國立中山大學 電機工程學系

碩士論文

多核心支援迴歸向量機應用於股價預測

A Multiple-Kernel Support Vector Regression Approach for Stock Market Price Forecasting

研究生：黃祺偉 撰

指導教授：李錫智 教授

中華民國 九十八 年 七 月 二十二 日

A Multiple-Kernel Support Vector Regression Approach for Stock Market Price Forecasting

Directed by: Professor Shie-Jue Lee

Graduate Student: Chi-Wei Huang

Department of Electrical Engineering

National Sun Yat-Sen University

Kaohsiung, Taiwan 804, R.O.C.

國立中山大學研究生學位論文審定書

本校電機工程學系碩士班

研究生：黃祺偉 (學號：M953010054) 所提論文

多核心支援迴歸向量機應用於股價預測

經本委員會審查並舉行口試，符合碩士學位論文標準。

學位考試委員簽章：

(召集人)

指導教授

系主任

A Multiple-Kernel Support Vector Regression Approach for Stock Market Price Forecasting

by

Chi-Wei Huang

A Thesis Submitted to the Graduate Division in Partial Fulfillment of the Requirements for the Degree of Master of Science

Department of Electrical Engineering , National Sun Yat-Sen University
Kaohsiung, Taiwan, Republic of China

(July 22, 2009)

Tzung-Pei Hong _____

_____ _____

_____ _____

_____ _____

Advisor : _____

Department Chairman : Ying-Chung Chen

致謝

光陰荏苒，日月如梭，轉眼間已將論文完成，期間的歡笑與汗水深深烙印在我心中。由於非本科系學生，一路走來有著許多困難，感謝許多人的協助與關心，讓我能順利的走完這漫長的道路。

首先要感謝我的指導教授 李錫智教授，感謝恩師敦敦教誨，其淵博的知識更令我在專業領域與為人處事上受益良多，令懵懵懂懂的我從摸索學習中獲益匪淺，更提供了良好的軟硬體設備，使得本論文得以順利完成。亦感謝恩師訓練我的研究能力，使我們研究團隊順利發表多篇期刊論文，並有一篇已被 IJICIC 所接受，這些機會是我學習路途上難得的經驗，學生將謹記在心，並致上最深的謝意。另外感謝口試委員 林文揚教授、 郭忠民教授、 洪宗貝教授、 謝朝和教授在百忙之中抽空參與，對於我的論文內容悉心的指導並提供許多寶貴的建議，使得本論文能更臻完備。

此外博士班學長葉吉原是我重要的貴人，他認真踏實的研究態度是我學習的最好榜樣，他總是不厭其煩的細心指導，帶領著我一步一步的探究學術領域。當我遇到許多困難、徬徨與無助時，給我許多有益的建議，有如此的學長帶領，令我在研究過程中受益良多，在此獻上最由衷的謝意。並感謝溫千慧學姐在我碩一剛進研究室時，細心的引領我踏入研究的道路，使我能夠專心於學術之上。亦感謝李婉瑞學姐，在我研究初期提供想法與建議，令我研究之路有了明確的方向。也感謝劉益賢學長能夠在帶領我接觸以往不熟悉的硬體領域，使我在研究道路上增添許多想法。最後謝謝博士班 志峰、忠益、永申、佳諺學長，已畢業學長 致瑩、經文、善成、開泰的關心與鼓勵;並謝謝士賢、仁嘉、杉榮、文彬、世達等同窗好友及學弟妹 文豪、連旺、憲奇、歐陽正、書嫻、旻宗、子典、耀瓏在研究的路上陪伴著我，讓我度過難忘豐收的碩士生涯。

最後，要感謝我的父母及關心我的人，在我身旁默默的支持，因為你們的關懷與體諒，是我堅持的最大力量，謝謝你們的包容與關心，使我順利的完成碩士學業，謝謝。

感謝大家!

黃祺偉 謹於

國立中山大學

中華民國 2009 年 夏

摘要

近年來，支援迴歸向量機已成功地被用來解決證券市場預測的問題。然而，支援迴歸向量機需要手動的調整核心函數的超參數。因此有學者提出多核心學習法來解決這類的問題，其中核心矩陣的權重向量與拉格朗日乘數常使用半正定規劃法來同時解得。但這類的演算法需要很大量的時間與空間，因此本論文提出一種結合最小序列優化法與梯度投影法的兩階段多核心學習演算法。

根據本演算法，使用者可以合併多個不同的超參數而使整個系統預測效果得到改善並且不需事先指定超參數的設置，更避免了過去需反覆實驗才可得到適合的超參數。本論文使用台灣加權指數加以實證，實驗結果顯示本方法效果優於其它的方法。

關鍵詞:證券市場預測、支援迴歸向量機、多核心學習、最小序列優化法、梯度投影法。

Abstract

Support vector regression has been applied to stock market forecasting problems. However, it is usually needed to tune manually the hyperparameters of the kernel functions. Multiple-kernel learning was developed to deal with this problem, by which the kernel matrix weights and Lagrange multipliers can be simultaneously derived through semidefinite programming. However, the amount of time and space required is very demanding. We develop a two-stage multiple-kernel learning algorithm by incorporating sequential minimal optimization and the gradient projection method.

By this algorithm, advantages from different hyperparameter settings can be combined and overall system performance can be improved. Besides, the user need not specify the hyperparameter settings in advance, and trial-and-error for determining appropriate hyperparameter settings can then be avoided. Experimental results, obtained by running on datasets taken from Taiwan Capitalization Weighted Stock Index, show that our method performs better than other methods.

Keywords: Stock market forecasting, support vector regression, multiple-kernel learning, SMO, gradient projection

目錄

摘要 i

Abstract ii

目錄 iii

圖目錄 iv

表目錄 v

第一章 導論 1

1.1 研究動機 1
1.2 研究目的 4
1.3 論文架構 5

第二章 基礎理論 7

2.1 支援向量機器 7
2.1.1 線性分割 7
2.1.2 硬性邊界支援向量機 9
2.1.3 軟性邊界支援向量機 13
2.1.4 以核心運算為基礎的支援向量機 17
2.2 支援迴歸向量機 24

第三章 研究方法 27

3.1 多核心支援迴歸向量機 27
3.2 兩階段多核心學習 29

第四章 實驗結果 33

4.1 實驗一 SKSVR 與 MKSVR 比較 33
4.2 實驗二 ARIMA、SKSVR 與 MKSVR 比較 37
4.3 實驗三 FNN、SKSVR 與 MKSVR 比較 42

第五章 結論與未來研究方向 46

5.1 結論 46
5.2 未來研究方向 46

參考文獻 47

圖目錄

圖 2.1 在二維空間中資料線性分類成兩類.....8
圖 2.2 有許多超平面可將資料分成兩類.....8
圖 2.3 引入邊界概念.....9
圖 2.4 硬性邊界支援向量機示意圖.....10
圖 2.5 引入鬆弛變數之軟性邊界支援向量機.....14
圖 2.6 特徵映射.....19
圖 3.1 兩階段多核心學習演算法.....29
圖 3.2 梯度投影法應用於兩階段多核心學習演算法.....32
圖 4.1 實驗一中不同超參數時 SKSVR 的預測效果.....35
圖 4.2 資料處理示意圖.....38
圖 4.3 實驗二中 ARIMA 於不同參數時的預測效果.....40
圖 4.4 實驗二中不同超參數時 SKSVR 的預測效果.....40
圖 4.5 實驗二 MKSVR 所預測的結果.....41
圖 4.6 實驗三 FNN 使用不同數量的隱藏節點之預測效果.....43
圖 4.7 實驗三不同超參數時 SKSVR 的預測效果.....43
圖 4.8 實驗三 MKSVR 所預測的結果.....45

表目錄

表 2.1 原始空間映射至特徵空間.....18

表 4.1 實驗一的資料區間.....34

表 4.2 SKSVR 與 MKSVR 在 RBF 核心於實驗一的結果.....35

表 4.3 SKSVR 與 MKSVR 在不同核心於實驗一的結果 36

表 4.4 實驗二、實驗三的資料區間.....37

表 4.5 ARIMA、SKSVR 與 MKSVR 於 RBF 核心在實驗二的效果.....40

表 4.6 SKSVR 與 MKSVR 在不同核心於實驗二的結果 41

表 4.7 FNN、SKSVR 與 MKSVR 在 RBF 核心於實驗三的效果.....44

表 4.8 SKSVR 與 MKSVR 在不同核心於實驗三的結果 44

第一章　導論

1.1 研究動機

近年世界經濟蓬勃發展，金融市場發展出許多投資理財的管道供消費者使用，各種金融產品也不斷的推陳出新，包括股票、共同基金、債券、期貨、選擇權等。眼見我國經濟發展也逐漸國際化和自由化，加上國民所得提高，使得投資理財一時蔚為風氣，儲蓄資金養老的觀念也漸漸被金融投資活動所影響，造成今日我國金融市場十分活躍，其中股票市場更是活絡，甚至成為一般民眾對於經濟景氣或投資的參考指標。根據台灣行政院金融監督管理委員會證卷期貨局[43]統計，至 98 年 5 月底止，上櫃公司已達 545 家，市值 13,763.9 億，上市公司更是達到 725 家， 市值高達 173,488.5 億，其中 97 年全年交易人數更有 3,032,342 人，相對於全國總人口數 22,934,997 的比例超過 13.2%，可以顯見股票市場在台灣儼然成為一種主要的投資工具。而一般民眾憑藉著在網路、電視、報章雜誌上所提供的許多資料做為投資參考，但面對如此龐大而混雜的資料，如何擷取出對投資有利的資訊便成為一種難題。因此這些年來，國內外學者對於股票價格及其報酬率之研究也多如牛毛，概括來說分為基本分析及技術分析兩大類。

基本分析是利用財務分析和經濟學上的研究來對企業價值評估或預測證券價值的走勢。這些被分析的基本資料大多為股票市場以外的資訊，可以包含一家公司所發布的財務報表和非財務的資訊，如商品需求成長性的預測、企業之間的比較，或是政策上新制度的影響及人口的改變等等的因素。投資者一般可使用基本分析來探討公司的財務狀況、營運狀態和經營方向，來了解公司的穩定性和未來潛力。探討的項目可能包括有股息的發放、公司資金的管理方法、公司債務分配和公司營利成長。投資者可利用基本分析的結果來決定要多頭或空頭策略。它通常和所謂的技術分析相對。

技術分析的專家認為歷史是不斷的重複，景氣也會不斷的循環，所以在研究證券價值的趨勢時，並不會到市場本身以外的因素來做預測，通常會用到證券市場的成交價格、成交量、價和量的變化以及產生這些變化經歷的時間等市場行為做為預測的分析基礎。所以技術分析的學者專家開始發展出許多工具來分析股票市場，如 K 線、移動平均指標、MACD、OBV、隨機指標等技術指標。

此外，數理經濟學及數理統計學的學者，也發展出許多模型來預測證券市場

的價值，像自迴歸模型(autoregressive, AR)[8]、自迴歸移動平均模型(autoregressive moving average, ARMA)[5]、自迴歸整合移動平均模型(autoregressive integrated moving average, ARIMA)[5]等時間序列工具，這些模型都是線性的。由於股票市場原本就是不斷演化而且呈現不規則變動的複雜系統，所含的雜訊及不穩定因素，會造成線性預測效果不佳。也因此近來的學者提出了許多非線性的預測工具，如自回歸條件異方差模型(autoregressive conditional heteroskedasticity, ARCH)[14]模型、一般化自回歸條件異方差模型(generalized autoregressive conditional heteroskedasticity, GARCH)[4] 模型等。

近年來資料探勘(data mining)的技術發展迅速常被應用於各領域來協助分析大量的資料，主要是希望能夠透過資料探勘從歷史中挖掘出富含價值的資料。此目標與技術分析專家的想法不謀而合，於是開始有學者將資料探勘的技術應用於證券市場之中，像是人工類神經網路(artificial neural networks, ANN)[18] [21] [23] [29] [41]、模糊類神經網路(fuzzy neural networks, FNN)[9] [25] [40]、支援向量機器(support vector machine, SVM)[10] [16] [34]、支援迴歸向量機(support vector regression, SVR)[6] [7] [10] [15] [16] [20] [27] [34] [36] [39]等。其中人工類神經網路(artificial neural networks, ANN)因發展的早，最廣為應用於證券市場這類的時間序列問題[20]，根據以往的研究結果來看，ANN 所建構的模型主要是經驗風險最小化(empirical risk minimization, ERM)，所以預測能力會優於傳統的統計模型[12] [37]。然而 ANN 存在的很複雜的訓練因素，很容易陷入區域的最佳解，且相關的參數缺少系統化的準則及隱藏層的數目多寡不定難以分析。

在 90 年代貝爾實驗室 Vapnik 博士，以統計及機器學習理論提出支援向量機器理論[10]，開始廣泛的應用於分類、分群、迴歸等領域之中。Vapnik 博士更提出了支援迴歸向量機[10]，SVR 不只擁有 ANN 的優點，更有許多不錯的特性[10] [12]:(1)使用結構風險最小化(Structural Risk Minimization ,SRM)學習理論，相較於類神經網路建構在經驗風險最小化學習理論，訓練良好的 SVR 對於複雜的迴歸方程式提供了更加的描述能力。(2)迴歸模型建立時的稀疏性質，及能自動控制模型複雜度的機制，讓其對於雜訊有良好的適應力。(3)訓練 SVM 的過程等價於解一個線性且有限制式的的二次規劃問題(quadratic programming, QP)，這表示 SVM 問題是一個凸規劃問題(convex problem)，它的解是唯一的且全域最佳解，解決了 ANN 會陷入區域最佳解的問題。也因此，SVR 往往表現出更佳的預測能

力。使用 SVR 需要事先決定核心函數(kernel function)的類型以及 SVR 要用的超參數(hyperparameters)。顯然，當選擇到不佳的核心函數或超參數將導致低落的預測效果[10] [13] [22]。大多數的研究人員都使嘗試錯誤法不斷的摸索而選擇出適當的數值，但這樣明顯地會花費大量的時間。此外使用單核心函數可能會無法完整解釋高複雜度的問題導致得不到讓人滿意的結果，尤其是證卷市場這一類的問題。

有學者採用了多核心來解決這類的問題[1] [2] [11] [17] [24] [26] [30] [31] [32] [33] [35] [38]。最簡單的想法便是結合多個核心然後將他們取平均。但這樣每個核心都擁有相同的權重，對於決策過程其實並不是很適當。因此產生了一個重要的議題，就是當結合多核心時要如何去決定每一個核心最佳的權重參數。Lanckriet et al.使用了一種線性矩陣結合方法來合併多個核心[24]。他們將這個最佳化問題轉換成半正定規劃(semidefinite programming, SDP)，如此一來問題就變成凸函數的形態，能夠找到全域的最佳解。此外，Bach et al. [1], Sonnenburg et al.[32], Rakotomamonjy et al.[30] [31], Szafranski et al.[33], and Gönen et al.[17]等人也提出了其他的多核心學習演算法。這些方法針對解決大型分類問題提出了重複執行最小序列優化法(sequential minimal optimization, SMO)演算法[28]來依次更新拉格朗乘數跟核心的權重，換言之就是先固定核心函數權重來更新拉格朗日乘數(Lagrange multipliers)，然後再依據固定的拉格朗日乘數更新核心函數權重，如此反覆的迭代。雖然這樣的演算法相較於 SDP 快速很多，但它們很容易得到區域最佳解。於是基於 hyperkernels 的多核心演算法紛紛被提出來[26] [35]。像 Tsang 和 Kwok[35]便將這個問題轉化成 second-order cone programming(SOCP)的方法表示。而 Crammer et al.[11]和 Bennett et al.[2]則使用 boosting 方法來結合異質的(heterogeneous)核心矩陣。

本論文提出了一種結合多核心學習跟 SVR 的迴歸模型來處理證券市場預測的問題。本論文發展出兩階段多核心學習演算法使 SVR 多核心矩陣合併能夠最佳化。此學習演算法重複地應用 SMO 演算法[28]及梯度投影法(gradient projection)[3]，來得到拉格朗日乘數與最佳的核心權重。藉由此演算法，有助於結合不同的超參數設定且大體上預測系統的效果都可以被改善。此外，使用者不需要預先明確的設定超參數而且可避免大量的應用嘗試錯誤法來決定超參數設定。

1.2 研究目的

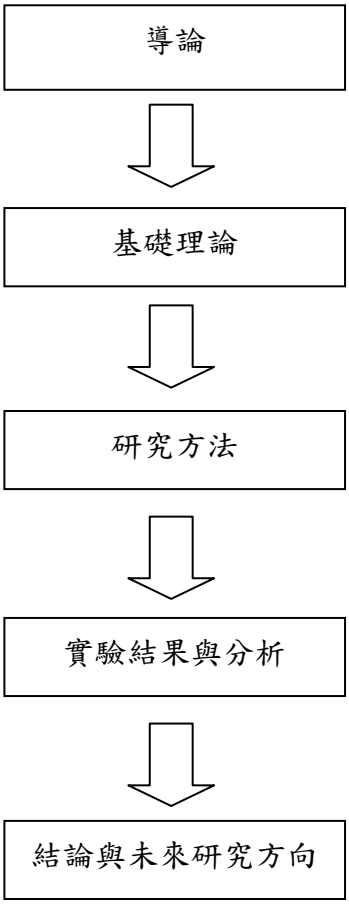
本研究使用兩階段多核心學習演算法結合 SVR，而研究資料為臺灣證券交易所發行量加權股價指數（簡稱加權股價指數、加權指數、TAIEX）其由臺灣證券交易所[42]依照上市公司之股價所編製的股價指數，是台灣最為人熟悉的股票指數，被視為是呈現台灣經濟走向的指標。因此本研究目的有二:

1.與原始單核心 SVR 模型比較，證實本方法可有效結合不同的超參數，並找到較佳的結果。

2.跟 ARIMA、FNN 不同之模型比較實驗效果，以證實本方法能有較佳的預測能力。

1.3 論文架構

在接下來的文章中將會更完整的介紹本研究所提出的方法，本論文的架構圖如下:



一、導論:

第一章，介紹研究動機，分析以往學者的研究方法，並引入本論文所要解決的問題且介紹實驗目的以及實驗架構。

二、基礎理論:

第二章，將介紹單核心支援向量機、支援迴歸向量機，以上都是本研究會用到的基本方法跟概念。

三、研究方法:

第三章，介紹本論文的研究動機以及方法，清楚說明本演算法法對於其他方法的優點及不同之處。

四、實驗結果與分析

第四章，將本論文提出的演算法與參考文獻中的方法進行比較，以驗證是否符合在第三章的看法，並檢視有無達到實驗的目的。

五、結論與未來研究方向:

第五章,把本論文做總結,說明本論文的貢獻以及未來需要繼續改進的方向。

第二章　基礎理論

本論文主要的方法屬於支援迴歸向量機，所要應用的問題為臺灣證券交易所發行量加權股價指數預測，主要將改良 SVR 核心矩陣的配置方法。茲將 SVM 與 SVR 的主要想法介紹如下:

2.1 支援向量機器

本節將簡介支援向量機的相關原理[24]，首先介紹線性分類的支援向量機，包含使用硬性邊界、軟性邊界的支援向量機，接下來介紹支援向量機器如何應用在非線性的問題上面，並介紹其主要精神。

2.1.1 線性分割

在兩分類的分類問題當中，資料的輸入可表示成向量的型態 $\mathbf{x} = \left[x_1, x_2, x_3..., x_n\right]^{\mathrm{T}} \in \mathbb{R}^n$，而資料的輸出 $y = \{+1, -1\}$。原始 SVM 的想法，就是想在有限維度 $\mathbb{R}^n$ 中找一個超平面(hyperplane)來對資料做線性分割，一個超平面 $H_{w,b}$ 可以如此表示:

$$\begin{aligned} &w_1x_1 + w_2x_2 + w_3x_3 + \ldots + w_nx_n + b = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0 \\ &w_i, x_i, b \in \mathbb{R}, \\ &\mathbf{x} = \left[x_1, x_2, x_3..., x_n\right]^{\mathrm{T}}, \mathbf{w} = \left[w_1, w_2, w_3..., w_n\right]^{\mathrm{T}} \in \mathbb{R}^n \end{aligned} \tag{2.1.1}$$

其中，$\mathbf{w}$ 為權重向量(vector of weight)，b 稱為偏移量(bias)。

此後利用這個超平面作為分類的依據，並藉由此超平面可以對資料分成正(positive)類別以及負(negative)類別(如圖 2.1)，因此我們可以由訓練資料(training data)得到超平面而訂出決策函數(decision function)(或稱分類器)，函式如下:

$$\begin{aligned} &f_{w,b}(\mathbf{w}) = w_1x_1 + w_2x_2 + w_3x_3 + \ldots + w_nx_n + b = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b \\ &w_i, x_i, b \in \mathbb{R}, \\ &\mathbf{x} = \left[x_1, x_2, x_3..., x_n\right]^{\mathrm{T}}, \mathbf{w} = \left[w_1, w_2, w_3..., w_n\right]^{\mathrm{T}} \in \mathbb{R}^n \end{aligned} \tag{2.1.2}$$

當令輸入資料 $\mathbf{x} = \left[x_1, x_2, x_3..., x_n\right]^{\mathrm{T}} \in \mathbb{R}^n$，而資料分類為 $y \in \{-1, +1\}$，此時若有一組訓練資料 $\mathbf{S} = \{(\mathbf{x}_i, y_i), i = 1...n\}$，存在一超平面 $H_{w,b}$ 可將這組訓練資料正確的分類，即將此組資料稱為線性可分割(linear separable)，因此當超平面 $H_{w,b}$ 被正確

計算出來後，將訓練資料帶入決策函數，若得到 $f_{w,b}(\mathbf{x}) > 0$ 時，將資料分為正類別亦即 $y_i = +1$，當 $f_{w,b}(\mathbf{x}) < 0$ 時，將資料列為負類別即 $y_i = -1$，如此一來就可將資料分成兩個類別。如圖 2.1 為二維平面，中間實線為一超平面 $H_{w,b}$，超平面將資料線性分割成兩個類別，其中上方為正類別，下方為負類別，**w** 為是超平面的法向量，b 是超平面的平行偏移量。因此，給定一組訓練資料 **S**，存在著許多個超平面可以將資料正確的分類，如圖 2.2。

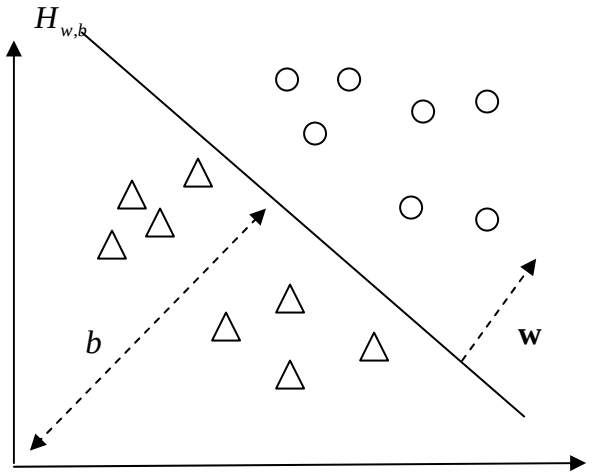


圖 2.1 在二維空間中資料線性分類成兩類

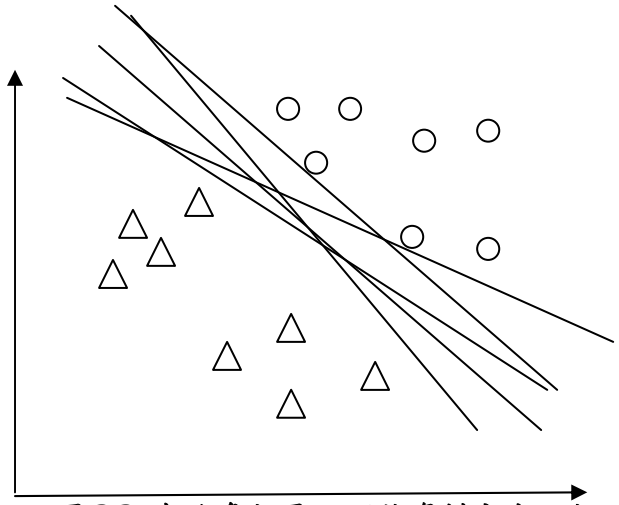
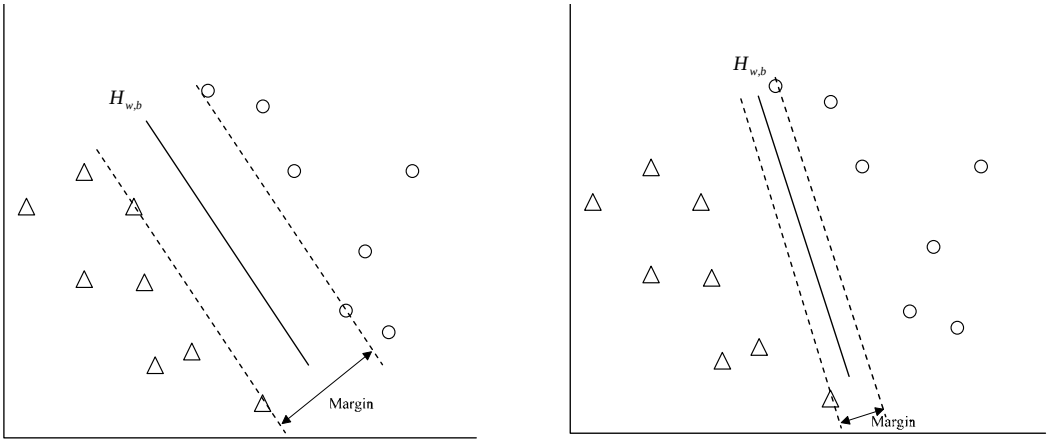


圖 2.2 有許多超平面可將資料分成兩類

2.1.2 硬性邊界支援向量機

給定一組訓練資料 **S**，假設存在兩個超平面可以將資料正確地分為兩類，如圖 2.3(a)(b)所示，其中 2.3(a)擁有較大的邊界(margin)，而 2.3(b)有擁有較小邊界，SVM 提出有最大硬性邊界(hard margin)的超平面為最佳分類器，如圖 2.3(a)，此超平面將擁有較佳的泛化能力(generalization ability)。



(a) (b)

圖 2.3 引入邊界概念

給定輸入訓練資料 $\mathbf{S}=\{(\mathbf{x}_i,\mathbf{y}_i), i=1...n\}$，其中 $\mathbf{x}=\{x_1,x_2,x_3...,x_n\}^{\mathrm{T}}\in\mathbb{R}^n$，$y\in\{-1,+1\}$，如圖 2.4，當定義 $H_{w,b}:\mathbf{w}^{\mathrm{T}}\mathbf{x}+b=0$，**w** 為是超平面的法向量，$b$ 是超平面的平行偏移量時，對於 $i=1,...,n$ 可以得到:

$$\mathbf{w}^{\mathrm{T}}\mathbf{x}_i+b\begin{cases}>0 & \text{for } y_i=1,\\ <0 & \text{for } y_i=-1,\end{cases} \tag{2.1.3}$$

因為是線性可分割，所以不會有訓練資料滿足 $\mathbf{w}^{\mathrm{T}}\mathbf{x}+b=0$。而為了控制最佳的分割的位置，對於(2.1.3)式改寫成以下的不等式:

$$\mathbf{w}^{\mathrm{T}}\mathbf{x}_i+b\begin{cases}\geq 1 & \text{for } y_i=1,\\ \leq -1 & \text{for } y_i=-1,\end{cases} \tag{2.1.4}$$

在這裡，不等式右邊的 1 與-1，可以改寫成常數 a(>0)和-a。而藉由運算(2.1.4)式，可以得到一個等價的方程式如下:

$$\mathrm{y}_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i+b)\geq 1 \text{ for } i=1,...,n \tag{2.1.5}$$

超平面函數可設為:

$$f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}+b = c \text{ for } -1 < c < 1 \tag{2.1.6}$$

就可形成一個可線性分割的超平面來分割訓練資料$\mathbf{x}_i(i = 1,...,n)$。當 $c = 0$ 時，指的是 $c = 1$ 及 $c = -1$ 的兩個超平面 H_1, H_2 中間的超平面 $H_{w,b}$(如圖 2.4)。

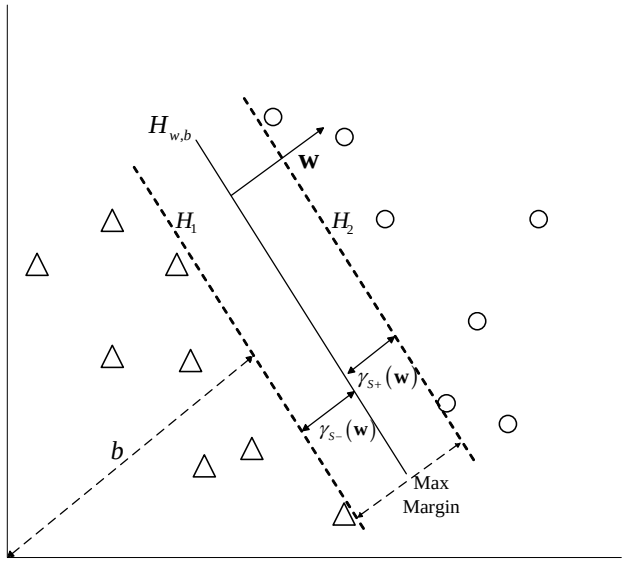


圖 2.4 硬性邊界支援向量機示意圖

藉由此步驟，就可以在無限多個可分割的超平面中，找到一個使得 H_1, H_2 之間距離最大的超平面，即是最佳的超平面(optimal separating hyperplane)(如圖 2.4)，而 H_1, H_2 稱做函數邊界(functional margin)或支援邊界(support hyperplane)，而 H_1, H_2 上的點稱為支援向量(support vectors)。

現在考慮最佳的超平面如何決定，在歐幾里德距離(Euclidean distance)中，一筆訓練資料 **x** 到可線性分割的超平面，距離可寫成 $|f(\mathbf{x})|/\|\mathbf{w}\|$。而 H_1, H_2 可以寫成

$$\begin{aligned} \langle \mathbf{w}\cdot\mathbf{x}_i^{+}\rangle + b &= +1 (\text{正類別的邊界}) \\ \langle \mathbf{w}\cdot\mathbf{x}_i^{-}\rangle + b &= -1 (\text{負類別的邊界}) \end{aligned} \tag{2.1.7}$$

這兩邊界到超平面的幾何距離(geometric margin)可表示如下

$$\begin{aligned} \gamma_{S+}(\mathbf{w}) &= \frac{1}{2}\left(\left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|_2}\cdot\mathbf{x}^{+} \right\rangle - \left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|_2}\cdot\mathbf{x}^{-} \right\rangle \right) \\ &= \frac{1}{2\|\mathbf{w}\|_2}\left(\langle \mathbf{w}\cdot\mathbf{x}^{+}\rangle - \langle \mathbf{w}\cdot\mathbf{x}^{-}\rangle \right) = \frac{1}{\|\mathbf{w}\|_2} = \gamma_{S-}(\mathbf{w}) \end{aligned} \tag{2.1.8}$$

得到兩邊界的距離為 $\frac{2}{\|\mathbf{w}\|_2}$，而當邊界距離越大，就能提供更佳的泛化能力(generalization ability)。因此在這裡我們希望能夠讓距離盡可能的大，目標為 maximize $\frac{2}{\|\mathbf{w}\|_2}$，即等價於 minimize $\|\mathbf{w}\|_2$。加上希望能將所有的資料都能夠正確的分類，所以的使用(2.1.5)式當成限制式，因此得到硬性邊界支援向量機的最佳化問題為:

$$\begin{aligned} &\min_{\mathbf{w},b} \ \frac{1}{2}\|\mathbf{w}\|^2 \\ &s.t. \quad y_i\left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b\right) \geq 1, \ i = 1,...,n. \end{aligned} \tag{2.1.9}$$

上式稱為原始問題(primal promblem)，為求解上式，我們可以利用二次規劃方法(quadratic programming, QP)，但由於 $\mathbf{x}_i (i = 1, ..., n)$ 本身可能是高維度空間，甚至有時更會是無限多維度。此時我們利用拉格朗日法(Lagrange)，如此一來就可將具有限制式的最佳化問題(constrained optimization problem)轉化成不具限制式的最佳化問題(unconstrained optimization problem)，因此引入拉格朗日乘數(Lagrange multipliers) 得到:

$$\begin{aligned} &L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} - \sum_{i=1}^{n}\alpha_i\left[y_i\left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b\right) - 1\right] \\ &\text{where} \quad \alpha_i \geq 0, \ i = 1...n. \end{aligned} \tag{2.1.10}$$

其中 α_i 為拉格朗日乘數。於是，原始的問題可改寫成:

$$\min_{\mathbf{w},b}\max_{\boldsymbol{\alpha}} L(\mathbf{w}, b, \boldsymbol{\alpha}) \tag{2.1.11}$$

為求解(2.1.11)式極值問題，將 $L(\mathbf{w}, b, \alpha)$ 分別對 $\mathbf{w}, b$ 偏微分，得到:

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i = 0, \Rightarrow \mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i. \tag{2.1.12}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} = -\sum_{i=1}^{n}\alpha_i y_i = 0, \Rightarrow \sum_{i=1}^{n}\alpha_i y_i = 0. \tag{2.1.13}$$

將(2.1.12)與(2.1.13)式代回原來的 $L(\mathbf{w}, b, \alpha)$，得到:

$$\begin{aligned} L(\mathbf{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle - \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle + \sum_{i=1}^{n} y_i \alpha_i b + \sum_{i=1}^{n}\alpha_i \\ &= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle \end{aligned} \tag{2.1.14}$$

因此可得到對偶問題(dual problem):

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle \\ &s.t \quad \sum_{i=1}^{n} y_i \alpha_i = 0,\ \alpha_i \geq 0,\ i = 1,...,n. \end{aligned} \tag{2.1.15}$$

由以上可以得知 SVM 問題是一個標準的凸規劃問題，目標函數可依據訓練資料的內積來解決最大化問題。

假設 α^* 為 (2.1.15) 式對偶問題的解，則此最佳化問題將滿足 Karush-Kuhn-Tucker(KKT)條件[10]，推導出 KKT 條件如下:

$$\begin{aligned} &\frac{\partial L(\mathbf{w}^*, b^*, \boldsymbol{\alpha}^*)}{\partial \mathbf{w}^*} = 0 \Rightarrow \mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \boldsymbol{x}_i, \\ &\frac{\partial L(\mathbf{w}^*, b^*, \boldsymbol{\alpha}^*)}{\partial b} = 0, \Rightarrow \sum_{i=1}^{n} \alpha_i^* y_i = 0, \\ &\text{Constraint}: \ y_i \left(\left\langle \mathbf{w}^*, \mathbf{x}_i \right\rangle + b^* \right) \geq 1, \\ &\text{Multiplier Condition}: \ \alpha_i^* \geq 0, \\ &\text{KKT 互補條件}: \ \alpha_i^* \left[y_i \left(\left\langle \mathbf{w}^*, \mathbf{x}_i \right\rangle + b^* \right) - 1 \right] = 0. \end{aligned} \tag{2.1.16}$$

定義 $I_{sv} := \{ i = 1,...,n : \alpha_i > 0 \}$ 令 $p \in I_{sv}$，因為 $\alpha_i^* \geq 0$，可以由(2.1.16)式中得到:

$$y_p \left(\left\langle \mathbf{w}^*, \mathbf{x}_p \right\rangle + b^* \right) = 1 \tag{2.1.17}$$

由於 $y_p = \pm 1$，且 $y_p^2 = 1$，因此上式左右同乘 y_p 後可寫成:

$$\left\langle \mathbf{w}^*, \mathbf{x}_p \right\rangle + b^* = y_p \tag{2.1.18}$$

經過整理可以得到 b^*:

$$b^* = y_p - \left\langle \mathbf{w}^*, \mathbf{x}_p \right\rangle = y_p - \left\langle \sum_{i \in I_{sv}} \alpha_i^* y_i \mathbf{x}_i, \mathbf{x}_p \right\rangle = y_p - \sum_{i \in I_{sv}} \alpha_i^* y_i \left\langle \mathbf{x}_i, \mathbf{x}_p \right\rangle \tag{2.1.19}$$

於是得到最佳的決策函數:

$$f(\mathbf{x}) = \left\langle \mathbf{w}^*, \mathbf{x} \right\rangle + b^* = \sum_{i \in I_{sv}} \alpha_i^* y_i \left\langle \mathbf{x}_i, \mathbf{x} \right\rangle + b^* \tag{2.1.20}$$

2.1.3 軟性邊界支援向量機

當訓練資料 **S** 常常維度很高或是包含著許多雜訊，因此資料在輸入空間(input space)$\mathbb{R}^n$很容易為不可線性分割(not linearly separable)，導致硬性邊界支援向量機可能很難以收斂，於是希望在原本的空間裡使用線性分類的方法處理時，允許有一定程度的錯誤容許度，並給予適當的懲罰值。如此可避免當資料隱含著雜訊時，訓練資料分類的很正確，但預測效果偏差很多。因此軟性邊界支援向量機(soft margin support vector machine)引入鬆弛變數(slack variable)，如此就變成軟性邊界分類器(soft margin classifier)，可以擁有較高的抗雜訊能力。

當給予一個分類問題，有 n 筆訓練資料$(\mathbf{x}_1, y_1),(\mathbf{x}_2, y_2),(\mathbf{x}_3, y_3),...,(\mathbf{x}_n, y_n)$，其中$\mathbf{x}_i \in \mathbb{R}^n, y_i \in \{+1,-1\}, i=1,...,n$，因此得到軟性邊界支援向量機的最佳化問題如下(如圖 2.5):

$$\begin{aligned} \min_{w,b} \ & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ s.t. \ & y_i\left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b\right) \geq 1-\xi_i, \\ & \xi_i \geq 0, \ i=1,...,n. \end{aligned} \tag{2.1.21}$$

其中 **w** 為是權重向量，ξ_i 是鬆弛變數。若給定$\gamma > 0$，則資料點$(\mathbf{x}_i, y_i)$對於超平面$H_{w,b}$及區間距離γ的鬆弛變數ξ_i的可定義如下:

$$\xi_i = \max\left(0, \gamma - y_i \cdot \left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b\right)\right) \tag{2.1.22}$$

如此一來，便可讓原來的硬性邊界變成有彈性的軟性邊界，軟性邊界允許資料點落在分類不正確的一邊，但是會給予適當的懲罰值，而資料落在正確的一邊時則不予懲罰。當 C 設的很大時，資料點落在分類不正確的一邊會得到較大的懲罰值，即注重分類的正確性，相對的兩邊界距離會比較小。反之，若 C 值很小時，則允許資料分類錯誤的容忍度會較高，但相對的兩邊界距離會比較大。如此便可在求得最大距離區間與允許資料分類錯誤容忍度中做較佳的取捨(trade-off) 。

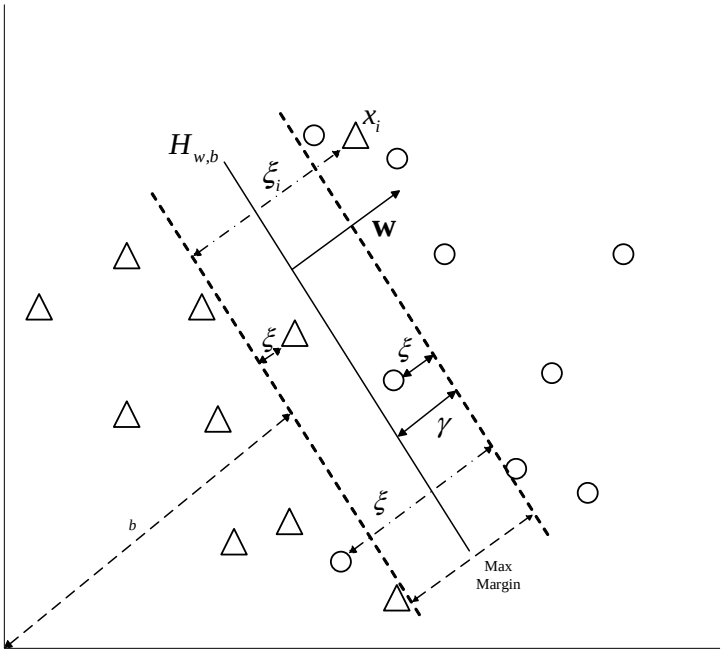


圖 2.5 引入鬆弛變數之軟性邊界支援向量機

(2.1.22)式經過拉格朗日法運算，可以得到:

$$L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})=\frac{1}{2}\mathbf{w}\cdot\mathbf{w}+C\sum_{i=1}^{n}\xi_i-\sum_{i=1}^{n}\alpha_i\left[y_i\left(\langle\mathbf{w},\mathbf{x}_i\rangle+b\right)-1+\xi_i\right]-\sum_{i=1}^{n}\beta_i\xi_i \tag{2.1.23}$$

其中 $\alpha_i,\beta_i\geq 0.i=1...n$

其原始問題可寫成:

$$\min_{\mathbf{w},b,\xi}\max_{\boldsymbol{\alpha},\boldsymbol{\beta}}L(\mathbf{w},b,\xi,\boldsymbol{\alpha},\boldsymbol{\beta}) \tag{2.1.24}$$

將使用拉格朗日法後的(2.1.23)式分別對原始變數$\boldsymbol{w},b,\xi$偏微分，得到:

$$\frac{\partial L}{\partial\mathbf{w}}=\mathbf{w}-\sum_{i=1}^{n}\alpha_i y_i\mathbf{x}_i=0,\Rightarrow\mathbf{w}=\sum_{i=1}^{n}\alpha_i y_i\mathbf{x}_i. \tag{2.1.25}$$

$$\frac{\partial L}{\partial b}=-\sum_{i=1}^{n}\alpha_i y_i=0,\Rightarrow\sum_{i=1}^{n}\alpha_i y_i=0. \tag{2.1.26}$$

$$\frac{\partial L}{\partial\xi_i}=C-\alpha_i-\beta_i=0,\Rightarrow\alpha_i=C-\beta_i. \tag{2.1.27}$$

再將(2.1.25)、(2.1.26)、(2.1.27)式代入(2.1.23)式取代原始變數$(\boldsymbol{w},b,\boldsymbol{\xi})$並將拉格朗日乘數$(\beta_i)$相消。運算如下:

$$
\begin{aligned}
L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}) &= \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle + \sum_{i=1}^{n} (C-\alpha_i-\beta_i)\xi_i \\
&\quad - \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle - \sum_{i=1}^{n} y_i \alpha_i b + \sum_{i=1}^{n} \alpha_i \\
&= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle
\end{aligned}
\tag{2.1.28}
$$

整理後得到

$$
\min_{w,b,\xi} L(\mathbf{w},b,\xi,\boldsymbol{\alpha},\boldsymbol{\beta}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle \tag{2.1.29}
$$

由(2.1.27)式，因 $\alpha,\beta \geq 0$ 且 $\alpha_i + \beta_i = C$ ，得到限制式 $C \geq \alpha_i \geq 0$ ，可推出對偶問題為:

$$
\begin{aligned}
&\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle \\
&s.t \quad \sum_{i=1}^{n} y_i \alpha_i = 0,\ C \geq \alpha_i \geq 0,\ i = 1,...,n.
\end{aligned}
\tag{2.1.30}
$$

由結果(2.1.30)式發現鬆弛變數 ξ_i 及拉格朗日乘數 (β_i) 都不復存在。反而是限制式 $C \geq \alpha_i \geq 0$ 的關係限制了落在正確邊界以外的點對結果的影響。

假設 α^* 為 (2.1.30) 式對偶問題的解，則此最佳化問題將滿足 Karush-Kuhn-Tucker(KKT)條件[10]，推導出 KKT 條件如下:

$$
\begin{aligned}
&\frac{\partial L}{\partial \mathbf{w}^*} = 0 \Rightarrow \mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x}_i, \\
&\frac{\partial L}{\partial b^*} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i^* y_i = 0 \\
&\frac{\partial L}{\partial \xi_i^*} = 0 \Rightarrow C - \alpha_i^* - \beta_i^* = 0 \\
&y_i\left(\left\langle \mathbf{w}^*, \mathbf{x}_i \right\rangle + b^*\right) - 1 + \xi_i^* \geq 0 \\
&\xi_i^* \geq 0 \\
&\text{Multiplier Condition : } \alpha_i^* \geq 0 \\
&\text{Multiplier Condition : } \beta_i^* \geq 0
\end{aligned}
\tag{2.1.31}
$$

KKT 互補條件：$\alpha_i^*\left[y_i\left(\left\langle \mathbf{w}^*,\mathbf{x}_i\right\rangle+b^*\right)-1+\xi_i^*\right]=0$

$$\mu_i\xi_i=0$$

整理 KKT 條件，假設$\xi_i^*>0$則$\beta_i^*=0$，由(2.1.27)式$\alpha_i^*=C-\beta_i^*=C$。若假設$C>\alpha_i^*$，那麼$\beta_i^*=C-\alpha_i^*>0$，則得到$\xi_i^*=0$。於是新的 KKT 條件可以寫成:

$$\begin{aligned} &y_i\left(\left\langle \mathbf{w}^*,\mathbf{x}_i\right\rangle+b^*\right)-1+\xi_i^*\geq 0\\ &\xi_i^*\geq 0\\ &\alpha_i^*\geq 0\\ &\left(C-\alpha_i^*\right)\xi_i^*=0 \end{aligned} \tag{2.1.32}$$

由(2.1.32)式可以得到$C\geq\alpha_i^*\geq 0$的點，它的$\xi_i^*=0$即表示被正確分類在邊界裡面或邊界上。定義$I_{sv}:=\{i=1,...,n:\alpha_i>0\}$可以得到

$$\mathbf{w}^*=\sum_{i=1}^{n}\alpha_i^*y_i\mathbf{x}_i=\sum_{i\in I_{sv}}\alpha_i^*y_i\mathbf{x}_i \tag{2.1.33}$$

令$p\in I_{sv}$，因為$C\geq\alpha_i^*\geq 0$，可以由(2.34)、(2.35)式中得到$\xi_i^*=0$及

$$y_p\left(\left\langle \mathbf{w}^*,\mathbf{x}_p\right\rangle+b^*\right)=1-\xi_i^*=1-0=1 \tag{2.1.34}$$

由於$y_p=\pm 1$，且$y_p^2=1$，因此上式左右同乘y_p後可寫成:

$$\left\langle \mathbf{w}^*,\mathbf{x}_p\right\rangle+b^*=y_p \tag{2.1.35}$$

經過整理可以得到b^*:

$$b^*=y_p-\left\langle \mathbf{w}^*,\mathbf{x}_p\right\rangle=y_p-\left\langle \sum_{i\in I_{sv}}\alpha_i^*y_i\mathbf{x}_i,\mathbf{x}_p\right\rangle=y_p-\sum_{i\in I_{sv}}\alpha_i^*y_i\left\langle \mathbf{x}_i,\mathbf{x}_p\right\rangle \tag{2.1.36}$$

於是得到最佳的決策函數:

$$f(\mathbf{x})=\left\langle \mathbf{w}^*,\mathbf{x}\right\rangle+b^*=\sum_{i\in I_{sv}}\alpha_i^*y_i\left\langle \mathbf{x}_i,\mathbf{x}\right\rangle+b^* \tag{2.1.37}$$

2.1.4 以核心運算為基礎的支援向量機

以上所討論方法，資料還是能以近乎線性的分類方法來解決，然而遇到資料太複雜，使用線性分割無法分類時，可以透過特徵映射的方法，將資料投射到更高維度的特徵空間(feature space)，如此我們就可以在高維度的空間中，把資料線性的分類。

支援向量機是一種以核心運算為基礎的演算法(kernel-based algorithm)，使用核心函數來處理資料不但容易能以線性的方法處理非線性的分類問題，更可以避免實際坐標做映射運算時的大量複雜運算，這個技巧被稱為 kernel trick，如此便可透過統計學理論使用前面介紹的線性支援向量機在特徵空間中尋找把訓練資料分類正確的最佳超平面。

一般的狀況下，會先找到一個適當的映射函數，然後將資料一筆筆的映射到特徵空間當中，再對資料做分類的處理。在這裡定義 ϕ 為特徵映射，運作如下

$$\mathbf{x} = (x_1, x_2, x_3..., x_n) \Rightarrow \phi(\mathbf{x}) = (\phi(x_1), \phi(x_2), \phi(x_3)..., \phi(x_n)). \tag{2.1.38}$$

如此的轉換，便可將資料從輸入空間 X(input space)映射到新的特徵空間 F 當中。在這裡舉一個例子，假設有資料 $\mathbf{x} = (x_1, x_2)$, $\mathbf{z} = (z_1, z_2)$ 令特徵映射 $\phi: \mathbb{R}^2 \Rightarrow \mathbb{R}^3$

$$\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2), \phi(\mathbf{z}) = (z_1^2, z_2^2, \sqrt{2}z_1z_2) \tag{2.1.39}$$

$$\begin{aligned} \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle &= \langle (x_1^2, x_2^2, \sqrt{2}x_1x_2), (z_1^2, z_2^2, \sqrt{2}z_1z_2) \rangle \\ &= x_1^2z_1^2 + x_2^2z_2^2 + 2x_1x_2z_1z_2 \end{aligned} \tag{2.1.40}$$

此時令多項式函數 $k(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle^2$ ，可以得到

$$\begin{aligned} K(\mathbf{x}, \mathbf{z}) &= \langle \mathbf{x}, \mathbf{z} \rangle^2 \\ &= \langle (x_1, x_2), (z_1, z_2) \rangle^2 \\ &= (x_1z_1 + x_2z_2)^2 \\ &= x_1^2z_1^2 + x_2^2z_2^2 + 2x_1x_2z_1z_2 \end{aligned} \tag{2.1.41}$$

觀察(2.1.40)及(2.1.41)式可以發現，兩式的結果是一樣的。於是可以得到以下定

義，令 $F=\{\phi(\mathbf{x})\,|\,\mathbf{x}\in X\}$ 為特徵空間，擁有一個實數內積空間且 $X\in\mathbb{R}^n$ ，而核心(kernel)是一個投射到 $X\times X$ 的實數函數，使得

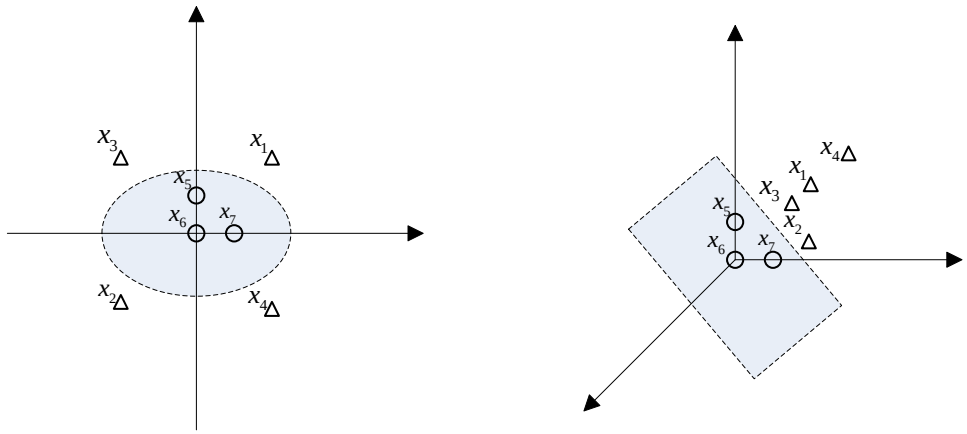
$$K(\mathbf{x},\mathbf{z})=\langle\phi(\mathbf{x}),\phi(\mathbf{z})\rangle,\ \mathbf{x},\mathbf{z}\in X \tag{2.1.42}$$

如此只要把資料點映射到特徵空間時的內積運算用核心函數取代，就可以用線性的支援向量機處理非線性的問題且不需大量的計算。

在此舉一個例子說明，若存在七個點 $x_1=(2,2)$、$x_2=(-2,-2)$、$x_3=(2,-2)$、$x_4=(-2,2)$、$x_5=(0,1)$、$x_6=(0,0)$、$x_7=(1,0)$，其中 x_1、x_2、x_3、x_4 為正，x_5、x_6、x_7 為負，此七點在原始空間中難以線性分類(如圖 2.6(a))，若以(2.1.39)做映射可得到 $\phi(x_1)=(2^2,2^2,\sqrt{2}\times2\times2)=(4,4,4\sqrt{2})$，其餘表列在表 2.1，如此便可將資料映射至特徵空間，並運用線性方式的將資料分類(如圖 2.6(b))。

表 2.1 原始空間映射至特徵空間

資料維度 2	資料維度 3
$x_1=(2,2)\Rightarrow y_1=+1$	$\phi(x_1)=(4,4,4\sqrt{2})\Rightarrow y_1=+1$
$x_2=(-2,-2)\Rightarrow y_1=+1$	$\phi(x_2)=(4,4,4\sqrt{2})\Rightarrow y_2=+1$
$x_3=(2,-2)\Rightarrow y_1=+1$	$\phi(x_3)=(4,4,-4\sqrt{2})\Rightarrow y_3=+1$
$x_4=(-2,2)\Rightarrow y_1=+1$	$\phi(x_4)=(4,4,-4\sqrt{2})\Rightarrow y_4=+1$
$x_5=(0,1)\Rightarrow y_1=-1$	$\phi(x_5)=(0,1,0)\Rightarrow y_5=-1$
$x_6=(0,0)\Rightarrow y_1=-1$	$\phi(x_6)=(0,0,0)\Rightarrow y_6=-1$
$x_7=(1,0)\Rightarrow y_1=-1$	$\phi(x_7)=(1,0,0)\Rightarrow y_7=-1$



(a) (b)

圖 2.6 特徵映射: $\phi: \mathbb{R}^2 \Rightarrow \mathbb{R}^3$

而 x_1 、 x_2 映射到特徵空間時內積運算為

$$\left\langle \phi\left(X_1\right), \phi\left(X_2\right) \right\rangle = \left\langle \left(2^2, 2^2, \sqrt{2} \times 2 \times 2\right), \left(\left(-2\right)^2, \left(-2\right)^2, \sqrt{2} \times \left(-2\right) \times \left(-2\right)\right) \right\rangle$$

$$= 2^2 \left(-2\right)^2 + 2^2 \left(-2\right)^2 + 2 \cdot 2 \cdot 2 \cdot \left(-2\right) \cdot \left(-2\right) = 64$$

若直接套用核心函式則可直接在二維空間運算

$$K\left(X_1, X_2\right) = \left\langle X_1, X_2 \right\rangle^2 = \left\langle \left(2, 2\right), \left(-2, -2_2\right) \right\rangle^2 = \left(-4 + -4\right)^2 = 64$$

顯而易見的，兩個方法可得到同樣答案，而且使用核心函數可以節省做映射以及內積的運算。上述的方法帶來很大的方便，由於特徵空間是個想像的空間，理論上可以有無限的維度，因此再複雜的問題都可以增加維度來使得分類變得簡單。透過在特徵空間中資料點與資料點間的內積來描述彼此的關係，而不使用所有的資料點之坐標來運算，最大的優點是計算更方便，而在特徵空間中運算資料的內積會比在輸入空間對實際坐標做映射容易很多。而且特徵映射不必求得資料映射後的坐標 $\phi\left(\mathbf{x}\right)$ ，可以不需知道實際的映射函數 ϕ 之數值。

首先在一般的狀況下，會先給定一個核心函數(kernel function)用來描述資料點之間的相似度，下面列出較常用的核心函數，其中 $\mathbf{x}, \mathbf{z} \in \mathbb{R}^n$ ，

Polynomial kernel function:

$$K\left(\mathbf{x}, \mathbf{z}\right) := \left(\left\langle \mathbf{x}, \mathbf{z} \right\rangle + c\right)^d = \left(\mathbf{x}^T \mathbf{z} + c\right)^d, c \geq 0, d \geq 2; \qquad (2.1.43)$$

RBF(Radio basis function) Kernel function:

$$K(\mathbf{x},\mathbf{z}) := \exp\left(-\sigma^{-2}\|\mathbf{x}-\mathbf{z}\|^2\right); \tag{2.1.44}$$

Mahalanobis Kernel function:

$$K(\mathbf{x},\mathbf{z}) := \exp\left(-(\mathbf{x}-\mathbf{z})^T \sum{}^{-1}(\mathbf{x}-\mathbf{z})\right); \tag{2.1.45}$$

Sigmoid kernel function:

$$K(\mathbf{x},\mathbf{z}) := \tanh\left(c\cdot\langle\mathbf{x}\cdot\mathbf{z}\rangle^d + \vartheta\right), c>0, \vartheta<0; \tag{2.1.46}$$

Spectral angle-base kernel:

$$K(\mathbf{x},\mathbf{z}) := \cos^{-1}\left(\frac{\mathbf{x}\cdot\mathbf{z}}{\|\mathbf{x}\|\|\mathbf{z}\|}\right); \tag{2.1.47}$$

如此一來便可透過非線性的轉換函數將資料投射到較高維度的空間，以求得更好的效果。接下來，先介紹一般所稱的硬性邊界支援向量機(hard margin support vector machines)，當使用映射時其最佳化問題可改寫為:

$$\begin{aligned} &\min_{\mathbf{w},b} \ \frac{1}{2}\|\mathbf{w}\|^2 \\ &s.t. \quad y_i\left(\langle\mathbf{w},\phi(\mathbf{x}_i)\rangle + b\right) \geq 1, \ i=1,...,n. \end{aligned} \tag{2.1.48}$$

其對偶問題(dual problem):

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n y_i y_j \alpha_i \alpha_j \langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_j)\rangle \\ &s.t \quad \sum_{i=1}^n y_i\alpha_i = 0, \ \alpha_i \geq 0, \ i=1,...,n. \end{aligned} \tag{2.1.49}$$

接下來用核心函數取代資料點映射的內積，上式可改寫成

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i,\mathbf{x}_j) \\ &s.t \quad \sum_{i=1}^n y_i\alpha_i = 0, \ \alpha_i \geq 0, \ i=1,...,n. \end{aligned} \tag{2.1.50}$$

假設α^*為(2.1.50)式對偶問題的解，則此最佳化問題將滿足 KKT 條件，可以得到相似於(2.1.32)式的 KKT 條件，定義$I_{sv} := \{i=1,...,n : \alpha_i > 0\}$，則

$$\mathbf{w}^* = \sum_{i=1}^n \alpha_i^* y_i \phi(\mathbf{x}_i) = \sum_{i\in I_{sv}} \alpha_i^* y_i \phi(\mathbf{x}_i) \tag{2.1.51}$$

對於任何 $p \in I_{sv}$，因為 $\alpha_i^* \geq 0$，藉由同(2.1.34)、(2.1.35)式的方法，經過整理可以得到 b^*:

$$b^* = y_p - \sum_{i \in I_{sv}} \alpha_i^* y_i \left\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_p) \right\rangle = y_p - \sum_{i \in I_{sv}} \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}_p) \tag{2.1.52}$$

於是得到最佳的決策函數:

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i \in I_{sv}} \alpha_i^* y_i \left\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \right\rangle + b^* \\ &= \sum_{i \in I_{sv}} \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b^* \end{aligned} \tag{2.1.53}$$

在這類問題中，資料點可以分為兩類：

(1) 若 $\alpha^* = 0$ 則資料點須滿足

$$y_i \left(\left\langle \mathbf{w}^*, \phi(\mathbf{x}_i) \right\rangle + b^* \right) \geq 1 \tag{2.1.54}$$

表示這些資料點在特稱空間中會被正確的分類並落在邊界正確的一邊，或是落在邊界上，而對建構超平面沒有影響。

(2) 若 $\alpha^* > 0$ 則資料點須滿足

$$y_i \left(\left\langle \mathbf{w}^*, \phi(\mathbf{x}_i) \right\rangle + b^* \right) = 1 \tag{2.1.55}$$

即表示這些資料點投射到特徵空間時，會落在被正確分類的邊界上，並對建構超平面有影響，這些資料點就稱為此問題的支援向量(support vector)。

接下來，同樣將在軟性支援向量機中，將資料點投射到特徵空間當中，則最佳化問題的目標函是跟限制式可改寫為:

$$\begin{aligned} \min_{\mathbf{w},b} \ & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \\ s.t. \ & y_i \left(\left\langle \mathbf{w}, \phi(\mathbf{x}_i) \right\rangle + b \right) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \ i = 1, ..., n. \end{aligned} \tag{2.1.56}$$

其對偶問題(dual problem):

$$\begin{aligned} \max_{\boldsymbol{\alpha}} & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \\ s.t \ & \sum_{i=1}^{n} y_i \alpha_i = 0, \ C \geq \alpha_i \geq 0, \ i = 1, ..., n. \end{aligned} \tag{2.1.57}$$

在高維度的特徵空間中，資料點將隨著維度的增加分佈的更為稀疏。同時軟性邊界支援向量機允許某些程度上的錯誤，如此一來便可解決無法線性分割的問題，並得到較好的結果。

假設 α^* 為(2.1.57)式對偶問題的解，可以得到 KKT 條件，定義 $I_{sv} := \{i = 1,...,n : \alpha_i > 0\}$ 經由 KKT 條件可以得到

$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \phi(\mathbf{x}_i) = \sum_{i \in I_{sv}} \alpha_i^* y_i \phi(\mathbf{x}_i) \tag{2.1.58}$$

對於任何 $p \in I_{sv}$，因為 $\alpha_i^* \geq 0$，藉由前面的方法，經過整理可以得到 b^*:

$$b^* = y_p - \sum_{i \in I_{sv}} \alpha_i^* y_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_p) \rangle = y_p - \sum_{i \in I_{sv}} \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}_p) \tag{2.1.59}$$

於是可得到最佳的決策函數:

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i \in I_{sv}} \alpha_i^* y_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle + b^* \\ &= \sum_{i \in I_{sv}} \alpha_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b^* \end{aligned} \tag{2.1.60}$$

在這類問題中，資料點可以分為三類：

(1) 若 $\alpha^* = 0$ 則 $\xi_i = 0$，而資料點須滿足

$$y_i \left(\langle \mathbf{w}^*, \phi(\mathbf{x}_i) \rangle + b^* \right) \geq 1 - \xi_i^* = 1 - 0 = 1 \tag{2.1.61}$$

表示這些資料點在特稱空間中會被正確的分類並落在邊界正確的一邊，或是落在邊界上，而對建構超平面沒有影響。

(2) 若 $C > \alpha^* > 0$ 則 $\xi_i = 0$，而資料點須滿足

$$y_i \left(\langle \mathbf{w}^*, \phi(\mathbf{x}_i) \rangle + b^* \right) \geq 1 - \xi_i^* = 1 - 0 = 1 \tag{2.1.62}$$

即表示這些資料點投射到特徵空間時，會落在被正確分類的邊界上，並對建構超平面有影響，這些資料點為此問題的支援向量。

(3) 若 $\alpha^* = C$ 則資料點須滿足

$$\begin{aligned} & y_i \left(\langle \mathbf{w}^*, \phi(\mathbf{x}_i) \rangle + b^* \right) - 1 + \xi_i^* = 0 \\ & \Rightarrow y_i \left(\langle \mathbf{w}^*, \phi(\mathbf{x}_i) \rangle + b^* \right) = 1 - \xi_i^* < 1 \end{aligned} \tag{2.1.63}$$

表示這些資料點在特徵空間中，可能落在超平面與邊界之間，或是並沒有落在正確分類的那一邊，被分類在超平面錯誤的另一邊。

按照以上的方法，便可將問題轉換成最小化目標函式後，要求解的最佳化問題就變成一個凸規劃問題，而目標函數為凸函數(convex function)且滿足限制式的解集合(feasible set)會是一個凸集合(convex set)，這樣一類的問題可以透過梯度下降法(gradient descent)[3]或是序列最小優化法(sequential minimal optimization, SMO)[3]來求解。

2.2 支援迴歸向量機

一般來說，支援向量機不但可以處理分類問題，更可以衍生解決關於迴歸的問題。在迴歸問題當中，我們給定訓練資料$\mathbf{S}=\{(\mathbf{x}_i,\mathrm{y}_i), i=1...n\}$，其中$\mathbf{x}_i \in \mathbb{R}^n$，$y_i \in \mathbb{R}, i=1,...,n$，而每一個$y_i$是相對於輸入值$\mathbf{x}_i$的歸屬值、目標值或輸出值。也就是說分類問題與迴歸問題最大的差別在於輸出的部分，分類問題的輸出目標為整數，而迴歸問題的輸出目標則是實數，因此也可以把分類問題看做是迴歸問題的一種特例。一個迴歸模型便是從這些樣本(patterns)當中，以一組向量來對目標做預測。而支援迴歸向量機(support vector regression, SVR)是一種基於非線性核心的迴歸模型，主要是要在高維度的特徵空間中找出擁有最小風險的迴歸超平面能夠準確預測資料的分佈，並希望能具有良好的函數逼近(function approximation)功能以及泛化能力(generalization capabilities)。

在眾多的支援迴歸向量機模型裡，最常見的就是使用ε-insensitive band[12]來找迴歸超平面的$\varepsilon-$SVR，支援迴歸向量機中超平面的邊界是由ε來控制，如此一來硬性支援迴歸向量機的最佳化問題便成為:

$$\begin{aligned} &\min_{w,b} \ \frac{1}{2}\|\mathbf{w}\|^2 \\ &s.t. \quad \left\| y_i - \left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b\right)\right\| \le \varepsilon, \ i=1,...,n. \end{aligned} \tag{2.2.1}$$

其中$\varepsilon \ge 0$，用來表示 SVR 預測值與實際值差距，定義ε-insensitive band 在於能允許ε的錯誤損失，避免訓練模型時為了減少實驗誤差所造成 overfitting 的現象，而$\varepsilon-$SVR 的名稱也是由此而來。

如同分類問題，輸入的資料點往往不容易恰好落在ε-insensitive band 之上或正確的區域。因此，為了讓支援迴歸向量機擁有更佳的解釋能力，對於落在ε-insensitive band 之外，對於最佳化不利的點，引入鬆弛變數給予懲罰值。此外與 SVM 相同，為了解決複雜的迴歸問題，亦會將資料點投射到特徵空間中，透過核心函數在特徵空間中描述資料點與資料點之間的關係，由此尋找超平面。接

下來論文當中為了方便，將使用 SVR 來代表$\varepsilon-$SVR ，而經過修正的 SVR 目標函式與限制式如下

$$\begin{aligned}\min_{w,b}\ & \frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{n}\left(\xi_i+\hat{\xi}_i\right)\\ s.t.\ & \left(\left\langle \mathbf{w},\phi(\mathbf{x}_i)\right\rangle+b\right)-y_i\le\varepsilon+\xi_i,\\ & y_i-\left(\left\langle \mathbf{w},\phi(\mathbf{x}_i)\right\rangle+b\right)\le\varepsilon+\hat{\xi}_i,\\ & \xi_i,\hat{\xi}_i\ge 0,\ i=1,...,n.\end{aligned} \tag{2.2.2}$$

其中，n 是訓練樣本的數量，C 是在模型複雜度與允許資料錯誤容忍度中做較佳取捨(trade-off)的參數，ξ_i 與 $\hat{\xi}_i$ 則是鬆弛變數可調整是否允許預測值與目標值差距大於 ε 。在這裡定義 $\phi: X\to F$ 為一個非線性的映射函數，可將資料從輸入空間 X(input space)映射到新的特徵空間 F 當中。而迴歸超平面的決策函數可以表示如下:

$$f(\mathbf{x})=\left\langle \mathbf{w},\mathbf{x}\right\rangle+b \tag{2.2.3}$$

其中，$\mathbf{w}$ 是權重向量，b 是偏移量。

為了解決(2.2.2)式，同樣地可以使用拉格朗日法，得到其對偶問題:

$$\begin{aligned}\max_{\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}}}\ & \sum_{i=1}^{n}y_i(\hat{\alpha}_i-\alpha_i)-\varepsilon\sum_{i=1}^{n}(\hat{\alpha}_i+\alpha_i)-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\hat{\alpha}_i-\alpha_i)(\hat{\alpha}_j-\alpha_j)K\left(\mathbf{x}_i,\mathbf{x}_j\right)\\ s.t\ & \sum_{i=1}^{n}(\hat{\alpha}_i-\alpha_i)=0,\\ & C\ge\alpha_i,\hat{\alpha}_i\ge 0,\ i=1,...,n.\end{aligned} \tag{2.2.4}$$

其中 $\alpha_i,\hat{\alpha}_i, i=1,...,n$ 是拉格朗日乘數，而 $\boldsymbol{\alpha}=[\alpha_1,\alpha_2,...,\alpha_n]$ 且 $\hat{\boldsymbol{\alpha}}=[\hat{\alpha}_1,\hat{\alpha}_2,...,\hat{\alpha}_n]$。並且使用 $K\left(\mathbf{x}_i,\mathbf{x}_j\right)$ 當核心函數來避免使用 $\left\langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_j)\right\rangle$ 內積算法時會造成太大的運算量。本實驗將採取最廣為使用的 RBF(Radial Basis Function)核心，此核心定義如下:

$$\begin{aligned}K\left(\mathbf{x}_i,\mathbf{x}_j\right)&=\left\langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_j)\right\rangle\\ &=\exp\left(-\gamma\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2\right)\end{aligned} \tag{2.2.5}$$

這裡的 γ 為 RBF 核心的寬度參數。接下來使用序列最小優化法(sequential

minimal optimization, SMO)解(2.2.4)式，假設得到最佳化的解為 $\alpha_i, \hat{\alpha}_i, i = 1,...,n$ ，則根據 KKT 條件，需要滿足以下條件:

$$\begin{aligned}
&\alpha_i^* \left(\left\langle \mathbf{w}^*, \phi(\mathbf{x}_i) \right\rangle + b^* - y_i - \varepsilon - \xi_i^* \right) = 0 \\
&\hat{\alpha}_i^* \left(y_i - \left\langle \mathbf{w}^*, \phi(\mathbf{x}_i) \right\rangle - b^* - \varepsilon - \hat{\xi}_i^* \right) = 0 \\
&\xi_i^* \hat{\xi}_i^* = 0, \alpha_i^* \hat{\alpha}_i^* = 0, \\
&\left(\alpha_i^* - C \right) \xi_i^* = 0, \left(\hat{\alpha}_i^* - C \right) \hat{\xi}_i^* = 0, i = 1,...,n.
\end{aligned} \tag{2.2.6}$$

由 KKT 關係式可以推得 $\alpha_i^* = C$ 時，ξ_i^* 才有可能不為零，而，$\alpha_i^* \hat{\alpha}_i^* = 0$ 說明 α_i^* 與 $\hat{\alpha}_i^*$ 不可能同時不為零，因此得到:

$$\begin{aligned}
&\varepsilon - y_i + \left\langle \mathbf{w}^*, \phi(\mathbf{x}_i) \right\rangle + b^* \geq 0 \text{ and } \xi_i^* = 0 \text{ if } \alpha_i^* < C \\
&\varepsilon - y_i + \left\langle \mathbf{w}^*, \phi(\mathbf{x}_i) \right\rangle + b^* \leq 0 \text{ if } \alpha_i^* > 0
\end{aligned} \tag{2.2.7}$$

同理可以推出 $\hat{\alpha}_i^*$ 的情形。因此經過整理可以得到 b^*:

$$\begin{aligned}
&\max\left\{ -\varepsilon + y_i - \left\langle \mathbf{w}^*, \phi(\mathbf{x}_i) \right\rangle \mid \alpha_i^* < C \text{ or } \hat{\alpha}_i^* > 0 \right\} \leq b^* \leq \\
&\max\left\{ -\varepsilon + y_i - \left\langle \mathbf{w}^*, \phi(\mathbf{x}_i) \right\rangle \mid \alpha_i^* > 0 \text{ or } \hat{\alpha}_i^* < C \right\}
\end{aligned} \tag{2.2.8}$$

最後，可以得到最佳的決策函數:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \left(\hat{\alpha}_i^* - \alpha_i^* \right) K \left\langle \mathbf{x}_i, \mathbf{x} \right\rangle + b^* \tag{2.2.9}$$

第三章　研究方法

本章將先介紹多核心支援迴歸向量機(multiple-kernel support vector regression)的主要想法，以及如何用兩階段多核心學習演算法(two-stage multiple-kernel learning algorithm)來得到最佳的核心權重以及拉格朗日乘數的數值。

3.1 多核心支援迴歸向量機

在早期提出的 SVR 模型中，通常使用單一個映射函數ϕ，而有個核心函數K。但如遇到資料集有區域性的複雜分佈特性時，使用單核心函數難以將多變的分佈特性詮釋良好。因此對於過去使用單核心時無法處理的部分，引入核心融合(Kernel fusion)的技巧來協助解決這樣的問題，取而代之的是將多個映射函數合併起來，同時來做映射。如果要將有 M 個映射函數向量融合的話，簡單直接的想法如下

$$\Phi(\mathbf{x})=\left[\phi_1(\mathbf{x})\ \phi_2(\mathbf{x})\ ...\ \phi_M(\mathbf{x})\right] \tag{3.1.1}$$

如此就可將輸入資料映射到特徵空間。但這樣只有每個映射函數對於資料都擁有相同的權重，對於問題的描述性依舊很差。於是我們採取了加權總和的概念來融合這些映射函數，可將(3.1.1)改寫如下

$$\Phi(\mathbf{x})=\left[\sqrt{\mu_1}\phi_1(\mathbf{x})\ \sqrt{\mu_2}\phi_2(\mathbf{x})\ ...\ \sqrt{\mu_M}\phi_M(\mathbf{x})\right] \tag{3.1.2}$$

其中$\mu_1,\mu_2,...,\mu_M$是函數成份的權重。所以現在迴歸問題需要最佳化的部分有兩個地方，一個是迴歸的超平面$f(\mathbf{x})$，而另外一部分是權重向量$\boldsymbol{\mu}=\left[\mu_1,\mu_2,...,\mu_M\right]$。為了讓(3.1.2)的映射函數$\Phi$是可行的，這些權重必須大於或等於零[12]。因此我們也希望在搜尋空間將這些權重的範圍限制在相加總和為1，來避免產生 overfitting。因此將(3.1.2)式改寫，得到目標式與限制式如下:

$$
\begin{aligned}
&\min_{\boldsymbol{\mu}} \min_{w,b} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\left(\xi_i + \hat{\xi}_i\right) \\
&s.t. \quad \left(\left\langle \mathbf{w}, \Phi(\mathbf{x}_i)\right\rangle + b\right) - y_i \le \varepsilon + \xi_i, \\
&\qquad y_i - \left(\left\langle \mathbf{w}, \Phi(\mathbf{x}_i)\right\rangle + b\right) \le \varepsilon + \hat{\xi}_i, \\
&\qquad \xi_i, \hat{\xi}_i \ge 0,\ i = 1,...,n, \\
&\qquad \mu_s \ge 0,\ s = 1,...,M, \\
&\qquad \sum_{s=1}^{M} \mu_s = 1
\end{aligned}
\tag{3.1.3}
$$

其中 Φ 是如(3.1.2)式的函數映射向量。

接下來藉由拉格朗日法，可以將(3.1.3)式轉換成對偶形式:

$$
\begin{aligned}
&\min_{\boldsymbol{\mu}} \max_{\alpha,\hat{\alpha}} \sum_{i=1}^{n} y_i(\hat{\alpha}_i - \alpha_i) - \varepsilon\sum_{i=1}^{n}(\hat{\alpha}_i + \alpha_i) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j)\widetilde{K}\left(\mathbf{x}_i, \mathbf{x}_j\right) \\
&s.t \quad \sum_{i=1}^{n}(\hat{\alpha}_i - \alpha_i) = 0, \\
&\qquad C \ge \alpha_i, \hat{\alpha}_i \ge 0,\ i = 1,...,n, \\
&\qquad \mu_s \ge 0,\ s = 1,...,M, \\
&\qquad \sum_{s=1}^{M} \mu_s = 1
\end{aligned}
\tag{3.1.4}
$$

其中

$$
\begin{aligned}
\widetilde{K}\left(\mathbf{x}_i, \mathbf{x}_j\right) &= \left\langle \phi(\mathbf{x}_i), \phi\left(\mathbf{x}_j\right)\right\rangle \\
&= \mu_1\left\langle \phi_1(\mathbf{x}_i), \phi_1\left(\mathbf{x}_j\right)\right\rangle + \mu_2\left\langle \phi_2(\mathbf{x}_i), \phi_2\left(\mathbf{x}_j\right)\right\rangle + \cdots + \mu_M\left\langle \phi_M(\mathbf{x}_i), \phi_M\left(\mathbf{x}_j\right)\right\rangle \\
&= \mu_1 K_1\left(\mathbf{x}_i, \mathbf{x}_j\right) + \mu_2 K_2\left(\mathbf{x}_i, \mathbf{x}_j\right) + \cdots + \mu_M K_M\left(\mathbf{x}_i, \mathbf{x}_j\right) \\
&= \sum_{s=1}^{M} \mu_s K_s\left(\mathbf{x}_i, \mathbf{x}_j\right)
\end{aligned}
\tag{3.1.5}
$$

是 M 個核心函數 $K_1, K_2, ..., K_M$ 依據映射函數 $\phi_1, \phi_2, ..., \phi_M$ 產生的加權總和。假設 $\boldsymbol{\mu}^*$ 、$\boldsymbol{\alpha}^*$ 、$\hat{\boldsymbol{\alpha}}^*$ 和 b^* 為(3.1.4)之解，則可以得到迴歸超平面應該為:

$$
f(\mathbf{x}) = \sum_{i=1}^{n}\left(\hat{\alpha}_i^* - \alpha_i^*\right)\widetilde{K}\left\langle \mathbf{x}_i, \mathbf{x}\right\rangle + b^* \tag{3.1.6}
$$

其中 $b^* = y_k + \varepsilon - \sum_{i=1}^{n}(\hat{\alpha}_i^* - \alpha_i^*)\widetilde{K}\left(\mathbf{x}_i, \mathbf{x}_k\right)$,$\forall \alpha_k^*, 0 < \alpha_k^* < C$

3.2 兩階段多核心學習

為了解決(3.1.4)式所產生的最佳化問題，我們提出了兩階段的最佳化演算法來處理。這個演算法由 SMO(sequential minimal optimization)最佳化方法和梯度投影法(gradient projection)[3]兩個階段所組成。如圖 3.1，這些階段會重複的執行，一直到滿足停止條件為止，其中 t 用來代表重複次數。

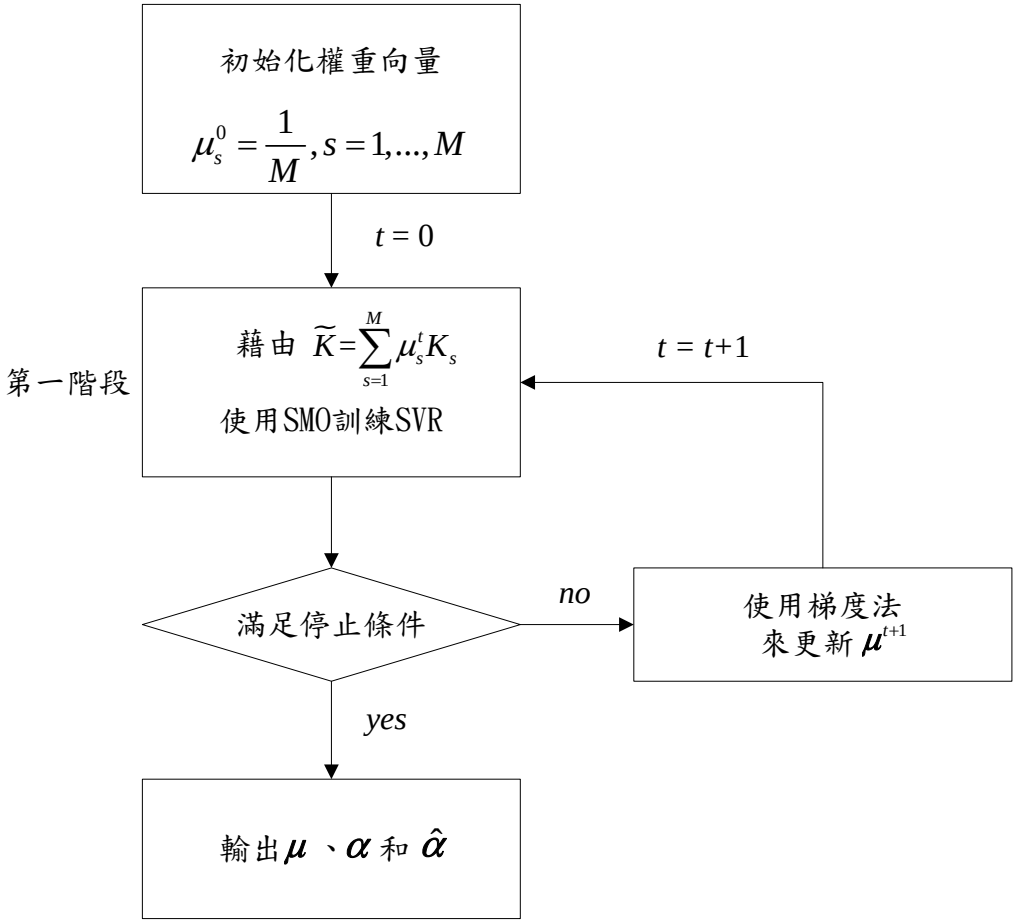


圖 3.1 兩階段多核心學習演算法

首先將 μ 初始化，設定權重為平均數。在第一階段，先保持 μ 為固定，然後求得 $\widetilde{K}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{s=1}^{M} \mu_s K_s(\mathbf{x}_i, \mathbf{x}_j)$，接著(3.1.4)式會變成

$$\begin{aligned} &\max_{\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}}} \sum_{i=1}^{n} y_i(\hat{\boldsymbol{\alpha}} - \alpha_i) - \varepsilon \sum_{i=1}^{n} (\hat{\alpha}_i + \alpha_i) - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j) \widetilde{K}(\mathbf{x}_i, \mathbf{x}_j) \\ &s.t \quad \sum_{i=1}^{n} (\hat{\alpha}_i - \alpha_i) = 0, \\ &\quad C \geq \alpha_i, \hat{\alpha}_i \geq 0, \; i = 1,...,n. \end{aligned} \tag{3.2.1}$$

可以很明顯的看出，上式與(3.1.4)式型態非常的類似，且同樣的可以用 SMO 來

求解。接下來在第二階段，保持拉格朗日乘數 $\boldsymbol{\alpha}$ 和 $\hat{\boldsymbol{\alpha}}$ 固定，並藉著梯度投影法[3]來更新權重向量 $\boldsymbol{\mu}$。由於 SMO 是一個常用來處理對偶問題的標準演算法，在此將不詳細說明，詳細的描述可以參考文獻[28]。接下來我們將敘述如何在第二階段使用梯度投影法[3]來最佳化權重向量 $\boldsymbol{\mu}$。

在第二階段時，拉格朗日乘數已由前一階段求出，因此可以將(3.1.4)重寫如下:

$$\begin{aligned} &\min_{\boldsymbol{\mu}} \ J(\boldsymbol{\mu}) \\ &s.t \quad \mu_s \geq 0,\ s=1,...,M, \\ &\qquad \sum_{s=1}^{M} \mu_s = 1 \end{aligned} \tag{3.2.2}$$

此處

$$J(\boldsymbol{\mu}) = \sum_{i=1}^{n} y_i(\hat{\alpha}_i - \alpha_i) - \varepsilon \sum_{i=1}^{n}(\hat{\alpha}_i + \alpha_i) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j)\widetilde{K}(\mathbf{x}_i, \mathbf{x}_j). \tag{3.2.3}$$

注意，$J(\boldsymbol{\mu})$ 只隨著 $\boldsymbol{\mu}$ 改變。接下來使用梯度投影法[3]我們可以得到

$$\boldsymbol{\mu}^{k+1} = \boldsymbol{\mu}^k + \eta^k(\hat{\boldsymbol{\mu}}^k - \boldsymbol{\mu}^k) \tag{3.2.4}$$

其中 $\boldsymbol{\mu}^k$ 是第 k 次的迭代的權重向量，而 $0 < \eta^k \leq 1$ 為步長(step-size)且 $\hat{\boldsymbol{\mu}}^k$ 定義為:

$$\hat{\boldsymbol{\mu}}^k = \begin{cases} \mathbf{z} & \text{如果 } \mathbf{z} \text{ 落在可行區域,} \\ \mathbf{z}_{\perp} & \text{其他,} \end{cases} \tag{3.2.5}$$

$$\mathbf{z} = \boldsymbol{\mu}^k - s^k \nabla J(\boldsymbol{\mu}^k) \tag{3.2.6}$$

其中 s^k 為正純量，且 $\mathbf{z}_{\perp}$ 代表 $\mathbf{z}$ 在可行區域(feasible region)的投影。合理區域包中所有的向量 $\mathbf{v} = [v_1, v_2, ..., v_n]$ 須滿足 $v_s \geq 0, 1 \leq s \leq M$ 且 $\sum_{s=1}^{M} v_s = 1$。而 $\nabla J(\boldsymbol{\mu}^k)$ 是下列的梯度:

$$\begin{aligned} \nabla J(\boldsymbol{\mu}_s^k) &= \frac{\partial J}{\partial \boldsymbol{\mu}_s^k} \\ &= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j)K_s(\mathbf{x}_i, \mathbf{x}_j),\ s = 1,...,M \end{aligned} \tag{3.2.7}$$

而 $\mathbf{z}$ 在合理區域的投影 $\mathbf{z}_{\perp}$，可以用以下的約束問題來求得:

$$\begin{aligned} &\min_{\mathbf{z}_{\perp}} \ \left\| \mathbf{z} - \mathbf{z}_{\perp} \right\|^2 \\ &s.t \quad \text{所有} \mathbf{z}_{\perp} \text{的組成都是非負整數且總和為} 1 \end{aligned} \tag{3.2.8}$$

而上式可以重新改成如下面形式的二次規劃問題(quadratic programming):

$$\begin{aligned} \min_{\mathbf{z}_\perp} \quad & \frac{1}{2}\left(\mathbf{z}_\perp\right)^T H\ \mathbf{z}_\perp - \mathbf{z}^T\mathbf{z}_\perp \\ s.t \quad & \mathrm{k}_s^T\mathbf{z}_\perp \geq 0, 1 \leq s \leq M, \mathbf{e}^T\mathbf{z}_\perp = 1 \end{aligned} \tag{3.2.9}$$

其中 H 是 rank 為 M 的單位矩陣(identity matrix)，k_s 代表一組 M 個向量，而此向量第 s 個為 1 其他為 0。**e** 則是一組 M 個向量，此向量全部由 1 所組成。此外，步長 η^k 在合理區域中使用 Armijo 規則來確定。因此藉由 $0<\beta,\sigma<1$，我們可以設定 $\eta^k=\beta^{m_k}$，此處 m_k 是下列式子的第一非負整數 m

$$J\left(\boldsymbol{\mu}^{k+1}\right) - J\left(\boldsymbol{\mu}^{k+1} + \beta^m\left(\hat{\boldsymbol{\mu}}^{k+1} - \boldsymbol{\mu}^{k+1}\right)\right) \geq -\sigma\beta^m \nabla J\left(\boldsymbol{\mu}^{k+1}\right)^T\left(\hat{\boldsymbol{\mu}}^{k+1} - \boldsymbol{\mu}^{k+1}\right) \tag{3.2.10}$$

詳細的梯度投影演算法的運作方式可以參考圖 3.2。注意，此處的迭代次數用 k 來表示。而在每一次的開始時將 k 重設為 0，並將權重向量設定為 $\boldsymbol{\mu}^t$。接著將(3.2.4)式重複的執行至達成停止條件。當演算法終止時，最後所求的權重設定給 $\boldsymbol{\mu}^{t+1}$ 。

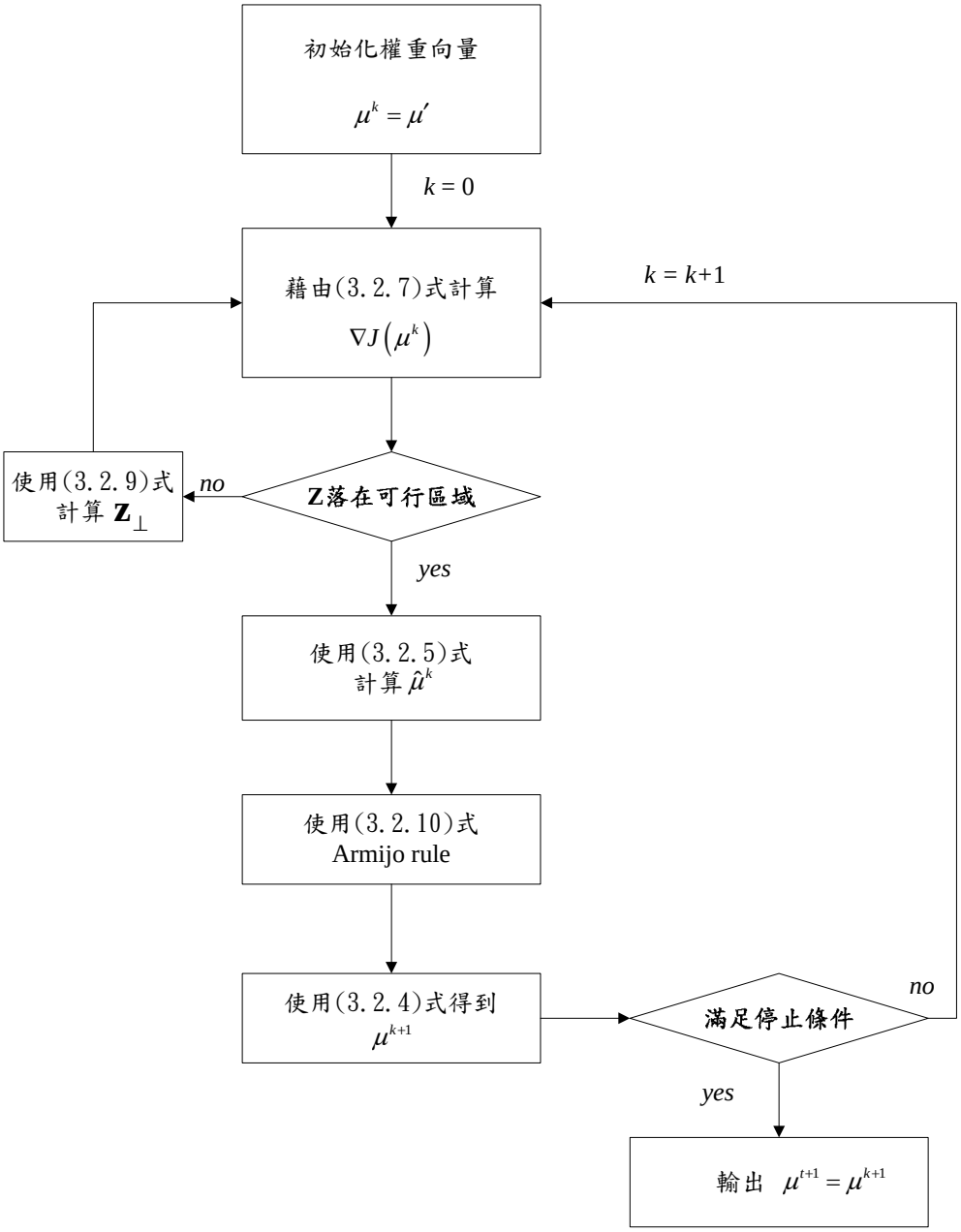


圖 3.2 梯度投影法應用於兩階段多核心學習演算法

第四章　實驗結果

為了測試我們所提出模型預測效果，我們實驗資料引用臺灣證券交易所發行量加權股價指數（TAIEX），其由臺灣證券交易所編制。臺灣證券交易所使用 Passche 加權平均[42]，其計算公式為:

指數=當期總發行市值÷基值×100

其中當期總發行市值為各個採樣股票乘上發行股數所得市值之總和。因此發行公司的市值越高，其指數占的權重就越高。發行量加權指數使用 1986 年當基期，基期指數設定為 100，其中採樣樣本除了特別股、全額交割股外，其於上市股票都包含其中。

此外，我們也將本研究預測效果與其他方法做比較，如單核心支援迴歸向量機(single-kernel support vector regression, SKSVR)[34]、自迴歸整合移動平均模型(Autoregressive Integrated Moving Average, ARIMA)[5]、TSK 型態的模糊類神經網路(FNN)[9]。為了方便起見，以下將我們的多核心支援迴歸向量機(mutiple-kernel support vector regression)簡稱為 MKSVR。

4.1 實驗一 SKSVR 與 MKSVR 比較

首先，我們比較原始的 SKSVR[34]與我們所提出的 MKSVR，在這個實驗中我們取用 TIAEX 從 2002 年十月到 2005 年十二月的每日收盤價，並以三個月一季當成一個測試區間，將資料分為訓練/驗證/測試三部分，其中訓練資料為四季，驗證及測試資料使用一季。這些資料即分別為 DS-I、DS-II、DS-III、DS-IV。舉例來說 DS-I 資料包含了 2002 年十月到 2004 年九月，取用 2002 年十月到 2004 年九月每日收盤價為訓練資料，而 2004 年十月到 2004 年十二月的每日收盤價為驗證資料，最後把 2005 年一月到 2005 三月當做測試資料[34]。依照這樣的方法，分別把 DS-I、DS-II、DS-III、DS-IV 四個周期建立出來，如表 4.1。並以訓練資料建立模型，再以驗證資料決定模型最佳參數，最後用測試資料檢驗預測效果。假設給定原始的每日收盤價為 $\mathbf{p}=\{p_1, p_2, ..., p_t, ...\}$，遵循[34]訂定訓練的樣本為 $(\mathbf{x}_t, y_t)$ 使用在 SKSVR 及 MKSVR 中。

表 4.1 實驗一的資料區間

資料集	訓練	驗證	測試
DS-I	2002/10 ~ 2004/09	2004/10 ~ 2004/12	2005/01 ~ 2005/03
DS-II	2003/01 ~ 2004/12	2005/01 ~ 2005/03	2005/04 ~ 2005/06
DS-III	2003/04 ~ 2005/03	2005/04 ~ 2005/06	2005/07 ~ 2005/09
DS-IV	2003/07 ~ 2005/06	2005/07 ~ 2005/09	2005/10 ~ 2005/12

首先第 t 天的 n 天的實驗移動平均 $\mathrm{EMA}_n(t)$ 定義如下:

$$\mathrm{EMA}_n(t) = \mathrm{EMA}_n(t-1) + \alpha\left(p_t - \mathrm{EMA}_n(t-1)\right) \tag{4.1.1}$$

其中 p_t 為第 t 天的當日收盤價且 $\alpha = \frac{2}{1+n}$。而輸出變數 y_t 定義如下:

$$y_i = \mathrm{RDP}_{+5}(t) = \frac{\mathrm{EMA}_3(t) - \mathrm{EMA}_3(t-5)}{\mathrm{EMA}_3(t-5)} \times 100 \tag{4.1.2}$$

在輸入資料 $\mathbf{x}_t$ 的部分，$\mathbf{x}_t$ 包含了五個特徵，即 $\mathbf{x}_t = \left[x_{t,1}\ x_{t,2}\ x_{t,3}\ x_{t,4}\ x_{t,5}\right]$。輸入的變數包含有 $x_{t,1}$=$\widehat{\mathrm{EMA}}_{15}(t-5)$、$x_{t,2}$=$\mathrm{RDP}_{-5}(t-5)$、$x_{t,3}$=$\mathrm{RDP}_{-10}(t-5)$、$x_{t,4}$=$\mathrm{RDP}_{-15}(t-5)$、$x_{t,5} = \mathrm{RDP}_{-20}(t-5)$ 並定義如下:

其中 $\widehat{\mathrm{EMA}}_n(t)$ (exponential moving average)為把收盤價經過轉換，將每日收盤價減去 n 天的指數移動平均:

$$\widehat{\mathrm{EMA}}_n(t) = p_t\text{-}\mathrm{EMA}_n(t) \tag{4.1.3}$$

$\mathrm{RDP}_{-n}(t)$ (lagged relative difference in percentage)為相對延遲百分比:

$$\mathrm{RDP}_{-n}(t) = \frac{p_t - p_{t-n}}{p_{t-n}} \times 100 \tag{4.1.4}$$

而效能比較方法採用平均方根誤差(root mean squared error ,RMSE)，而其定義如下:

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(y_t - \hat{y}_t\right)^2} \tag{4.1.5}$$

其中 y_t 與 $\hat{y}_t$ 分別代表實際的輸出與預測的輸出。

對於 SKSVR 來說，當使用 RBF 核心時，需要預先給定三個參數，即 C、ε 與 γ。我們使用 C=1，ε=0.001 來測驗 SKSVR 的效果。此外我們使用 37 個不同

的超參數γ設定，從 0.01 到 0.09 間隔為 0.01;從 0.1 到 0.9 間隔為 0.1;從 1 到 9 間隔為 1;從 10 到 100 間隔為 10，並分別把四個資料集不同參數的預測效果繪製於圖 4.1 中。從圖中我們可以看見在 SVSVR 中，不同的資料集如果要擁有好的表現，需要不同的γ設定。在 DS-I 中，最好的效果出現在$0.01 \le \gamma \le 0.05$。在 DS-II 中，最好的效果出現在$0.1 \le \gamma \le 0.5$。在 DS-III 中，最好的效果出現在$50 \le \gamma \le 100$。在 DS-IV 中，最好的效果出現在$0.01 \le \gamma \le 0.05$。而每個資料區間，最佳的 RMSE 整理在表 4.2。

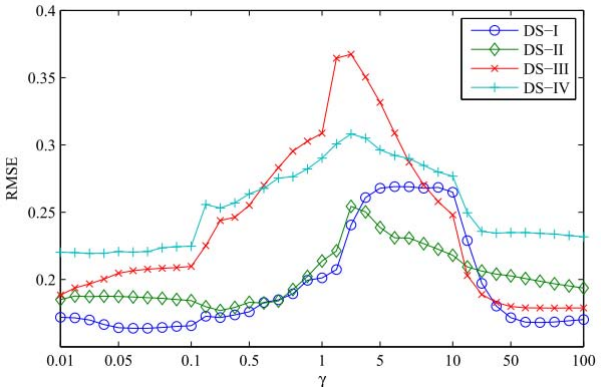


圖 4.1 實驗一中不同超參數時 SKSVR 的預測效果

表 4.2 SKSVR 與 MKSVR 在 RBF 核心於實驗一的結果

	資料集			
方法	DS-I	DS-II	DS-III	DS-IV
SKSVR	0.170	0.179	0.188	0.234
MKSVR	0.161	0.174	0.179	0.219

而在多核心學習方法中，我們希望把上面所提到的 37 個不同的 RBF 核心合併成一個核心，即$\gamma \in \{0.01,0.02,...,0.09,0.1,0.2,...,0.9,1,2,...,9,10,20,...,90,100\}$。因此，合併的核心矩陣會把 37 個核心矩陣加權總和起來，即$\widetilde{K} = \mu_1 K_1 + \mu_2 K_2 + \cdots + \mu_{37} K_{37}$，其中$\mu_1$表示第一個核心矩陣$\gamma = 0.01$的核心權重，$\mu_2$表示第二個核心矩陣$\gamma = 0.02$的核心權重，以此類推。而四個資料集，藉由 MKSVR 實驗得到的 RMSE 結果表列在表 4.2。

接下來，將 SKSVR 套用不同的核心函式: Linear、Polynomial、RBF、Sigmoid，並將其最好的結果列在表 4.3。在多核心學習方法中，將以上四種不同的核心函

數，以兩階段多核心學習法合併，即 $\widetilde{K} = \mu_1 K_1 + \mu_2 K_2 + \cdots + \mu_4 K_4$ ，其中 μ_1 表示 Linear 核心矩陣的權重，μ_2 表示 Polynomial 核心矩陣的權重，以此類推。而四個資料集，藉由 MKSVR 實驗得到的 RMSE 結果列在表 4.3。

表 4.3 SKSVR 與 MKSVR 在不同核心於實驗一的結果

		資料集			
	kernel	DS-I	DS-II	DS-III	DS-IV
SKSVR	Linear	0.164	0.196	0.212	0.234
	Poly	0.216	0.183	0.248	0.225
	Sigmoid	0.169	0.182	0.185	0.219
	RBF	0.170	0.179	0.188	0.234
MKSVR	MKL	0.147	0.173	0.185	0.218

很明顯的，從表 4.2、表 4.3 每一個資料集都可以看見 MKSVR 的效果皆優於 SKSVR，而且我們不需要事先特別給定某一個超參數，也避免大量使用錯誤嘗試法來尋找出適當的超參數。

4.2 實驗二 ARIMA、SKSVR 與 MKSVR 比較

在這個實驗當中，我們將比較 MKSVR 與 ARIMA[5]的效果。在這個實驗中我們取用 TIAEX 從 2004 年一月到 2005 年十二月的每日收盤價，並以三個月一季當成一個測試區間，將資料分為訓練/驗證/測試三部分，其中訓練 資料為四季，驗證及測試資料使用一季。並把資料建成了四個資料集: DS-V、DS-VI、DS-VII、DS-VIII [27]。舉例來說 DS-V 資料包含了 2004 年一月到 2005 年三月，取用 2004 年一月到 2004 九月每日收盤價當訓練資料，而 2004 年十月到 2004 年十二月的每日收盤價當驗證資料,最後把 2005 年一月到 2005 三月當做測試資料。依照這樣的方法，分別把 DS-V、DS-VI、DS-VII、DS-VIII 四個周期建立出來，如表 4.4。並以訓練資料建立模型，再以驗證資料決定模型最佳參數，最後用測試資料檢驗預測效果。

表 4.4 實驗二、實驗三的資料區間

資料集	訓練	驗證	測試
DS-V	2004/01 ~ 2004/09	2004/10 ~ 2004/12	2005/01 ~ 2005/03
DS-VI	2004/04 ~ 2004/12	2005/01 ~ 2005/03	2005/04 ~ 2005/06
DS-VII	2004/04 ~ 2005/07	2005/04 ~ 2005/06	2005/07 ~ 2005/09
DS-VIII	2004/10 ~ 2005/06	2005/07 ~ 2005/09	2005/10 ~ 2005/12

給定原始的每日收盤價為 $\mathbf{p}=\{p_1,p_2,...,p_t,...\}$，遵循[27]訂定訓練的樣本為 $(\mathbf{x}_t,y_t)$。在這個實驗中為使資料成平穩時間序列(stationary time series)，如圖 4.2 (c)，首先將原始的每日收盤價為 $\mathbf{p}=\{p_1,p_2,...,p_t,...\}$ 使用自然對數轉換，得到另一個時間序列 $\mathbf{y}'=\{y_1',y_2',...,y_t',...\}$ 其中 $y_t'=\ln(p_t)$。輸出序列則訂定為 $\mathbf{y}=\{y_1,y_2,...,y_t,...\}$ 並把 y_t 定義如下:

$$y_t=y_t'-y_{t-1}' \tag{4.2.1}$$

如此一來，我們就可得到一個比較平穩的輸出變數。在圖 4.2 中有一個簡單的範例，可以明白這不同的序列有何不同。而輸入向量 $\mathbf{x}_t$ 包含了三個部份，分別是自迴歸模型、差分處理及移動平均模型。而由三個參數 p,d,q 來代表其特徵，分別為自迴歸階數、差分階數、移動平均階數。因此使用 ARIMA(p,d,q)，來區分不同的模型。每個輸入向量包含了(p+q)個特徵，即 $\mathbf{x}_t=\left[x_{t,1}\ x_{t,2}\ ...\ x_{t,p+q}\right]$。

其中自迴歸模型(autoregressive, AR)，簡單來說就是現在的某一變數數值，和同一變數過去的變數數值有關(可能是上一期，也可能是上二、三期等，這裡以 p 來表示)，所以可以把 AR 用下列函數型態來表示:

$$y_t = c + \sum_{i=1}^{p} \alpha_i y_{t-i} + \varepsilon_t \tag{4.2.2}$$

其中 c：常數的截距項；p：落後的期數；ε_t：白噪音；α_i：y_{t-i} 的係數(常數)。由於 AR 比較像是一種數學模型，無法準確的模擬出實際發生的現象，為了使預測與真實更符合，需要讓預測跟隨機項產生關聯，於是使用移動平均模型(moving average, MA)，讓模型含有誤差修正的特性。移動平均模型表示如下:

$$y_t = c + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i} + \varepsilon_t \tag{4.2.3}$$

其中 c：常數的截距項；q ：移動平均的期數；ε_t：誤差項；β_i：ε_{t-i} 的係數(常數)。

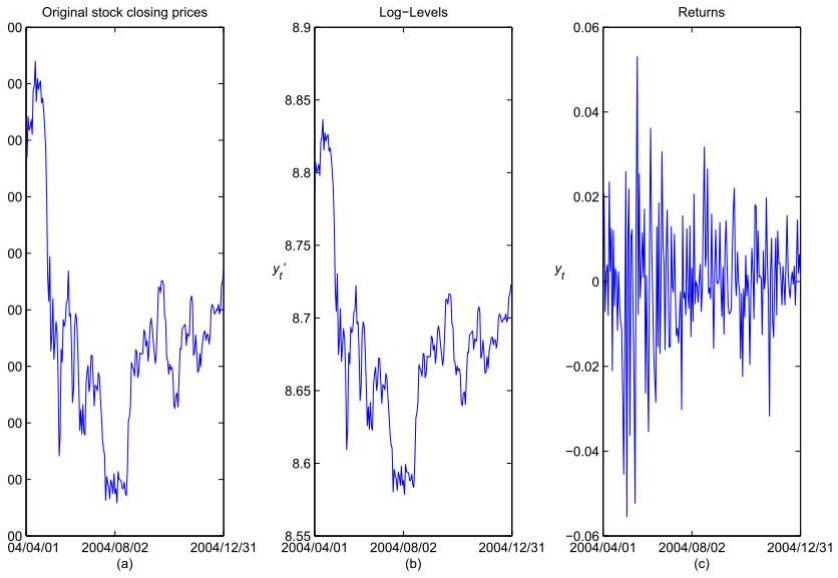


圖 4.2 資料處理示意圖取自 TAIEX (2004/04/01 ~ 2004/12/31)分別為 (a) p , (b) y′ , (c) y .

一般證券市場的時間序列問題，時常是非平穩性的（non-stationary），因此一般的做法是對其取 d 階差分至序列平穩化(stationary)，序列穩定之後便可開始訓練模型。

$$y_t^d = y_t^{d-1} - y_{t-1}^{d-1} \tag{4.2.4}$$

其中 y_t^d 代表經過 d 階差分後的第 t 天，y_t^{d-1} 代表經過 d-1 階差分後的第 t 天，並

把 y_t^0 為未經過差分的第 t 天。

因此，ARIMA 模型使用流程為:

1. 根據自相關函數(autocorrelation function, ACF)及偏自相關函數(partial autocorrelation function, PACF) 圖形識別其平穩性。
2. 若為非平穩之時間序列，則將其進行平穩化處哩，直到 ACF 及 PACF 的數值趨近於零為止。
3. 依據所 ACF 及 PACF 圖形，識別出所須對應的時間序列模型。本研究所使用的為 ARIMA 模型，模型函數如下:

$$y_t^d = \sum_{i=1}^{p} \alpha_i y_{t-i}^d + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i}^d + \mu_t^d \tag{4.2.5}$$

其中 d：代表 d 階差分後，所求得之數值；μ_t^d ：常數的截距項；q：移動平均的期數；p：落後的期數；ε_t^d：誤差項；α_i：y_{t-i} 的係數(常數)；β_i：ε_{t-i} 的係數(常數)。

舉例來說，對於 ARIMA(2,1,3)我們有 $x_{t,1} = y_{t-1}$、$x_{t,2}=y_{t-2}$、$x_{t,3}=\varepsilon_{t-1}$、$x_{t,4}=\varepsilon_{t-2}$、$x_{t,5}=\varepsilon_{t-3}$，其中 ε_{t-1}、ε_{t-2}、ε_{t-3} 為預測誤差。而使用的 RMSE 定義如下:

$$\begin{aligned} RMSE &= \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(p_t - \hat{p}_t\right)^2} \\ &= \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\exp\left(y_t'\right) - \exp\left(\hat{y}_t'\right)\right)^2} \\ &= \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\exp\left(y_t'\right) - \exp\left(\hat{y}_t' + y_{t-1}'\right)\right)^2} \end{aligned} \tag{4.2.6}$$

其中 $\hat{y}_t'$ 是預測的輸出變數。

為了比較 ARIMA 及 MKSVR，我們考慮的 25 種關於 ARIMA(m,1,n)的模型，其中 $m \in \{1,2,3,4,5\}$，$n \in \{1,2,3,4,5\}$。而 ARIMA 在四個資料集的預測的成果，繪製於圖 4.3。明顯地，不同的參數設定的 ARIMA 只有很小的變化。

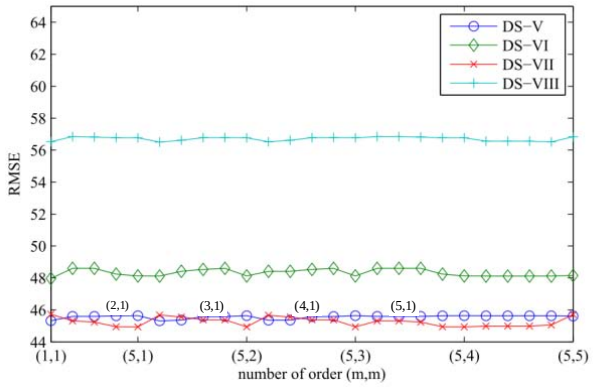


圖 4.3 實驗二中 ARIMA 於不同參數時的預測效果

我們在四個資料集上實驗 SKSVR 與 MKSVR 時，也使用與實驗一同樣的方法，特徵則取 ARIMA 在各資料集最佳的參數做為輸入變數。同樣的也將不同參數時 SKSVR 的預測效果繪製於圖 4.4。從圖中可以明顯的觀察到，在不同的資料集中，欲擁有最佳的預測效果，需使用相異的γ參數設定。

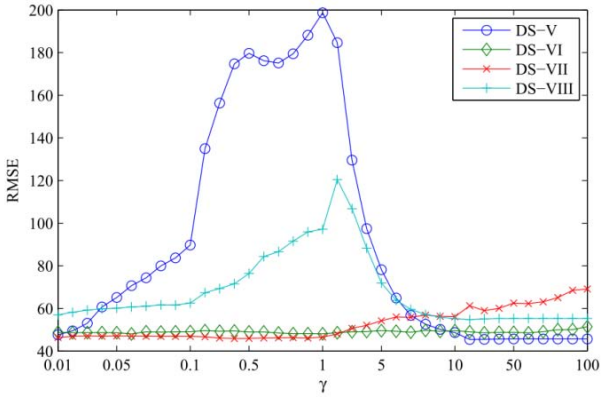


圖 4.4 實驗二中不同超參數時 SKSVR 的預測效果

表 4.5 ARIMA、SKSVR 與 MKSVR 於 RBF 核心在實驗二的效果

	資料集			
方法	DS-V	DS-VI	DS-VII	DS-VIII
ARIMA	45.421	48.400	45.674	56.957
SKSVR	45.686	48.667	46.401	55.294
MKSVR	45.634	47.297	44.142	54.882

而 ARIMA 及 SKSVR 在每個資料區間，不同的參數設定下，最佳的結果表列在表 4.5。而 MKSVR 在不同資料集的 RMSE 值亦同樣表列在表 4.5。明顯地，在不同的資料集裡，

接下來，將 SKSVR 套用不同的核心函式: Linear、Polynomial、RBF、Sigmoid，

並將其最好的結果列在表 4.6。在多核心學習方法中，將以上四種不同的核心函數，以兩階段多核心學習法合併，即 $\widetilde{K} = \mu_1 K_1 + \mu_2 K_2 + \cdots + \mu_4 K_4$ ，其中 μ_1 表示 Linear 核心矩陣的權重，μ_2 表示 Polynomial 核心矩陣的權重，以此類推。而四個資料集，藉由 MKSVR 實驗得到的 RMSE 結果表列在表 4.6。

表 4.6 SKSVR 與 MKSVR 在不同核心於實驗二的結果

		資料集			
	kernel	DS-I	DS-II	DS-III	DS-IV
SKSVR	Linear	48.552	49.518	44.982	57.379
	Poly	104.369	47.728	49.707	56.792
	Sigmoid	46.832	49.151	44.950	56.609
	RBF	45.686	48.667	46.401	55.294
MKSVR	MKL	45.686	48.137	44.984	56.390

以上都可以看出 MKSVR 可以表現的與最佳的 ARIMA、SKSVR 一樣好，或是更好。但我們卻不需要擔心 MKSVR 中關於超參數的設定。

圖 4.5 繪製出對於資料集 DS-V 到 DS-VIII 由 MKSVR 所預測的結果。

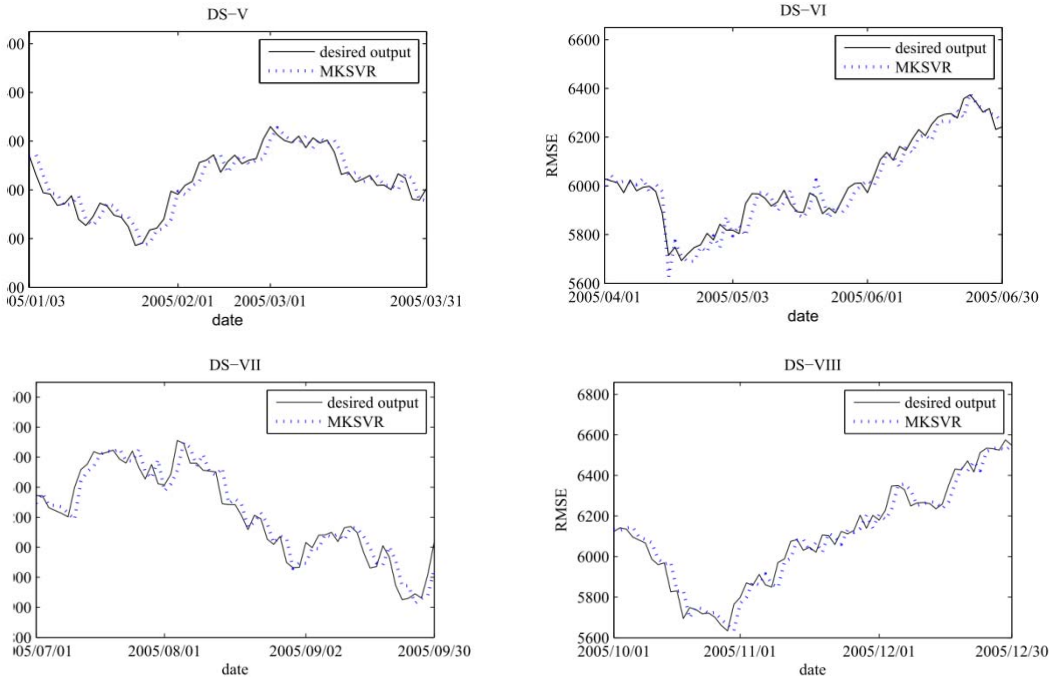


圖 4.5 實驗二 MKSVR 所預測的結果

4.3 實驗三 FNN、SKSVR 與 MKSVR 比較

在這個實驗當中，我們將比較 MKSVR 與 FNN[9]的預測效果。我們使用與實驗二相同的資料集如表 4.3。當給定原始的每日收盤價為$\mathbf{p}=\{p_1,p_2,...,p_t,...\}$，遵循[9]訂定訓練的樣本為$(\mathbf{x}_t,y_t)$使用在實驗當中。在這個實驗中以$y'$代表$p_t$，即令$y'_t=p_t$。並使用兩種股市指標 SMA 及 BIAS 產生輸入資料$\mathbf{x}_t$。其中 SMA 是簡單移動平均的縮寫(simple moving average)，是用來強調移動的趨勢，並將輸出值及變動平滑化。而第 t 天的 n 天 SMA 定義如下:

$$\mathrm{SMA}_n(t)=\frac{\sum_{i=t}^{t-5}p_i}{n}. \tag{4.3.1}$$

BIAS 是觀察收盤價與移動平均線差距的比例，而第 t 天的 n 天 BIAS 定義如下:

$$\mathrm{BIAS}_n(t)=\frac{p_t-\mathrm{SMA}_n(t)}{\mathrm{SMA}_n(t)}\times 100 \tag{4.3.2}$$

因此定義輸入變數$x'_{t,1}$=$\mathrm{SMA}_6(t-1)$、$x_{t,2}$=BIAS_6(t-1)。現在將下面的資料集以 k-means[19](一種很常用的分群演算法)分成 K 群，然後輸出變數y_t可定義為:

$$y_t=\frac{y'_t-\bar{y}'_j}{\sigma_{y'_j}} \tag{4.3.3}$$

其中y'_t是屬於第 j 群而$\bar{y}'_j$、$\sigma_{y'_j}$分別是在第 j 群中y'方向的平均數與標準差。而輸入向量$\mathbf{x}_t=\left[x_{t,1}\ x_{t,2}\right]$可以由下式得到:

$$xi_t=\frac{x'_{t,i}-\bar{x}'_{j,i}}{\sigma_{x'_{j,i}}} \tag{4.3.4}$$

其中$i=1,2$，且$\left[x'_{t,1}\ x'_{t,2}\right]$屬於第 j 群而$\bar{x}'_{j,i}$、$\sigma_{x'_{j,i}}$分別是在第 j 群中第 i 方向的平均數與標準差。而在此處 RMSE 的定義如下:

$$\begin{aligned}RMSE&=\sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(p_t-\hat{p}_t\right)^2}\\&=\sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(y'_t-\hat{y}'_t\right)^2}\\&=\sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(y'_t-\left(\hat{y}'_t\times\sigma_{y'_j}+\bar{y}'_j\right)\right)^2}\end{aligned} \tag{4.3.5}$$

其中$\hat{y}'_t$為預測輸出且 j 是同一群的指標。

對於 FNN，使用標準的三層網絡。而在輸入層有兩個節點，輸出層有一個節點。為了測試不同的結構所造成的影響，設定隱藏節點數量從 2 到 15，而且將隱藏層中每個間隔設為 1。而 FNN 在不同數量的隱藏節點之預測效果繪製於圖 4.6。從圖中可以發現，FNN 在不同的資料集中，需要不同數量的隱藏節點才會有良好的預測效果。

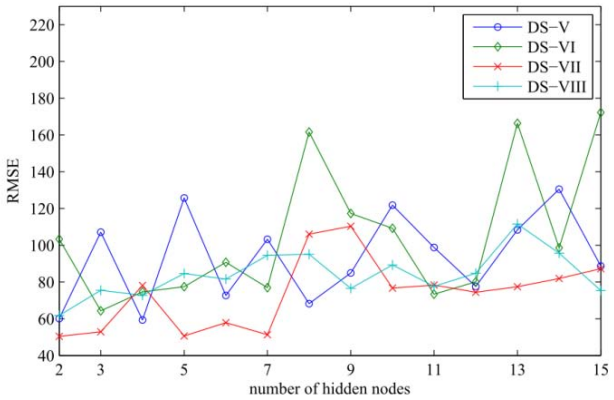


圖 4.6 實驗三 FNN 使用不同數量的隱藏節點之預測效果

我們使用與實驗一相同的方法來運作 SKSVR 與 MKSVR，而輸入變數與 FNN 相同。並將使用不同超參數的 SKSVR 預測效果繪製於圖 4.7。從圖中可以明顯的觀察到，在這些的資料集中最好的預測效果，其所使用的的γ參數皆不同。

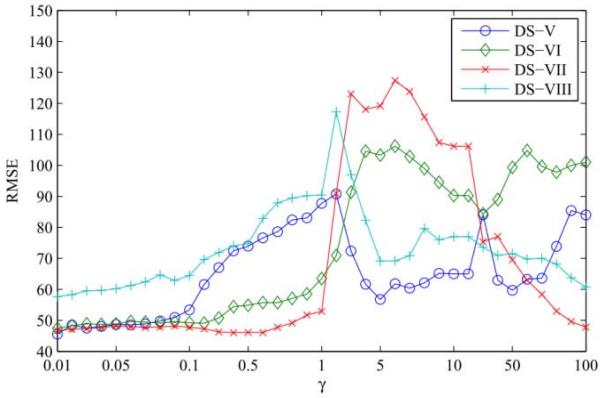


圖 4.7 實驗三不同超參數時 SKSVR 的預測效果

而 FNN 與 SKSVR 在不同參數中，最佳的預測效果列於表 4.7。而 MKSVR 在四個資料集的預測成果，同樣的也列在表 4.7。接下來，將 SKSVR 套用不同的核心函式: Linear、Polynomial、RBF、Sigmoid，並將其最好的結果列在表 4.8。在多核心學習方法中，將以上四種不同的核心函數，以兩階段多核心學習法合併，即$\widetilde{K} = \mu_1 K_1 + \mu_2 K_2 + \cdots + \mu_4 K_4$，其中$\mu_1$表示 Linear 核心矩陣的權重，$\mu_2$表示 Polynomial 核心矩陣的權重，以此類推。而四個資料集，藉由 MKSVR 實驗得到

的 RMSE 結果表列在表 4.8。

表 4.7 FNN、SKSVR 與 MKSVR 在 RBF 核心於實驗三的效果

	資料集			
方法	DS-V	DS-VI	DS-VII	DS-VIII
FNN	59.260	64.232	50.395	61.774
SKSVR	45.543	47.434	46.669	57.625
MKSVR	45.531	47.398	45.907	57.301

表 4.8 SKSVR 與 MKSVR 在不同核心於實驗三的結果

		資料集			
	kernel	DS-I	DS-II	DS-III	DS-IV
SKSVR	Linear	45.691	48.568	50.099	57.573
	Poly	55.582	48.464	47.476	64.660
	Sigmoid	45.758	47.476	46.550	56.415
	RBF	45.544	47.434	46.669	57.625
MKSVR	MKL	45.570	47.260	46.550	56.204

明顯地，對於每一個資料集，MKSVR 都表現出優於 FNN 及 SKSVR 的預測效果。而且在使用 MKSVR 時不需要使用到錯誤嘗試法來尋找超參數。同樣的，我們將 MKSVR 的預測結果繪製於圖 4.8。

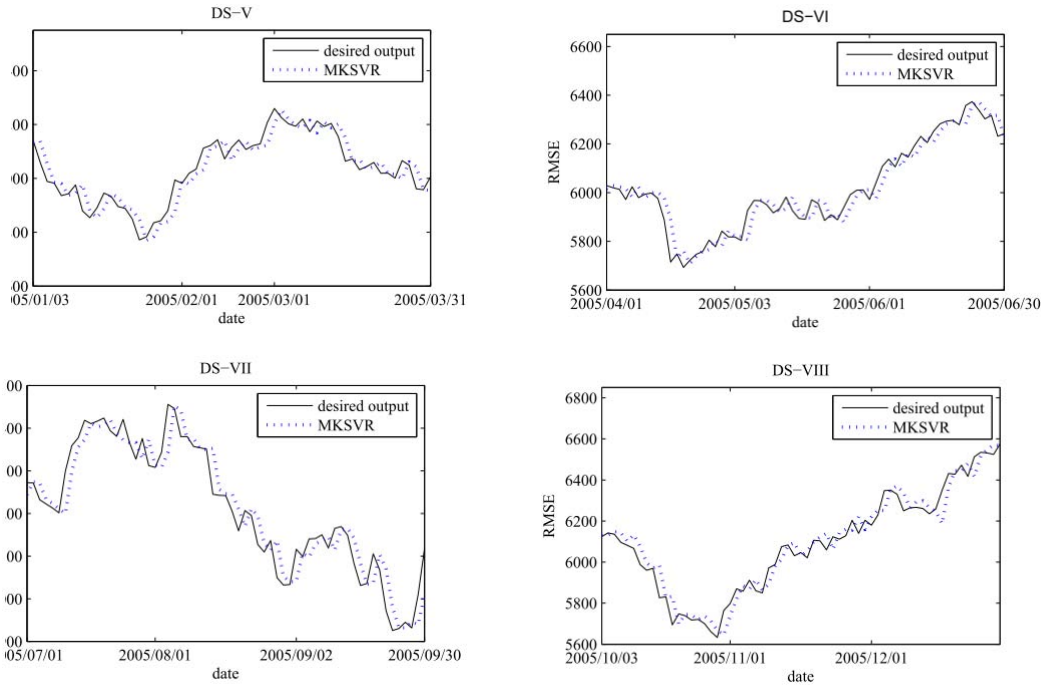


圖 4.8 實驗三 MKSVR 所預測的結果

第五章　結論與未來研究方向

5.1 結論

在本論文中，我們提出了一個多核心支援迴歸向量應用於股價預測。並使用兩階段多核心學習演算法來合併在支援迴歸向量機中的多個核心矩陣。此學習演算法應用 SMO 優化法以及梯度投影法來得到拉格朗日乘數的數值及最佳的核心權重。藉由此演算法，有助於結合不同的超參數設定，並改善使整個系統的的效能。此外，使用者也不需要預先給予特定的超參數，如此可避免以往總是大量使用錯誤嘗試法來尋找可用的超參數設定，造成大量的時間浪費。而使用 TAIEX 當實驗資料的研究結果顯示，我們的方法能夠表現出比其他模型更佳的的效果。

5.2 未來研究方向

本論文雖然在錯誤率以及自動化方面，有著顯著的效果，但在實驗的過程中，發現仍有幾點改善空間，這也是我們未來需要研究的方向。

1. 當使用梯度投影法時，需要初始化梯度投影法的 step-size，由於每個資料集彼此的分佈十分不同，導致各自初始設定的 setp-size 皆不盡相同。在未來的研究當中，可以針對支援向量點的分佈特性，尋找使用梯度投影法時，step-size 較佳的初始值。
2. 在實驗二中也發現，當給定的超參數分佈不夠廣時，也許還是不能找到能將資料完整解釋的核心矩陣。在未來的研究當中，可以朝當核心矩陣依舊不足對資料有良好解釋時，是否能夠往合適的超參數設定區間找尋更合適的核心矩陣。

參考文獻

[1] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proceedings of the 21th International Conference on Machine Learning*, pp. 6-13, 2004.

[2] K. P. Bennett, M. Momma, and M. J. Embrechts, "MARK: A boosting algorithm for heterogeneous kernel models," in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 24-31, 2002.

[3] D. P. Bertsekas, *Nonlinear Programming*, Second Edition, Athena Scientific, Massachusetts, USA, 1999.

[4] T. Bollerslev, "Generalized autoregressive conditional heteroscedasticity," *Journal of Econometrics*, vol. 31, no. 3, pp. 307-327, 1986.

[5] G. E. P. Box and G. M. Jenkins, *Time series analysis: Forecasting and control*, 3rd Edition, Prentice Hall, Englewood Cliffs, 1994.

[6] L. Cao and F. E. H. Tay, "Financial forecasting using support vector machines," *Neural Computing & Applications*, vol. 10, no. 2, pp. 184-192, 2001.

[7] L. Cao and F. E. H. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," *IEEE Transactions on Neural Networks*, vol. 14 no. 6, pp. 1506-1518, 2003.

[8] D. G. Champernowne, "Sampling theory applied to autoregressive schemes," *Journal of the Royal Statistical Society*: *Series B*, vol. 10, pp. 204-231, 1948.

[9] P.-C. Chang and C.-H. Liu, "A TSK type fuzzy rule based system for stock price prediction," *Expert Systems with Application*, vol. 34, no. 1, pp. 135-144, 2008.

[10] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 131-159, January 2002.

[11] K. Crammer, J. Keshet, and Y. Singer, "Kernel design using boosting," S. Becker, S. Thrun, and K. Obermayer (Eds.), in *Advances in Neural Information Processing Systems*, MIT Press, Cambridge, MA, USA, vol. 15, pp. 537-544, 2003.

[12] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, Cambridge, UK, 2000.

[13] K. Duan, S. Keerthi, and A. N. Poo, "Evaluation of simple performance measures for tuning SVM hyperparameters," *Neurocomputing*, vol. 51, pp. 41-59, 2003.

[14] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation," *Econometrica*, vol. 50, no. 4, pp. 987-1008, 1982.

[15] P.-C. Fernando, A. A.-R. Julio, and G. Javier, "Estimating GARCH models using support vector machines," *Quantitative Finance*, vol. 3, no. 3, pp. 163-172, 2003.

[16] T. V. Gestel, J. A. K. Suykens, D. E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. D. Moor, and J. Vandewalle, "Financial time series prediction using least squares support vector machines within the evidence framework," *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 809-821, 2001.

[17] M. Gönen and E. Alpaydin, "Localized multiple kernel learning," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 352-359, 2008.

[18] J. V. Hansen and R. D. Nelson, "Neural networks and traditional time series methods: A synergistic combination in state economic forecasts," *IEEE Transactions on Neural Networks*, vol. 8, no. 4, pp. 863-873, 1997.

[19] J. A. Hartigan and M. A. Wong, "A K-means clustering algorithm," *Applied Statistics*, vol. 28, pp. 100-108, 1979.

[20] V. Kecman, *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*, MIT Press, Cambridge, MA, USA, 2001.

[21] K. J. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert Systems with Application*, vol. 19, no. 2, pp. 125-132, 2000.

[22] J. T.-Y. Kwok, "The evidence framework applied to support vector machines," *IEEE Transactions on Neural Networks*, vol. 11, no. 5, pp. 1162-1173, 2000.

[23] Y.-K. Kwon and B.-R. Moon, "A hybrid neurogenetic approach for stock forecasting," *IEEE Transactions on Neural Networks*, vol. 18, no. 3, pp. 851-864, 2007.

[24] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27-72, 2004.

[25] S.-K. Oh, W. Pedrycz, and H.-S. Park, "Genetically optimized fuzzy polynomial neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 1, pp. 125-144, 2006.

[26] S. Ong, A. J. Smola, and R. C. Williamson, "Learning the kernel with hyperkernels," *Journal of Machine Learning Research*, vol. 6, pp. 1043-1071, 2006.

[27] P.-F. Pai and C.-S. Lin, “A hybrid ARIMA and support vector machines model in stock price forecasting,” *Omega: The International Journal of Management Science*, vol. 33, no. 6, pp. 497-505, 2005.

[28] J. C. Platt, “Fast training of support vector machines using sequential minimal optimization,” B. Schölkopf, C. J. C. Burges, and A. J. Smola (Eds.), in *Advances in Kernel Methods: Support Vector Learning*, MIT Press, Cambridge, MA, USA, vol. 11, pp. 185-208,1999.

[29] M. Qi and G. P. Zhang, “Trend time-series modeling and forecasting with neural networks,” *IEEE Transactions on Neural Networks*, vol. 19, no. 5, pp. 808-816, 2008.

[30] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, “More efficiency in multiple kernel learning,” in *Proceedings of the 24th International Conference on Machine Learning*, pp. 775-782, 2007.

[31] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, “SimpleMKL,” *Journal of Machine Learning Research*, vol. 9, pp. 2491-2521, 2008.

[32] S. Sonnenburg, G. Ratsch, C. Schäfer, and B. Schölkopf, “Large scale multiple kernel learning,” *Journal of Machine Learning Research*, vol. 7, pp. 1531-1565, 2006.

[33] M. Szafranski, Y. Grandvalet, and A. Rakotomamonjy, “Composite kernel learning,” in *Proceedings of the 25th International Conference on Machine Learning*, pp. 1040-1047, 2008.

[34] F. E. H. Tay and L. Cao, “Application of support vector machines in financial time series forecasting,” *Omega: The International Journal of Management Science*, vol. 29, no. 4, pp. 309-317, 2001.

[35] I. W.-H. Tsang and J. T.-Y. Kwok, “Efficient hyperkernel learning using second-order cone programming,” *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 48-58, 2006.

[36] G. Valeriy and B. Supriya, “Support vector machine as an efficient framework for stock market volatility forecasting,” *Computational Management Science*, vol. 3, no. 2, pp. 147-160, 2006.

[37] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, New York, USA, 1995.

[38] Z. Wang, S. Chen, and T. Sun, “MultiK-MHKS: A novel multiple kernel learning algorithm,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 348-353, 2008.

[39] H. Yang, L. Chan, and I. King, “Support vector machine regression for volatile stock market prediction,” in *Proceedings of the 3rd International Conference on Intelligent Data Engineering and Automated Learning*, pp. 391-396, 2002.

[40] M. H. F. Zarandi, B. Rezaee, I. B. Turksen, and E. Neshat, “A type-2 fuzzy rule-based expert system model for stock price analysis,” *Expert Systems with Application*, vol. 36, no. 1, pp. 139-154, 2009.

[41] D. Zhang and L. Zhou, “Discovering golden nuggets: Data mining in financial application,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C, Applications and Reviews*, vol. 34, no. 4, pp. 513-522, 2004.

[42] Taiwan Stock Exchange Corporation.[Online]. URL http://www.twse.com.tw/

[43] 臺灣行政院金融監督管理委員會證卷期貨局.[Online]. URL http://www.sfb.gov.tw/