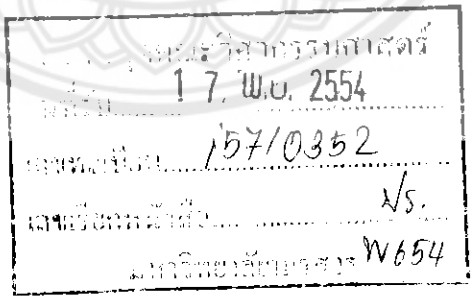
การเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล

ONLINE LEARNING WITH KERNEL METHOD

นายพิเชษฐ์ เทือกถา รหัส 48380151

นายภูษิต พิสกุล รหัส 48380153



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2552

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน	การเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล
ผู้ดำเนินโครงงาน	นายพิเชษฐ์ เทือกถา รหัส 48380151
	นายภูษิต พิสกุล รหัส 48380153
ที่ปรึกษาโครงงาน	ดร. ศุภวรรณ พลพิทักษ์ชัย
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2552

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

..... ที่ปรึกษาโครงงาน
(ดร. ศุภวรรณ พลพิทักษ์ชัย)

..... กรรมการ
(ดร. นิพัทธ์ จันทรมินทร์)

..... กรรมการ
(ดร. มุฑิตา สงฆ์จันทร์)

ชื่อหัวข้อโครงงาน	การเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล
ผู้ดำเนินโครงงาน	นายพิเชษฐ์ เทือกถา รหัส 48380151
	นายภูษิต พิสกุล รหัส 48380153
ที่ปรึกษาโครงงาน	ดร. ศุภวรรณ พลพิทักษ์ชัย
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2552

.....

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้ศึกษาการเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล ซึ่งนำมาประยุกต์ใช้สำหรับหาความสัมพันธ์ในรูปแบบฟังก์ชันระหว่างข้อมูลขาเข้าและขาออก สิ่งที่ได้จะเรียกว่าฟังก์ชันการประมาณค่าที่ประกอบไปด้วยผลบวกของฟังก์ชันเคอร์เนลที่แต่ละเทอมถูกให้ค่าน้ำหนักด้วยพารามิเตอร์ที่เรียกว่าแอลฟา ในการเรียนรู้วิธีนี้จะเป็นการแก้ข้อเสียของการเรียนรู้แบบข้อมูลทั้งหมด ที่ใช้ข้อมูลทั้งหมดในเวลาเดียวกัน ซึ่งจะมีปัญหาในการคำนวณถ้าข้อมูลที่ต้องใช้มีปริมาณเมาก ๆ ดังนั้นวิธีออนไลน์จึงถูกนำมาใช้แทน วิธีการเรียนรู้นี้จะใช้ข้อมูลแค่ทีละตัวในการปรับค่าแอลฟาที่กล่าวมาแล้ว ดังนั้นจึงไม่มีปัญหาในการใช้งานกับข้อมูลจำนวนมาก ๆ

การหาฟังก์ชันการประมาณค่าด้วยวิธีเคอร์เนลจะใช้กลุ่มข้อมูลที่เรียกว่ากลุ่มสอน ส่วนประสิทธิภาพของการเรียนรู้จะถูกวัดด้วยค่าความผิดพลาดเฉลี่ยกำลังสองผ่านกลุ่มข้อมูลที่เรียกว่ากลุ่มทดสอบ ซึ่งในการทดลองจะหาค่าความผิดพลาดรวมโดยใช้วิธีที่เรียกว่าครอสวาลิเดชั่น นอกจากนั้นการทดลองยังแสดงให้เห็นถึงผลของค่าพารามิเตอร์แต่ละตัว เช่น อัตราการเรียนรู้ เรกกูลาไรเซชั่นพารามิเตอร์ ความกว้างของเคอร์เนลฟังก์ชัน ว่ามีผลต่อความผิดพลาดเฉลี่ยกำลังสองอย่างไรบ้าง

Project title	Online Learning with Kernal Method
Name	Mr. Pichet Tuektha ID. 48380151
	Mr. Phusit Phitsakun ID. 48380153
Project advisor	Supawan Phonphitakchai, Ph.D.
Major	Electrical Engineering
Department	Electrical and Computer Engineering
Academic year	2009

Abstract

This online learning method is introduced to overcome a disadvantage of batch learning in which a whole data set is processed once causing a computational expensive. Online learning uses only one data pair to be processed by updating the parameters and then discarded. For this reason, the online method is able to apply with a huge data set without a computation problem.

Estimating our model is applied with the training data set whereas testing the model performance is measured with the testing data set through mean square error. Moreover, the performance is also investigated with cross validation. In this study, the experiments reveal the affects of each parameter, learning rate, regularization parameter and kernel width. In order to get the approximation function, our model is trained with train data whereas the model performance is measured with mean square error via test data. The performance is also investigated with cross validation. In our experiments, the effects of each parameter (learning rate, regularization parameter and kernel width) on mean square error are presented.

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปด้วยความกรุณาเป็นอย่างยิ่งจาก ดร. ศุภวรรณ พลพิทักษ์ชัย ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน และได้คอยชี้แนะแนวทางตลอดการทำโครงงาน คณะผู้ดำเนินโครงงานขอกราบขอบพระคุณเป็นอย่างสูงและขอระลึกถึงความกรุณาของท่านไว้ตลอดไป

ขอขอบคุณคณาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ให้กับคณะผู้ดำเนินโครงงาน นอกจากนี้ยังต้องขอขอบคุณภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ที่ให้ยืมอุปกรณ์และเครื่องมือต่าง ๆ จนทำให้โครงงานนี้สำเร็จลุล่วงไปได้

เหนือสิ่งอื่นใด คณะผู้ดำเนินโครงงานขอกราบขอบพระคุณบิดามารดา ผู้มอบความรัก เมตตา สติปัญญา รวมทั้งเป็นผู้ให้ทุกสิ่งทุกอย่างตั้งแต่วัยเยาว์จวบจนถึงปัจจุบัน คอยเป็นกำลังใจทำให้ได้รับความสำเร็จอย่างทุกวันนี้ และขอบคุณทุก ๆ คนในครอบครัวของคณะผู้ดำเนินโครงงานที่ไม่ได้กล่าวไว้ ณ ที่นี้ด้วย

นายพิเชษฐ์ เทือกถา
นายภูษิต พิสกุล

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญรูป

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เครื่องการเรียนรู้ (Machine learning) เป็นกลุ่มงานวิจัยหลายแขนงคือรวมเอาหัวข้อหลายอย่างไว้จึงค่อนข้างยากที่จะนิยามมันให้ชัดเจน อาจจะบอกได้ว่าเป็นกลุ่มของเทคนิคในการทำให้คอมพิวเตอร์ทำงานบางอย่างได้ดีขึ้นโดยการใช้ข้อมูลพื้นฐานข้อมูลฝึกสอนหรือกลุ่มของเทคนิคในการทำให้คอมพิวเตอร์ทำงานได้อย่างอัตโนมัติ งานบางอย่างที่กล่าวถึงนี้ก็มี การจำแนกประเภท (Classification), การทำนายค่า (Prediction หรือ Regression), การคำนวณการกระจายของข้อมูล (Density estimation) และการเลือกค่าแบบจำลอง (Model selection) ทฤษฎีด้านสถิติมักจะตั้งสมมติฐานว่ามีข้อมูลจำนวนมากไม่จำกัดขณะที่งานด้านทฤษฎีส่วนมากในเครื่องการเรียนรู้ จะสนใจการวิเคราะห์ข้อมูลที่มีจำกัด

การเรียนรู้หมายถึงการหาความสัมพันธ์ระหว่างข้อมูลขาเข้า (Input) และ ขาออก (Output) ซึ่งข้อมูลที่ได้นั้นจะต้องมีค่าความผิดพลาดน้อยที่สุดและเป็นค่าที่ดีที่สุดสามารถนำไปใช้งานได้จริง

ส่วนใหญ่แล้วการเรียนรู้จากข้อมูลจะใช้แบบข้อมูลทั้งหมด (Batch learning) ในการสอน (Train) เพียงครั้งเดียว อย่างไรก็ตามวิธีนี้จะมีข้อเสียที่ข้อมูลมีจำนวนมาก คือมีข้อมูลเท่าไรก็จะป้อนเท่านั้น การทำแบบนี้จะทำให้คอมพิวเตอร์มีการประมวลผลช้าค่อนข้างที่จะสับสน และอาจจะทำให้มีค่าความผิดพลาดที่มากหรืออาจจะทำให้ระบบหยุดการทำงานได้เลย

ดังนั้น เราจึงใช้การเรียนรู้แบบออนไลน์ (Online learning) ซึ่งวิธีนี้จะใช้ข้อมูลเข้าไปทีละตัว โดยจะมีการสร้างแบบจำลอง (Model) ของแต่ละข้อมูลเอาไว้เพื่อจะหาค่าความผิดพลาดว่ามีค่ามากน้อยเพียงใด และจะทำให้คอมพิวเตอร์ของเราทำงานได้อย่างมีประสิทธิภาพ เนื่องจากจะป้อนข้อมูลไปทีละตัว นอกจากนั้นยังมีการเรียนรู้ชนิดการเรียนรู้แบบมีผู้สอน (Supervised learning) และการเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning) ซึ่งรู้แบบการเรียนรู้แบบมีผู้สอนจะเริ่มด้วยการส่งสิ่งเร้าที่ใช้ในการสอนเข้าไปเป็นอินพุตของระบบ เมื่อสร้างแบบจำลองขึ้นมาเพื่อทำนายเอาต์พุตแล้วเปรียบเทียบผลความคลาดเคลื่อนที่เกิดขึ้นเพื่อนำไปคำนวณการปรับแต่งค่าต่างๆ เพื่อลดความคลาดเคลื่อนลงให้เหลือน้อยที่สุด ส่วนการเรียนรู้แบบไม่มีผู้สอนนั้น ไม่จำเป็นต้องมีค่าเป้าหมายของแต่ละข้อมูลตัวอย่าง ในระหว่างการเรียนรู้จะได้รับข้อมูลกระตุ้นในรูปแบบต่างๆ และจะทำการจัดกลุ่มรูปแบบต่าง ๆ เหล่านั้นเองตามต้องการ ผลตอบของการเรียนรู้แบบไม่มีผู้สอนนี้

จะเป็นการระบุกลุ่มของข้อมูลที่ใส่เข้าไปโดยจะอิงกับวิธีการจัดกลุ่มซึ่งได้เรียนรู้จากข้อมูลที่เคยพบมา ในปี 1980 ได้มีผู้เสนอการเรียนรู้วิธีเคอร์เนล (Kernel method) ซึ่งมีข้อแตกต่างกับวิธีการเรียนรู้แบบเดิมเช่น วิธีโครงข่ายประสาท (Neural Network) หรือโครงข่ายใยประสาทเสมือน (Artificial Neural Network: ANN) และแง่ของการหาค่าที่เหมาะสมที่สุด (Optimization) สามารถทำได้ด้วยวิธีเคอร์เนลและยังไม่มีปัญหาด้านค่าที่ต่ำสุดเฉพาะที่ (local minima) ในปัจจุบันนี้การเรียนรู้วิธีเคอร์เนล ถูกนำไปใช้ในงานต่าง ๆ อาทิเช่น เอสวีเอ็ม (SVM-support vector machine) การวิเคราะห์รูปแบบของงานคือการค้นหาและการศึกษาทั่วไปประเภทของความสัมพันธ์ (เช่นกลุ่ม, การจัดอันดับ, องค์ประกอบหลัก, การจำแนกประเภท) ในทั่วไปประเภทของข้อมูล (เช่น การเรียงลำดับข้อความเอกสารชุดรูปภาพ ฯลฯ) เป็นต้น

โครงงานนี้เป็นการศึกษาทฤษฎีของการเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล โดยจะมุ่งเน้นในการศึกษาการนำข้อมูลต่าง ๆ ซึ่งเป็นข้อมูลจริงมาวิเคราะห์ประสิทธิภาพที่ได้โดยใช้โปรแกรมแมทแลป (MATLAB) ซึ่งประสิทธิภาพของการเรียนรู้จะถูกแสดงให้เห็นว่าขึ้นอยู่กับการปรับค่าพารามิเตอร์ต่าง ๆ ของแบบจำลองและจะใช้วิธีครอสวาลิเดชั่น (Cross validation) เป็นตัวสร้างแบบจำลองวิธีเคอร์เนลและทดสอบประสิทธิภาพ

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อศึกษาทฤษฎีของการเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล
2) เพื่อศึกษาประสิทธิภาพของการเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล โดยการนำไปใช้วิเคราะห์ข้อมูลต่าง ๆ และใช้การสอนแบบครอสวาลิเดชั่น
3) เพื่อศึกษาถึงพารามิเตอร์ต่าง ๆ ของแบบจำลองวิธีเคอร์เนล

1.3 ขอบเขตของโครงงาน

1) ศึกษาเรื่องทฤษฎีของ การเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล
2) นำข้อมูลมาวิเคราะห์ด้วยวิธีเคอร์เนลโดยสร้างขั้นตอนวิธีบนโปรแกรมแมทแลป
3) ศึกษาผลกระทบที่ได้จากการปรับค่าของพารามิเตอร์แต่ละตัว

1.4 ขั้นตอนและแผนการดำเนินงาน

รายละเอียด	ปี 2552						ปี 2553		
	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1. รวบรวมข้อมูล									
2. ศึกษาทฤษฎีทั่วไปของ การเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล									
3. ดำเนินการวิเคราะห์ผลที่ได้จากวิธีการเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล									
4. สรุปผลการดำเนินงาน									
5. จัดทำปริญญานิพนธ์ฉบับสมบูรณ์									

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน

1) เข้าใจถึงเรื่องทฤษฎีของ การเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล
2) เข้าใจถึงผลกระทบของพารามิเตอร์แต่ละตัวในการเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนล
3) สามารถนำความรู้ด้านทฤษฎีของ การเรียนรู้แบบออนไลน์ด้วยวิธีเคอร์เนลไปใช้ในขั้นสูงต่อไป

1.6 งบประมาณ

1) ค่าถ่ายเอกสารและค่าเข้าเล่มรายงานฉบับสมบูรณ์	เป็นเงิน	1,000 บาท
2) ค่าพิมพ์เอกสาร	เป็นเงิน	500 บาท
3) ค่าวัสดุคอมพิวเตอร์	เป็นเงิน	500 บาท
รวมเป็นเงินทั้งสิ้น (สองพันบาทถ้วน)		**2,000 บาท**

หมายเหตุ: ถัวเฉลี่ยทุกรายการ

บทที่ 2
การประมาณค่าฟังก์ชั่นด้วยวิธีเคอร์เนล

ในบทนี้จะนำเสนอแนวทางในการแก้ไขปัญหาที่สามารถใช้ในงานแตกต่างกัน อาทิเช่น การประมวลผลสัญญาณ การควบคุมเครื่องกลและการประมาณค่าฟังก์ชัน โดยส่วนใหญ่แล้วเราจะมองถึงปัญหาเหล่านี้ในรูปแบบของการประมาณการระบบที่ไม่รู้จักโดยอาศัยข้อมูลตัวอย่าง ในปัจจุบันนี้วิธีอาร์เคเอชเอส (RKHS – Reproducing kernel Hilbert Spaces) ได้ถูกศึกษาและนำไปใช้กับปัญหาต่างๆ ที่ได้กล่าวมาแล้วดังนั้นในบทนี้จะเป็นการแสดงทฤษฎีที่เกี่ยวข้องกับการแก้ปัญหาด้วยวิธีเคอร์เนลแบบออนไลน์ นั้นคือเป็นการหาฟังก์ชันการประมาณค่าที่เป็นตัวแทนของระบบที่เราไม่รู้จักจากจำนวนข้อมูลจำกัด ซึ่งสามารถเก็บค่าได้จากระบบนั้น ๆ

2.1 การประมาณค่าฟังชันในอาร์เคเอชเอส

สมมติว่าฟังก์ชันไม่รู้ค่า f ซึ่งสามารถสังเกตค่าจำนวนจำกัดได้ เป็นส่วนหนึ่งของอาร์เคเอชเอส $\mathcal{F}$ และ f นิยามบนเซต X ที่สามารถถือได้ว่าเป็นข้อมูลขาเข้าโดยที่ $x \in X$ ดังนั้น $f(x)$ จะสามารถแสดงถึงการประมาณค่าของ f ที่ x และเซตข้อมูลขาเข้า X จะถูกมองว่าเป็นซับเซตปริภูมิยูคลิเดียน (Euclidian space) นั่นคือ $X \subseteq R^n$ โดยที่ x แต่ละตัวเป็นเวกเตอร์มิติ n

เซตจำกัดของการสังเกตหรือข้อมูลขาออกของฟังก์ชัน $\{z_i\}_{i=1}^N$ จะสอดคล้องกับข้อมูลขาเข้า $\{x_i\}_{i=1}^N$ โดยเราสามารถกล่าวได้ว่า เซตของข้อมูลขาออกอยู่ในปริภูมิ Z ถ้าไม่มีข้อผิดพลาดค่าการสังเกตจะแสดงได้ดังนี้

$$z_i = L_i f \tag{2.1}$$

เมื่อ $\{L_i\}_{i=1}^N$ เป็นเซตของการประเมินผลฟังก์ชันนัลแบบเชิงเส้น (Linear evaluation functional) กำหนดบน $\mathcal{F}$ ซึ่งสอดคล้องกับ f เราสามารถแสดงค่าทั้งหมดของการสังเกต $\{z_i\}_{i=1}^N$ ได้ดังนี้

$$z = Lf = \sum_{i=1}^{N} (L_i f) s_i \tag{2.2}$$

โดยที่ $s_i \in R^N$ เป็นเวกเตอร์มาตรฐาน (Standard basis vector) ที่ลำดับ i โดยในทางปฏิบัติ สามารถเขียนได้เป็น

$$z_i = f(x_i) \tag{2.3}$$

ซึ่งสามารถนำไปใช้ในปัญหาการประมาณค่า

ปัญหาฟังก์ชันการประมาณค่า ซึ่งเป็นตัวแทนฟังก์ชันที่ไม่รู้ค่า สามารถกระทำได้โดยกำหนดคลาสของ $\mathcal{F}$ ของฟังก์ชัน และค่าสังเกต $\{z_i\}_{i=1}^N$ ของฟังก์ชันนัล L_i ซึ่งถูกกำหนดบน $\mathcal{F}$ ดังนั้นจะสามารถหาฟังก์ชัน f บน $\mathcal{F}$ ซึ่งสอดคล้องกับสมการที่ (1) และ (3) ได้

ตามหลักแล้วเราสามารถกำหนดอาร์เคเอชเอส ให้เป็นปริภูมิฮิลเบิร์ต (Hilbert space) ของฟังก์ชันบน $\mathcal{X}$ ด้วยคุณสมบัติที่ว่าในแต่ละ $x \in \mathcal{X}$ ฟังก์ชันนัล L_i ซึ่งเชื่อมโยง f ด้วย $f(x_i)$ นั่นคือ $L_i \to f(x_i)$ จะเป็นฟังก์ชันนัลแบบเส้นตรงที่ถูกกำหนดขอบเขตและการมีขอบเขตหมายถึงการมีของค่า M โดยที่

$$|L_i f| = |f(x_i)| \le M\|f\| \quad \text{สำหรับทุกๆ } f \text{ ใน RKHS}$$

เมื่อ $\|\cdot\|$ เป็นค่านอร์มในอาร์เคเอชเอส

เมื่อฟังก์ชันนัล L_i ถูกกำหนดขอบเขต ตามทฤษฎีบทของรีซ (Riesz representation theorem) เราสามารถแสดงค่าการสังเกตได้ดังนี้

$$L_i f = \langle f, k_i \rangle, \quad i = 1,\ldots,N \tag{2.4}$$

เมื่อ $\langle \cdot,\cdot \rangle$ หมายถึงผลคูณภายใน (inner product) ใน $\mathcal{F}$ และ $\{k_i\}_{i=1}^N$ เป็นชุดของฟังก์ชันที่เรียกว่ารีโพรดิวซิ่งเคอร์เนล (reproducing kernels) ซึ่งแต่เป็นส่วนหนึ่งของ $\mathcal{F}$ และจะไม่ถูกกำหนดซ้ำกันโดยฟังก์ชันนัล L_i

ดังนั้นการแก้ปัญหาฟังก์ชันการประมาณค่าจะสามารถกำหนดใหม่ได้อีกแบบนั่นคือ กำหนดปริภูมิฮิลเบิร์ต $(\mathcal{F})$ ของเซตของฟังก์ชัน $\{k_i\}_{i=1}^N \subset \mathcal{F}$ และค่าการสังเกต $\{z_i\}_{i=1}^N$ จากเงื่อนไขดังกล่าวมาแล้วสามารถหาฟังก์ชันการประมาณค่า $f \in \mathcal{F}$ ที่สอดคล้องกับสมการที่ (4) ได้

ในทุกอาร์เคเอชเอส จะมีฟังก์ชันบวกแน่นอน (Positive – definite function) ที่เรียกว่ารีโพรดิวซิ่งเคอร์เนล (k) ที่กำหนดบน $\mathcal{X} \times \mathcal{X}$ ในการให้คำนิยามของอาร์เคเอชเอสในรูปฟังก์ชันนัลที่จำกัดเขตเชิงเส้นเราจะใช้คุณสมบัติดังนี้ ซึ่งกำหนดบนอาร์เคเอชเอส นั่นคือทุก ๆ $x, x' \in \mathcal{X}$ จะได้ $k(\cdot, x')$ เป็นฟังก์ชันที่กำหนดไว้บน $\mathcal{X}$ โดยที่ค่าที่ x มีค่าเท่ากับ $k(x, x')$ สำหรับทุก f ใน $\mathcal{F}$ จะได้

(i) $k(\cdot, x') \in \mathcal{F}$; และ

(ii) $\langle f, k(\cdot, x') \rangle_{\mathcal{F}} = f(x')$

จากคุณสมบัติที่กล่าวมาแล้วแสดงให้เห็นว่า ฟังก์ชันรีโพรดิวซิ่งเคอร์เนล k_i เป็นฟังก์ชันบน $\mathcal{X} \times \mathcal{X}$ แต่อย่างไรก็ตาม เราสามารถแสดงเคอร์เนลฟังก์ชันในรูปแบบของฟังก์ชันบน $\mathcal{X}$

เท่านั้นได้ด้วยการพิจารณาฟังก์ชันเคอร์เนล $k(x,x_i)$ เมื่อ x_i เป็นจุดที่ไม่เปลี่ยนแปลงค่าและ $x_i \in X$ ดังนั้นการเขียน $k(x,x_i)=k_i(x)$ (หรือ k_i) จะสามารถสรุปได้ว่า $k_i \in F$ บน X มีจุดศูนย์กลางบน x_i โดยส่วนใหญ่แล้วจุด x_i มักจะถูกกำหนดให้เป็นศูนย์กลางของฟังก์ชันเคอร์เนลและยังเป็นไฮเปอร์พารามิเตอร์ (Hyper parameter) ของฟังก์เคอร์เนลอีกด้วย ซึ่งโดยทั่วไปแล้วศูนย์กลางเหล่านี้จะสอดคล้องกับข้อมูลขาเข้า ดังนั้นเราจึงสามารถแสดงฟังก์ชัน f ทั่ว ๆ ไปบน อาร์เคเอชเอส F ด้วยรีโพรดิวซิ่งเคอร์เนล k ได้ดังนี้

$$f(x)=\sum_i \alpha_i k(x,x_i)=\sum_i \alpha_i k_i(x) \tag{2.5}$$

เมื่อ $\alpha_i \in R, i \in N$ เพราะฉะนั้นการแก้ปัญหาการประมาณค่าฟังก์ชัน จะเปลี่ยนรูปเป็นการประมาณค่าที่เหมาะสมสำหรับตัวแปร α_i ในสมการที่ (5)

ฟังก์ชันที่ถือว่าเป็นฟังก์ชันเคอร์เนลมีหลายชนิดที่นิยมใช้ ยกตัวอย่างเช่น ฟังก์ชันเกาเสียนเรเดียลเบสิสที่ใช้ในระบบข่ายประสาท

$$k(x,x')=\exp(-\beta\|x-x'\|^2) \tag{2.6}$$

โดยที่ $\beta>0$

ฟังก์ชันโพลิโนเมียลของค่าชุดของค่า x กำลัง d แสดงได้โดย

$$k(x,x')=\sum_{k=0}^{d}\frac{1}{k!\rho_k}(xx')^k \tag{2.7}$$

เมื่อ ρ_k คือเซตของค่าคงที่ที่เรียกว่าค่าน้ำหนักและมีค่าเท่ากับ

$$k(x,x')=(1+xx')^d \tag{2.8}$$

ฟังก์ชันเคอร์เนลสามารถแสดงได้ด้วยปริภูมิพาเลย์ – เวียเนอร์ของฟังก์ชันจำกัดแถบ (Band limited function)

$$k(x,x')=\frac{\sin\pi(x-x')}{\pi(x-x')} \tag{2.9}$$

2.2 การประมาณค่าฟังก์ชันในอาร์เคเอชเอสแบบออนไลน์

ในกรณีนี้ได้ตั้งสมมติฐานว่าในแต่ละรอบของการทำซ้ำ เราจะทราบค่าสังเกตเพียงแค่ค่าเดียว นั่นคือ z_n ดังนั้น

$$L_n f = z_n \tag{2.10}$$

จากนั้นเราจะได้ฟังก์ชันนัล $\hat{g}_{reg}$ ที่เวลา n ที่เป็นฟังก์ชันไม่เป็นค่าลบ โดยที่ $\hat{g}_{reg} : Z \to R$

$$\hat{g}_{reg}(f_n) = \frac{1}{2}\left\| L_{n+1} f_n - z_{n+1} \right\|^2 + \frac{\rho}{2}\left\| f_n \right\|^2 \tag{2.11}$$

ตั้งค่าเริ่มต้น f_0 ซึ่งโดยส่วนใหญ่จะให้มีค่าเท่ากับศูนย์ จากนั้นเราจะหาค่า f_n ที่จะทำให้สมการที่ (2.11) มีค่าต่ำที่สุดโดยอาศัยวิธีสโตแคสติกเกรเดียนเดสเซนต์ (Stochastic gradient descent, SGD) ที่แสดงโดย

$$f_{n+1} = f_n - \eta_n \nabla \hat{g}_{reg}(f_n) \tag{2.12}$$

เมื่อที่ $\nabla \hat{g}_{reg}(f_n)$ เป็นเกรเดียน ณ เวลาปัจจุบันของ $\hat{g}_{reg}$ เทียบกับ f_n ดังนั้น

$$\begin{aligned} \hat{g}_{reg}(f_n) &= L_{n+1}^* L_{n+1} f_n - L_{n+1}^* z_{n+1} + \rho f_n \\ &= L_{n+1}^* (L_{n+1} f_n - z_{n+1}) + \rho f_n \end{aligned}$$

แทนค่าในสมการที่ (12) จะได้ว่า

$$f_{n+1} = (1 - \eta_n \rho) f_n - \eta_n L_{n+1}^* (L_{n+1} f_n - z_n) \tag{2.13}$$

โดยที่สำหรับค่าคงที่ใด ๆ a จะสามารถแสดงได้ว่า $L_{n+1}^* a = k_{n+1} a$ และ $L_{n+1} f_n = f_n(x_{n+1})$ ดังนั้น

$$\nabla \hat{g}_{reg}(f_n) = k_{n+1}[f_n(x_{n+1}) - z_{n+1}] + \rho f_n \tag{2.14}$$

เพราะฉะนั้นสมการที่ (12) สามารถเขียนได้เป็น

$$f_{n+1} = (1 - \eta_n \rho) f_n - \eta_n k_{n+1}[f_n(x_{n+1}) - z_{n+1}] \tag{2.15}$$

สมมติว่าการประมาณค่าฟังก์ชัน ณ เวลาปัจจุบันมีเคอร์เนลจำนวน p เทอม ฟังก์ชันการประมาณค่าเหล่านี้สามารถเขียนใหม่ได้เป็น

$$f_{n+1}(x) = (1-\eta_n\rho)\sum_{i=1}^{p}\alpha_n^i k_i(x) - \eta_n e_{n+1}k_{n+1}(x)$$

$$= \sum_{i=1}^{p+1}\alpha_{n+1}^i k_i(x) \qquad (2.16)$$

เมื่อ α_{n+1}^i ถูกคำนวณมาก่อนแล้วดังสมการต่อไปนี้

$$\alpha_{n+1}^i = (1-\eta_n\rho)\alpha_n^i \qquad \text{โดยที่ } i \le p \qquad (2.17)$$

และ

$$\alpha_{n+1}^i = -\eta_n e_{n+1} \qquad \text{โดยที่ } i = p+1 \qquad (2.18)$$

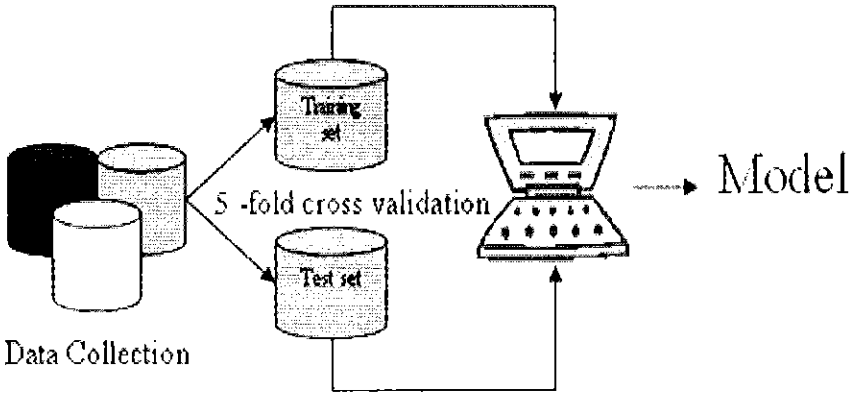
จากสมการที่ผ่านมาเราสามารถสรุปได้ว่า เมื่อค่าน้ำหนัก (α) ค่าใหม่ถูกเพิ่มให้กับแบบจำลอง (f_{n+1}) โดยมีการคูณเข้ากับฟังก์ชันเคอร์เนล ค่าน้ำหนักค่าเก่าจะถูกปรับปรุงโดยการคูณด้วยเทอม ($1-\eta_n\rho$) ซึ่งหมายถึงการลดลงของค่าน้ำหนักค่านั้น ๆ

2.3 ครอสวาลิเดชั่น (Cross validation)

คือวิธีการในการคาดการค่าความผิดพลาดของโมเดลหรือวิธีการที่เรานำเสนอ โดยพื้นฐานของวิธีการครอสวาลิเดชั่นคือการสุ่มตัวอย่าง (Resampling) โดยเริ่มจากแบ่งชุดข้อมูลออก เป็นส่วน ๆ และนำบางส่วนจากชุดข้อมูลนั้นมาตรวจสอบผลลัพธ์จากการทำครอสวาลิเดชั่นมักถูกใช้เป็นตัวเลือกในการกำหนดโมเดล อาทิเช่น สถาปัตยกรรมเครือข่ายการสื่อสาร (Network architecture) โมเดลในการคัดแยกประเภท (Classification model)

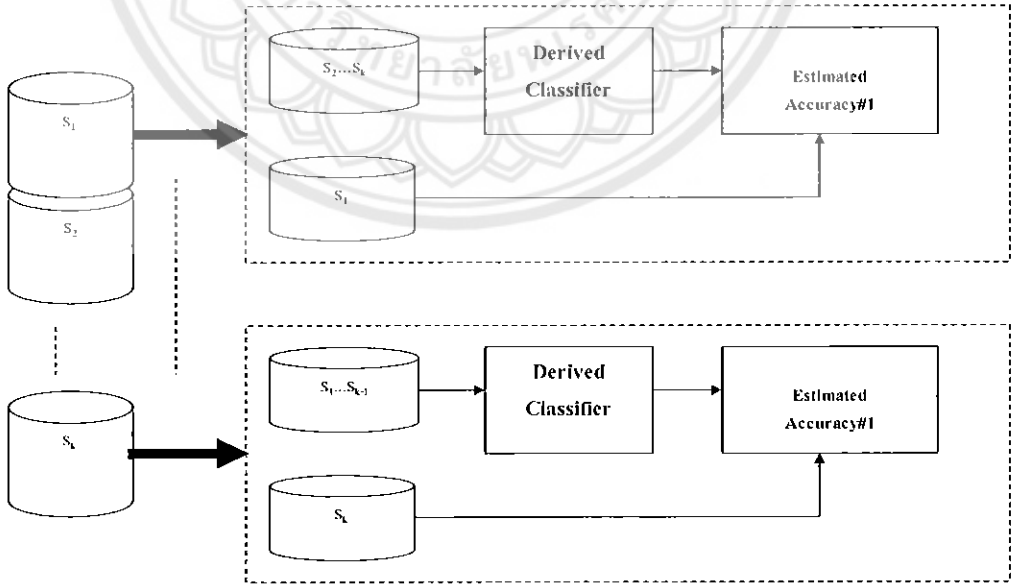
ในการแบ่งแยกข้อมูล โดยใช้เทคนิคของ ดาต้าไมนิ่ง (Data mining) เช่น โครงข่ายประสาทเทียม นั้นจะต้องมีการแบ่งข้อมูลออกเป็นชุดสอน (Train) และชุดทดสอบ (Validation) แต่ในบางครั้งอาจเกิดปัญหาจากการเลือกข้อมูลที่ดีและง่ายมาเป็นข้อมูลชุดทดสอบทำให้ผลการแบ่งแยกนั้นดีเกินจริง ดังนั้นจึงมีการใช้วิธีครอสวาลิเดชั่นแบบเคโฟล์ด (k-fold cross validation) ขึ้นมาแก้ปัญหานการแบ่งข้อมูลเป็น

วิธีนี้จะแบ่งข้อมูลออกเป็นกลุ่มจำนวน k กลุ่ม (k-fold) ในตอนแรกเลือกข้อมูลกลุ่มที่ 1 เป็นข้อมูลชุดทดสอบ และข้อมูลชุดที่เหลือจะเป็นข้อมูลชุดสอน นำข้อมูลไปใช้ จากนั้นจะสลับข้อมูลกลุ่มที่ 2 มาเป็นชุดทดสอบและข้อมูลกลุ่มอื่น ๆ ที่เหลือเป็นชุดสอน สลับอย่างนี้ไปเรื่อย ๆ จนครบ k กลุ่ม ในขั้นตอนสุดท้ายจะหาค่าเฉลี่ยของค่าความผิดพลาดในแต่ละกลุ่ม วิธีการนี้ข้อมูลทุกตัวอย่างจะได้เป็นทั้งชุดสอนและชุดทดสอบ



รูปที่ 2.1 แสดงแผนภาพวิธีครอสวาลิเดชั่นแบบเคโฟล์ด

วิธีวัดประสิทธิภาพแบบนี้จะเหมาะกับกรณีข้อมูลมีจำนวนน้อย วิธีการนี้จะทำให้จำนวนข้อมูลทั้งหมดได้ผ่านขั้นตอนการแบ่งแยกซึ่งแตกต่างจากการใช้วิธีโฮเอาท์ (Hold-out method) ที่กลุ่มสอนจะไม่ได้นำมาทดสอบด้วย โดยส่วนใหญ่แล้วจะนิยมใช้ค่า k เป็น 10 เนื่องจากได้ความถูกต้องเป็นที่น่าพอใจ



รูปที่ 2.2 แสดงการวัดความแม่นตรงแบบครอสวาลิเดชั่นแบบเคโฟล์ด

บทที่ 3
วิธีการประมาณค่าฟังก์ด้วยวิธีเคอร์เนล

โครงงานนี้ได้มุ่งเน้นศึกษาการออกแบบวิธีการประมาณค่าฟังก์ชันของระบบที่ไม่รู้ค่าด้วยวิธีเคอร์เนล โดยอาศัยทฤษฎีที่ได้กล่าวมาแล้วในบทที่ 2 จากนั้นฟังก์ชันการประมาณค่าหรือแบบจำลองที่ได้ จะถูกหาค่าความผิดพลาดโดยใช้ค่าความผิดพลาดเฉลี่ยกำลังสอง (MSE-mean square error) ด้วยวิธีในครอสวาลิเดชันแบบเคโฟล์ด

3.1 หลักการออกแบบการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนล

การออกแบบระบบด้วยวิธีการหาค่าความผิดพลาดของโมเดลโดยการสุ่มตัวอย่าง เป็นการระบุค่าไฮเปอร์พารามิเตอร์ของระบบจากข้อมูลหลาย ๆ ข้อมูลมาทำการหาค่าเฉลี่ยความผิดพลาด ซึ่งจะมีหลักการในการเลือกแบบจำลองที่เหมาะสมของระบบโดยพิจารณาจากค่าที่ผิดพลาดน้อยที่สุด

การประมาณค่าฟังก์ชันมีขั้นตอนการทำดังนี้

1. กำหนดค่าเริ่มต้นค่า β, η, ρ และฟังก์ชัน $f_0 = 0$
2. กำหนดค่า n เริ่มที่ $n = 0$
3. รับค่าข้อมูล (x_{n+1}, z_{n+1}) จากชุดข้อมูลที่ใช้สอน
4. หาค่าความผิดพลาดโดยใช้สมการ $e_{n+1} = f_n(x_{n+1}) - z_{n+1}$
5. หาค่า α จากสมการที่ (2.17) และ (2.18)
6. หาค่าฟังก์ชันการประมาณค่าที่ได้จากสมการ $f_{n+1}(x) = \sum_{i=1}^{P+1} \alpha_{n+1}^{i} k_i(x)$ โดยใช้ฟังก์ชันเคอร์เนล $k(x, x') = \exp(-\beta \|x - x'\|^2)$
7. ทำซ้ำตามขั้นตอนที่ 3-6

ดังนั้นเมื่อเสร็จสิ้นกระบวนการจะได้ฟังก์ชันการประมาณค่า f เพื่อที่จะนำไปวิเคราะห์หาข้อผิดพลาดจากข้อมูลที่ใช้ทดสอบต่อไป

3.2 ขั้นตอนการหาค่าความผิดพลาดของการประมาณค่าฟังก์ชันด้วยวิธีครอสวาลิเดชั่นแบบเคโฟล์ด

ในกรณีการทำครอสวาลิเดชั่นแบบเคโฟล์ดเราจะแบ่งข้อมูลออกเป็น k ชุดเท่า ๆ กัน และทำการคำนวณค่าความผิดพลาด k รอบ โดยแต่ละรอบการคำนวณข้อมูลชุดหนึ่งจากข้อมูล k ชุดจะเลือกออกมาเพื่อเป็นข้อมูลทดสอบและข้อมูลอีก k-1 ชุดจะใช้เป็นข้อมูลสำหรับการสอนดังตัวอย่างต่อไปนี้ที่ทำครอสวาลิเดชั่นแบบเคโฟล์ด (k=5)

ชุดข้อมูลหลังจากแบ่งข้อมูลออกเป็น 5 ชุดข้อมูลย่อยเท่า ๆ กันโดยแต่ละกล่องคือชุดข้อมูลย่อย 1 ชุด

1st data test	2nd data test	3rd data test	4th data test	5th data test

รูปที่ 3.1 แสดงการแบ่งข้อมูล

หลังจากนั้นทำการคำนวณค่าความผิดพลาดเป็นจำนวน k รอบเมื่อ k = 5 โดยกำหนดให้กล่องที่พิมพ์ข้อมูลตัวหนาคือข้อมูลทดสอบและกล่องที่พิมพ์ข้อมูลตัวปกติคือข้อมูลสำหรับการสอน

รอบที่ 1

1st data test	2nd data train	3rd data train	4th data train	5th data train

ค่าความผิดพลาดรอบที่ 1 (e1)

รอบที่ 2

1st data train	**2nd data test**	3rd data train	4th data train	5th data train

ค่าความผิดพลาดรอบที่ 2 (e2)

•

•

รอบที่ 5

1st data train	2nd data train	3rd data train	4th data train	**5th data test**

ค่าความผิดพลาดรอบที่ 5 (e5)

รูปที่ 3.2 แสดงการหาค่าความผิดพลาดของแต่ละรอบ

หลังจากวิธีการข้างต้นนั้นเราจะได้ค่าความผิดพลาดของแต่ละรอบการคำนวณซึ่งประกอบด้วย e1, e2, e3, e4 และ e5 โดยปกตินั้นเราจะหาค่าเฉลี่ยความผิดพลาดและใช้ค่านั้นเป็นตัวแทนของความผิดพลาดของโมเดลหรือวิธีที่นำเสนอ ซึ่งการหาค่าเฉลี่ยความผิดพลาด สามารถแสดงได้ดังสมการต่อไปนี้

$$\text{ค่าเฉลี่ยความผิดพลาด (Average Error)} = \frac{(e1 + e2 + ... + ek)}{k} \qquad (3.1)$$

จากตัวอย่างข้างต้นนั้น ข้อดีของวิธีการนี้คือข้อมูลในแต่ละชุดที่ทำการแบ่งจะถูกทดสอบอย่างน้อย 1 ครั้ง และถูกเรียนรู้ทั้งหมด k-1 ครั้ง โดยในขั้นตอนเหล่านี้เราสามารถกำหนดได้ว่าต้องการข้อมูลขนาดใด และต้องการคำนวณเป็นจำนวนรอบเท่าใด แต่อย่างไรก็ตามเมื่อมองในมุมกลับกันวิธีนี้ใช้การคำนวณเป็น k เท่า

บทที่ 4
ผลการทดลองและการวิเคราะห์ผล

ในบทนี้จะกล่าวถึงผลการทดลองของวิธีการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลและการประมาณค่าที่ได้ด้วยวิธีครอสวาลิเดชั่นแบบเคโฟล์ดที่ได้ศึกษามา ซึ่งจะทำการปรับค่าพารามิเตอร์ต่าง ๆ และดูผลที่เกิดขึ้น โดยแต่ละค่าพารามิเตอร์ที่แตกต่างกันจะให้ผลของค่าความผิดพลาด (ค่าความผิดพลาดเฉลี่ยกำลังสองหรือ MSE) ที่แตกต่างกันออกไป แต่สิ่งที่ต้องการคือการลดลงของค่าความผิดพลาดต้องค่อย ๆ ลดลงและค่าความผิดพลาดที่ได้ต้องน้อยที่สุด ซึ่งเป็นสิ่งที่ระบบต้องการ

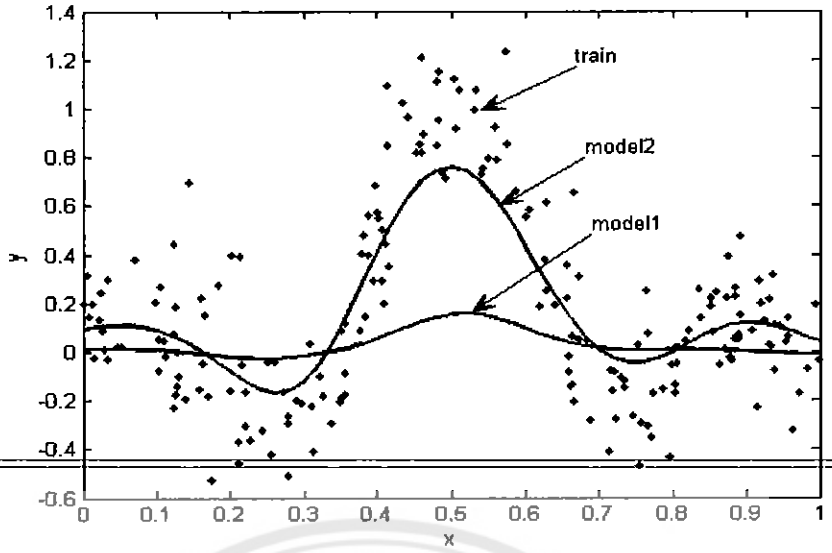
4.1 ผลการทดลองการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนล

การประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลนั้นสามารถทำได้โดยใช้สมการที่ (2.16) และจะใช้ฟังก์ชันเกาเสียนเรเดียลเบสิสดังสมการที่ (2.6) เป็นฟังก์ชันเคอร์เนล ในการทดลองหาการประมาณค่าฟังก์ชัน จะมีขั้นตอนตามที่ได้กล่าวมาแล้วในบทที่ 3 หัวข้อที่ 3.1 ซึ่งจะใช้โปรแกรมแมทแลปในการประมาณค่าฟังก์ชัน และการหาค่าความผิดพลาดจากข้อมูลที่ใช้ทดสอบ สามารถหาได้ดังสมการต่อไปนี้

$$\text{ค่าความผิดพลาด (MSE)} = \frac{(e1)^2 + (e2)^2 + \ldots + (en)^2}{n} \tag{4.1}$$

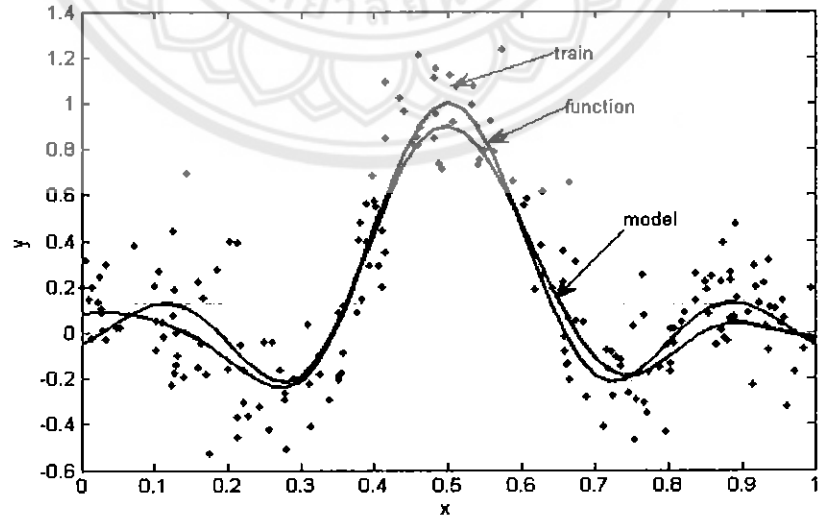
เมื่อ n คือจำนวนของข้อมูล

ข้อมูลที่ใช้ในการทดลองถูกสร้างขึ้นโดยใช้โปรแกรมแมทแลป ซึ่งแบ่งข้อมูลออกเป็น 2 ชุด คือชุดสอน จำนวน 200 ค่า และชุดทดสอบ จำนวน 50 ค่า



รูปที่ 4.1 แสดงการเปรียบเทียบข้อมูลที่ใช้สอนและที่ได้จากการทดลอง

จากรูปที่ 4.1 เป็นการแสดงแบบจำลองที่ได้จากการประมาณค่าฟังก์ชัน โดยกำหนดให้ข้อมูลชุดสอนจำนวน 200 ค่า จะเห็นได้ว่าที่แบบจำลอง 1 ได้จากการคำนวณหาการประมาณค่าฟังก์ชันจำนวน 10 รอบ และที่แบบจำลอง 2 จำนวน 100 รอบ ซึ่งจะเห็นได้ว่าเมื่อเพิ่มรอบในการคำนวณมากขึ้นจะทำให้แบบจำลองที่ได้ใกล้เคียงกับฟังก์ชันจริง ดังรูปที่ 4.2



รูปที่ 4.2 แสดงการเปรียบเทียบฟังก์ชันจริงของข้อมูลที่ใช้สอนและที่ได้จากการทดลอง

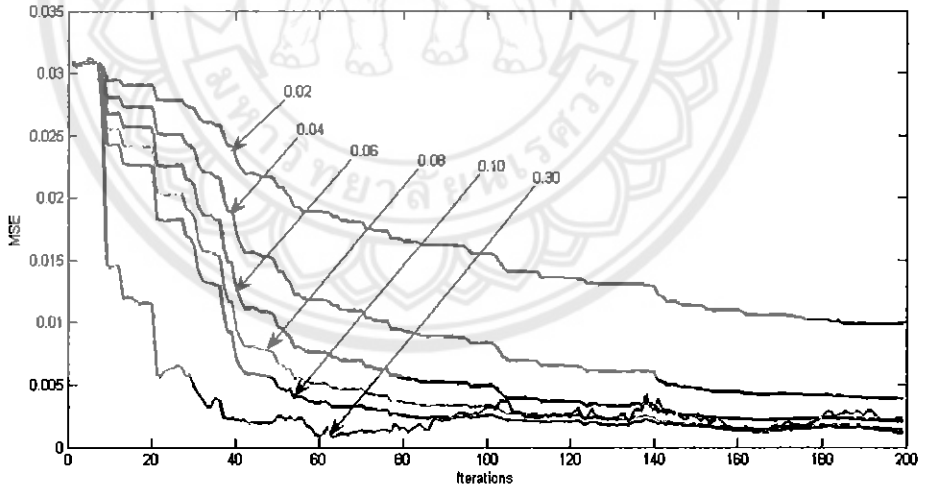
จากรูปที่ 4.2 เป็นการแสดงการเปรียบเทียบฟังก์ชันจริงของข้อมูลที่ใช้สอนและแบบจำลองที่ได้จากการคำนวณการประมาณค่าฟังก์ชันจำนวน 199 รอบ ซึ่งจะเห็นได้ว่าแบบจำลองที่ได้มีค่าใกล้กับฟังก์ชันจริงมาก

การประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลจะมีการทดลองปรับค่าพารามิเตอร์ต่าง ๆ ให้ได้ค่าพารามิเตอร์ที่ดีที่สุด เพื่อหาค่าความผิดพลาดที่ดีที่สุด ซึ่งจะมีหลักการในการเลือกโดยพิจารณาจากค่าความผิดพลาดที่น้อยที่สุดของระบบ โดยจะมีการปรับค่าพารามิเตอร์ 3 ตัว คือค่าอัตราการเรียนรู้ ค่าแรกกูลาไรเซชัน และค่าเบต้า ซึ่งมีวิธีในการปรับค่าต่าง ๆ ดังต่อไปนี้

4.1.1 ผลของการปรับค่าอัตราการเรียนรู้ (η)

ในการปรับค่าอัตราการเรียนรู้นั้นเราจะกำหนดให้ค่าแรกกูลาไรเซชั่นเท่ากับ 0.001 และค่าเบต้าเท่ากับ 100 โดยสองค่านี้จะให้คงที่ แล้วทำการปรับค่าอัตราการเรียนรู้

ผลที่ได้จากการปรับค่าอัตราการเรียนรู้ แล้วหาค่าความผิดพลาดจากสมการที่ (4.1) สามารถแสดงได้ดังรูปที่ 4.3



รูปที่ 4.3 แสดงการเปรียบเทียบผลของค่าความผิดพลาดของการปรับค่า η

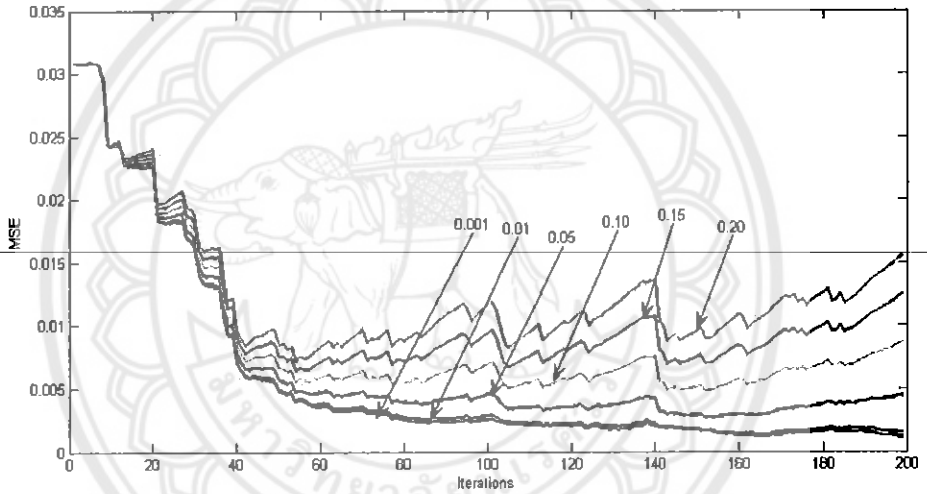
จากรูปที่ 4.3 จะเป็นการนำผลของค่าความผิดพลาดที่ได้จากการปรับค่าอัตราการเรียนรู้ต่าง ๆ มาเปรียบเทียบกัน ซึ่งจะเห็นได้ว่าที่ $\eta = 0.10$ จะมีการลดลงของค่าความผิดพลาดดีที่สุด คือ

ในช่วงแรกจะลดลงอย่างฉับพลัน แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิดพลาดนั้นมีลักษณะคงที่ แต่เมื่อปรับค่า η เท่ากับ 0.02, 0.04, 0.06, 0.08 และ0.30 จะมีการลดลงของค่าความผิดพลาดที่ไม่ดีคือค่าความผิดพลาดที่ได้จะมีค่ามากขึ้น จะเห็นได้จากรูปการทดลองที่ 4.3

4.1.2 ผลของการปรับค่าแรกกูลาไรเซชั่น (ρ)

ในการปรับค่าแรกกูลาไรเซชันนั้นเราจะกำหนดให้ค่าอัตราการเรียนรู้เท่ากับ 0.10 และค่าเบต้าเท่ากับ 100 โดยสองค่านี้จะให้คงที่ แล้วทำการปรับค่าแรกกูลาไรเซชั่น

ผลที่ได้จากการปรับค่าแรกกูลาไรเซชั่น แล้วหาค่าความผิดพลาดจากสมการที่ (4.1) สามารถแสดงได้ดังรูปที่ 4.4



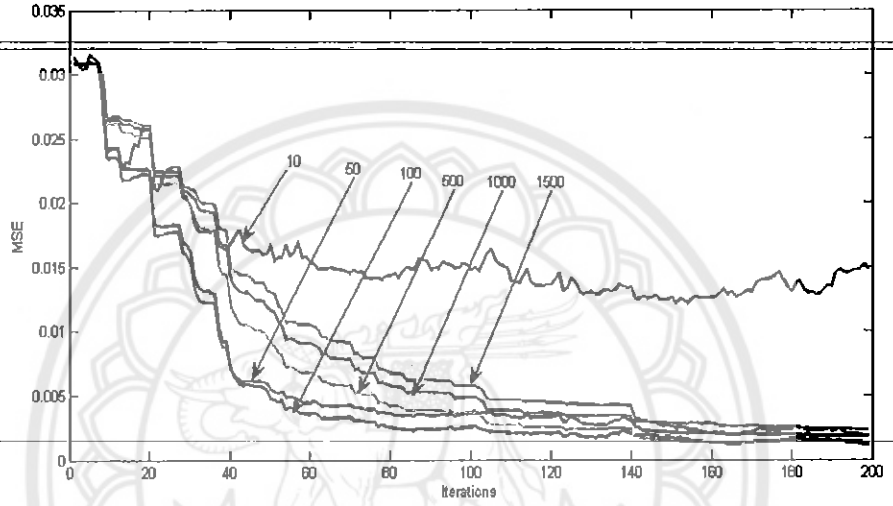
รูปที่ 4.4 แสดงการเปรียบเทียบผลของค่าความผิดพลาดของการปรับค่า ρ

จากรูปที่ 4.4 จะเป็นการนำผลของค่าความผิดพลาดที่ได้จากการปรับค่าแรกกูลาไรเซชั่นต่าง ๆ มาเปรียบเทียบกัน จะเห็นว่าค่าความผิดพลาดที่ $\rho = 0.001$ มีลักษณะการลดลงของค่าความผิดพลาดที่ดีที่สุด โดยในช่วงแรกจะลดลงอย่างฉับพลันตั้งแต่รอบที่ 0-50 แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิดพลาดนั้นมีลักษณะคงที่ ส่วนที่ค่า $\rho = 0.05$, 0.10, 0.15 และ 0.20 ค่าความผิดพลาดมีการลดลงที่ไม่ดี คือในช่วงแรกค่าความผิดพลาดจะลดลงอย่างฉับพลัน แล้วหลังจากนั้นจะค่อย ๆ เพิ่มขึ้นอีก ดังรูปที่ 4.4

4.1.3 ผลของการปรับค่าเบต้า (β)

ในการปรับค่าเบต้านั้นเราจะกำหนดให้ค่าเรกกูลาไรเซชันเท่ากับ 0.001 และค่าอัตราการเรียนรู้เท่ากับ 0.10 โดยสองค่านี้จะให้คงที่ แล้วทำการปรับค่าเบต้า

ผลที่ได้จากการปรับค่าค่าเบต้า แล้วหาค่าความผิดพลาดจากสมการที่ (4.1) สามารถแสดงได้ดังรูปที่ 4.5



รูปที่ 4.5 แสดงการเปรียบเทียบผลของค่าความผิดพลาดของการปรับค่า β

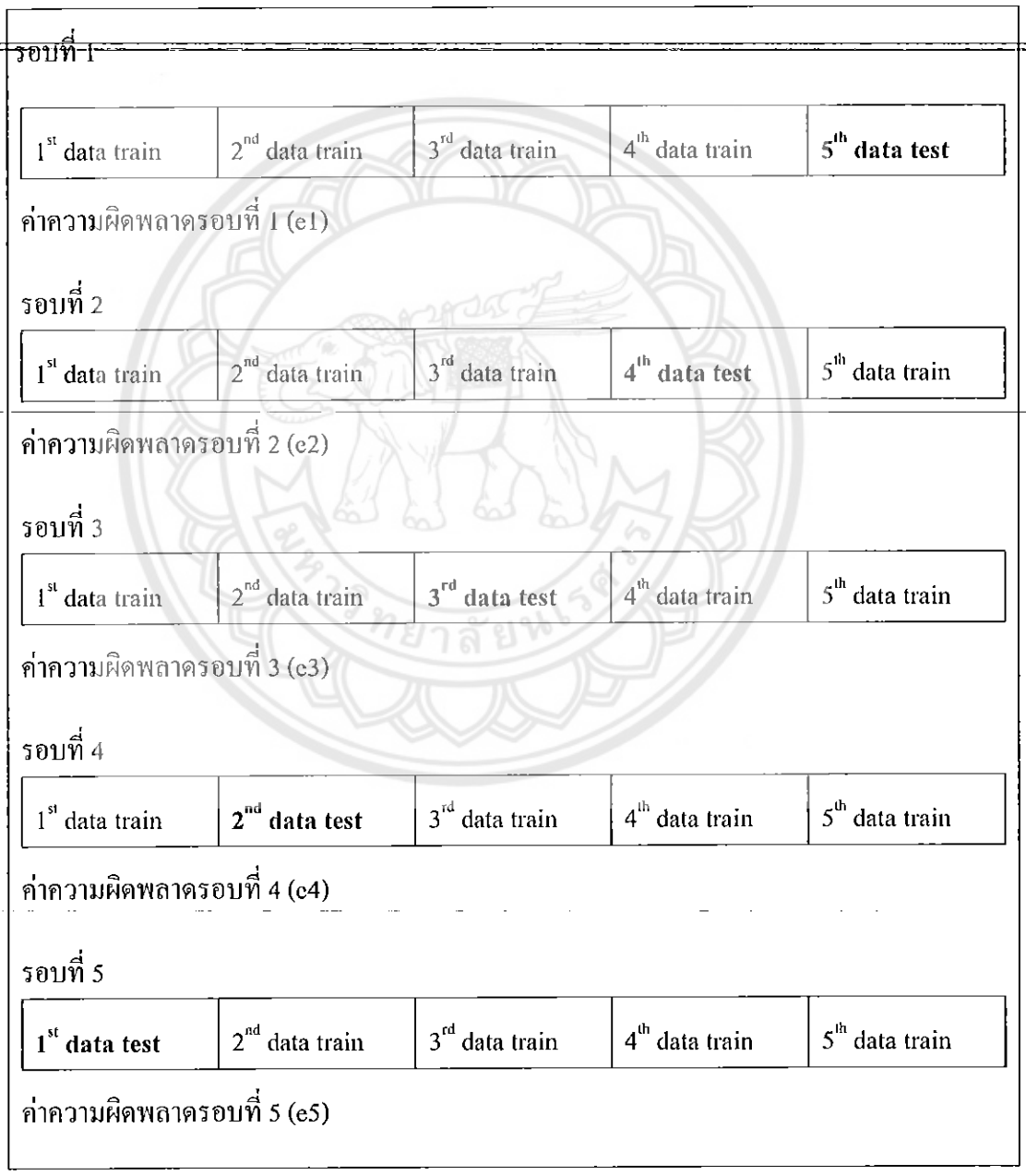
จากรูปที่ 4.5 จะเป็นการนำผลของค่าความผิดพลาดที่ได้จากการปรับค่าเบต้าต่าง ๆ มาเปรียบกัน จะเห็นว่าค่าความผิดพลาดที่ $\beta = 100$ มีการลดลงของค่าความผิดพลาดที่ดี โดยในช่วงแรกจะลดลงอย่างฉับพลันตั้งแต่รอบที่ 0–50 แล้วหลังจากนั้นจะค่อย ๆ ลดลงจนค่าความผิดพลาดนั้นมีลักษณะคงที่ ส่วนที่ค่า β = 10, 50, 500, 1000 และ 1500 ค่าความผิดพลาดที่ได้จะมีค่ามากและการลดลงของค่าความผิดพลาดไม่ดี ดังรูปที่ 4.5

4.2 ค่าความผิดพลาดของการประมาณค่าฟังก์ชันด้วยวิธีครอสวาลิเดชั่นแบบเคโฟล์ด

วิธีครอสวาลิเดชั่นนี้เราจะสร้างข้อมูลโดยใช้โปรแกรมแมทแลปจำนวน 1000 ค่า แล้วทำการแบ่งข้อมูลออกเป็น 5 ชุด นั่นคือเราให้ค่า k เป็น 5 ชุด ชุดละเท่า ๆ กัน ในแต่ละชุดนั้นเราจะให้เป็นชุดสอนและชุดทดสอบ จำนวน 200 ค่า แล้วทำการคำนวณหาค่าความผิดพลาด k รอบ โดยแต่ละรอบการคำนวณข้อมูลชุดหนึ่งจะถูกเลือกออกมา เพื่อเป็นข้อมูลทดสอบ และข้อมูลที่เหลืออีก 4

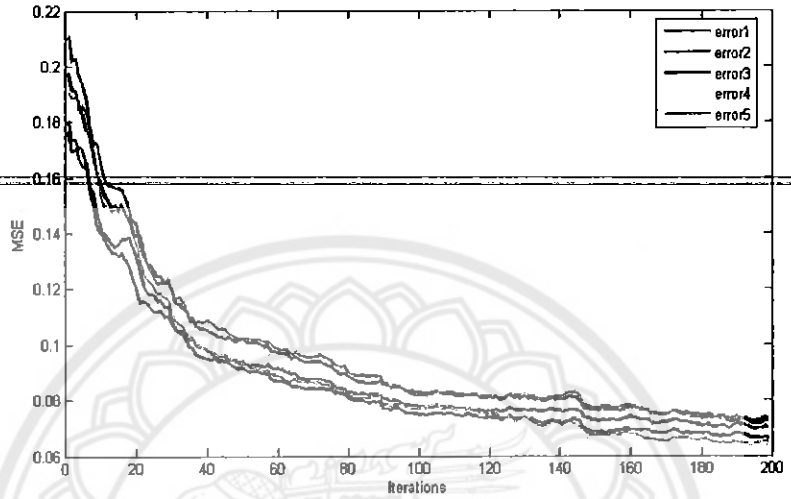
ชุดจะใช้เป็นข้อมูลสำหรับสอน ซึ่งจะสลับชุดข้อมูลที่ใช้ทดสอบไปเรื่อย ๆ จนครบทั้ง 5 ชุด จากนั้นหาค่าความผิดพลาดในแต่ละรอบ แล้วนำค่าที่ได้มาคำนวณหาค่าเฉลี่ยความผิดพลาด จากสมการที่ (3.1)

การแบ่งชุดข้อมูลที่ใช้ในการคำนวณหาค่าความผิดพลาดของแต่ละรอบ สามารถแสดงได้ดังรูปที่ 4.6



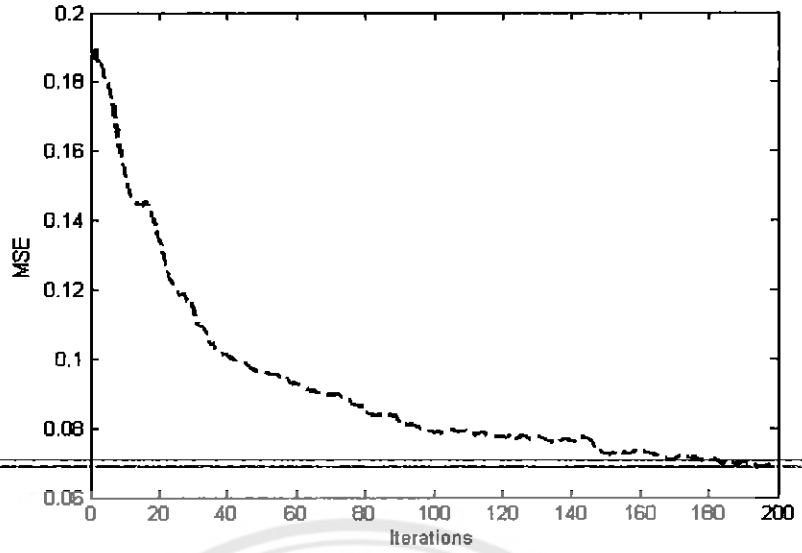
รูปที่ 4.6 แสดงการแบ่งข้อมูล 5 ชุด

ในการทดลองเราจะเลือกใช้ค่าอัตราการเรียนรู้เท่ากับ 0.10 ค่าเรกกูลาไรเซชั่นเท่ากับ 0.001 และค่าเบต้าเท่ากับ 100 ซึ่งค่าพารามิเตอร์เหล่านี้ทำให้ได้ผลของค่าความผิดพลาดดีที่สุด จะเห็นได้จากการทดลองที่ 4.1



รูปที่ 4.7 แสดงผลการเปรียบเทียบค่าความผิดพลาดทั้ง 5 รอบ

เมื่อนำค่าความผิดพลาดที่ได้จากข้อมูลทั้ง 5 ชุดมาเปรียบเทียบกัน สามารถแสดงได้ดังรูปที่ 4.7 จะเห็นว่าค่าความผิดพลาดที่ได้จะมีลักษณะคล้าย ๆ กันแต่ไม่เหมือนกันถึงแม้ว่าจะมาจากข้อมูลที่ได้จากระบบเดียวกัน และผลที่เกิดขึ้นจากการหาค่าเฉลี่ยความผิดพลาดทั้ง 5 รอบ สามารถแสดงได้ดังรูปที่ 4.7



รูปที่ 4.8 แสดงผลของค่าเฉลี่ยความผิดพลาดทั้ง 5 รอบจากข้อมูล 5 ชุด

จากรูปที่ 4.8 ค่าเฉลี่ยความผิดพลาดที่ได้จะมีลักษณะการลดลงที่ดี คือในช่วงแรกมีการลดลงอย่างฉับพลัน แล้วหลังจากนั้นค่าความผิดพลาดจะค่อย ๆ ลดลงจนเกือบคงที่

บทที่ 5
สรุปผลและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงผลของการทดลองที่เกิดขึ้นในการหาค่าความผิดพลาดด้วยวิธีการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลและการประมาณค่าฟังก์ชันที่ได้ด้วยวิธีครอสวาลิเดชั่นแบบเคโฟล์ดที่ได้ศึกษามา ว่าจะได้ค่าความผิดพลาดที่เพิ่มขึ้นหรือลดลงอย่างไร เพื่อที่จะหาวิธีการที่สามารถลดค่าความผิดพลาดที่ดีที่สุด

5.1 สรุปผลการทดลอง

ในโครงงานนี้ได้ศึกษาการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลและการประมาณค่าฟังก์ชันที่ได้ด้วยวิธีครอสวาลิเดชั่นแบบเคโฟล์ด เพื่อหาค่าความผิดจากข้อมูลที่ใช้ทดลอง แล้วเลือกค่าความผิดพลาดที่มีค่าน้อยที่สุดมาใช้เป็นแบบจำลองต่อไป

5.1.1 การหาค่าพารามิเตอร์ที่ดีที่สุดของการประมาณค่าฟังก์ชัน

จากการทดลองในบทที่ 4 จะมีการปรับค่าพารามิเตอร์ 3 ตัว คือ ค่าอัตราการเรียนรู้ ค่าเรกกูลาไรเซชัน และค่าเบต้า เพื่อหาค่าความผิดพลาดที่ดีที่สุด ซึ่งผลการทดลองที่ได้จะเห็นว่าที่ค่าอัตราการเรียนรู้เท่ากับ 0.10 ค่าเรกกูลาไรเซชันเท่ากับ 0.001 และค่าเบต้าเท่ากับ 100 เป็นค่าที่ดีที่สุด แสดงให้เห็นว่าค่าพารามิเตอร์ทั้ง 3 ตัวนี้มีผลต่อการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนล โดยที่ค่าอัตราการเรียนรู้คือช่วงของการลู่เข้าสู่ค่าความผิดพลาดที่ศูนย์ ค่าเรกกูลาไรเซชันคือความซับซ้อนของแบบจำลอง และค่าเบต้าคือความกว้างของเคอร์เนลฟังก์ชัน

5.1.2 การประมาณค่าฟังก์ชันที่ได้ด้วยวิธีครอสวาลิเดชั่นแบบเคโฟล์ด

วิธีนี้จะทำการแบ่งข้อมูลออกเป็นชุด ชุดละเท่า ๆ กันแล้วทำการหาค่าความผิดพลาดในแต่ละรอบ จากนั้นนำค่าความผิดพลาดของแต่ละรอบที่ได้มาหาค่าเฉลี่ยความผิดพลาด

จากการทดลองในหัวข้อ 4.2 ค่าเฉลี่ยความผิดพลาดของการประมาณค่าฟังก์ชันด้วยวิธีครอสวาลิเดชันแบบเคโฟล์ดจะมีการลดลงอย่างต่อเนื่องจนค่าที่ลดลงนั้นมีลักษณะค่อนข้างคงที่และค่าความผิดพลาดที่ได้มีค่าน้อย ซึ่งเป็นสิ่งที่ระบบต้องการ

การประมาณค่าฟังก์ชันที่ได้ด้วยวิธีครอสวาลิเดชันแบบเคโฟล์ดนี้ จะมีข้อดีคือข้อมูลในแต่ละชุดที่ทำการแบ่งจะถูกทดสอบอย่างน้อย 1 ครั้ง และถูกสอนทั้งหมด k-1 ครั้งโดยในขั้นตอนเหล่านี้เราสามารถกำหนดได้ว่าต้องการข้อมูลขนาดใดและต้องการคำนวณเป็นจำนวนรอบเท่าใด

และขั้นตอนสุดท้ายจะหาค่าเฉลี่ยของค่าความผิดพลาดในแต่ละกลุ่มออกมา วิธีการนี้ข้อมูลทุกตัวจะได้เป็นทั้งชุดสอนและชุดทดสอบ

5.2 ปัญหาและแนวทางแก้ไข

ในการทดลองการประมาณค่าฟังก์ชันด้วยวิธีเคอร์เนลนั้น เมื่อข้อมูลที่ใช้ในการทดลองมีจำนวนมากแล้วใช้โปรแกรมแมทแลปในการคำนวณหาค่าความผิดพลาด จะทำให้การประมวลผลของคอมพิวเตอร์ล่าช้ามาก แนวทางแก้ไขคือค้นคว้าหาวิธีการคำนวณที่ลดการคำนวณที่ซ้ำซ้อนกันให้มากที่สุด

สำหรับโปรแกรมการทดลองทั้งหมดจะรวมอยู่ในแผ่นซีดีที่แนบมาพร้อมกับปริญญานิพนธ์เล่มนี้แล้ว ซึ่งสามารถนำไปใช้ในการทดลองหรือนำไปประยุกต์ใช้งานต่อไป

เอกสารอ้างอิง

[1] J.Platt, "**A resource - allocating network for function interpolation**,"*Neural Computation,* vol. 3, pp. 213-225, 1991.

[2] N. Aronszajn,"**Theory of reproducing kernels**," *Transactions of the American Mathemati cal Socity*, vol. 68, pp. 337-404, 1950.

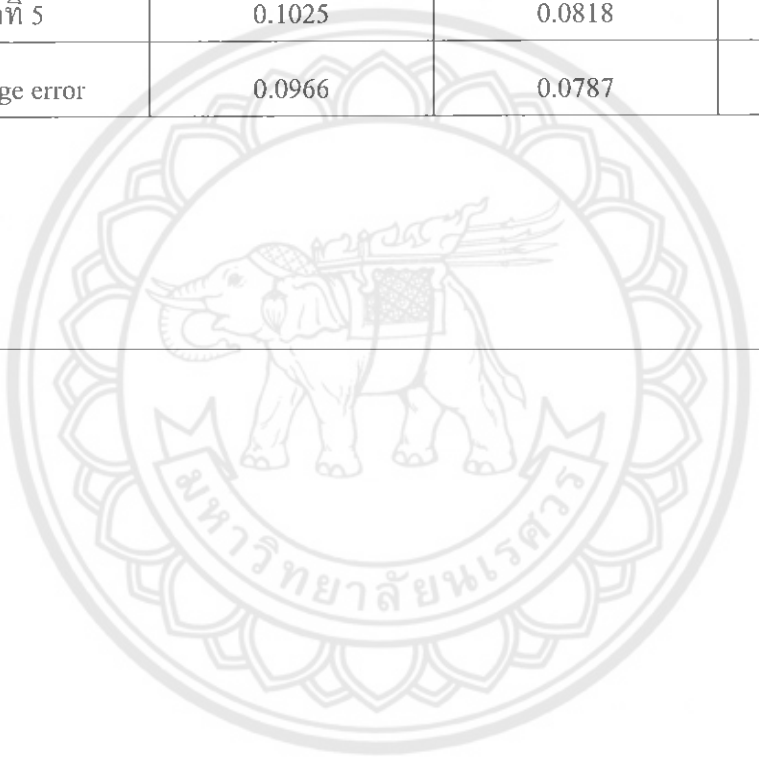
[3] T.Podd, "**Gradient descent approach to approximation in reproducing kernel Hilbert spaces**," Department of Automatic Control and Systems Engineering University of Shef field, UK, Tech. Rep. 821, 2002.

[4] สัญฉกร วุฒิสิทธิกุลกิจ และคณะ. " **MATLABการประยุกต์ใช้งานทางวิศวกรรมไฟฟ้า", พิมพ์** ครั้งที่ 3, สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, 2551.

ภาคผนวก ก
แสดงข้อมูลที่ได้จากการทดลอง

	η	ρ	β	จำนวนรอบในการทดลอง		
				50	100	200
	0.02	0.001	100	0.0215	0.0155	0.0099
	0.04	0.001	100	0.0150	0.0083	0.0039
	0.06	0.001	100	0.0106	0.0049	0.0020
ปรับค่า η	0.08	0.001	100	0.0075	0.0033	0.0014
	0.10	0.001	100	0.055	0.0025	0.0012
	0.30	0.001	100	0.0020	0.0029	0.0023
	0.10	0.001	100	0.0055	0.0025	0.0012
	0.10	0.01	100	0.057	0.0028	0.0016
	0.10	0.05	100	0.0065	0.0044	0.0045
ปรับค่า ρ	0.10	0.10	100	0.0076	0.0068	0.0087
	0.10	0.15	100	0.0086	0.0920	0.0125
	0.10	0.20	100	0.0096	0.0113	0.0156
	0.10	0.001	10	0.0160	0.0152	0.0148
	0.10	0.001	50	0.0060	0.0037	0.0019
	0.10	0.001	100	0.055	0.0025	0.0012
ปรับค่า β	0.10	0.001	500	0.0097	0.0036	0.0014
	0.10	0.001	1000	0.0124	0.0048	0.0018 ,
	0.10	0.001	1500	0.0138	0.0058	0.0023

ข้อมูล	จำนวนรอบในการทดลอง		
	50	**100**	**199**
ชุดที่ 1	0.0915	0.0748	0.0663
ชุดที่ 2	0.0939	0.0775	0.0701
ชุดที่ 3	0.1019	0.0827	0.0735
ชุดที่ 4	0.0931	0.0769	0.0649
ชุดที่ 5	0.1025	0.0818	0.0724
Average error	0.0966	0.0787	0.0695



ภาคผนวก ข
โปรแกรมที่ใช้ในการทดลอง

โปรแกรมที่ใช้ในการทดลองในบทที่ 4

1) โปรแกรมที่ใช้ในการปรับค่าอัตราการเรียนรู้

```
clear all; clc
rand('seed',1032423);
randn('seed',42434123);

x_train = rand(200,1); %ข้อมูลที่ใช้สอน (x)
y_train = sinc((20*x_train-10)/pi);
y_train = y_train + 0.2*randn(size(x_train )); %ข้อมูลที่ใช้สอน (y)
x_test  = [0.1:0.1:5]';  %ข้อมูลที่ใช้ทดสอบ (x)
y_idealtest = sinc((20*x_test-10)/pi); %ข้อมูลที่ใช้ทดสอบ (y)

no_data = size(x_train,1);
%กำหนดค่าพารามิเตอร์
eta  = 0.02;  %ปรับค่าอัตราการเรียนรู้เท่ากับ 0.02, 0.04, 0.06, 0.08, 0.10 และ0.30 ตามลำดับ
rho  = 0.001; %กำหนดให้ค่าเรกกูลาไรเซชั่นคงที่
beta = 100;   %กำหนดให้ค่าเบต้าคงที่


 for n=1
    f=[0]; %กำหนดค่าเริ่มต้น
    e(n)=-y_train(n);
    alpha=(-eta)*e(n);
  for i=1:50
      xtest = x_test(i);
      yidealtest = y_idealtest(i);
      f(i)=alpha*exp(-beta*(abs((x_train(n)-xtest))).^2); %kernel(x_train(n),xtest)
      ff=[f(i)]';
      error(i)=((ff-yidealtest).^2);

   end
```

```
    MSE=sum(error)/length(y_idealtest);
    f_predict=alpha*exp(-beta*(abs(x_train(n)-x_train(n+1))).^2);
end

for n=2:199
  e(n)=f_predict-y_train(n);
  alpha=[alpha*(1-eta*rho); -eta*e(n)];

  for i=1:50
    xtest = x_test(i);
    yidealtest = y_idealtest(i);
    for j=1:n
      ff(j)=alpha(j)*exp(-beta*(abs(x_train(j)- x_test(i))).^2);
    end
    ff= sum(ff);

    error(i)=((ff-y_idealtest(i)).^2);

  end
    MSE(n)=sum(error)/length(y_idealtest);
  for j=1:n
    ff(j)=alpha(j)*exp(-beta*(abs(x_train(n+1)-x_train(j))).^2) ;
  end
    f_predict=sum(ff);
end
MSE; %ค่าเฉลี่ยความผิดพลาดกำลังสอง
plot(MSE) %กราฟของค่าความผิดพลาด
```

2) โปรแกรมที่ใช้ในการปรับค่าแรกกูลาไรเซชั่น

```
clear all; clc
rand('seed',1032423);
randn('seed',42434123);

x_train = rand(200,1); %ข้อมูลที่ใช้สอน (x)
y_train = sinc((20*x_train-10)/pi);
y_train = y_train + 0.2*randn(size(x_train )); %ข้อมูลที่ใช้สอน (y)
x_test = [0.1:0.1:5]'; %ข้อมูลที่ใช้ทดสอบ x
y_idealtest = sinc((20*x_test-10)/pi); %ข้อมูลที่ใช้ทดสอบ (y)

no_data = size(x_train,1);
%กำหนดค่าพารามิเตอร์
eta  = 0.10;   %กำหนดให้ค่าอัตราการเรียนรู้คงที่
rho  = 0.001; %ปรับค่าแรกกูลาไรเซชั่น เท่ากับ 0.001, 0.01, 0.05, 0.10, 0.15 และ0.20
                ตามลำดับ
beta = 100;   %กำหนดให้ค่าเบต้าคงที่


 for n=1
    f=[0]; %กำหนดค่าเริ่มต้น
    c(n)=-y_train(n);
    alpha=(-eta)*e(n);
   for i=1:50
      xtest = x_test(i);
      yidealtest = y_idealtest(i);
      f(i)=alpha*exp(-beta*(abs((x_train(n)-xtest))).^2); %kernel(x_train(n),xtest)
      ff=[f(i)]';
      error(i)=((ff-yidealtest).^2);

   end
      MSE=sum(error)/length(y_idealtest);
```

```
    f_predict=alpha*exp(-beta*(abs(x_train(n)-x_train(n+1))).^2);
end

for n=2:199
  e(n)=f_predict-y_train(n);
  alpha=[alpha*(1-eta*rho); -eta*e(n)];

  for i=1:50
    xtest = x_test(i);
    yidealtest = y_idealtest(i);
    for j=1:n
      ff(j)=alpha(j)*exp(-beta*(abs(x_train(j)- x_test(i))).^2);
    end
    ff= sum(ff);

    error(i)=((ff-y_idealtest(i)).^2);

  end
    MSE(n)=sum(error)/length(y_idealtest);
  for j=1:n
    ff(j)=alpha(j)*exp(-beta*(abs(x_train(n+1)-x_train(j))).^2) ;
  end
    f_predict=sum(ff);
end
MSE; %ค่าเฉลี่ยความผิดพลาดกำลังสอง
plot(MSE) %กราฟของค่าความผิดพลาด
```

3) โปรแกรมที่ใช้ในการปรับค่าเบต้า

```
clear all; clc
rand('seed',1032423);
randn('seed',42434123);

x_train = rand(200,1); %ข้อมูลที่ใช้สอน (x)
y_train = sinc((20*x_train-10)/pi);
y_train = y_train + 0.2*randn(size(x_train )); %ข้อมูลที่ใช้สอน (y)
x_test  =[0.1:0.1:5]';  %ข้อมูลที่ใช้ทดสอบ (x)
y_idealtest = sinc((20*x_test-10)/pi); %ข้อมูลที่ใช้ทดสอบ (y)

no_data = size(x_train,1);
%กำหนดค่าพารามิเตอร์
cta  = 0.10;   %กำหนดให้ค่าอัตราการเรียนรู้คงที่
rho  = 0.001; %กำหนดให้ค่าแรกกูลาไรเซชั่นคงที่
beta = 10;    %ปรับค่าเบต้าเท่ากับ 10, 50, 100, 500, 1000 และ1500 ตามลำดับ

 for n=1
     f=[0]; %กำหนดค่าเริ่มต้น
     e(n)=-y_train(n);
     alpha=(-cta)*e(n);
    for i=1:50
       xtest = x_test(i);
       yidealtest = y_idealtest(i);
       f(i)=alpha*exp(-beta*(abs((x_train(n)-xtest))).^2); %kernel(x_train(n),xtest)
       ff=[f(i)]';
       error(i)=((ff-yidealtest).^2);

    end
       MSE=sum(error)/length(y_idealtest);
       f_predict=alpha*exp(-beta*(abs(x_train(n)-x_train(n+1))).^2);
 end
```

```
for n=2:199
   e(n)=f_predict-y_train(n);
   alpha=[alpha*(1-eta*rho); -eta*e(n)];

   for i=1:50
      xtest = x_test(i);
      yidealtest = y_idealtest(i);
      for j=1:n
         ff(j)=alpha(j)*exp(-beta*(abs(x_train(j)- x_test(i))).^2);
      end
      ff= sum(ff);

      error(i)=((ff-y_idealtest(i)).^2);

   end
      MSE(n)=sum(error)/length(y_idealtest);
   for j=1:n
      ff(j)=alpha(j)*exp(-beta*(abs(x_train(n+1)-x_train(j))).^2) ;
   end
      f_predict=sum(ff);
end
MSE; %ค่าเฉลี่ยความผิดพลาดกำลังสอง
plot(MSE) %กราฟของค่าความผิดพลาด
```

4) โปรแกรมที่ใช้ทดลองด้วยวิธีครอสวาลิเดชั่น

```
clear all; clc
rand('seed',1032423);
randn('seed',42434123);

x = rand(1000,1);  %ข้อมูลที่ใช้สอน(x)
y = sinc((20*x-10)/pi);
y = y + 0.2*randn(size(x));  %ข้อมูลที่ใช้สอน(y)

%การแบ่งข้อมูล
x1=x(1:200);   %ข้อมูลชุดที่ 1
y1=y(1:200);

x2=x(201:400);  %ข้อมูลชุดที่ 2
y2=y(201:400);

x3=x(401:600);  %ข้อมูลชุดที่ 3
y3=y(401:600);

x4=x(601:800);  %ข้อมูลชุดที่ 4
y4=y(601:800);

x5=x(801:1000); %ข้อมูลชุดที่ 5
y5=y(801:1000);

x_train = x1; %ข้อมูลชุดสอน(x)
y_train = y1; %ข้อมูลชุดสอน(y)

x_test  = x5;     %ข้อมูลชุดทดสอบ(x)
y_idealtest = y5;  %ข้อมูลชุดทดสอบ(y)

no_data = size(x_train,1);
```

```
%กำหนดค่าของพารามิเตอร์
beta =100;
eta  =0.10;
rho  =0.001;

 for n=1
   f=[0];  %กำหนดค่าเริ่มต้น
   e(n)=-y_train(n);
   alpha=(-eta)*e(n);
  for i=1:51;
    xtest = x_test(i);
    yidealtest = y_idealtest(i);
    f(i)=alpha*exp(-beta*(abs((x_train(n)-xtest))).^2); %kernel(x_train(n),xtest)
    ff=[f(i)]';

    error(i)=((ff-yidealtest).^2);

 end
    MSE=sum(error)/length(y_idealtest);
    f_predict=alpha*exp(-beta*(abs(x_train(n)-x_train(n+1))).^2);
end

for n=2:100
   e(n)=f_predict-y_train(n);
   alpha=[alpha*(1-eta*rho); -eta*e(n)];

   for i=1:51
     xtest = x_test(i);
     yidealtest = y_idealtest(i);
     for j=1:n
       ff(j)=alpha(j)*exp(-beta*(abs(x_train(j)- x_test(i))).^2);
     end
```

```
        ff= sum(ff);
        yff(i)=ff;

        error(i)=((ff-y_idealtest(i)).^2);

    end
        MSE(n)=sum(error)/length(y_idealtest);
    for j=1:n
        ff(j)=alpha(j)*exp(-beta*(abs(x_train(n+1)-x_train(j))).^2) ;
    end
        f_predict=sum(ff);
 end
data11=MSE; %เก็บค่าข้อมูลที่หาได้แต่ละชุดข้อมูล
```

5) โปรแกรมที่ใช้หาความผิดพลาดในแต่ละรอบ

```
%ข้อมูลที่เก็บได้ของแต่ละชุดข้อมูล
data11; data12; data13; data14;
data21; data22; data23; data25;
data31; data32; data34; data35;
data41; data43; data44; data45;
data52; data53; data54; data55;

%คำนวณหาค่าเฉลี่ยความผิดพลาดจากข้อมูลที่ใช้ทดสอบในแต่ละรอบ
MSE_av1 = (data11+data12+data13+data14)/4;
MSE_av2 = (data21+data22+data23+data25)/4;
MSE_av3 = (data31+data32+data34+data35)/4;
MSE_av4 = (data41+data43+data44+data45)/4;
MSE_av5 = (data52+data53+data54+data55)/4;

%คำนวนหาค่าเฉลี่ยความผิดพลาดทั้ง 5 รอบ
```

```
ae=(MSE_av1+MSE_av2+MSE_av3+MSE_av4+MSE_av5)/5;

plot(MSE_av1,'b') % กราฟของค่าความผิดพลาดในรอบที่ 1
hold on;

plot(MSE_av2,'r') % กราฟของค่าความผิดพลาดในรอบที่ 2

plot(MSE_av3,'k') % กราฟของค่าความผิดพลาดในรอบที่ 3

plot(MSE_av4,'c') % กราฟของค่าความผิดพลาดในรอบที่ 4

plot(MSE_av5,'g') % กราฟของค่าความผิดพลาดในรอบที่ 5
figure
plot(ae,'--')%กราฟของค่าความเฉลี่ยความผิดพลาด
```