการหาน้ำหนักในการเชื่อมต่อ

ขั้นตอนของการสอน (Training) โครงข่ายประสาทเทียมนั้นส่วนมากจะหมายถึง การหาค่าน้ำหนักที่เหมาะที่สุด (Optimal Weight) โดยวัดจากค่าผิดพลาดกำลังสองเฉลี่ยที่น้อยที่สุด และกระบวนการในการปรับหาค่าน้ำหนักที่ใช้ในการเชื่อมต่อในแต่ละชั้นนั้น จะใช้วิธีที่มีพื้นฐานมาจากการลดระดับเกรเดียนต์ (Gradient Descent) ซึ่งโดยมากแล้วค่าที่ได้จะเป็นค่าต่ำสุดเฉพาะที่

(Local Minimum) เท่านั้นไม่ได้เป็นค่าต่ำสุดในวงกว้าง (Global Minimum) จากปัญหาดังกล่าว ได้มีการคิดค้นและพัฒนาแก้ปัญหามาโดยลำดับ ซึ่งก็มีทั้งการนำเอาขั้นตอนวิธีวิวัฒนาการ (Evolutionary Algorithm) [4] มาประยุกต์ใช้สำหรับหาค่าน้ำหนักเริ่มต้นก่อน กล่าวคือ นำขั้นตอนวิธีวิวัฒนาการมาหาค่าน้ำหนักที่ใกล้เคียงกับค่าต่ำสุดในวงกว้าง จากนั้นจึงใช้วิธีการลดระดับเกรเดียนต์ในการหาค่าต่ำที่สุดแบบละเอียดต่อไป

จากที่กล่าวมา ในการแก้ปัญหาจะเห็นได้ว่าจะต้องใช้ทั้งขั้นตอนวิธีวิวัฒนาการ และวิธีการลดระดับเกรเดียนต์ร่วมกัน ซึ่งขั้นตอนวิธีวิวัฒนาการนั้น ต้องอาศัยเวลาในการทำงานมากหรือหากผู้เขียนโปรแกรมออกแบบโปรแกรมได้ไม่ดี ก็ไม่สามารถหาคำตอบที่มีประสิทธิภาพได้

3.3 ซัพพอร์ตเวกเตอร์แมทชีน

ซัพพอร์ตเวกเตอร์แมทชีนคือ โครงข่ายประสาทเทียมรูปแบบใหม่ที่ได้ถูกพัฒนาขึ้นในช่วงต้นทศวรรษที่ 60 [14,15,34] แนวความคิดของการพัฒนาซัพพอร์ตเวกเตอร์แมทชีนคือ เพื่อแก้ไขปัญหาข้อบกพร่องต่างๆ ของโครงข่ายประสาทเทียมแบบดั้งเดิม โดยได้อาศัยพื้นฐานความรู้มาจากเรื่องของทฤษฎีการเรียนรู้ทางสถิติ (Statistical Learning Theory) [42]

ซัพพอร์ตเวกเตอร์แมทชีนได้ถูกนำเสนอเป็นครั้งแรกที่ประเทศสหภาพโซเวียตในปี ค.ศ. 1963 โดย *Vapnik* ในระยะแรกนั้น ซัพพอร์ตเวกเตอร์แมทชีนถูกนำมาใช้กับงานด้านการรู้จำรูปแบบ (Pattern Recognition) หรือใช้สำหรับการแบ่งกลุ่ม (Classification) [1,2,5,13,37] แต่ในระยะต่อมา ได้มีการนำซัพพอร์ตเวกเตอร์แมทชีนไปประยุกต์ใช้สำหรับงานด้านการประมาณค่าฟังก์ชัน (Function Approximation) [8,16-19,21,23,25,27,31,38-44]

โครงสร้างของซัพพอร์ตเวกเตอร์แมทชีน จะมีความคล้ายคลึงกับโครงข่ายประสาทเทียมแบบที่ต้องมีการเตรียมข้อมูลสำหรับการเรียนรู้มาก่อน หรือที่เรียกว่าเป็นการเรียนรู้แบบมีผู้ช่วยสอน (Supervised Learning) [28,34-36] จึงสามารถทำการแบ่งกลุ่ม รู้จำ หรือทำการวิเคราะห์การถดถอย (Regression Analysis) ข้อมูลที่นำมาทดสอบได้

3.3.1 คุณประโยชน์ของซัพพอร์ตเวกเตอร์แมทชีน

Vapnik ได้นำเสนอซัพพอร์ตเวกเตอร์แมทชีน [14] ซึ่งเป็นอัลกอริทึมใหม่ (Novel Algorithm) ที่ได้พัฒนาขึ้นจากการผสมผสานเทคนิควิธีการหลายรูปแบบเข้าด้วยกันคือ ทฤษฎีการเรียนรู้ทางสถิติ เพื่อการเลือกแบบจำลองที่ดีที่สุด อัลกอริทึมการเรียนรู้ของเคอร์เนล (Kernel-Based Learning Algorithm) [10,26,29] ใช้ในการแมป (Map) ข้อมูลอินพุตเพื่อช่วยในการสร้างระนาบเกิน (Hyperplane) อัลกอริทึมของโครงข่ายประสาทเทียม [20] และ ทฤษฎีการหาค่าเหมาะที่สุด

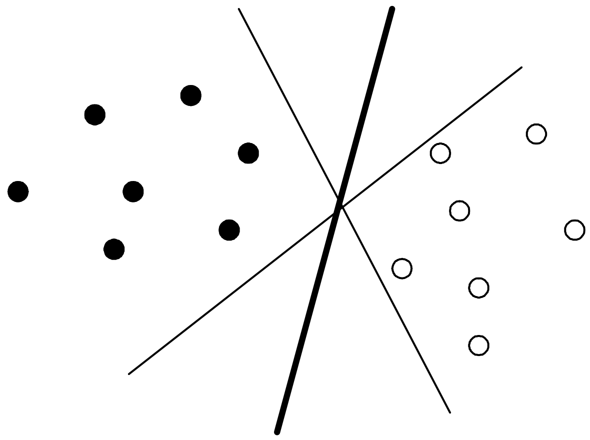
(Optimization Theory) [9] เพื่อใช้สำหรับหาคำตอบของการสร้างระนาบเกินที่เหมาะที่สุด (Optimal Hyperplane: OHP)

ที่มาของหลักการสร้างสมการซัพพอร์ตเวกเตอร์แมทชีนจะใช้หลัก Structural Risk Minimization [30,34] ซึ่งมีสมรรถนะสูงกว่าโครงข่ายประสาทเทียมแบบธรรมดาทั่วไปที่ใช้หลัก Empirical Risk Minimization [19] ด้วยข้อแตกต่างนี้ ทำให้ซัพพอร์ตเวกเตอร์แมทชีนมีความสามารถโดยทั่วไปที่เหนือกว่าโครงข่ายประสาทแบบดั้งเดิม หรืออาจกล่าวได้ว่าซัพพอร์ตเวกเตอร์แมทชีนคือ โครงข่ายประสาทเทียมชนิดใหม่ที่ถูกพัฒนาขึ้นมาเพื่อแก้ไขข้อบกพร่องต่างๆ ของโครงข่ายประสาทเทียมแบบดั้งเดิมจนเป็นที่ยอมรับ และมีการนำไปประยุกต์ใช้กับงานทางด้านวิศวกรรมกันอย่างกว้างขวาง ตัวอย่างเช่น ได้มีการนำซัพพอร์ตเวกเตอร์แมทชีนไปประยุกต์ใช้สำหรับงานที่เกี่ยวกับด้านการรู้จำรูปแบบ โดยเรียกซัพพอร์ตเวกเตอร์แมทชีนที่ใช้ในงานทางด้านนี้ว่า ซัพพอร์ตเวกเตอร์แมทชีนสำหรับการแบ่งกลุ่ม (Support Vector Classification) และยังมีการนำซัพพอร์ตเวกเตอร์แมทชีนไปประยุกต์ใช้กับงานทางด้านการประมาณค่าฟังก์ชัน ทั้งที่เป็นแบบเชิงเส้น (Linear) และแบบไม่เชิงเส้น (Non-Linear) หรือประยุกต์ใช้กับงานด้านการพยากรณ์ต่างๆโดยเรียกซัพพอร์ตเวกเตอร์แมทชีนที่ใช้สำหรับงานด้านนี้ว่า ซัพพอร์ตเวกเตอร์แมทชีนสำหรับการถดถอย (Support Vector Regression)

3.3.2 การใช้ซัพพอร์ตเวกเตอร์แมทชีนสำหรับการแบ่งกลุ่ม

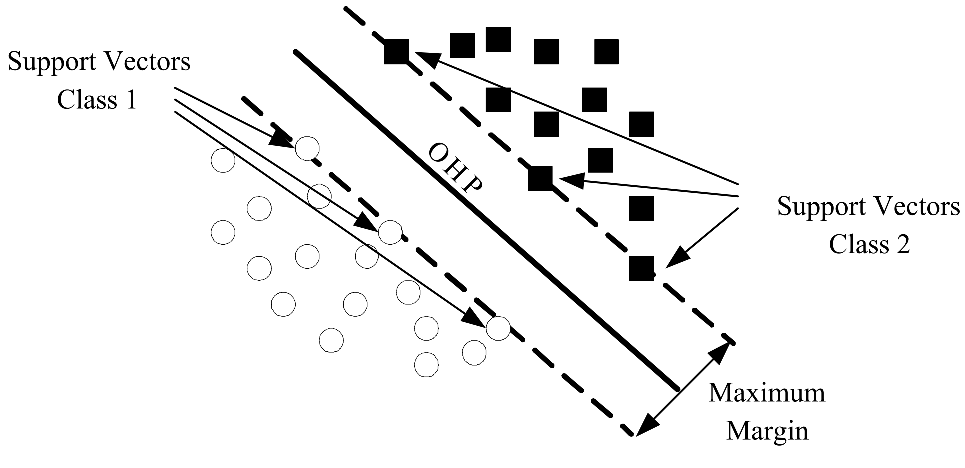
ในระยะแรกเริ่มแรกนั้น ซัพพอร์ตเวกเตอร์แมทชีนได้ถูกพัฒนาขึ้นมาเพื่อการแก้ปัญหาสำหรับงานด้านการแบ่งกลุ่ม สิ่งที่เป็นข้อแตกต่างระหว่างซัพพอร์ตเวกเตอร์แมทชีนสำหรับแบ่งกลุ่ม กับโครงข่ายประสาทเทียมแบบดั้งเดิมคือ ซัพพอร์ตเวกเตอร์แมทชีนจะพยายามหาระนาบเกินที่เหมาะที่สุด [13,19,32] สำหรับใช้แบ่งกลุ่มข้อมูลเพื่อเพิ่มสมรรถนะให้สามารถใช้งานได้เป็นอย่างดี กับข้อมูลที่โครงข่ายไม่รู้จักมาก่อน [36,37] จากรูป 3.1 จะเห็นได้ว่ามีการสร้างระนาบเกินที่ใช้สำหรับแบ่งกลุ่มข้อมูลได้หลายแบบด้วยกัน แต่จะมีระนาบเกินที่เหมาะที่สุดที่สามารถรักษาระยะห่างมากที่สุดระหว่างข้อมูล 2 กลุ่มที่อยู่ใกล้กันมากที่สุดเพียงระนาบเดียวเท่านั้น

หลักการโดยคร่าวๆ ของการหาระนาบเกินสำหรับแบ่งกลุ่มข้อมูลที่เหมาะที่สุดคือการหาตำแหน่งข้อมูลที่เป็น ซัพพอร์ตเวกเตอร์ (Support Vector) ของแต่ละกลุ่มข้อมูล ซึ่งก็คือตำแหน่งของข้อมูลที่อยู่ใกล้กับระนาบแบ่งกลุ่มมากที่สุด จากนั้นจะสนใจเฉพาะข้อมูลที่เป็นซัพพอร์ตเวกเตอร์ โดยถือว่าซัพพอร์ตเวกเตอร์ของข้อมูลแต่ละกลุ่มจะใช้แทนข้อมูลในกลุ่มนั้นทั้งหมด และกำหนดให้ตำแหน่งของซัพพอร์ตเวกเตอร์แต่ละกลุ่มคือแนวขอบระนาบทั้งสองฝั่ง



รูป 3.1 ระนาบเกินที่เหมาะที่สุดสำหรับการแบ่งกลุ่มข้อมูล

ในทางทฤษฎีจะต้องไม่มีข้อมูลเกินเข้ามาในระหว่างขอบระนาบทั้งสอง จากนั้นจึงหาระนาบที่รักษาระยะห่างจากขอบมากที่สุด (Maximum Margin) และถือว่าระนาบดังกล่าว คือ ระนาบเกินสำหรับการแบ่งกลุ่มที่เหมาะที่สุด แสดงดังรูป 3.2



รูป 3.2 การหาระนาบเกินที่เหมาะที่สุดสำหรับการแบ่งกลุ่มโดยอาศัยซัพพอร์ตเวกเตอร์

กล่าวโดยสรุปก็คือ ซัพพอร์ตเวกเตอร์แมทชีนสำหรับการแบ่งกลุ่มได้ถูกพัฒนาขึ้นเพื่อเพิ่มความถูกต้องสำหรับงานด้านการจัดแบ่งกลุ่มข้อมูล ด้วยเทคนิคการออกแบบอย่างมีหลักการ ทำให้ซัพพอร์ตเวกเตอร์แมทชีนมีสมรรถนะที่เหนือกว่าโครงข่ายประสาทเทียมแบบดั้งเดิม [19, 22, 28, 36, 37]

3.3.3 การใช้ซัพพอร์ตเวกเตอร์แมทชีนสำหรับการถดถอย

เมื่อไม่นานมานี้ได้มีการนำซัพพอร์ตเวกเตอร์แมทชีนมาพัฒนาประยุกต์ใช้สำหรับงานด้านการประมาณค่า (Approximation) หรือการหาค่าที่ไม่ทราบ [11,12,16-19, 21,25,27,33,38-44] สำหรับการประยุกต์ใช้ซัพพอร์ตเวกเตอร์แมทชีนให้สามารถทำการประมาณค่าได้นั้น จะต้องทำการสอนซัพพอร์ตเวกเตอร์แมทชีนเพื่อให้ได้ฟังก์ชัน $f(\mathbf{x})$ ซึ่งเป็นฟังก์ชันเชิงเส้นที่ใช้แทนสมการของระนาบเกินที่เหมาะที่สุด ที่สามารถใช้แทนกลุ่มของข้อมูลที่นำมาฝึกสอนได้ทั้งหมด ดังแสดงในสมการ (3.1)

$$f(\mathbf{x}) = \langle \mathbf{w} \bullet \mathbf{x} \rangle + b \tag{3.1}$$

โดยที่ $\mathbf{w}$ คือ เวกเตอร์น้ำหนัก

b คือ ค่าไบอัส

$\langle \bullet \rangle$ คือ การทำผลคูณจุด (Dot Product)

$\mathbf{x}$ คือ เวกเตอร์ข้อมูลอินพุตในรูปของเวกเตอร์คอลัมน์ $(x_1,\ldots,x_n)^T$ ขนาดเท่ากับ n, $x_i \in \mathbf{R}$; $i = 1,\ldots,n$

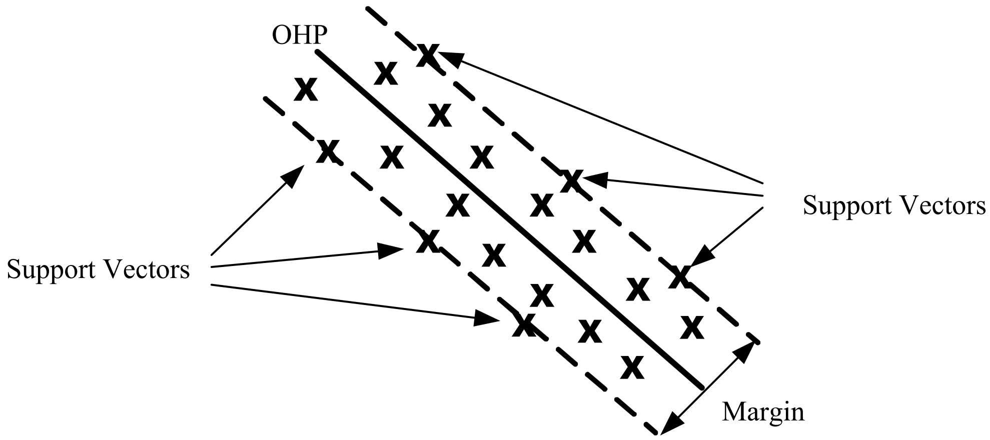
$\mathbf{R}$ คือ เซตของจำนวนจริง

หลักการในการหาระนาบเกินสำหรับใช้แทนกลุ่มของข้อมูลที่เหมาะที่สุด ในกรณีของซัพพอร์ตเวกเตอร์แมทชีนสำหรับการถดถอย จะตรงกันข้ามกับกรณีของการแบ่งกลุ่มตรงที่ ซัพพอร์ตเวกเตอร์แมทชีนสำหรับการถดถอยจะหาตำแหน่งของข้อมูลที่เป็นซัพพอร์ตเวกเตอร์ ซึ่งจะเป็นข้อมูลในตำแหน่งที่อยู่ห่างจากระนาบเกินมากที่สุด และตำแหน่งดังกล่าวจะอยู่บนขอบของระนาบที่กำหนดขึ้น ข้อมูลที่กำหนดให้เป็นซัพพอร์ตเวกเตอร์จะใช้แทนกลุ่มของข้อมูลทั้งหมด ในทางทฤษฎีจะพยายามให้ข้อมูลทั้งหมดอยู่ภายในขอบระนาบ และสร้างระนาบเกินที่ใช้แทนกลุ่มข้อมูลขึ้น โดยระนาบดังกล่าวจะอยู่ในตำแหน่งที่รักษาระยะห่างที่มากที่สุด ระหว่างตำแหน่งข้อมูลที่เป็นซัพพอร์ตเวกเตอร์ และถือว่าระนาบดังกล่าวเป็นระนาบเกินที่เหมาะที่สุดสำหรับใช้แทนกลุ่มข้อมูล ดังแสดงรายละเอียดในรูป 3.3

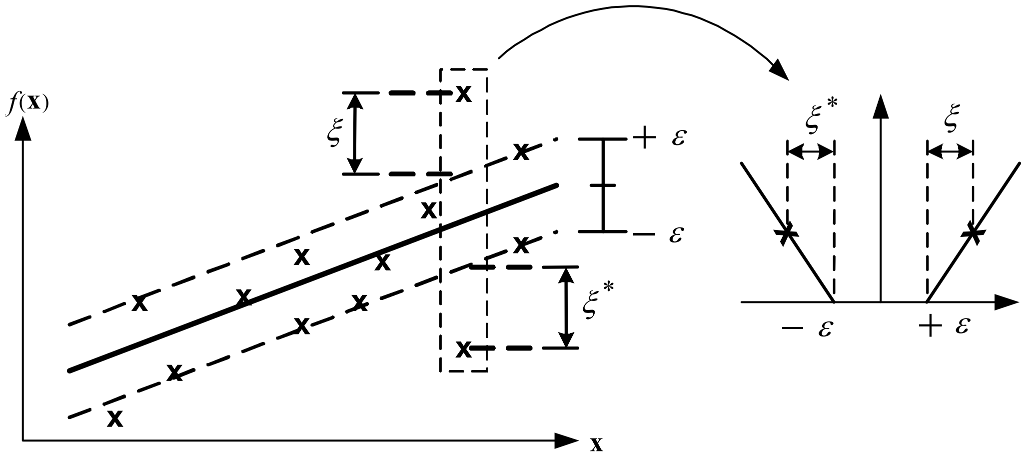
ในการกำหนดความกว้างของขอบระนาบจะขึ้นอยู่กับการกำหนดค่าคลาดเคลื่อนที่ยินยอมได้ (Error Insensitive) ซึ่งได้มาจากการกำหนดค่าพารามิเตอร์ ε โดยอาศัยคุณสมบัติของฟังก์ชันการสูญเสีย (Loss Function) โดย *Vapnik* ได้เสนอฟังก์ชันการสูญเสียแบบ ε-insensitive [42] ซึ่งมีความสัมพันธ์ดังสมการ (3.2)

$$L(\mathrm{y}, f(\mathbf{x})) = \begin{cases} 0 & ; \quad |y - f(\mathbf{x})| \leq \varepsilon \\ |y - f(\mathbf{x})| - \varepsilon & ; \quad |y - f(\mathbf{x})| > \varepsilon \end{cases} \tag{3.2}$$

ความหมายของฟังก์ชันการสูญเสียคือ ข้อมูลที่อยู่ระหว่างขอบระนาบทั้งสองนี้จะถือว่าไม่มีค่าคลาดเคลื่อน (Error = 0) แต่สำหรับข้อมูลที่อยู่นอกเหนือจากขอบระนาบนั้น จะกำหนดค่าคลาดเคลื่อนในการสร้างระนาบโดยการแทนด้วยตัวแปร ξ (Slack) แสดงในรูป 3.4



รูป 3.3 การหาระนาบเกินที่เหมาะที่สุดสำหรับใช้แทนกลุ่มข้อมูล



รูป 3.4 การกำหนดแนวขอบระนาบเกินด้วยฟังก์ชัน ε-insensitive

ขั้นตอนการสอนซัพพอร์ตเวกเตอร์แมทชีนเพื่อหาฟังก์ชัน $f(\mathrm{x})$ สำหรับใช้แทนสมการของระนาบเกินที่เหมาะที่สุดนั้น เริ่มจากการเตรียมเซตตัวอย่างข้อมูลสำหรับการสอน

$$\{(\mathbf{x}_i, y_i)\}_{1 \le i \le \ell},\ \mathbf{x}_i \in X \subset \mathbf{R}^n,\ y_i \in Y \subset \mathbf{R}$$

โดยที่ $\mathbf{x}_i$ คือ ตัวอย่างข้อมูลอินพุต (Input Data) ที่ i

y_i คือ ตัวอย่างข้อมูลเอาต์พุตที่ต้องการ (Desired Output) ที่ i

X คือ ปริภูมิอินพุต (Input Space)

Y คือ ปริภูมิเอาต์พุต (Output Space)

ℓ และ n คือ จำนวน และ มิติของข้อมูลที่นำมาสอนตามลำดับ

ในการสอนซัพพอร์ตเวกเตอร์แมทชีนเพื่อให้สามารถหาระนาบเกินที่เหมาะที่สุดได้นั้น จะต้องทำการหาค่าเวกเตอร์น้ำหนัก (**w**) ที่เหมาะที่สุด โดยปัญหาดังกล่าวเขียนแทนด้วยสมการ (3.3) และอสมการ (3.4)

$$\Phi(\mathbf{w}, \xi_i, \xi_i^*) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{\ell}(\xi_i + \xi_i^*) \qquad (3.3)$$

เงื่อนไข

$$\left.\begin{array}{rcl} y_i - \langle \mathbf{w} \bullet \mathbf{x}_i \rangle - b & \le & \varepsilon + \xi_i \\ \langle \mathbf{w} \bullet \mathbf{x}_i \rangle + b - y_i & \le & \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* & \ge & 0 \end{array}\right\} \qquad (3.4)$$

เมื่อ $\|\bullet\|$ คือ นอร์มแบบยุคลิด (Euclidean Norm)

C คือ ค่าคงที่ ($C > 0$) สำหรับคุมค่าคลาดเคลื่อน (Regularization Parameter) ในการสร้าง $f(\mathrm{x})$

ξ, ξ^* คือ ค่าคลาดเคลื่อนของข้อมูลจากขอบระนาบ บน-ล่าง ตามลำดับ

ปัญหาดังกล่าวเป็นปัญหาที่มีเงื่อนไขบังคับ (Constrains) ถ้าสามารถทำให้สมการ (3.3) มีค่าน้อยมากๆ ภายใต้เงื่อนไข (3.4) จะส่งผลให้สามารถประมาณค่าของฟังก์ชันได้ดีและ

ถูกต้องมากยิ่งขึ้น จากการหาคำตอบของสมการ (3.3) ภายใต้เงื่อนไข (3.4) โดยใช้ทฤษฎีการหาค่าเหมาะที่สุด (Optimization Theory) [9,14] ทำให้ได้ความสัมพันธ์ของ **w** ดังสมการ (3.5)

$$\mathbf{w} = \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i)\,\mathbf{x}_i \tag{3.5}$$

โดยที่ α_i^*, α_i คือตัวคูณลากรานจ์ (Lagrange Multipliers) ซึ่งสามารถหาค่าได้โดยใช้ขั้นตอนวิธี การโปรแกรมกำลังสอง (Quadratic Programming) ทำการแก้ปัญหา (3.6)

$$\min_{\alpha_i, \alpha_i^*} \frac{1}{2} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) - \sum_{i=1}^{\ell} y_i (\alpha_i^* - \alpha_i) + \varepsilon \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) \tag{3.6}$$

ตามเงื่อนไข

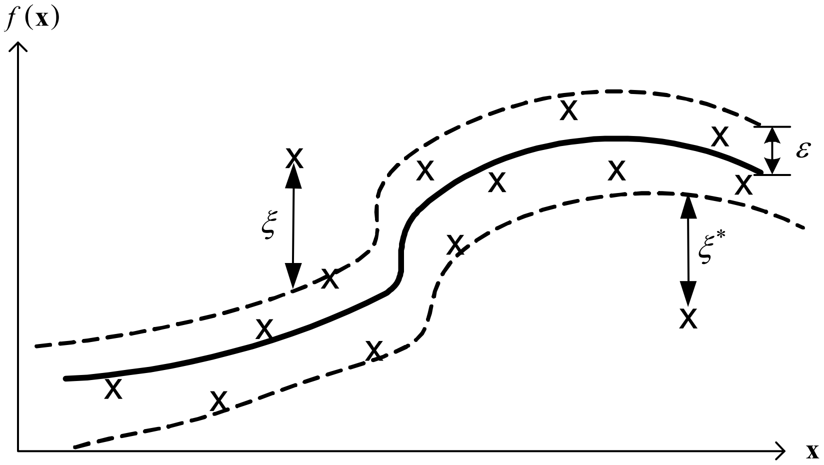
$$\begin{gathered} \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) = 0 \\ 0 \le \alpha_i^*, \alpha_i \le C \end{gathered} \tag{3.7}$$

หลังจากหาค่า α_i^* และ α_i ที่เหมาะที่สุดได้แล้ว แทนค่า **w** ตามสมการ (3.5) ในสมการ (3.1) ทำให้ได้คำตอบของสมการระนาบเกินที่เหมาะที่สุดสำหรับแทนกลุ่มข้อมูลได้ดังนี้

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) \langle \mathbf{x} \bullet \mathbf{x}_i \rangle + b \tag{3.8}$$

เมื่อ $\mathbf{x}_i$ คือ ข้อมูลที่เป็นซัพพอร์ตเวกเตอร์ที่เกิดจากข้อมูลที่ใช้ในการฝึกสอน ซึ่งเป็นตำแหน่งของข้อมูลที่มีค่า $|\alpha_i^* - \alpha_i| > 0$ ตามข้อกำหนดของ *Karush-Kuhn-Tucker* (KKT) [14,33]

สำหรับกรณีปัญหาแบบไม่เชิงเส้น (Non-Linear) ดังแสดงในรูป 3.5 ทำให้ไม่สามารถใช้วิธีการแบบเชิงเส้นกับกรณีดังกล่าวได้ ดั้งนั้นจึงต้องใช้ฟังก์ชันเคอร์เนล (Kernel Function) ทำการแมปข้อมูลที่ไม่เชิงเส้น เพื่อให้สามารถใช้วิธีการแบบเชิงเส้นได้กับกรณีดังกล่าว



รูป 3.5 การวิเคราะห์การถดถอยแบบไม่เชิงเส้น

3.3.4 ปริภูมิลักษณะเด่นและการแมปข้อมูลโดยใช้ฟังก์ชันเคอร์เนล

สำหรับกรณีปัญหาแบบไม่เชิงเส้น จะใช้ฟังก์ชันเคอร์เนลทำการแมปข้อมูลอินพุตให้มีมิติเพิ่มมากขึ้นในปริภูมิลักษณะเด่น (Feature Space) ดังรูป 3.6 เพื่อจัดระเบียบข้อมูลเบื้องต้นให้สะดวกต่อการสร้างระนาบเกินสำหรับใช้แทนกลุ่มข้อมูลได้อย่างเหมาะที่สุด และให้สามารถใช้วิธีการแบบเชิงเส้นได้ในปริภูมินี้ ฟังก์ชันเคอร์เนลแสดงดังสมการ (3.9)

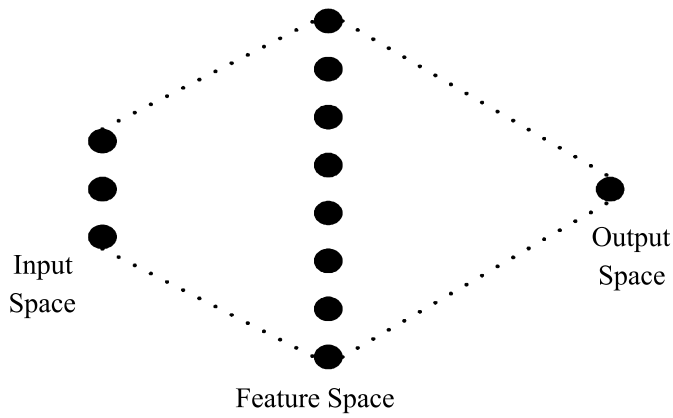
$$K(\mathbf{x},\mathbf{x}_i) = \langle \Phi(\mathbf{x}) \bullet \Phi(\mathbf{x}_i) \rangle \tag{3.9}$$

โดยที่ $\mathbf{x}$ คือ เวกเตอร์ข้อมูลอินพุต

$\mathbf{x}_i$ คือ ซัพพอร์ตเวกเตอร์

Φ คือ ฟังก์ชันการแมป (Mapping Function)

สมการ (3.9) หมายถึงการนำข้อมูล 2 ชุดมาผ่านฟังก์ชันการแมป ($\Phi(\bullet)$) จากนั้นนำมาทำผลคูณจุดที่ปริภูมิลักษณะเด่นเพื่อให้มีมิติที่เพิ่มมากขึ้น ซึ่งจะให้ผลเหมือนกับการนำข้อมูล 2 ชุดมาผ่านฟังก์ชันเคอร์เนลที่ปริภูมิอินพุต ในทางปฏิบัติจะนิยมใช้วิธีหลังมากกว่าเนื่องจากมีความง่ายและสะดวกกว่า ฟังก์ชันเคอร์เนลนั้นมีอยู่หลายชนิดด้วยกัน [19,26,29] แต่ฟังก์ชันที่จะใช้ทำหน้าที่เป็นฟังก์ชันเคอร์เนลได้นั้นจะต้องเป็นไปตามเงื่อนไขของ *Mercer* [14]



รูป 3.6 การแมปข้อมูลจากปริภูมิอินพุตให้อยู่ในปริภูมิลักษณะเด่น

จากกรณีใช้ฟังก์ชันเคอร์เนลมาช่วยสร้าง $f(\mathbf{x})$ ทำให้สมการ (3.5) มีการเพิ่มเทอมของฟังก์ชันเคอร์เนลเข้าไปในสมการ ทำให้ได้สมการใหม่คือสมการ (3.10)

$$\mathbf{w} = \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) K(\mathbf{x}_i) \tag{3.10}$$

แทนค่า $\mathbf{w}$ ตามสมการ (3.10) ในสมการ (3.1) ทำให้ได้คำตอบของสมการระนาบเกินที่เหมาะที่สุดดังนี้

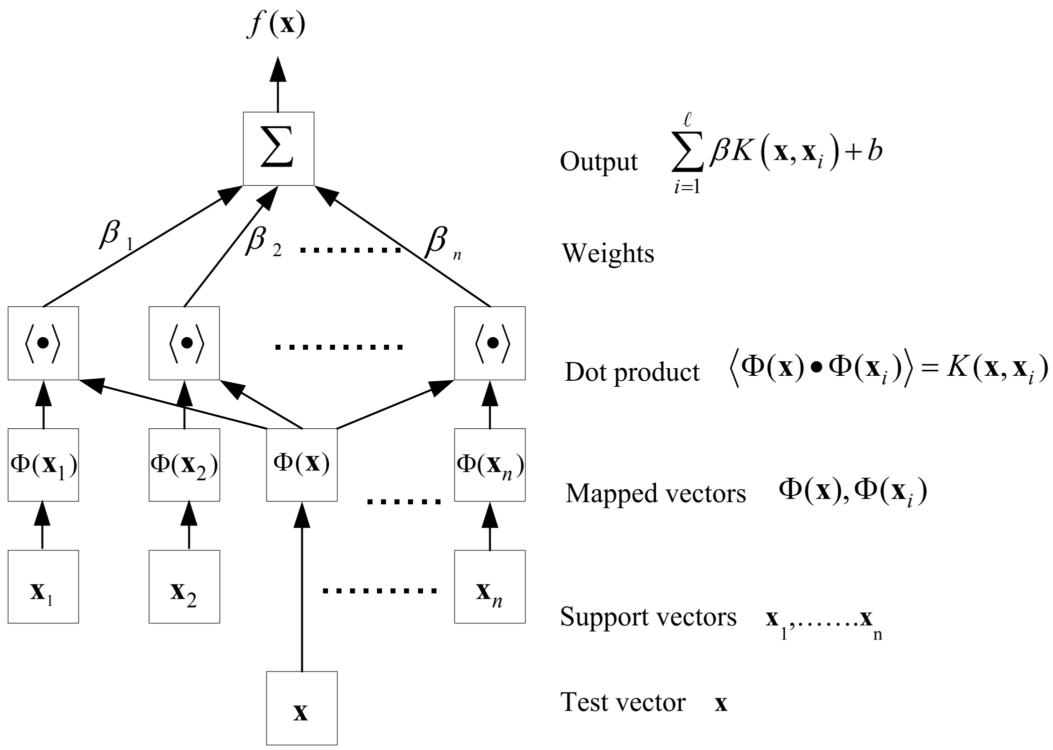
$$f(\mathbf{x}) = \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) K\left(\mathbf{x}, \mathbf{x}_i\right) + b \tag{3.11}$$

จากสมการ (3.11) เมื่อให้ $(\alpha_i^* - \alpha_i) = \beta_i$ ทำให้สมการจัดอยู่ในรูปอย่างง่ายคือ

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} \beta_i K\left(\mathbf{x}, \mathbf{x}_i\right) + b \tag{3.12}$$

ในกรณีที่เลือกใช้ฟังก์ชันเคอร์เนลชนิดที่มีการรวมเทอมของไบอัส (b) เข้าไว้ด้วยแล้ว จะไม่มีการคิดค่าของไบอัสอีกต่อไปทำให้ได้สมการของระนาบเกินใหม่ดังนี้

$$f(\mathbf{x}) = \sum_{i=1}^{\ell} \beta_i K\left(\mathbf{x}, \mathbf{x}_i\right) \tag{3.13}$$



รูป 3.7 สถาปัตยกรรมของซัพพอร์ตเวกเตอร์แมทชีนสำหรับการถดถอย

ขั้นตอนการทำงานโดยสรุปของซัพพอร์ตเวกเตอร์แมทชีนสำหรับการถดถอยแสดงดังรูป 3.7 โดยมีลำดับการทำงานดังนี้

(1) เริ่มจากนำข้อมูลที่ต้องการทดสอบป้อนเข้ายังปริภูมิอินพุตของระบบ

(2) ทำการแมปข้อมูลอินพุต และซัพพอร์ตเวกเตอร์ที่ได้จากการฝึกสอน

(3) ทำการหาผลคูณจุดระหว่างข้อมูลที่ผ่านการแมปแล้ว เพื่อให้มีมิติเพิ่มมากขึ้นในปริภูมิลักษณะเด่น ซึ่งได้ผลเหมือนกับการใช้ฟังก์ชันเคอร์เนลที่ปริภูมิอินพุต

(4) นำผลคูณจุดของข้อมูลจากปริภูมิลักษณะเด่นมาคูณกับค่าน้ำหนักที่ได้เตรียมไว้ ซึ่งให้ผลเหมือนกับการแมปข้อมูลด้วยฟังก์ชันเคอร์เนลที่ปริภูมิอินพุต แล้วคูณกับค่าน้ำหนักที่ได้เตรียมไว้ เช่นกัน

(5) คำตอบที่ได้คือการรวมผลคูณทั้งหมดเข้าด้วยกัน โดยไม่ต้องใช้ฟังก์ชันตัดสินใจ (Decision Function) สำหรับกรณีของการประมาณค่าฟังก์ชัน

กล่าวโดยสรุปจุดเด่นของซัพพอร์ตเวกเตอร์แมทชีนนั้น อยู่ตรงที่มีการออกแบบอย่างมีหลักการมากขึ้น โดยอาศัยหลัก Structural Risk Minimization ที่มีพื้นฐานมาจากทฤษฎีการเรียนรู้

ทางสถิติ ซึ่งแตกต่างจากโครงข่ายประสาทเทียมแบบดั้งเดิมที่ใช้หลัก Empirical Risk Minimization ดังที่ได้กล่าวมาแล้วในข้างต้น

ในส่วนของการออกแบบโครงสร้างนั้น จะสามารถทำได้ง่ายกว่าโครงข่ายประสาทเทียมแบบดั้งเดิม ซึ่งซัพพอร์ตเวกเตอร์แมทชีนไม่ต้องมีการกำหนดจำนวนของโนดของชั้นซ่อน และไม่ต้องกำหนดว่าจะมีการใช้ชั้นซ่อนกี่ชั้น เนื่องจากซัพพอร์ตเวกเตอร์แมทชีนจะมีชั้นซ่อนเพียงชั้นเดียว แต่จะเรียกแทนว่าปริภูมิลักษณะเด่น ซึ่งเป็นปริภูมิที่ใช้สำหรับการสร้างระนาบเกินที่เหมาะที่สุดสำหรับการแบ่งกลุ่ม หรือ การแทนกลุ่มข้อมูล ทั้งนี้ขึ้นอยู่กับลักษณะของงานที่นำไปใช้ ซึ่งจะเห็นได้ว่าซัพพอร์ตเวกเตอร์แมทชีนสามารถเลือกใช้งานได้ง่ายกว่าเนื่องจากมีอยู่ 2 แบบด้วยกัน แต่ถ้าเป็นโครงข่ายประสาทเทียมแบบดั้งเดิมนั้นจะมีให้เลือกใช้อยู่หลายแบบมาก ทั้งนี้ขึ้นอยู่กับประสบการณ์ของผู้ใช้งานแต่ละคนในการเลือกใช้งานให้เหมาะสม ข้อแตกต่างอีกประการหนึ่งก็คือ ซัพพอร์ตเวกเตอร์แมทชีนจะใช้ข้อมูลอินพุตตัวอย่าง หรือใช้ซัพพอร์ตเวกเตอร์ที่เกิดจากการฝึกสอน เป็นเวกเตอร์น้ำหนักระหว่างชั้นอินพุตกับชั้นซ่อน แทนเวกเตอร์น้ำหนักแบบทั่วไปที่ใช้กับโครงข่ายประสาทเทียมแบบดั้งเดิม นอกจากนี้ซัพพอร์ตเวกเตอร์แมทชีนยังใช้เวลาในขั้นตอนการเรียนรู้ที่น้อยกว่า และยังให้ผลลัพธ์ที่มีความถูกต้องมากกว่าด้วย

ในส่วนของข้อด้อยนั้นจะอยู่ตรงที่ซัพพอร์ตเวกเตอร์แมทชีน จะให้เอาต์พุตได้เพียง 1 ค่าเท่านั้น ปัญหาอีกประการหนึ่งก็คือ จะต้องเลือกใช้ฟังก์ชันเคอร์เนลที่มีความเหมาะสมกับงานและจะต้องปรับแต่งค่าพารามิเตอร์บางตัวให้มีความเหมาะสม โดยจะกล่าวถึงรายละเอียดในส่วนของหัวข้อการออกแบบ และการทดสอบแบบจำลองในบทต่อไป